Optimization of Incident Reporting System for Danish Healthcare

MASTER'S THESIS BY
MARLENA ANNA PLOCHARSKA
MIA HORSBOEL HANSEN
GROUP 15GR1075
JUNE 2015

AALBORG UNIVERSITY
DENMARK

AALBORG UNIVERSITET

Department of Health Science & Technology
Frederik Bajers Vej 7 D2
9220 Aalborg Ø
Telephone (+45) 9940 9940
Fax (+45) 9815 4008
http://www.hst.aau.dk

Title:
Optimization of Incident Reporting System for Danish Healthcare

Theme:
Master's thesis

Project period:
4^{th} Semester of the Master Programme
9^{th} February - 3^{rd} June 2015

Project group:
Group 15gr1075

Members:
Marlena Anna Plocharska
Mia Horsbøl Hansen

Supervisors:
Ole Hejlesen
Louise Pape-Haugaard

Report Pages: 123
Total no. of pages: 132
Completed: June 2, 2015

Synopsis:

10 % of all patients admitted to a hospital experience an incident with an adverse event, while 1/3 of these incidents result in death or disability. Therefore, is the procedure and analysis of the reported incidents important to prevent recurrence. Based on the analyzed literature and interviews regarding the Danish healthcare system, three main problem areas were identified regarding incident reporting, incident feedback, and data extraction and analysis. The problem areas provided knowledge for the development of a proof of concept IR system and IR database design. The validation results indicated possible improvements with the IR system, however further tests of the IR database are required. The IR system proved to be well accomplished regrading usability, and provided means for visualization and data extraction. However, several changes to the IR system have to be implemented and tested before it can be used in the Danish healthcare system.

The contents of this report is freely accessible, however publication (with source references) is only allowed upon agreement with the authors.

Preface

The master thesis has been developed by the group 15gr1075 during the 4^{th} semester of the master program in Biomedical Engineering & Informatics at Aalborg University, over a period from 9^{th} February to 3^{rd} June 2015. The title of the project is *"Optimization of Incident Reporting System for the Danish Healthcare"*.

The target groups for the project include students, supervisors, researchers and others with an interest in biomedical engineering, especially patient safety, health informatics, incident reports and web-based user interfaces.

Reading Guide

Within the project, source references will be listed according to the Harvard method, with given [Surname, Year] in the text. All references are collected in the bibliography at the end of the report and listed alphabetically. Tables and figures are numbered according to the chapter in question, e.g. first figure in Section 2, has a number of 2.1, the next one of 2.2, etc. Therefore there is a possibility of table 2.1 and figure 2.1 in same section. Used abbreviations are defined with first usage and placed in brackets. The image for the front page is downloaded from the website: $http://static1.squarespace.com/static/549ee8e3e4b0bcb26c236d93/t/54c6f5a2e4b05c4a4774f722/1422325163756/bigstock-medicine-doctor-working-with-m-43167979.jpg?format=1500w$.

Acknowledgments

The project group would like to thank our supervisors Ole Hejlesen and Louise Pape-Haugaard from Aalborg University for their guidance and help throughout the project. The project group would also like to thank Patientombuddet, risk managers from: Oncology Department, Aalborg University Hospital and Region North Jutland, and a patient for the interviews. The interviews provided the needed knowledge and understanding of the Danish incident reporting system, which was used doing the problem analyze and solution.

CONTENTS

Chapter 1

Introduction

Millions of people have experienced deaths or disabling injuries from the healthcare system worldwide. The injuries are in contrast to the general presumption, that the healthcare professionals do not harm their patients during their treatment, [Brasaite et al., 2014; Tingle, 2011; Emslie et al., 2002]. In fact, 10 % of all patients admitted to a hospital experience an incident with an adverse event, while a third of these incidents results in death or disability, [Brasaite et al., 2014]. Table 1.1 presents the prevalence of adverse event incidents distributed over selected counties.

Country	Adverse event admissions	Death and permanent harm admissions
Canada	7.5	1.6
Denmark	9.0	0.4
America	10.0	2.0
Australia	10.6	2.0
England	11.7	1.5
New Zealand	12.9	1.9

Table 1.1. The prevalence of incidents from different countries presented in percentage based on medical record reviews, [Runciman and Walton, 2007].

In Denmark, a maximum of 90 days from when an incident happens to its finalization is required. In the Danish healthcare system, an adverse event incident is defined as

> *"an incident which is a result of treatment or hospitalization, and not caused by the patient's illness, and at the same time is either harmful or could have been harmful, but previously was averted, or because of other circumstances not occurred ('near miss'). Adverse events include both previously known and unknown events and errors"*, [Sundhedsstyrelsen, 2003].

In 2013, 181.326 incidents were reported into the Danish Patient Safety Database (DPSD), which is 39.486 more reports than in 2012, [Patientombuddet, 2014a]. The amount of the reported incidents in Denmark is low comparing to the 850.000 annual adverse incidents in the United Kingdom Department of Health estimated by The World Health Organization (WHO), [Organization, 2015].

The figure 1.1 represents the distribution of the different incident types and shows that in Denmark medication error has the highest prevalence of all incidents, [Nordjylland, 2013; Patientombuddet, 2014a]. In the United States 44.000-98.000 patients die due to medication error every year, which makes the incidents the 5th to 8th leading cause of death with an annual cost of $38 billions, [Barach and Small, 2000; Ortiz et al., 2002; Ash et al., 2004].

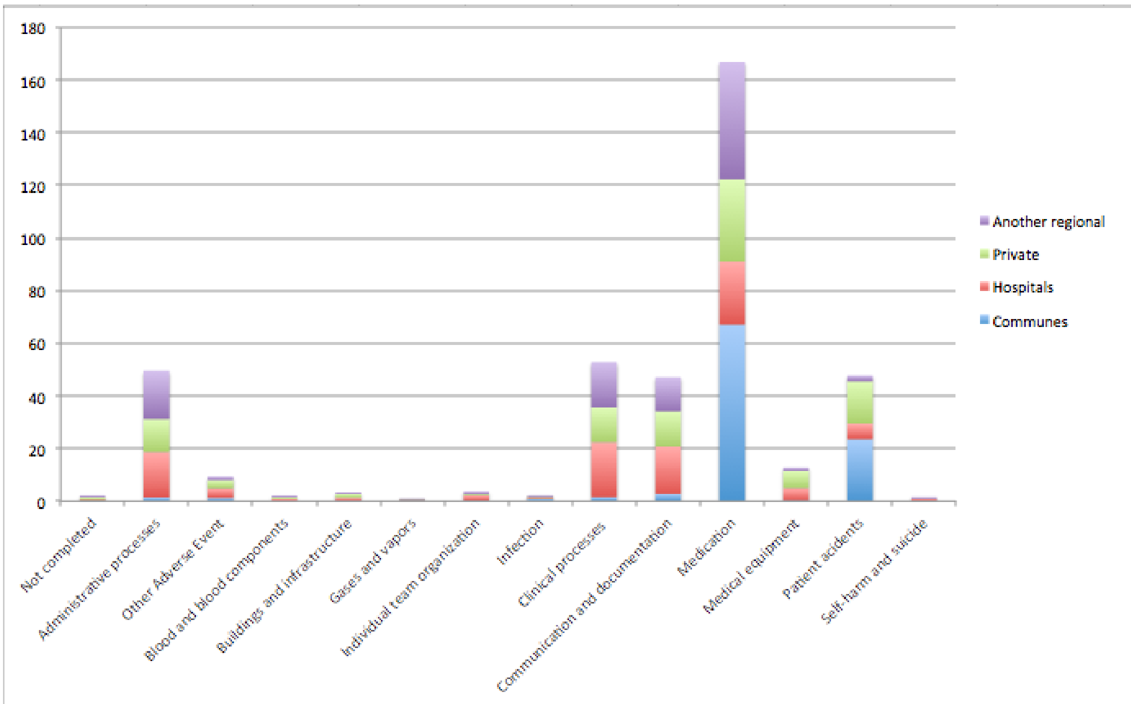


Figure 1.1. Distribution of the different incident types divided into percentage for communes, hospitals, private sector and another regional, which have been submitted to and completed by Danish Patientombuddet, edited from Patientombuddet [2014a].

Nevertheless, literature related to patient safety shows, that around 25% of healthcare professionals do not know how to access an incident report form, 40% of the registrars and consultants have never filled an incident report and 50-96% underreport the adverse event incidents, [Mahajan, 2010; Barach and Small, 2000; Lawton and Parker, 2002].

WHO has estimated, that in the European Union incidents with adverse events could be reduced by implementing systematic approaches strategies. The approach includes for instance an annual prevention of 50-70,2% medical errors (around 750.000), 3,2 million fewer hospital days, reduction of permanent disabilities by 260.000 and reduction of mortality rates by 95.000, [Organization, 2015].

Therefore, use of the incident reporting (IR) system makes it possible for the healthcare system to learn from the errors in the healthcare system, and thereby overcome the errors by finding possible weaknesses in the management process that will improve the quality and safety, [WHO, 2005; Magrabi et al., 2010; Cheung et al., 2011; Ratwani and Fong, 2014; Chai et al., 2013]. The IR system is a scheme for the healthcare professionals to report incidents, when patients experience 'near misses' (no harm) and serious adverse events. The stored incident data in the reporting system provides the possibility to share patterns and highlight trends, [Ratwani and Fong, 2014].

Since the healthcare professionals have to report an incident using the IR system, several problems appeared in relation to the healthcare professionals such as a believe that the reporting form takes too long to fill out, lack of feedback on what action was taken to prevent recurrence of the incident, or lack of understanding of importance to report near misses etc., [Ratwani and Fong, 2014; Evans et al., 2006].

The described introduction to incidents leads to the following initial statement:

> *What kind of problems are related to procedures and systems used within the healthcare system during reporting of an incident with focus on a hospital level?*

Part I

Problem Analysis

CHAPTER 2

INCIDENTS IN HEALTHCARE SYSTEM

Within this chapter, different aspects related to incident reporting (IR) were described. The reporting procedure of the incidents in Denmark was presented together with description of implemented database and learning methods used for feedback on the reported incidents. Relevant IR problems and their causes were identified and described to provide an understanding of the incident reporting culture both worldwide and in Denmark.

All healthcare interventions carry a possibility of risk for a patient, such as harm, reaction to drugs and infusions. The named risks are often caused by complex factors and processes in a healthcare system resulting in errors and incidents. Since incidents may have serious, or even fatal outcomes for the patient. Therefore, understanding of the contributing factors to the incident is crucial, [Tighe et al., 2006].

Recurrence of the incident has to be prevented. Therefore, an incident report is filled to provide a description and documentation of the incident. Based on the report, it is possible to classify the incident as either a near-miss or an adverse event, [Runciman and Walton, 2007].

By looking at the work environment alone, several possible causes of the incidents can be identified. The causes of the incidents include factors such as high work pressure, stress, lack of resources, distraction or confusion, [Patientsikkerhedsdatabase, 2009]. Increasing number of patient within the last few years in the healthcare system resulted in overload of the work pace on the healthcare professionals, that can lead to incidents, [Appendix A.2.2]. The overload contributes to the lack of resources to reduce possibilities and development of solutions preventing the incidents, [Appendix A.2.2].

The incidents result in additional financial costs for a hospital from e.g. patients' prolonged hospital stays. Additionally, the incidents can have an effect on involved staff, and leave a psychological burden or a physical harm to the involved patient, [Ahluwalia and Marriott, 2005].

2.1 Incident Reporting

Incident reporting allows to capture contextual information about the incidents, [Evans et al., 2006]. An IR system can improve organizational learning and enhance safety for the patients. The improvements can be achieved by giving, discussing and sharing information between the healthcare professionals about e.g. possible preventive actions, error types or causes, [Cheung et al., 2011].

In order to improve patient safety, the incidents are analyzed and learning is facilitated. The analysis may help with identifying risks before an incident may occur again, [Kessels-Habraken et al., 2010]. Some of the most common contributing factors to the incidents are communication problems, staff shortages or failure of checking the procedures, [Tighe et al., 2006]. Other factors affecting the healthcare are related to individuals, team, institution, environment, patient and task to be undertaken, [Ahluwalia and Marriott, 2005].

Nevertheless there are plenty of unreported incidents causing many hospitals to fail to learn from the errors. Moreover IR culture of the healthcare professionals can carry some biases, such as focusing IR on a specific incident types with a high severity rate, or on higher reporting of the incidents without direct relation to the tasks performed by the hospital staff. Many incidents, such as ones related to treatment, are found to be underreported, [Kessels-Habraken et al., 2010]. Underreporting of the incidents by healthcare professionals limits use of reports for being used as an epidemiological tool to measure frequency of incidents and effectiveness of the suggested interventions for patient safety improvements. A subjective nature of the IR causes reports to lack consistency and validation of the used classification for the incidents, [Evans et al., 2006].

The incidents are reported by the healthcare staff. There are many nurses, doctors and doctor's secretaries that report the incidents, [Appendix A.1.2]. Nevertheless, there is still difference within the staff on how often they report an incident. There are persons who are effective at reporting, but there are some, who never reported an incident, [Appendix A.2.1]. A study by Kessels-Habraken et al. [2010] documented that the doctors are less willing to report an incident compared to the nurses. A study by Evans et al. [2006] assessed awareness and use of the incident reporting system and identified inhibiting factors for incident reporting in hospitals. To improve reliability of the incident reporting, both nurses and doctors are required to report representative amount of incidents at the hospital. However, the results of the study showed, that the awareness and use of the reporting system varies between them. Even though both doctors and nurses are aware of the IR system's existence, it is mostly nurses who complete an incident report, know how to access IR form and what to do with the completed form, [Evans et al., 2006]. Detailed data from the study regarding differences between doctors and nurses is shown in figure 2.1.

The difference in reporting between doctors and nurses may be caused by few factors, such as lack of error recognition, feeling of shame or blame, doctor's attitude that the

	Doctors (%)		Nurses (%)	
	Yes	N	Yes	N
Awareness of hospital incident reporting system	93.6%	174	99.8%	586
Ever completed an incident report	64.6%	115	89.2%	520
Know how to locate/access an incident form	43.0%	77	88.3%	515
Know what to do with a completed incident form	49.7%	89	81.9%	476

Figure 2.1. Comparison between doctors' and nurses' awareness and use of the incident reporting system presented in percentages, [Evans et al., 2006].

errors are inevitable, want to keep errors in-house, unfamiliarity with a reporting system and analysis process, lack of feedback and time pressure, [Kessels-Habraken et al., 2010]. Moreover, lack of incident reporting is also caused by the time it takes to fill out the reporting form. The reporting procedure is more time-consuming for the healthcare staff that is inexperienced with the reporting procedure and if the person describes the incident with a lot of details, [Appendix A.2.3]. Therefore, ensure that the healthcare professionals are willing to report the incidents, the procedure requires to be simplified and less time-consuming, [Appendix A.1.8].

2.1.1 Incident Reporting System

An aim of the incident reporting system is to analyze and communicate knowledge about the incidents' causes. The IR system can deal with human, technical and organizational errors, which influence occurrence of the incidents during patient's treatment. Furthermore, the IR system must support quality development in the healthcare, and learning environment for the healthcare professionals, [Patientombuddet and for Sundhed og Forebyggelse, 2014]. The IR systems allows learning from the incidents, monitoring of the underlying trends and patterns, performing of the incident's processing and providing feedback of an accurate information about the reported incident, [Ahluwalia and Marriott, 2005]. The reports are gathered, analyzed and their results are used for learning from the reported incidents, [Sundhedsstyrelsen, 2011].

To provide a standard incident reporting form, a central IR system is established in hospitals, [Tighe et al., 2006]. In Denmark, it is the DPSD system, which aims to secure patient safety, [Sundhedsstyrelsen, 2011].

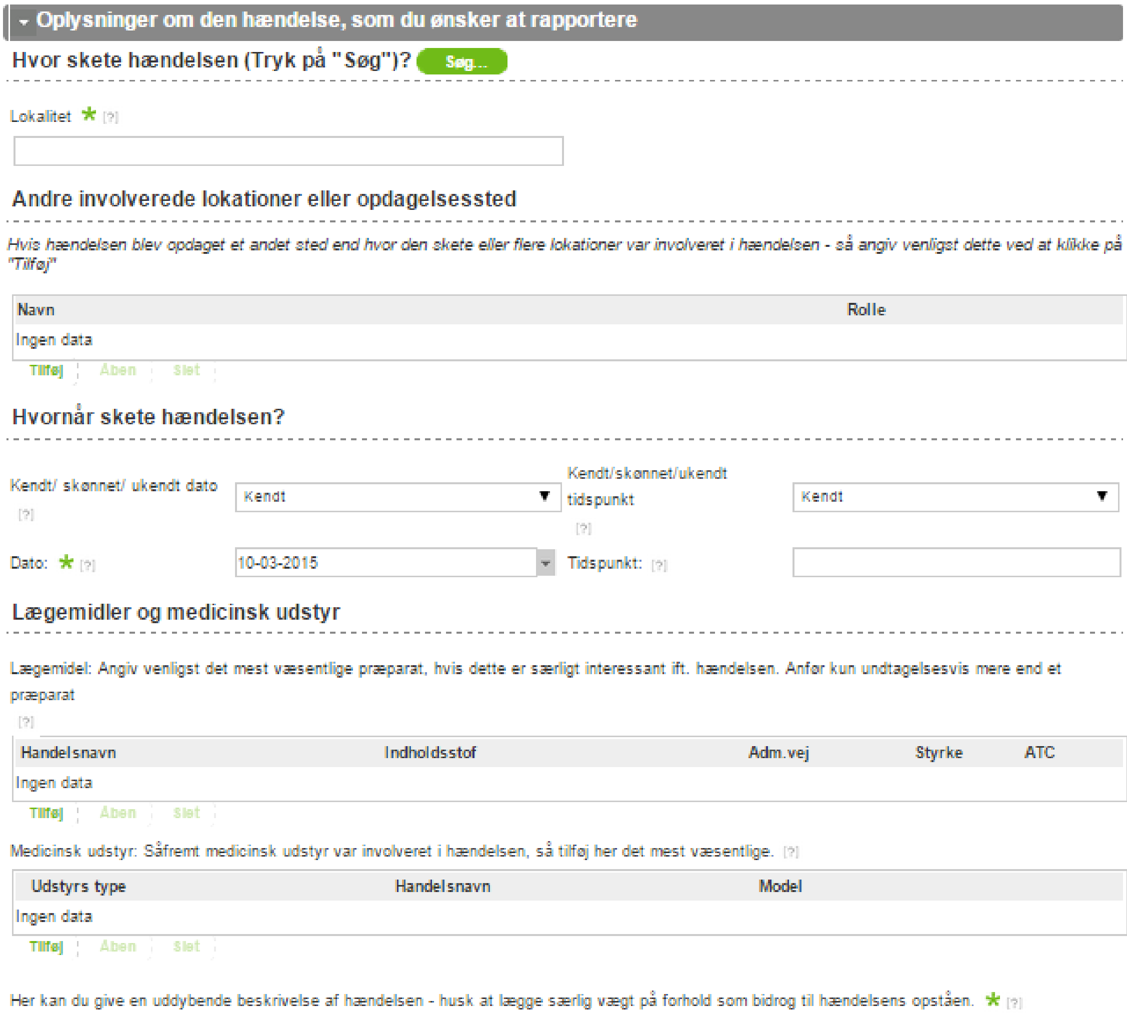
The IR system must justify the requirements for development of the healthcare quality and aid with creating an environment, where the incident can be analyzed and learned from, [Sundhedsstyrelsen, 2011]. It is required that the IR system is immediately available, easy and quick to complete to engage healthcare staff to report an incident, [Ahluwalia and Marriott, 2005]. Furthermore the IR system has to be, [Tighe et al., 2006]:

- Safe - Confidentiality of the incident reports is ensured
- Easy to use - A form for incident reporting have to be easy to use and be composed of tick-boxes and fields for free text
- Effective - Provides a feedback to the staff with newsletters and new policies

2.1.2 Reporting Procedure

The IR system in Denmark consists of the national DPSD system and database, [Patientombuddet and for Sundhed og Forebyggelse, 2014]. Reporting of the incidents to DPSD is performed online by all healthcare professionals, or by patients/relatives, [Kvalitetskontoret, 2011]. Reports from patients/relatives are however uncommon. Many of the reports from patients/relatives are from the ones that heard about or were in the incident program before, [Appendix A.1.8]. An interview with a patient, confirmed the statement, since the person did not know about the reporting possibility, or knew where and how to fill out an incident report, [Appendix A.3.1]. However, if the patients/relatives require help with reporting procedure, they can call a supervisor in Patientkontor/Patientdialog, whom helps them with filling the reporting form in DPSD, [Appendix A.1.8].

Danish healthcare system requires all incidents, which happened at a hospital to be reported latest seven days after their occurrence, [Kvalitetskontoret, 2011]. Additionally, an incident analysis has to be performed locally within a department where the incident occurred. A completed incident analysis requires to be sent to Patientombuddet latest 90 days after the incident report is received by a region, [Kvalitetskontoret, 2011]. However, due to ineffective organization or high load of the incident reports, it is possible that the incidents may become outdated, i.e. over 90 days old, [Appendix A.1.8].



Oplysninger om den hændelse, som du ønsker at rapportere

Hvor skete hændelsen (Tryk på "Søg")? Søg...

Lokalitet * [?]

Andre involverede lokationer eller opdagelsessted

Hvis hændelsen blev opdaget et andet sted end hvor den skete eller flere lokationer var involveret i hændelsen - så angiv venligst dette ved at klikke på "Tilføj"

Navn	Rolle
Ingen data	

Tilføj | Åben | Slet

Hvornår skete hændelsen?

Kendt/ skønnet/ ukendt dato [?] Kendt | Kendt/skønnet/ukendt tidspunkt [?] Kendt

Dato: * [?] 10-03-2015 | Tidspunkt: [?]

Lægemidler og medicinsk udstyr

Lægemidel: Angiv venligst det mest væsentlige præparat, hvis dette er særligt interessant ift. hændelsen. Anfør kun undtagelsesvis mere end et præparat [?]

Handelsnavn	Indholdsstof	Adm.vej	Styrke	ATC
Ingen data				

Tilføj | Åben | Slet

Medicinsk udstyr: Såfremt medicinsk udstyr var involveret i hændelsen, så tilføj her det mest væsentlige. [?]

Udstyrs type	Handelsnavn	Model
Ingen data		

Tilføj | Åben | Slet

Her kan du give en uddybende beskrivelse af hændelsen - husk at lægge særlig vægt på forhold som bidrog til hændelsens opståen. * [?]

Figure 2.2. A part of the IR form for healthcare professionals at DPSD website.

The incident reporting form at DPSD a reporting healthcare professionals, is shown in figure 2.2. The reporting form requires description of the incident with focus on what happened, when it happened, characterization of type and severity rate with description of consequences form the incident. Then, the user is asked to suggest a possible prevention method and to provide patient's information. At the end of the reporting form, there is a possibility to provide a contact information to the reporter, [Appendix A.1.4]. The reporter chooses who will receive the report, e.g. risk manager of a hospital, or a department where the incident occurred. The initial receiver verifies whether the copy of the incident report has to be forwarded, and the analysis of the incident report starts in the incident's target department at the hospital, [Kvalitetskontoret, 2011]. At Aalborg University Hospital (AalborgUH), there are 10 possible initial receivers, i.e. risk managers at AalborgUH and eight clinics, [Appendix A.1.1]. The eight clinics at AalborgUH are shown in figure 2.3.

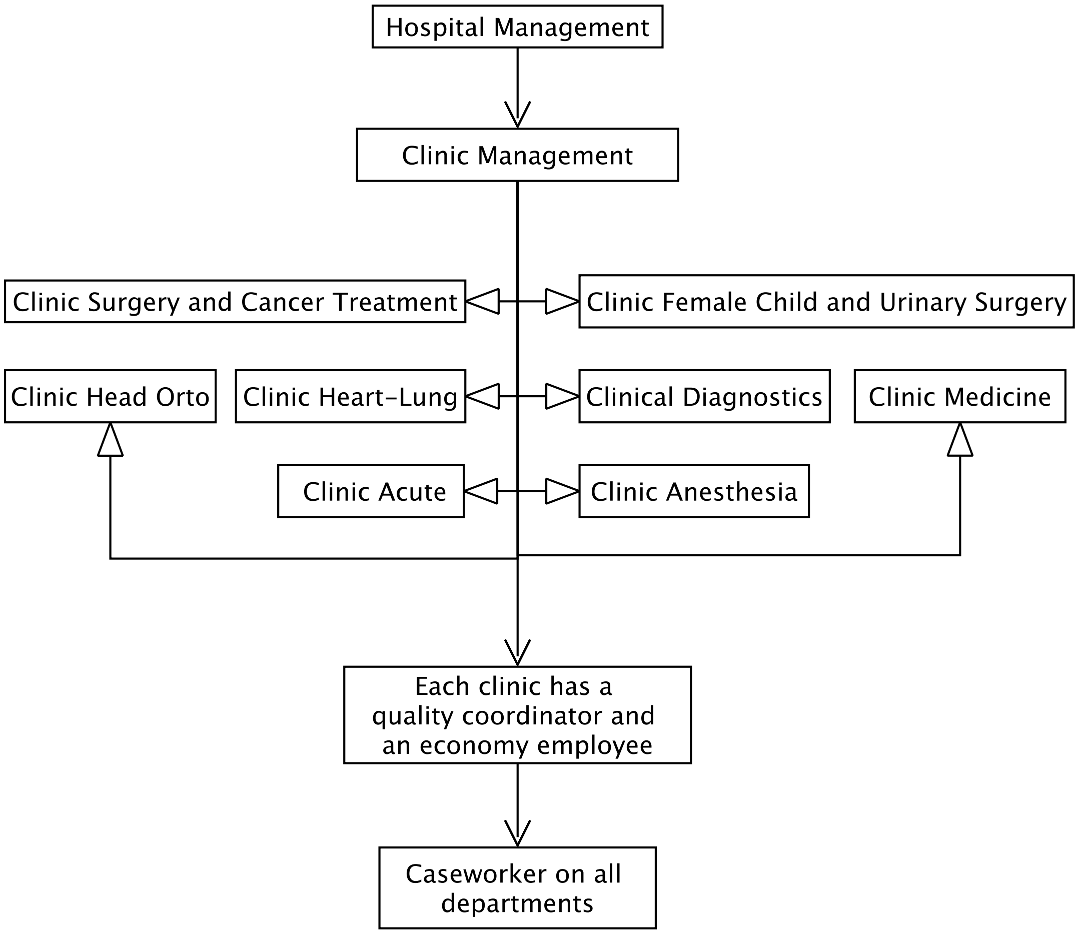


Figure 2.3. Work structure of the different departments, that processes the incident reports at AalborgUH, [Universitetshospital, 2014].

The incident is classified according to the type of incident and its contributing factors. It is also scored with likelihood of recurrence and degree of severity impact on the patient, [Tighe et al., 2006]. During the incident reporting, the reporters may choose to report as anonymous, or provide heir contact information. Anonymity may increase willingness to reporting, but reduces contact possibilities to the reporter in case of a need for further information about the incident, [Ahluwalia and Marriott, 2005].

Severity Rate

The severity rate is used to describe the actual consequence of the incident for the patient, [Appendix A.1.8]. The reporter provides the severity rate in the incident report, [Patientombuddet, 2014a]. Severity rate estimation consists of five levels described in table 2.1.

Severity rate	Consequences for the patient
No harm	No harm for patient.
Mild	Small temporary harm, which do not require increased treatment or care of patient
Moderate	Temporary harm, which require hospitalization, treatment from a general practitioner, increased care or treatment at a hospital
Serious	Permanent harm, which requires hospitalization, treatment from a general practitioner, increased care or treatment at a hospital, as other harm which require lifesaving treatment
Deadly	Patient's death.

Table 2.1. Five severity rates of the incidents in Denmark, [Patientombuddet, 2014a; Kvalitetskontoret, 2011].

There are some reporters who report the severity rate based on the potential outcome of the incident. It is also possible, that the reporter does not know the consequences of the incident while reporting, [Appendix A.1.8]. Therefore, the chosen severity rate is reviewed and verified by the risk managers at the chosen sector level. Based on their evaluation, the given severity rate can be adjusted and corrected to the more appropriate one, [Patientombuddet, 2014a].

In 2013, the highest prevalence of the incidents in the DPSD system had the incidents classified as no harm to patient, while incidents with severity rate defined as deadly had the lowest prevalence, [Patientombuddet and for Sundhed og Forebyggelse, 2014].

Incident Types

The incident types used for IR classification, are based on a WHO's international classification of incidents, [Patientombuddet, 2014a]. The incident types used for the DPSD system are presented in table 2.2.

Type of incident	Occurrence possibilities
Administrative procedures	Transitions of the patient between different sectors in healthcare, agreements and time reservation
Clinical Processes	Examination, diagnosis, treatment, rehabilitation and nursing
Communication and Documentation	Communication regarding patient between healthcare professionals
Infection	Obtained in contact with healthcare, such as sepsis or pneumonia.
Medication	Ordination, documentation, storage, dispensation etc..
Blood and Blood Components	Collection, testing, documentation, administration of blood and blood components etc.
Gases and oxygen for medical use	Prescribing, supply, documentation, administration, dispensation etc.
Medical devices	Use of medical devices for treatment of the patients.
Self-harm / suicide attempt / suicide	Situations where self-harm is present, patient attempts or commits suicide.
Patient Accidents	Incidents or accidents of a patient, e.g. fall.
Buildings and Infrastructure	Logistics, supply systems and transport, fire alarms etc.
Individual, Team and Organization	Insufficient resources or inadequate work organization.
Other	Other types of incidents, which cannot be classified i previous categories.

Table 2.2. Different types of incidents in the DPSD system, [Patientombuddet, 2014a; Sundhedsstyrelsen, 2011].

The types of incident can be used to identify tendencies and patterns in the reported incidents. Therefore, during incident reporting, the reporter is required to state an incident type. The incidnet type is further used for analyzing of the incidents and learning purposes, [Patientombuddet, 2014a]. In the DPSD system, the possible incident types change according to the chosen location of the incident, [Appendix A.1.4].

In 2013, the five most reported incident types were: medication, clinical processes, communication and documentation, administrative procedures and patient accidents, [Patientombuddet and for Sundhed og Forebyggelse, 2014].

Anonymity

During the reporting of an incident, contact details of the reporter may be provided, however are not required for submission of the report. Therefore, the reporter may choose to report an incident staying anonymous. Anonymity during reporting is a source of problems for the team responsible for processing and analyzing of the reported incident, because they cannot contact the reporter in case the report does not provide all necessary and relevant information regarding the incident, [Kvalitetskontoret, 2011]. However, all reports are made anonymous before sending them to Patientombuddet, since information about the reporter can be only disclosed to persons working on the incident report within the same region, commune, or at the same private hospital, e.g. disclosure of information may not be possible between a hospital and commune, or between hospital and another region, [Kvalitetskontoret, 2011]. At Patientombuddet, the only remaining information about the patient in the incident report are age and gender, [Appendix A.1.4].

Patient information, such as name or social security number, may be given further to another region, commune, private hospital or practice if required. Therefore, if an incident happens in e.g. Region North Jutland, the patient information may be given to other sections of the region or to other regions as well, [Kvalitetskontoret, 2011].

2.1.3 Analysis of the Incident Report

To reduce future risk of the incidnet's recurrence, an analysis of the incident report is performed. The aim of an incident analysis is to identify causative factors for the incident, asses risk for future recurrence, make recommendations for improvement, disseminate learning points and ensure to meet local and mandatory requirements, [Ahluwalia and Marriott, 2005].

In Denmark, the reported incidents may have an overlap between different sectors, e.g. between a region and a commune. The sector which receives the reported incident is responsible for analysis of the incident, closing the incident in the DPSD system and giving a feedback to the reporter, [Kvalitetskontoret, 2011]. The different sectors in Region North Jutland are presented in figure 2.4.

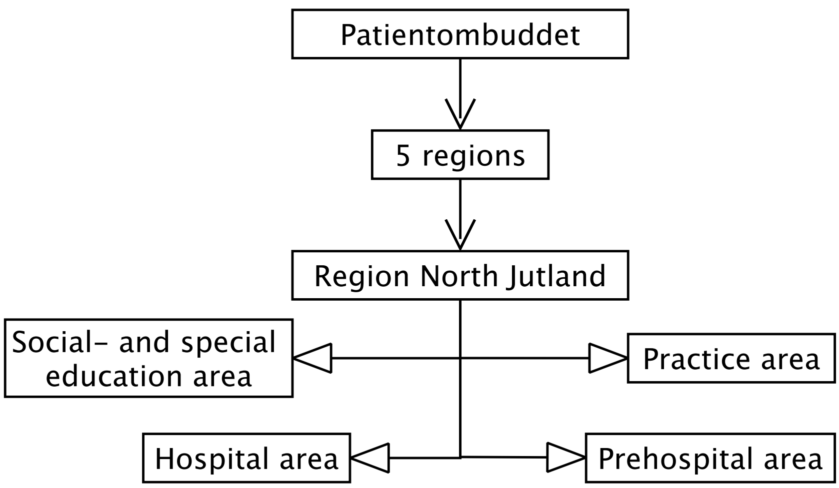


Figure 2.4. Work structure of the sectors analyzing the incident reports in Region North Jutland, [Kvalitetskontoret, 2011].

The overlap between the sectors can be considered in three main scenarios, [Kvalitetskontoret, 2011]:

1. An incident regards one sector, but was reported by another sector
2. An incident is faulty reported to a wrong sector
3. An incident involves several sectors

The incident is analyzed to identify *what happened*, *how it could happened* and *how to ensure that it will not happen again*. The analysis does not focus on who is guilty or responsible for the incident. Specific type of analysis is chosen by risk managers for analysis of the specific incident, [Kvalitetskontoret, 2011].

At a hospital, work on patient safety involves actors on different hospital levels and region. Each of the departments of the hospitals in Region North Jutland, has at least one person with a function as a key person for patient safety. At each of the hospitals in a region is a risk manager for the hospital. Such risk manager has insight into incident reports at the given hospital, and can locally coordinate work at patient safety. The regional levels have regional risk managers, which are responsible for education, development, implementation, administration of the users, coordination of the hospitals' work on patient safety both locally and between other regions, [Kvalitetskontoret, 2011].

Analysis of the incident reports is confidential and performed within boundaries of staff working on patient safety in a region, [Kvalitetskontoret, 2011]. Process of incident report analyzing is depicted in figure 2.5.

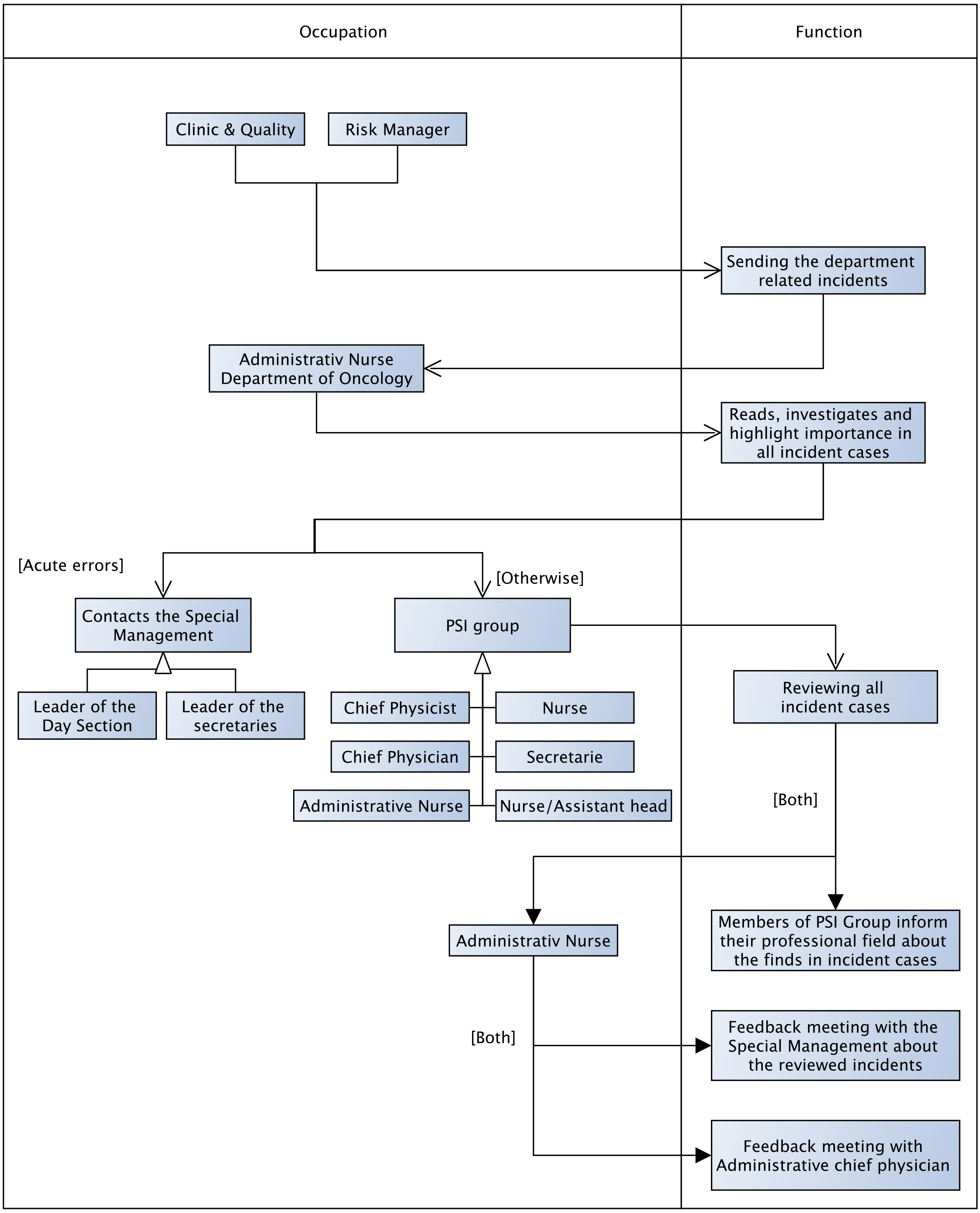


Figure 2.5. Working process of the incident report analyzing at the hospital department, based on the interviews with risk managers from appendix A.1.

The analysis method of the incident is decentralized for the whole hospital area, [Appendix A.1.5]. The hospital, where the incident occurred is responsible to choose a relevant analysis method. If the severity rate of the reported incident is classified as deadly, a root cause analysis is performed on the reported incident, [Kvalitetskontoret, 2011].

The initial receiver of the incident report checks the selected severity rate and classification of the incident. Additionally to each incident report, a description of the patient's hospitalization, prescriptions and notes are printed out. The relevant information for the incident are highlighted and presented during the meetings of the incident processing group. The incidents are presented on a big screen, where the processing group reads the incident reports. Sometimes information from Clinical Suite and OPUS are as well checked. Discovered problems from the incident reports are described according to the identified causes. The results of the analysis are marked and closed in DPSD and send to Patientombuddet, [Appendix A.1.5].

Results of the incident analysis are anonymous and send to administration of the involved department and hospital, [Kvalitetskontoret, 2011].

However, a research within patient safety area is challenging due to limitations of its methods. Examples of the methodological challenges include, [Shojania et al., 2001]:

1. Use of practices is evident to the participants and therefore they cannot be subjected to a double-blinded studies
2. All outcomes of near misses and actual harm are very difficult to be captured
3. An effective practice is multidimensional and therefore selecting a specific and working part of an intervention is challenging
4. Incidents generating highest concern are rare and representation of the improved safety procedure outcomes in a statistically meaningful manner cannot be made
5. Difficulty to establish links between the causes and incidents

2.2 Database

Patientombuddet is responsible for administration of the DPSD database, which is used for whole Denmark as a part of the DPSD reporting system to report the incidents, gather knowledge, identify causes of the incidents and learn from the reports, [Patientombuddet, 2014b; Patientombuddet and for Sundhed og Forebyggelse, 2014]. Based on the reported incidents and other available data in the database, Patientombuddet monitors healthcare system against patient safety issues, [Sundhedsstyrelsen, 2011].

The DPSD database was developed by a Canadian company, [Appendix A.1.8]. The currently implemented model of the DPSD database is a relational database. The DPSD database is divided into two main parts: application framework and information taxonomy, [Appendix A.4.1]. More details related to database design and structure are described in appendix A.4.1.

The DPSD data cannot be used for statistical or monitoring purposes, since it only reflects amount of reported incidents and not the real overview of the incidents' number or changes in patient safety, [Patientombuddet and for Sundhed og Forebyggelse, 2014].

When the DPSD was created, is was estimated that around 4.000 cases related to hospitals will be reported per year, but already during the first year 6.000 cases were reported. Since the same IR system is used by other sectors in healthcare, the number of cases is increasing. In 2013, over 180.000 cases of incidents were reported into DPSD, where over 110.000 cases were from the communal sectors. In 2014, the DPSD stored over 500.000 reported cases of incidents, [Patientombuddet and for Sundhed og Forebyggelse, 2014].

The DPSD system was developed to be used by one organization, however it has to cope with several units nowadays. It is challenging for the system, since it is used on a national scale in Denmark. Therefore, a work on framework is in progress, which should improve the DPSD database, [Appendix A.1.8].

Based on the DPSD data, it is possible to provide information about patterns in incidents to local patient safety initiatives in form of publications, which should give focus on specific areas. The knowledge should be used in local healthcare centers to improve quality and safety for the patients, [Patientombuddet, 2014b]. The data from the DPSD database is extracted using combination of the free text fields from the incident reporting form and provided classification details, e.g. incident type. The data can provide information to the management about the severity rate of the incidents and the working processes, which may caused the incidents, [Patientombuddet and for Sundhed og Forebyggelse, 2014].

Learning from the DPSD data is done to improve quality of patient safety in healthcare, [Patientombuddet, 2014b]. The DPSD data provides possibility to create a new knowledge about the unknown risk areas, causes and processes across all of the healthcare sectors. It is opposite to the traditional quality databases, which measure the risks only in the known areas, [Patientombuddet and for Sundhed og Forebyggelse, 2014].

Nevertheless, the DPSD data does not provide the whole spectrum of the safety related problems, and cannot be combined with local data, such as quality reports, to provide knowledge about the risks for the incidents' prevalence. Many of the incident types are underreported, such as ones happening over some time period, or the reports in the DPSD lack specification of the severity rates, [Patientombuddet and for Sundhed og Forebyggelse, 2014]. Statistics and data extraction from the DPSD database could be improved. Currently, the data search within the DPSD database can have a poor performance, to which precise causes remain unknown. Data extraction of specific sub-items from the database is also difficult, since, e.g. amount of the extracted data, may not be correct, [Appendix A.1.8]. Additionally, a study by Tighe et al. [2006] documented, that information stored in a reporting database may have several issues and weaknesses with reporting incidents, and limitations associated with information recorded on the database, [Tighe et al., 2006]. The issues may be related to, [Tighe et al., 2006]:

- Alternative classification of the incidents
- Inconsistent use of the categories in classification
- Duplication of the same incident

- Incomplete reports and missing information on incident, classification or contributory factors
- Lack of identification of the contributory factors in the majority of reports
- Extensive use of "other" classification type without specifying details masking serious problems

Information, which may lead to improvements in patient safety may be lost due to a lack of provided detailed information in the incident reports. All necessary information are often not available and incomplete at time of the incident reporting. However, it is still recommended to provide incomplete but correct data in the incident report, rather than complete but inaccurate, [Tighe et al., 2006].

2.3 Feedback

Improvement of the patient safety is obtained by reduction and prevention of future occurrences of the incidents, by establishing appropriate interventions and implementing them in healthcare, [Drupsteen et al., 2013]. Furthermore, it can be improved by enhancing incident reporting and by providing a feedback to the staff involved in the incident. The feedback can change healthcare professionals' risk perceptions, attitude and their behavior towards patients safety, [Kessels-Habraken et al., 2010].

Lack of feedback is the main reason for underreporting of incidents in a healthcare system, [Benn et al., 2009]. A study by Mahajan [2010] reported that 2/3 of the respondents used for the study, named the lack of feedback as the main obstacle for not reporting of the incidents, [Mahajan, 2010]. However, the reporting rate of the incidents increases by using newsletters or other feedback methods, [Mahajan, 2010; Benn et al., 2009].

Effectiveness of the given feedback depends on the data feedback's organizational context, data quality and timeliness, [Bradley et al., 2004]. A study Wallace et al. [2009] described five types of feedback for IR system. The feedback types are, [Wallace et al., 2009]:

1. Bounce back feedback: Provides information to the reporter, team or service in form of an automated response, telephone debriefing etc.
2. Rapid response feedback: An action is taken within the local work environments against serious and immediate threats to patient safety
3. Raise risk awareness feedback: Provides information to all front-line staff in form of publications, online bulletins, newsletters, etc.
4. Inform staff about taken actions feedback: Information is given back to the reporting community and reporter about actions to been taken and achieved progress based on the incident reports
5. Improve work systems safety feedback: Actions are taken within local work systems by implementing and developing measures to improve the work flow based on the reported issue

After an incident report analysis, lessons and guidance have to be given back to the individuals and department involved in the incident, [Ahluwalia and Marriott, 2005]. Therefore, feedback mechanisms may include regular or multidisciplinary meetings of the staff responsible for incident analysis, bulletins such as paper-based, individual e-mail bulletins or updates posted on a department's website. Furthermore, feedback may be given through targeted campaigns for a particular incident, or patterns leading to the incident, [Ahluwalia and Marriott, 2005].

Forms of the effective feedback for the incidents are described in a study by Benn et al. [2009]. To achieve the most beneficiary feedback on the incidents, the study suggested to incorporate multiple modes of both information and action processes. The given feedback must include corrective safety actions and be closed in order to ensure, that the vulnerabilities identified from reporting, analysis and investigation are timely and correctly addressed in the environment at the healthcare, [Benn et al., 2009].

In the past, there was a requirement to provide feedback to the reporter of the incident, but nowadays the requirement is difficult to be fulfilled. Therefore, a feedback is not expected, and a direct feedback to the reporter is not always provided, [Appendix A.1.7].

The responsibility to provide feedback is a staff's responsibility in a chosen form, e.g. the incident's outcome is discussed personally. Moreover, bulletins are published four times a year providing information about an incident group and changes made based on the incidents. A statistical overview can be as well provided, [Appendix A.1.7]. However, bulletins produced by local processing groups are read by the nurses rather, than the ones produced by the Patientombuddet or hospital's risk managers, [Appendix A.2.3].

It is important that the given feedback will be provided in a timely and balanced manner, and that whole team is involved in seeking solution to an incident, [Ahluwalia and Marriott, 2005]. Providing of balanced information is crucial, since an overload of information exists in healthcare in form of e.g. increasing numbers of journals and guidelines, which must be read and learned as a large number of patient data that need to be considered, [Hall and Walton, 2004]. The given information to the doctors and nurses therefore have to be short and concise. The balance between knowledge and time must be obtained, so that the healthcare professionals can learn from the given information, [Appendix A.1.8].

Impact of the taken action on the patient safety should be monitored and its results published to provide a feedback on patterns of the incident data to detect potential risk problems in healthcare, [Wallace et al., 2009].

2.4 Problems with the DPSD System

The analysis of the DPSD system described in previous sections, highlights problems and challenges with the DPSD system related to a variety of areas. However, some of the described problems in the problem analysis are not possible to be solved within this project, since they relate to other factors that are not directly linked to the DPSD system, such as lack of patients'/relatives' knowledge of the possibility to report incidents, only the same staff reporting the incidents and occupational health perspective among healthcare staff. Therefore, based on the described aspects and problems derived from the interviews and scientific literature dealing with the incident reporting systems, three main focus areas in this report have been identified as incident reporting, lack of feedback from the incidents, and incident data extraction and analysis.

2.4.1 Incident Reporting Area

The incident reporting area includes aspects related to reporting of the incidents with the DPSD system. It both describes reporting process as well as problems related to the reporting form in the DPSD system.

Problems related to the incident reporting area, gathered from the analysis are:

- Short incident description increases probability that the incident will be dismissed
- Lack of patient's and reporter's details in the report increases the risk of dismissing incident reports
- Possibility to define the gender in various ways, e.g. using 'K' and 'k', results in saving these designations differently in the DPSD database
- Lack of provided information in the incident reports requires data search in other healthcare related databases
- The information must be short and precise to obtain a balance between knowledge and time
- Lack of incident reporting
- Reporting of the incidents is time consuming
- Fear of incident reporting
- Lack of incident reporting examples

Moreover, several comments related to the design of the reporting form in the DPSD system were described in a report by Patientombuddet [2013]. Some of the provided comments regrading description of the incidents in the DPSD system point out lack of guidelines and help with filling the report form, which would ensure that the incidents will not be dismissed. Since in the existing reporting form, the description of the incident is divided into three fields, is also suggested to combine them into one description field, [Patientombuddet, 2013]. The title of the incident report could be based on the chosen classification of the incident, instead of requiring the reporter to write own title of the report. By doing so, search for the incidents could be optimized, [Patientombuddet, 2013]. The severity rate of the incident is often evaluated based on what could have happened, instead of what actually happened. Furthermore, the chosen severity rate could be evaluated too low. Therefore, an improvement to aid the reporter in choosing the severity rate is needed, [Patientombuddet, 2013]. An idea was to create a questionnaire allowing to assign correct severity to the report. However, the idea was evaluated as too time consuming for the user, since it would require more steps to choose the correct severity rate, [Patientombuddet, 2014c]. There are several problems with defining location of where the incident happened. In order to improve the search for a location, SOR codes identifying the different locations could be used, [Patientombuddet, 2013]. The possibility to choose the date works well in the DPSD system, however it is not possible to mark, that the date is unknown, [Patientombuddet, 2013]. The description of medications and medical devices is an important function to have. However, not many users know how to use them, since they are difficult to understand, [Patientombuddet, 2013]. At the end, most of the users of the DPSD commented that knowledge of how the incident report is processed, and how the different information are used, would help to understand what is important in filling the reporting form, [Patientombuddet, 2013].

2.4.2 Incident Feedback Area

The incident feedback area focuses on the problems related to feedback on the incident reports. The area includes both forms and procedures of providing feedback to the healthcare professionals.

The identified problems related to the feedback on incident reports are:

- Lack of personal feedback on healthcare staff's incident reports
- Lack of feedback is the reason for underreporting
- Providing feedback on the processed incident report is not a requirement
- Not all of the provided feedback forms are read by healthcare professionals

2.4.3 Analysis and Data Extraction Area

The analysis and data extraction area focuses on learning possibilities from the incident reports. Therefore, it includes both analysis of the incident reports and extraction of the data from the DPSD database for feedback.

The identified problems to the analysis and data extraction area are:

- Too many incident reports in proportion to the time available for their analysis
- Performance problems during data search
- Extraction of statistics is difficult and time consuming
- Extraction of specific sub-items, e.g. in location, is difficult
- Search for specific data is difficult and time consuming
- Questioning technique for the incident description, which has focused on increasing learning

2.4.4 Summary of Problem Areas

Amount of identified problems, described in sections 2.4.1, 2.4.2 and 2.4.3, vary between the different areas of the DPSD system. However, the three problem areas related to the existing DPSD system in Denmark, are crucial to be solved, since they deal with the procedures used in incident reporting, analyzing and evaluating of the reported incidents.

Therefore, those procedures are important ones, since they are the ones contributing to improvements in patient safety, feedback and awareness regarding the incident reports.

CHAPTER 3

AIM OF STUDY

The number of reported incidents to the DPSD system increases every year due to the broader application and utilization of the system. The DPSD is an online web system, where healthcare professionals, patients and relatives can report an incident. The reports are manually analyzed and classified with respect to what information has been reported in the incident report. The analysis of the reports contributes to identify causative factors and creation of future recommendations. The obtained knowledge from the incident reports could aid prevention of future errors, thus improve both safety and quality of healthcare services. Improvement of safety and quality was done by Patientombuddet, where the reported incident data was further analyzed to find patterns in the incidents on a national plane.

However, several issues related to the incident reporting systems have been highlighted concerning problems among the healthcare professionals, the DPSD system itself, and feedback from the reported incidents. The current reporting form at DPSD was regarded as time consuming, lacking guidance, and not clear enough for some of the healthcare professionals. The access to data exploration and statistical tools is limited to risk managers, whom have a high workload of tasks related to analyzing and processing of the incident reports within a limited time range. Extraction of the incident data was time consuming due to large amount of data in the DPSD database, and it may not always include all of the required incident data. Additionally, the learning possibilities are reduced, since the provided feedback on the reported incidents was not read by many of the healthcare professionals due to overflow of information in the healthcare system.

The summary of the problem analysis leads to the following problem formulation:

> *How could a system aiding reporting procedure and supporting feedback to healthcare professionals for the Danish healthcare system be designed, developed and validated?*

3.1 Focus of the Study

The study focused on development of an optimized IR system and database to solve the identified problem areas related to incident reporting, analysis and data extraction, and feedback on the incident reports.

The developed IR system was based on the DPSD system, and aimed to provide an easier, more intuitive and less time consuming reporting form, which would reduce healthcare professionals' fear and aversion for the incident reporting. Additionally, means for statistical analysis and data extraction of the incident data in the IR system were added. The means aimed to provide aid for feedback, analyzing and processing methods of the incident data. To maintain anonymity, enhance security and privacy of the incident data, a login function was required to aid with verification of the users' access rights to the different interfaces in the IR system.

Due to the identified problems with data extraction from the DPSD database, a new database structure was required to aid data extraction and monitoring of the incident reports by the risk managers and Patientombuddet. Furthermore, optimization of the database design aimed to provide the users with an efficient and accessible way to investigate reported incidents.

The chosen form for the IR system was a web-interface to match the current, online DPSD system. Furthermore, the web-interface provided a broad and quick accessibility for healthcare professionals, patients and relative to the IR system. Since most of the reports and problems were identified in the healthcare system, the study focused only on the healthcare professionals and risk managers in the hospital area.

Part II

Problem Solution

CHAPTER 4

SOLUTION STRATEGY

Within this chapter an explanation of how the described methods in appendix B were utilized during the system development was provided. The described solution strategy for the system development was applied during the problem solution part.

The problem statement described in chapter 3 provided a scope for the developed solution. In order to ensure that the proposed solution is well documented and of good quality, several methods described in appendix B for documentation of the proposed solution were used.

4.1 Interview

To gain a background knowledge and better understanding of daily challenges, problems, and work on incidents in Region North Jutland, several interviews on different organizational levels were performed according to methods described in appendix B.2. The results of the interviews were provided in appendix A, which contributed to the problem analysis and solution.
The obtained information from the interviews were compared with existing scientific literature regarding incident reporting systems and guidelines used within the Danish healthcare system. The results of the comparisons were provided within this report.

The selection of the interviewees was based on their competences and references from the different organizational level subjects within the incident reporting management. The references helped to ensure the trustworthiness of the interviewees. To achieve a quality in the obtained knowledge, the interviews execution and guide was develop based upon the theory described in appendix B.2.1.

The chosen levels in the healthcare system included Patientombuddet, Region North Jutland, Risk Management at Aalborg University Hospital and Department of Oncology at the Aalborg University Hospital, due to the different responsibilities for processing incident reports. The interview was performed in person and varied 30-45 minutes.

Additionally, an interview with a Canadian company - RL Solutions, was conducted regarding the DPSD database development to learn about the DPSD database structure, used models, and to identify possible optimization areas. The interview was executed using as an email correspondence due to the time zone differences and incompatibility for the planned meetings. The information from the interview were provided in appendix A.4.

To ensure credibility and confidentiality that the interviewees would not be expose, all the names of the interviewed subjects were removed. Instead, their job positions or associations were used to indicate their statements during analysis of the interviews provided in appendix A.

Prior to each in person interview, the interviewees were informed, both on text and verbally, about the project background and aim to provide them with background knowledge about the project group for the purpose of the interview. The interview questions were adjusted to fit the interviewee's area of expertise, and to receive the most truthful answer. The interview questions were designed as open questions to not reduce interviewee's opinions to "yes" or "no", and to give them freedom to answer freely and deeper from their point of view.

The interviews were translated from Danish to English, where the project group ensured that the translations were true with original information provided by the interviewees, and that their opinions were properly expressed. To focus and highlight the relevant information, the obtained information were grouped according to the focus points, and all the information not related to the researched subject was excluded. The transcription of the information created clear statements, since the interviewees sometimes got carried away under the interviews. Afterward, meaning condensation analysis was performed to clearly present the information given by the interviewees and to eliminated spoken form of language used by the interviewees. The sentences were therefore rephrased, which resulted in concise and precise information.

4.2 System Development

Elaboration of the proposed proof of concept IR system and database was described and presented to present an overview of the IR system capabilities, possible implementation and application in the Danish healthcare system. The description was used as a start point for the further documentation of the IR system and IR database, since it provided the focus points for the development process and highlighted the possibilities of the system. The described elements of the IR system and database development are based on table B.2 in appendix B.3.

4.2.1 Requirements

Based on the gathered knowledge from problem analysis and system description, the functionality of the IR system and IR database were specified to describe its functional and non-functional requirements as described in appendix B.3.2, to get an overview of the IR system and IR database capacity and features. The IR system functionality and behavior were presented using UML elements, such as use case diagram, use case specifications and activity diagrams. The use of UML, allowed to demonstrate the users' possible interactions with the IR system and the IR system limitations.

4.2.2 Analysis and Design

The functionality was analyzed to identify the design elements required during the design phase. Based on the results of the analysis, principles for interface design describe in appendix B.3.3 and B.3.3, initial mock-ups were created to illustrate the structure of the IR system. The usability theory and the theory principles for interface design were used to ensure a good design quality, user experience and easy to use interface of the IR system.

Analysis of the requirements for the IR databases, was translated into database design elements. The proposed design for the IR database was based on the theory for a database design from appendix B.4, Silberschatz et al. [2011] and the mail interview with the Canadian company, which was provided in appendix A.4. The IR database design focused on ensuring quality, robustness, improve performance of the database and structure of the database elements.

4.2.3 Implementation

The IR system and IR database described in section 4.2.2, were implemented using implementation tools for web-development and database implementation. The choices regarding the implemented functionality and relationships between the elements were explained to aided understanding of the chosen solution methods and implemented functionality. Furthermore, it provided an overview of the developed IR system and IR database, its abilities, functionality and limitations.

4.2.4 Test

The implemented IR system and IR database were tested to verify the functionality of the system and database, furthermore to identify possible errors in the IR system and IR database. The functionality test was performed to verify the IR system and IR database before validation by the prospective users. It was done to avoid the breakdown of the system and database due to unexpected errors, which could confuse and stress the validation participants. Since the UP is an iterative process, the tests were performed doing the system's and database's development. However, they were not described in the report, since they were used to identify minor functionality errors in development requiring immediate attention and correction.

4.2.5 Validation

The interfaces were validated by the prospective users to verify the implemented IR system's requirements and design choices according to the theory described in appendix B.3.5. Based upon the chosen validation from appendix B.3.5, the validation was performed as an usability test with the healthcare professionals, whom were considered as the prospective users of the system. The usability test was chosen, since the goal of the validation was to evaluate how the systems' design and functionality would be perceived by the users. The correct design choices were considered of lesser importance, due to project group's focus on system's functionality, such as an improvement of incident data input, data manipulation and extraction with possible improvements in learning and feedback on the incident data. Therefore, a heuristic evaluation was not performed. The performed validation of the system provided different types of feedback from the participants, which consisted of positive and negative feedback, and identification for possible missing features in the system. The IR database was validated with the existing DPSD database to verify advantages and disadvantages between the databases by comparing the test results. The IR database validation was performed to verify if the IR database was a proper solution for a new DPSD database design.

4.3 Synthesis

The used methods during the solution and obtained validation results were further discussed to compare the proposed system to existing solutions, identify other possible applications and features of the system, verify the used methods and described the limitations of the described study. Based on the discussed aspects, a conclusion was made to sum the study described within this report, and to conclude on the obtained solution and results.

CHAPTER 5

SYSTEM DEVELOPMENT

Within this chapter a description of the system development processes were provided. Based on the UP theory, the requirements, analysis and design, implementation and tests of the system were provided.

The aim of the study was to develop a web-based interface for healthcare professionals for incident reporting, and extraction of specific incident reports to investigate, analyze and provide feedback on the incident data. The incident reports may provide risk managers with data of possible improvements in the healthcare system. Thereby, these incidents can be used to implement solutions that will prevent further occurrences of incidents, and ensure quality and safety of the patients in the healthcare system.

The web-based interface functionality have the opportunity to provide healthcare professionals with a visual overview and data exploration of the reported incidents, which hopefully would increase the frequency of the incoming incident reports. Additionally, the risk managers at the hospital, regional and national level have the ability to monitor and find alarming outcomes of the incidents. The visualization of the incidents allows the user to select the type of incident, severity rate and time range to investigate and analyze trends and patterns. Furthermore, by extracting specific information from the incident reports, relevant statistical methods may be implemented in the healthcare system, subsequently sharing the results with other healthcare system.

The web development tools are chosen due to the already well established incident reporting DPSD system, and its board accessibility. The web system does not have specific requirements, accessibility is quick and easy, and can be available on any device at any time. It would give the user the freedom to choose where and when they would like to report an incident.

The IR system is designed in a way that simultaneously considers usability theory, principles of good interface and screen design, required information in the incident reporting form, and similarity to the existing DPSD system.

The IR system consists of three interfaces. Within the first interface, the user can report an occurred incident with detailed information. The second interface is for the visualization and exploration of the register incident data, in which the user has the opportunity to interact with the interface and investigate the number of incidents in various options and different visualization methods. The third interface provdes the access and ability to investigate and analyze the incoming incident data to find solutions to the reported incidents and implement the solutions at e.g. hospital, region, national and international level.

To accommodate incident data, and satisfy safety and privacy restriction requirements for registering of the incident data, the web-based interface requires identification of the user. The user identification is needed to verify if the user has the requested clearance to access the incident data. Therefore, the second and third interfaces are accessible for a specific selection of authorized healthcare professionals.

5.1 Requirements

To represent functionality of the developed IR system, a set of requirements is defined based on the knowledge obtained from the literature search and the conducted interviews. Based on theory described in section B.3.2, the requirements for the IR system and IR database are divided into functional and non-functional requirements, and prioritized using MoSCoW theory.

Functional Requirements

- The system must make incident reporting possible
- The system must allow search for location, patient, healthcare professional, medication and medical devices
- The system must include a login function
- The system must improve the database performance
- The system should make precised extraction of selected incident data
- The system should visualize the selected incident data
- The system should make it easy to apply changes in the database
- The system should process large amounts of data from the database
- The system could let the content of possible processes be dependent on chosen incident types
- The system wants to have an incident reporting scheme adjustable to either healthcare professionals or patients/relatives

Non-Functional Requirements

- The system must make connections to the required databases
- The system must consist of a new database adjusted to the system
- The system must provide a robust construction of the databases

- The system must be developed using Danish language
- The system should be based on theories related to proper and usable design of an interface
- The system should be able to work properly on different web-browsers

5.1.1 Use Case Diagram and Specifications

To provide an overview of user's possible interactions with the IR system, and present system's limitations, use case diagram and specifications were used. The use case diagram for the developed IR system is presented in figure 5.1. Within the diagram, three generalizations are used for actors and for `Statistics` and `Extract Data` use cases. The generalization of the actors indicates that the different groups of the healthcare professionals have an equal possibility to access the IR system. Since the modeling of the system is limited to healthcare professionals, the patients/relatives interactions are not shown in the use case diagram. Furthermore, the `Search` use case is a generalization for possible search functions and describes the search for `Medication and Medical Devices`, `Patient`, `Location` and `Healthcare Professionals` use cases.

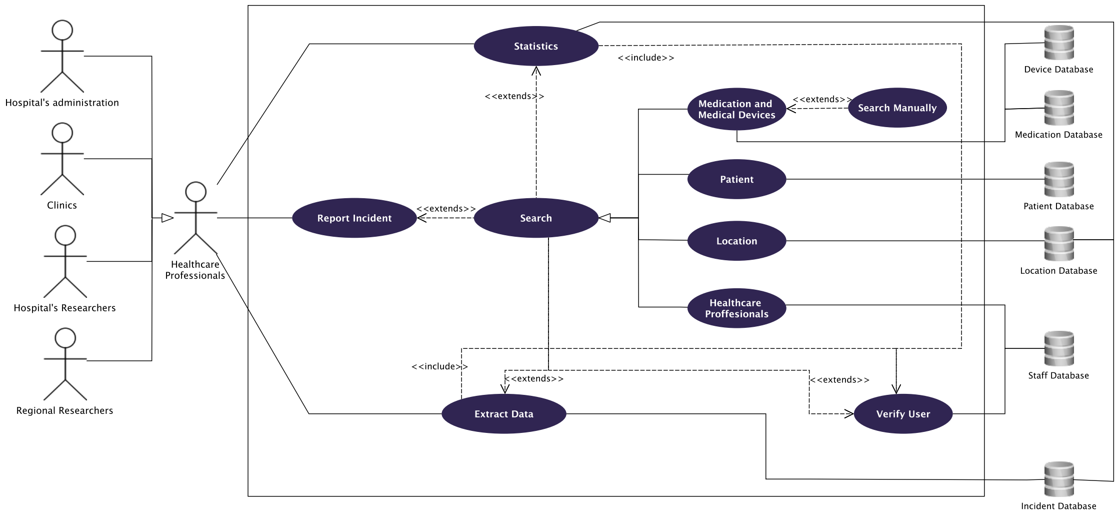


Figure 5.1. Use case diagram for the developed IR system.

In the presented use case diagram, the `«include»` describes the relationship of `Statistics` and `Extract Data` to `Verify User`. The relationship in the use cases is used to indicate, that inclusion use case `Verify User` is required from the actor to access the `Statistics` and `Extract Data` use cases.

The `«extend»` is used to describe relationships in the `Report Incident` use case, because during reporting of the incident not all information are necessary to be provided by the actor. Therefore, the `Search` use case is an extension use case to the `Report Incident`, `Statistics` and `Extract Data` use cases. The `Search Manually` use case is an extension use case to `Search` use case, since it is possible for the user to access the search for medication and medical devices manually.

Most of the identified use cases interact with specific IR databases. The Device Database and Medication Database are used within the Medication and Medical Devices use case, since from this use case search in the databases is possible. The Patient Database includes details about the patients and is searched from the Patient use case. The Staff Database stores details about the healthcare professionals and is accessed from the Healthcare Professionals and Verify User use cases. The Location Database provides details about the possible locations and is accesses from Statistics, Search Location and Extract Data use cases. The Incident Database stores entered information regarding the reported incidents and their status, and is accessed from Statistics and Extract Data use cases.

The use cases illustrated in figure 5.1 are further described and specified with a use case specification. Within this chapter only use case Report Incident and corresponding activity diagram are presented, since they represent most of the IR system functionality. The rest of the use cases and activity diagrams are described in appendix C.

Use case Report Incident

The use case Report Incident represents the situation, where the actor is reporting an incident to the IR system. Within the use case, the actor fills the required details about the incident. Additional details required during reporting can be provided using Search use case. The use case specifications for the Report Incident use case are described in table 5.1.

Use case name	`Report Incident`
Use case ID	UC1
Description	Allows the actor to report an incident to the IR system.
Primary actor	Healthcare Professionals
Secondary actor	Incident Database
Pre-conditions	The *Report Incident* tab in the IR system is open. Connection to the Internet and the database is established.
Main flow	1) Actor chooses as whom to report the incident. 2) Actor specifies the severity rate of the incident. 3) Actor selects a type of incident best suited to the reported incident. 4) Actor selects a process leading to the reported incident. 5) Actor describes *what* and *how* the incident happened. 6) Actor chooses the date, that the reported incident happened. 7) Actor selects a location of where the reported incident happened. 8) Actor submits the incident.
Post-conditions	The incident is submitted to the DPSD database.
Alternative flow	1.a) If the actor selects *Healthcare Professionals* button: i) The actor fills the reporting form required from a healthcare professional. 7.a) The use case `Search` is initiated. 8.a) The actor can choose to print the reporting form. 8.b) The actor can choose to reset the reporting form. 9.a) The actor can provide information about the medication and medical devices involved in the incident. i) The use case `Search` is initiated. 10.a) The actor can fill the patient data, by searching after the patient's CPR number in the Patient database. i) The use case `Search` is initiated. 11.a) The actor can fill the healthcare professional data, by searching after the staff ID number in the Staff database. i) The use case `Search` is initiated.

Table 5.1. Use case specification for `Report Incident`.

The activity flow of the `Report Incident` use case is illustrated in activity diagram in figure 5.2, where the user fills the incident form with information describing the incident.

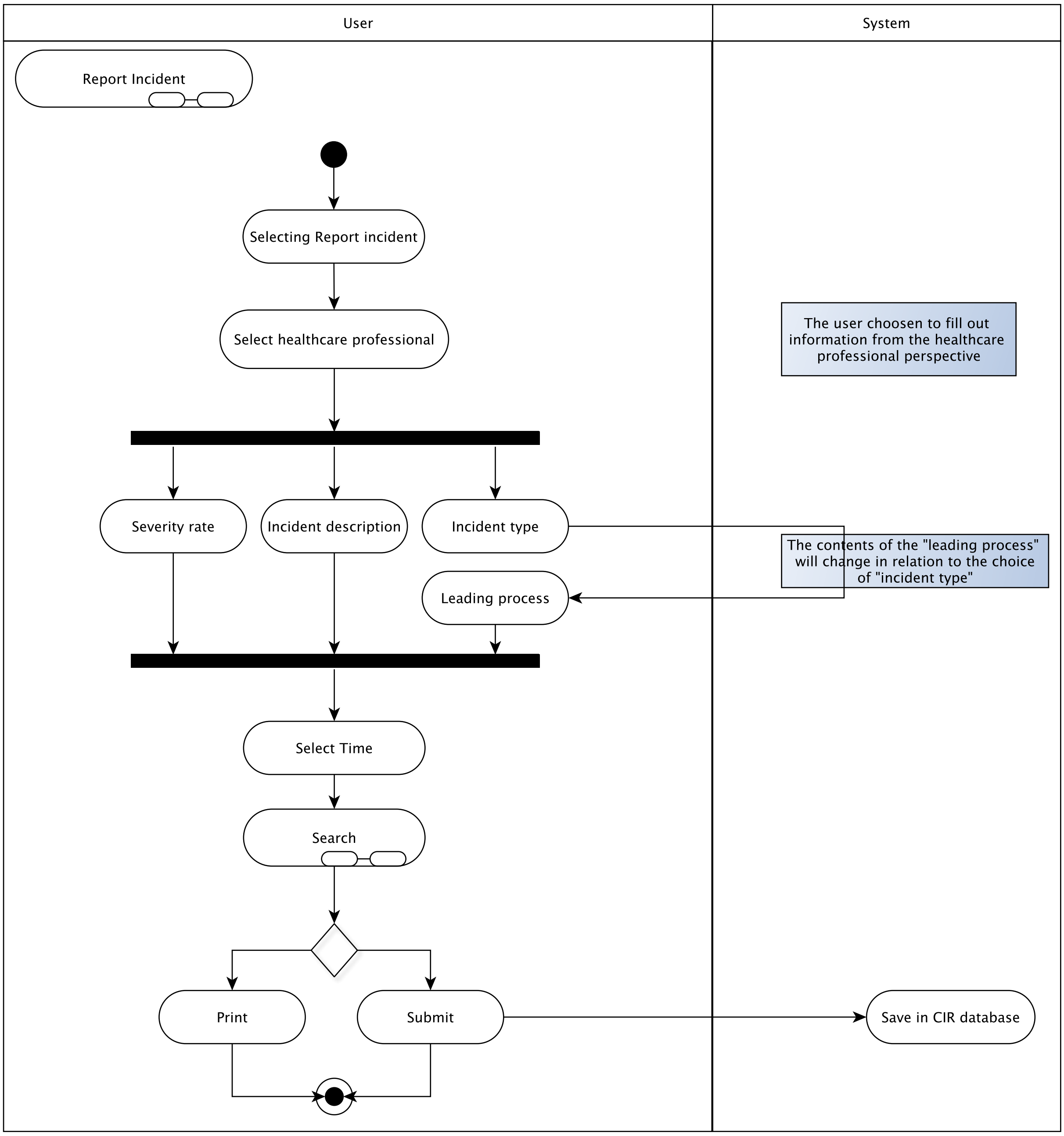


Figure 5.2.

5.2 Analysis and Design

The presented requirements in section 5.1 were analyzed and translated into design elements. Therefore, within this section, design of the IR system and IR database structure was described.

5.2.1 Design of the IR System

The design behind the IR system was presented in form of mock-ups and described, to demonstrate and explain the functionality of the proposed interfaces. The figure 5.3 presents the structure of the interfaces in the IR system.

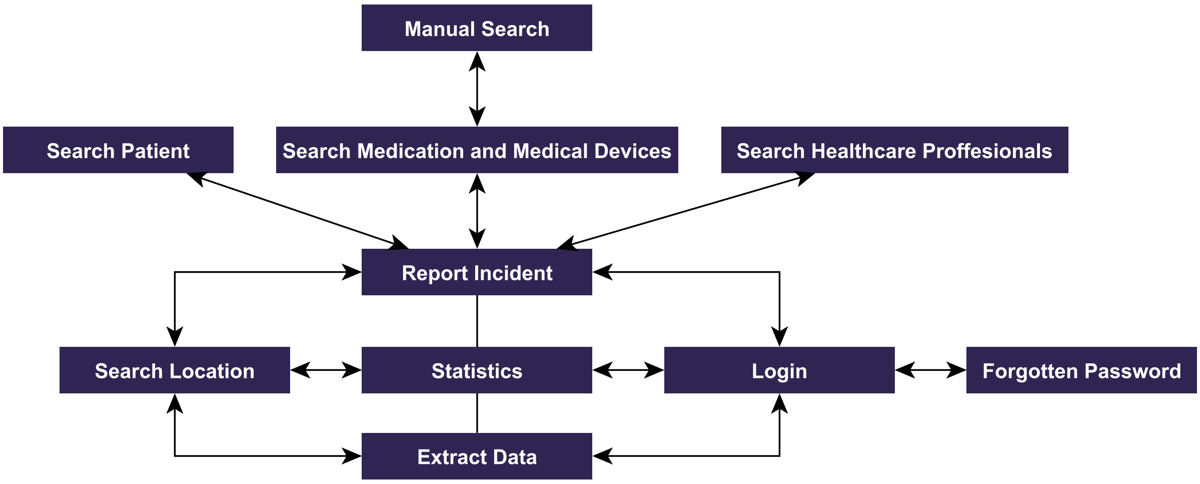


Figure 5.3. Structure of the interfaces in the IR system.

The choices of design for mock-ups were based upon the literature search, performed interviews, usability and interface/screen design requirements described in appendix B. The used language in the interfaces is an important factor, since it aids the user with understanding of the provided information and utilization of the IR system. Since the IR system has been developed as a supplement, or a possible replacement for the existing DPSD system, the used language was Danish. The written formulations were based on the current information in the DPSD system, and delivered in the IR system in a familiar and consistent way for the healthcare professionals.

To allow the IR system to be used by both experienced and inexperienced users, the system had to achieve an intuitive and user-friendly design. A simplicity, unity and predictability of the designed interfaces, were important factors in obtaining a clear, feasible and intuitive navigation. Additionally, a clear design could reduce errors and anxiety of the users during their use of the system and result in easier implementation of the IR system in the Danish healthcare system.

Mock-up: `Report Incident` Interface

The interface `Report Incident` represents the website which will meet the user visiting the IR system. The mock-up for the interface is shown in figure 5.4. The interface introduces a welcome message to ensure the users, that they are on the correct website. A short description of incident reporting concept with definition for an incident is provided to explain to the user how to complete the reporting form. The provided text would as well include an indication of the hover bubble *Help* functions, which would be attached to each of the design elements, to aid the user and eliminate doubt regarding a function of the incident form elements.

To ensure that the user is not confused about the presented information and the incident form, the only information that will be visible to the user is the information shown above the dotted line. Depending on whether the user chooses *Patient/Relatives* or *Healthcare Professionals* buttons, the content of the incident report form would change. The change in the reporting form is required, since the healthcare professionals know more precise information regarding the circumstances leading to the incident, than the patients or relatives.

The presented system is a proof of concept with focus on healthcare professionals. Therefore, the information displayed after the dotted line represents the information to be completed by the user, if the user selects the *Healthcare Professionals* button. The focus area is chosen due to the larger amount of incident reports being reported by the healthcare professionals, and it is where the improvement of the efficiency in incident reporting is missing. The points the healthcare professional has the opportunity to fill out are:

- Severity rate of the incident
- Incident type
- Process leading to the incident based on the selected incident type
- Description of the incident
- Occurrence time of the incident
- Location of the incident
- Medication and medical devices involved in the incident
- Patient information involved during the incident
- Reporting healthcare professional's contact information

The content of the incident reporting contains the same information as the current incident reporting form in the DPSD system, but suggests a more clear and structured design. The current DPSD website might be confusing by requiring a high amount of a very specific information regarding the incident from the user, which the user often does not have at the time of the reporting. In order to comply with the requirements for good interface/screen design, all the unnecessary information has been removed and typing requirements reduced. The order of the completion of the incident form was changed to make a more user friendly website, and to guide the user naturally throughout the different steps in the incident reporting form.

The severity rates, definition of the incident types and the processes leading to the incident have been left unchanged compared to the original DPSD system, since the radio buttons provide a good and clear overview of the various choices. The three steps have been moved up in the suggested incident reporting form as the importance of the steps is essential for the report to be analyzed properly. In the new edition of the DPSD system, the Patientombuddet wanted to create several questions for the severity rate instead of the five choices, as described in section 2.4.1. The solution was not chosen in the report, since the user would have to spend even more time on filling out the incident report form.

The free text description is a field, where the user has the freedom to write how the incident took place. There are no recommendations about what the description must contain, or how deep the incident should be documented. However, there are restrictions to the minimal length of the incident description, as described in section 2.4.1. Therefore, the *Help* function will be presented as a hover bubble, and provide the user with a guided advices on what the description should contain. The *Help* function gives the user an easy opportunity at any time to see what the next item in the specification of the incident should be. By implementing the descriptive *Help* hover bubble, it is possible to remove the other two free text fields, which make the original design messy and confusing.

The information about the patient and contact information to healthcare professionals has been changed from fill-in descriptions to *Search* functions accessible through buttons. The *Search* functions are a pop-up windows making incident reporting faster and complying with the terms described in appendix B.3.3. The functionality of the *Search* functions for the patient and healthcare professional information search were described in `Search Patient` and `Search Healthcare Professionals` sections.

In the original DPSD website, the sections `Search Location` and `Search Medication and Medical Devices` consist of buttons presenting pop-up windows requiring input of the information, or many selections from the user. However, in the suggested IR system, the functionality and information inside the pop-ups have been changed to *Search* functions to provide a faster and easier reporting procedure and to meet the requirements from performed interviews and literature search.

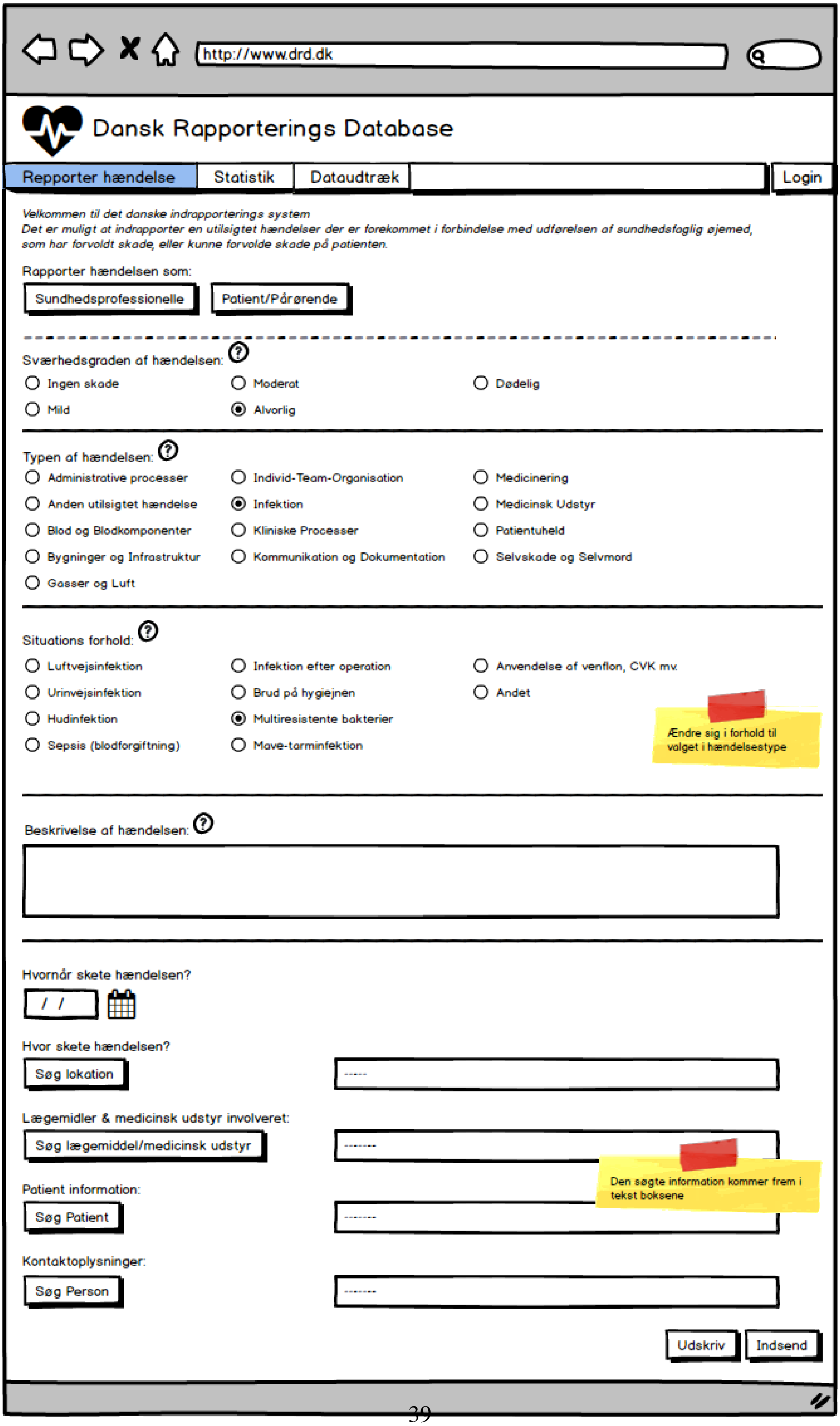


Figure 5.4. Mock-up for the `Report Incident` interface.

Mock-up: Pop-up Window for `Search Location`

The interface `Search Location` is a pop-up window, which will be presented when the user wishes to enter the location of the incident by pressing the *Search location* button in the `Report Incident` interface. The user can choose to find the correct location of the incident by either entering a department name or a department's SOR code. The user will be presented with some options in relation to the provided data. When the user clicks on the right department and the *OK* button, the pop-up window will disappear and the user will be navigated back to the `Report Incident` interface, where the selected department information is presented in the text field next to the *Search location* button. The user also has the possibility to enter several departments by pressing the *Add* button, and when all the involved departments are entered, the user presses the *OK* button and will be redirected to the `Report Incident`interface, where the all selected information is presented in the text field next to the *Search location* button. Additionally, the `Search Location` pop-up window is also accessible from `Statistics` and `Data Extraction` interfaces.

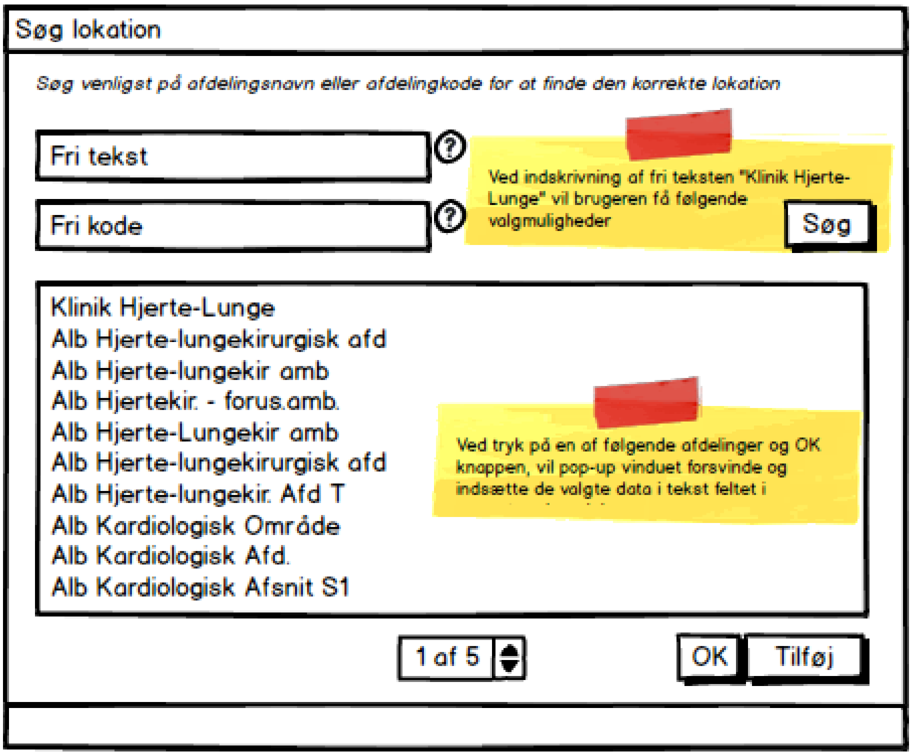


Figure 5.5. Mock-up for the `Search Location` pop-up window accessible from the `Report Incident`, `Statistics` and `Data Extraction` interfaces.

Since the healthcare professionals should know their department SOR code, it will make reporting faster, because the healthcare professionals will not have to go through the three pop-up windows to find the location of the incident, as it is in the existing DPSD system. Use of one of the two free text fields provides the user with a degree of freedom as the user have a feeling of control.

In the DPSD website, the search button for finding the location of the incident is problematic to find as shown in figure 2.2. It is due to a misleading field, that the user believes require to enter location of the incident, however it is not possible to write in the text field. Therefore, by removing the text field the confusion will be removed by only allowing the user to press the *Search Location* button.

The figure 5.5 shows a mock-up of the `Search Location` pop-up window with an example, where the user searches in the text field for ”Clinic cardiopulmonary”. The large text field at the bottom of the pop-up window, presented in figure 5.5, shows the different results from the searched text. When the user has chosen the involved department, the website will provide the user with immediate feedback by showing the chosen department in the empty text field in the `Report Incident` interface.

Additionally, each pop-up window in the DPSD system has a *Help* button marked as (*?*) in the right corner, which does not work while being pressed. Therefore, it is considered as an unnecessary and misleading button by the project group and has been removed in the suggested design.

Mock-up: Pop-up Window for `Search Medication and Medical Devices`

The pop-up window `Search Medication and Medical Devices` appears when the user clicks on the button *Search Medication/Medical Devices* in the `Report Incident` interface. The figure 5.6 presents the mock-up design for the pop-up window `Search Medication and Medical Devices` when the user would like to either add a medication (image to the left) or/and a medical device (image to the right) to the incident report. The pop-up window is constructed in a similar way as the mock-up for `Search Location` with search possibilities for medication/medical devices name and medication/medical devices code. By entering either text or code of the medication or medical devices, the user will be introduced to different options, and by marking the correct medication or medical devices and pressing the *OK* button, the selected medication or medical devices will be presented in the empty text field in the `Report Incident` interface. Display of the involved medication and/or medical devices will provide the user with immediate feedback showing their choices. If the user wants to select multiple medications/medical devices it is possible by pressing the *Add* button.

In cases where the user neither knows the name nor the code of the medication/medical devices, or cannot find the medication/medical device in the database, the user has the option to manually enter information about the medication/medical devices by pressing the *Manual* button. The possibility for the manual input is described further in this section and is presented in figure 5.7.

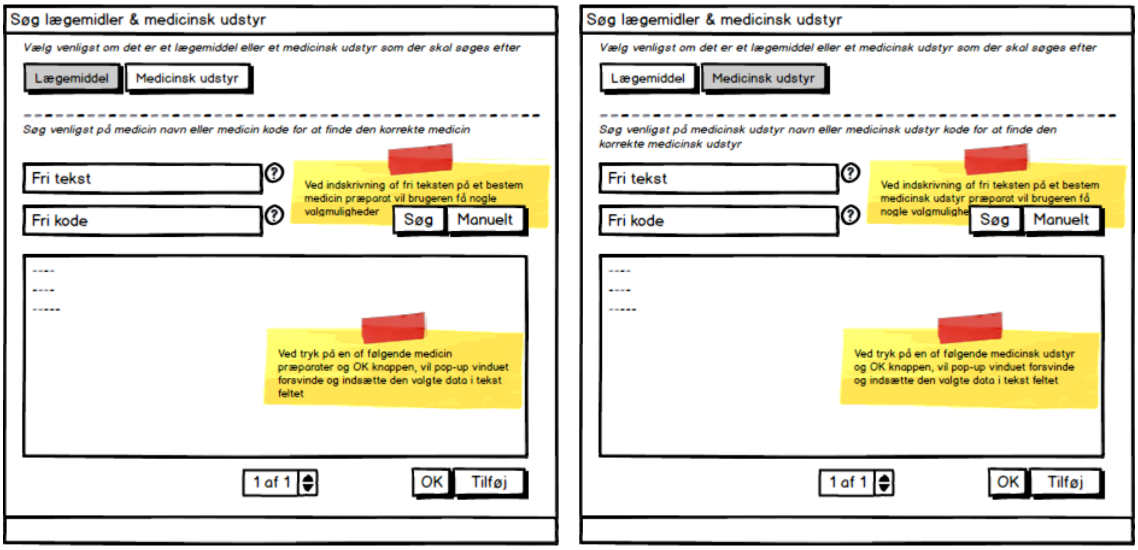


Figure 5.6. Mock-up for the `Search Medication and Medical Devices` pop-up window accessible from the `Report Incident` interface.

With the provided search text fields the search for medicine and medical devices is easier, faster and more straightforward than in the DPSD system. In the DPSD system the user could be confused, if it is required to fill the form manually or search for the medicine or medical devices. By using free text fields it is more obvious what the user should do and gives the user a degree of freedom and a feeling of control. Again the *Help* button in the pop-up window in the DPSD system does not work. Instead of the *Help* button the presentation text at the top in the `Search Medication and Medical Devices` pop-up window will provide the user with the necessary information to reduce doubt about the pop-up window's function.

Mock-up: Pop-up Window for `Manual enter of Medication and Medical Devices`
To create consistency in the design and make it easier and faster for the user to add a new medication and/or medical device, the manual input of the new medication and/or medical device has been transformed into a button, since it is considered rare that the medication and medical devices is not found during the automatic search function. The figure 5.7 presents the mock-up for the `Manual enter of Medication and Medical Devices` pop-up window. The image to the left shows interface when the user presses *Medication* button, and the right image represents the interface when the user chooses the *Medical devices* button. When the user has entered all the necessary text fields for the description of a new medication or medical device, the user can press either *OK*, *Add* or *Reset* buttons. The *OK* button is pressed by the user when the user is done with the input and wants to add new medication or medical device. The *Add* button gives the user possibility to add more new medication or medical devices and the *Reset* button can be used when the user wants to start over filling of the form.

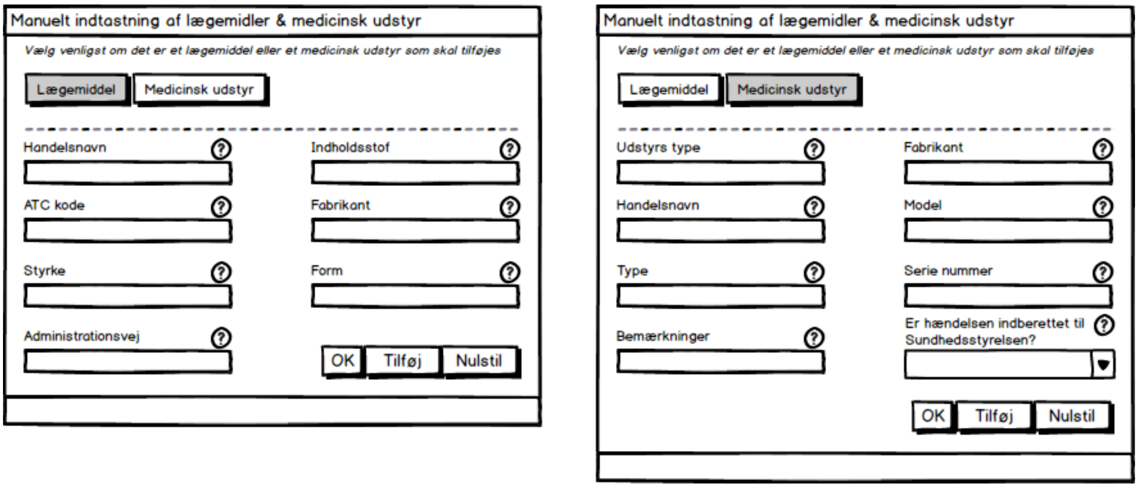


Figure 5.7. Mock-up for the `Manual enter of Medicinal and Medical Devices` pop-up window accessible from the `Search Medicinal and Medical Devices` pop-up window.

The *Help* button in the pop-up windows `Medical devices` and `Involved medication` on the DPSD website is not working properly and therefore excluded from the suggested design. Additionally, at each description field in existing reporting form, a help function is shown as a pop-up window, when the user manually enters new `Medical devices` or `Involved medication`. This involves a lot of unnecessary clicks for the user, and changing of the pop-ups to hover bubbles can solve the problem in the suggested design. The manually entered information will be saved in the database and provide feedback to the user by displaying the manually entered information in the empty text field in the `Report Incident` interface next to the *Search Medication/Medical Devices* button.

Mock-up: Pop-up Window for `Search Patient`

The figure 5.8 shows the mock-up design for the `Search Patient` pop-up window. To create consistency in the design and make it faster and easier to report the incident, the user only has to search for the patient's CPR (Det Centrale Personregister) number. When the user presses the *Search* button, the pop-up will vanish and navigate back to the `Report Incident` interface. All information about the patient will be shown in the empty text field in the `Report Incident` interface and therefore give the user immediate feedback. If the patient do not exist, the user will be provide with and error message in the pop-up window indicated with red text.

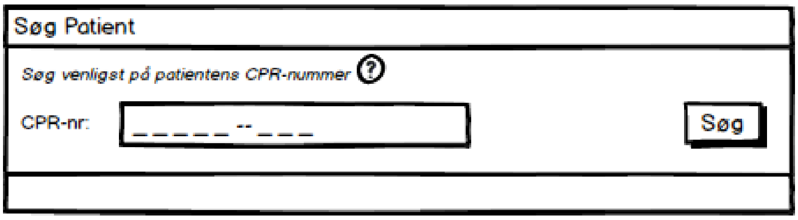


Figure 5.8. Mock-up for the `Search Patient` pop-up window accessible from the interface `Report Incident`.

The entire description of the patient is changed in the suggested design comparing to the DPSD system. All the fields have been replaced by one button and a pop-up window to create simplicity, but also because the typing requirement should be as low as possible. On the current DPSD website the user could manually enter the patient information or search by CPR number. The problem with the CPR number search is that the user does not know that it requires to press Enter on the keyboard before age and gender automatically will be filled out. Furthermore, the user still has to enter the name of the patient, and the website does not show an error when CPR number and gender/age are not matching, or when the CPR number is not existing. Additionally, if the user enters another CPR number and presses *Refresh*, the fields *Gender* and *Age* stay unchanged. By using the CPR number search, all the patient information could be extracted from the CPR register automatically, and all the *Help* pop-up windows in the DPSD website for each field description will disappear creating simplicity, consistency and usability in the suggested design.

Mock-up: Pop-up Window for `Search Healthcare Professionals` **accessible from the** `Report Incident` **interface.**

The design of the `Search Healthcare Professionals` pop-up window is presented in the figure 5.9. The pop-up window is constructed in the same way as the `Search Patient` pop-up window, but instead of searching for CPR number the user searches for staff ID. When the user enters the staff ID number and presses the *Search* button, the pop-up will disappear and move back to the `Report Incident` interface, creating consistency, usability and simplicity in the design. All the information about the healthcare professional will be presented in the empty text field in the `Report Incident` interface next to *Search Healthcare Professionals* button. If the staff ID do not exist the user will be provide with and error message in the pop-up window marked with red text.

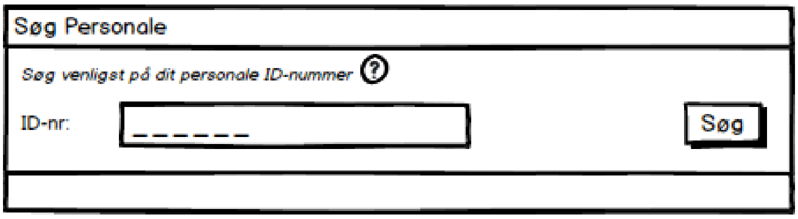


Figure 5.9. Mock-up for the `Search Healthcare Professionals` pop-up window accessible from the interface `Report Incident`.

In the DPSD website it is possible to enter information about a healthcare professional other than the reporting person. However, it is in contradiction with the information obtained during the performed interviews, since it is only possible to fill in the information deeply enough if the reporter involved in the incident himself/herself reports the incident due to specific reporting questionnaire, [Appendix A.2.3]. Therefore, use of the *Search Healthcare Professional* button reduces possibility to enter information about the incident and another healthcare professional. The button also eliminates the need for entering information into four text fields requiring from the user to only search for a staff number. The *Help* pop-up windows for each field description will disappear in the healthcare professional's description making the typing requirement as low as possible.

Mock-up: `Statistics` Interface

The `Statistics` interface will provide statistical information to selected users from the healthcare professionals group. The interface will be protected and require a login function from the user to allow only the approved users access to the `Statistics` interface.

Within the `Statistics` interface presented in figure 5.10, the user will have the possibility to investigate the reported incident data by selecting time range, incident type, severity rate, graph type and location. Based on the user's choice, the selected data will be extracted from the database and plotted as a graph to the right on the website when the user presses the *Search* button. The plot has *Time range* on the x-axis and *Number of Incidents* on the y-axis, with a legend to the shown data. The user could also reset and save the selected data by pressing either the *Reset* or *Save* button.

Each selecting option has a given number which tells in what order the user's selection should occur. The first is the selection of the time range, where the user selects the incident date by pressing the calendar icon. The calendar icon shows a calendar where the user can choose from one date to another (e.g. 01.01.2014 to 01.01.2015). The second selection is the incident type where the user can choose check boxes one, multiple or all incident types. The third option is severity level where the user again can choose one, multiple or all severity levels. The fourth option is the graph type where the user can choose from three different types pie, bar and curve diagram. The last option is the search location where the user can select a specific department by pressing the *Search*

Department button, the pop-up window Search Location will appear. Each option has a help function which will be presented for the user as hover bubbles which makes it possible to guide the user through the plotting of the incident data.

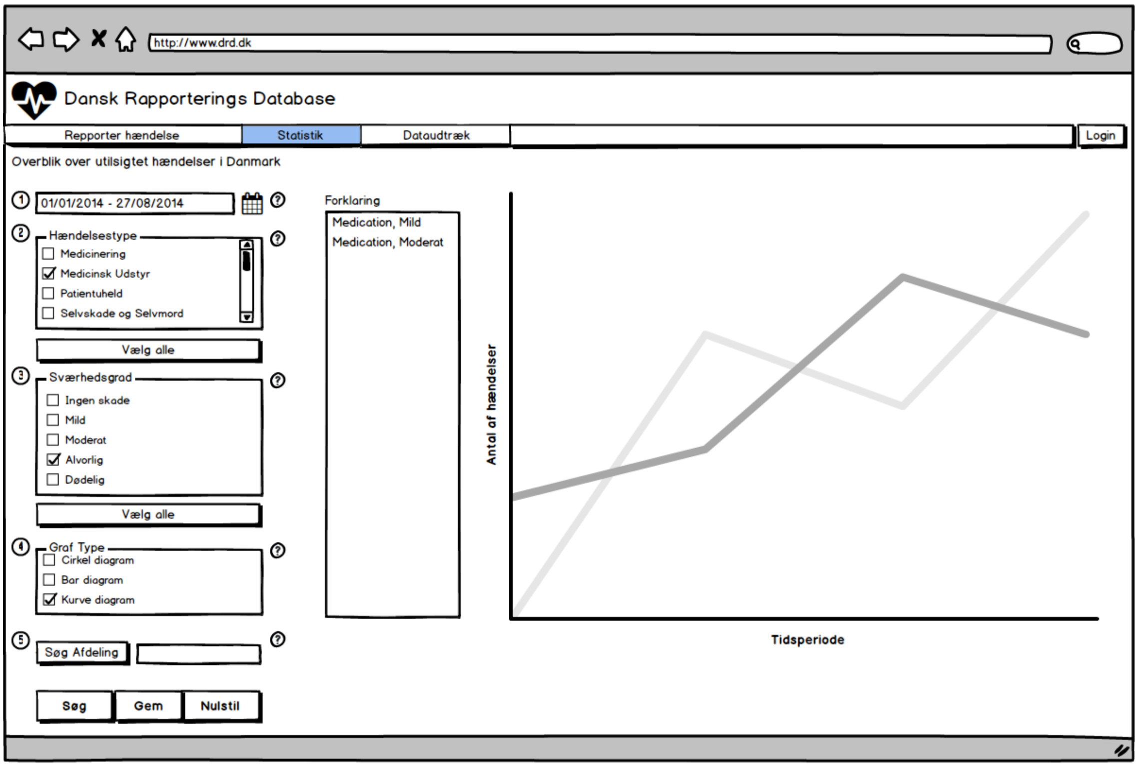


Figure 5.10. Mock-up for the Statistics interface.

Based on the literature search and conducted interviews the interface Statistics has been created. As described in section 2.3, 2/3 of the responders believed, that lack of feedback was the reason why healthcare professionals did not report incidents and by represent information visual gives more successful and faster problem solving. The interviews confirm this claim as it appeared that not all rapporteurs receive a feedback on the reported incident. To reduce the lack of reporting and to motivate healthcare professionals to report incidents, the interface Statistics has been created. It will also provide healthcare professionals with the opportunity to investigate and analyze the patterns, trends of the reported incidents providing the user with a feeling of control.

By having all selection options on the left side and the displayed graph to the right allows the construction to be more clearly structured, and with a comprehensive description of what the plot displays, the user will not feel unsure of what the purpose of the interface is. Therefore, the interface has been structured to be simple and straightforward.

Mock-up: Data Extraction Interface

The design of the Data Extraction interface also requires a login function, but with more security authentication than the interface Statistics by only allowing the access for risk manager responsible for processing of the reported incidents. The strict login function allows to maintain anonymity, confidentiality of the reporter and to enhance security and privacy of the incident data. The interface `Data Extraction` is structured similarly to the interface `Statistics` to give a more consistency in the overall design with the options on the left and the extracted incident data on the right. The design of the interface is presented in figure 5.11. The user has several selection options: severity rate, incident type, location and status of the reports. When the user has chosen the possible selection options and pressed the *Search* button, the incident data is extracted from the database and presented to the user in a table as feedback of the user's selected options.

The table allows the user to display the entire description of a single incident by clicking on the incident, where the user is able to *Print* and *Save* the table or a single incident. The table allows the user to investigate and analyze each simple incident in the user's department, e.g. to prevent recurrence of the incidents.

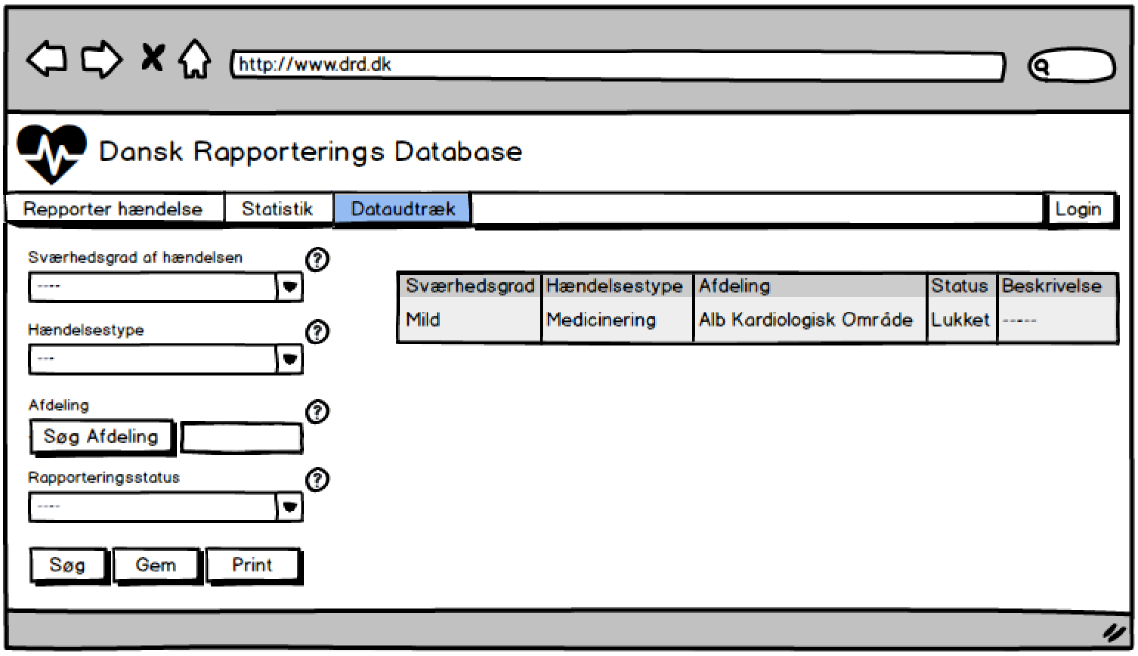


Figure 5.11. Mock-up for the `Data Extraction` interface.

The performed interviews revealed, that data extraction from the existing database is difficult, see section A.1.8. Nowadays, the risk manager receives the incident reports in a comprehensive file from the risk manager at the hospital. Instead of the risk manager having to send the incident report to individual departments, the risk manager has the possibility to extract the data himself/herself. It provides the user with a degree of freedom and a feeling of control to select between the different options which will make the incident reports more manageable.

Mock-up: Pop-up Windows for `Login` and `Forgotten Password`

The pop-up windows `Login` and `Forgotten Password` are used to verify the user in the `Data Extraction` and `Statistics` interfaces. The simple design of the `Login` pop-up window, presented in figure 5.12, requires the user to enter his/her username and password to access the `Data Extraction` and `Statistics` interfaces. The login function is used to maintain anonymity, confidentiality of the reporter, as well as enhancing security and privacy of the incident data.

The pop-up window *Forgotten Password*, presented in figure 5.13, appears to the user when the *Forgotten Password* button is pressed. As shown in figure 5.13, the user must contact the IT Support to receive a password, since the IT support has to identify the user and verify whether the user has the authorisation to access the `Data Extraction` and `Statistics` interfaces.

Figure 5.12. Mock-up for the `Login` pop-up window used in the interfaces `Data Extraction` and `Statistics`.

Figure 5.13. Mock-up for the `Forgotten Password` pop-up window.

5.2.2 Design of the IR Database

Based on the structure of the IR system's reporting form, the database must be able to both retrieve and save information. There are several predefined variables that should be retrieved into the form when the user requires to find and search for the location of the incident, the involved medication or/and the medical devices, the affected patient and the involved healthcare professionals. Furthermore, in the cases where the medication and the medical devices could not be found in the databases, it should be possible for the user to manually enter the information about the involved product, where the product is stored in either the `Medication Database` or the `Devices Database`.

The proposed new solution for the IR system requires a new database design. The IR database must be able to process large amount of data, improve the nowadays performance of the system, be able to easily apply changes and will provide security of the reported incident data. The choice of the database design lies between two types of databases - a relational and a non-relational database. Since a relational database is a well established database with several advantages, while a non-relational (NOSQL) database offers modern and effective solutions for databases, these two database types will be compared. The information used for comparison are based on sections B.4.1 and B.4.2.

Property	Relational model	Non-Relational model
Query Language	SQL	Not specific
Data Reliability	Higher with ACID	Lower with BASE
Performance	Good but decreases with increasing volume of data	Good on large amount of data
Data Structure	Only structured data	Both structured and unstructured data
Scaling	Possible, but problematic	Possible

Table 5.2. Comparison of relational and non-relational database models.

Based upon the comparison from table 5.2 and the requirements in 5.1, the most suitable IR database type is a non-relational model. The characteristics description in the table B.4 describes the document-based stories as the most suitable database type for the required database for the IR system. Additionally, the document-based stories contains two database models: the MongoDB and the CouchDB. The MongoDB model was chosen based upon the fact, that it is currently the most used and available NoSQL database, [Moniruzzaman and Hossain, 2013].

The `Data Extraction` and `Statistics` interfaces are retrieving information from the IR database to visualize the reported incidents in different ways. As the information retrieved is dependent on the user's choices and selections, the query will search the IR database for information, which matches the request. The data will then be prepared to be shown in plots or in a table according to the selected interface.

The theory described in appendix D illustrates the structure of the MongoDB. Therefore, in order to be able to apply the MongoDB theory to fit the IR system's reporting form, a IR database design is developed in figure 5.14.

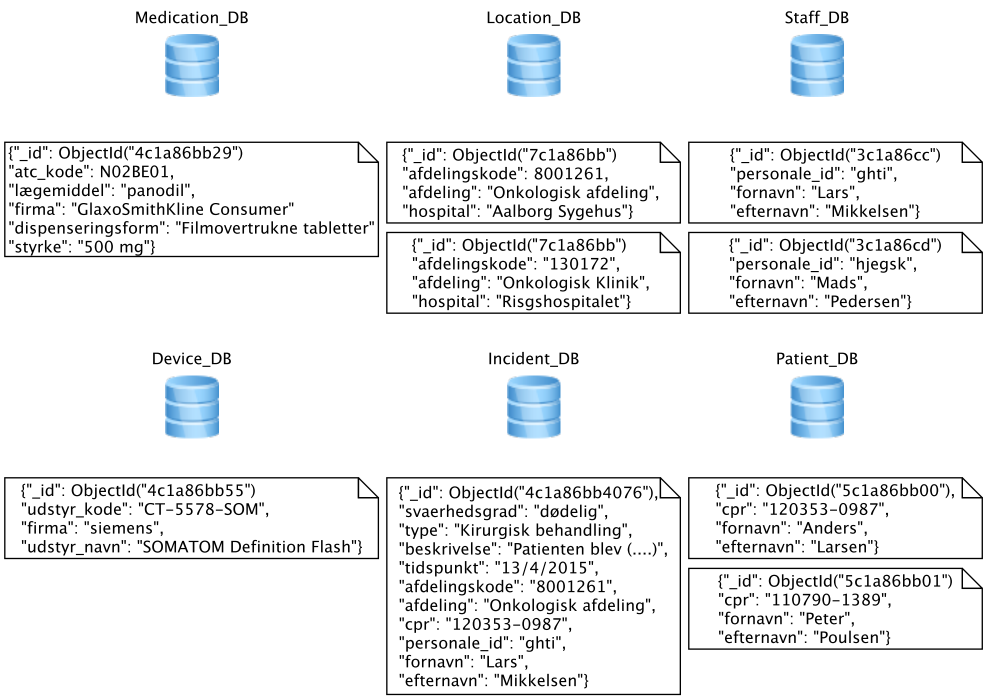


Figure 5.14. Design of the IR database.

The structure of the IR database in figure 5.14 is constructed with six databases. Five of the databases - (`Medication Database`, `Device Database`, `Patient Database`, `Location Database` and `Staff Database`) are used to extract predefined information, while the last database (`Incident Database`) save and retrieve information regarding the reported incidents.

Each database contains a collection of documents where each document, for instance in `Patient Database`, represents one patient. Therefore, the `Patient Database` includes millions of documents, a document for each person with a CPR number. The information of a document in the patient database might look as follows:

```
{"_id":  ObjectId("5c1a86bb00"),
..."cpr":  "120353-0987",
..."fornavn":  "Anders",
..."efternavn":  "Larsen"}
```

More examples of documents in a collection of the different databases are presented in the figure 5.14.

5.3 Implementation

Implementation of the IR system and database design was based on strategies described in section 5.2. Due to lack of incident data, the IR database design was not connected to the implemented IR system. Therefore, within this section practical implementation and expected functionality of the IR system and IR database is provided and explained.

5.3.1 Implementation of the IR System

The developed IR system was implemented using web development methods described in section B.3.4.

Since the implemented IR system is aimed for Danish healthcare system, it is developed using Danish language for the interface, as described in requirements presented in section 5.1. The layout and the structure of the designed interfaces are based on the mock-ups presented in section 5.2.1 and implemented using HTML. The visual properties of the system's presentation layer were coded using a CSS file. Since the CSS file require specification of many different elements and their different states, the project group decided to utilize a free Bootstrap framework during the implementation. The framework provided a ready to use CSS file, which was adjusted by the project group to fit the required design. The functionality of the used elements was specified with use of the JavaScript and mainly jQuery library. The jQuery, allowed the system functionality to work on broader amount of browsers and their different versions. Some of the JavaScript functionality, such as interactive tables or calendar functions, were as well based on elements of the Bootstrap framework.

The IR system is divided into three main interfaces: `Report Incident`, `Statistics` and `Extract Data`. All three interfaces and `Login` function are accessible to the user at any time trough a navigation bar presented in figure 5.15.

Figure 5.15. The navigation bar of the IR system

Since the IR system lacks the connection to the database, it is implemented as a proof of concept with main focus on visual optimization of the design and layout. Many of the search functions allowed in the IR system are used across the interfaces. Therefore, the description of the implemented IR system is based on the individual elements, rather than the interfaces.

Implementation of the `Report Incident` Interface

Implemented `Report Incident` interface is presented in figure 5.16 and is the interface, which the user can use to report the incident. The implemented interface provides possibility to choose between reporting as a healthcare professionals and as a patient/relatives by pushing the respective buttons. However, since main focus of the project was on healthcare professionals, an interface for reporting as a patient/relatives was not implemented. Therefore, when the user presses *Patient/Relatives* button, a pop-up appears informing the user that the interface is not implemented. When the user chooses to report as the healthcare professional, by pressing *Healthcare Professionals* button, a reporting form appears. All of the fields in the reporting form are provided with a help hover bubble, which appears when user moves the mouse over the green icon with a question mark.

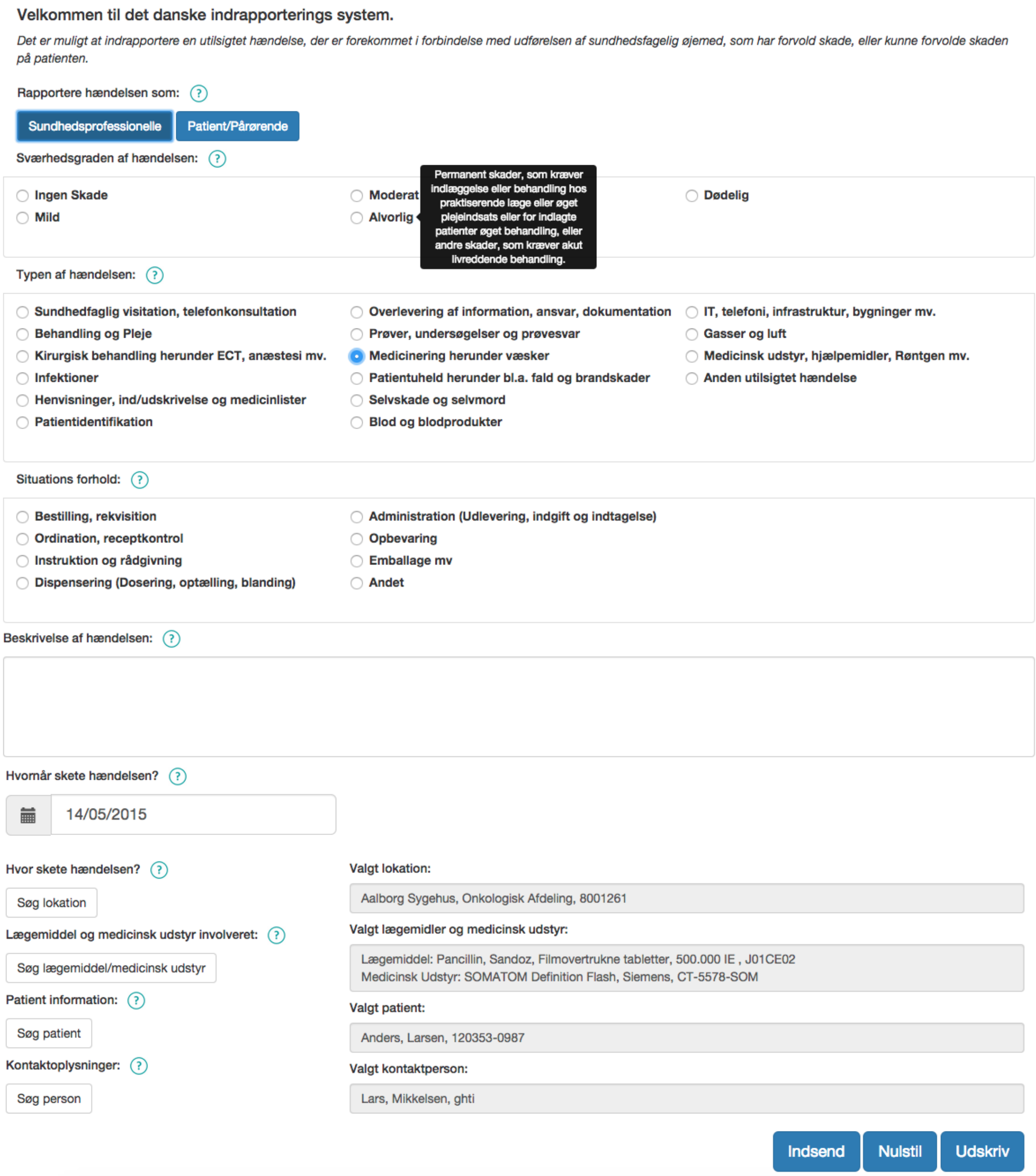


Figure 5.16. Implemented `Report Incident` Interface.

The user has a chance to select required severity rate and incident rate using radio buttons. Based on the chosen incident type, a possible selections of situations contributing to the incident appear in the form as radio buttons. Since the implemented interface is programmed as a proof of concept, only situations related to medication errors are presented. Medication errors were chosen based on the analysis described in section 2.1.2, which states that it was the most common incident types in Denmark. The radio buttons for selection of the severity rate, incident type and contributing situation were used to ensure, that the user will not choose more than one variable in each of the three sections. Furthermore, all of the used terms for severity rate, incident type and contributing situation have implemented help hover bubbles attached to them. The help hover bubbles appear when the user moved the mouse cursor over the variable's name and they provide user with correct definition and information about the selected variable, as shown in figure 5.16.

As previously described in section 2.4.1, the suggestion to merge the different text fields regarding incident description, was implemented in the `Report Incident` interface. The merged version of the incident description text field, provided the user with only one text field, which gives the possibility to the user to provide full description of the incident, together with consequences and possible improvements. It also prevents the information regarding the incident to be duplicated.

Selection of the incident's occurrence date is implemented using a calendar function, as shown in figure 5.17. The implemented calendar allows the user to choose among the dates before and up to current date. These calendar settings are applied to avoid a possible error of choosing a future date as an incident occurrence date.

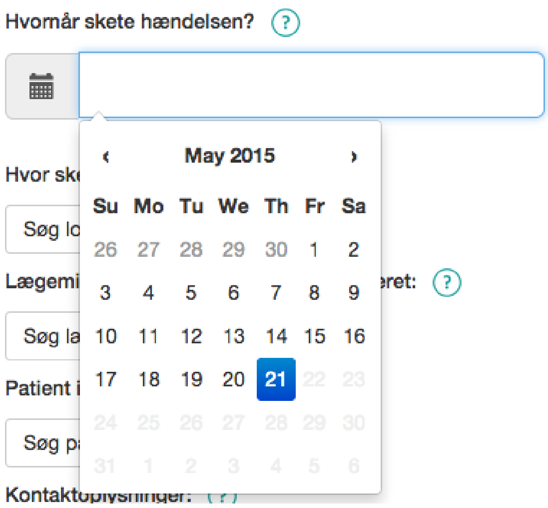


Figure 5.17. Implemented calendar function in `Report Incident` interface.

Selection of additional information for the incident report is performed as a search functions for the location of the incident, involved medication and medical devices, involved patient and contact details to the person reporting the incident. The implementation as a search functions was made to provide the user with quicker reporting procedure, and to ensure that the provided details have always the same format without lacking information. All of the search functions are implemented as a pop-up windows and are described in the following sections. The selected information from the search functions are collected and shown in the `Report Incident` interface in the not writable text fields next to the corresponding search functions.

At the end of the incident reporting form, the user has a chance to submit, reset or print the form. The *Submit* button is designed to submit the filled form to the database. Since the database was not connected to the interface in the project, the information provided in the form are shown as console logs, which aids verification if the selected information are correctly collected from the reporting form. The *Reset* button clears all the selections and texts in the chosen reporting form, so the user does not need to choose as whom the report is filled as. The *Print* button provides printing function of the incident report, if the user would like to print a copy of the filled report. However, the printing function is not implemented in the developed system.

From the `Report Incident` interface, the user has a possibility to navigate to other interfaces such as `Statistics` and `Data Extraction`.

Implementation of the `Search Location` Pop-Up

The `Search Location` pop-up window is implemented as presented in figure 5.18. The pop-up allows the user to search for the location, where the incident has occurred. The user has a chance to search a location database after the location's name or SOR code. The user then selects the wanted location from the table by checking a checkbox or by clicking on the respected table row to select the location. After the selection, the user must press the *Add* button to confirm the selection and then *OK* button to close the pop-up window. Additionally, if the selection was wrong, the user may press the *Reset* button to clear the added variables in the text field.

The `Search Location` pop-up window is used for location search in all of the implemented interfaces - `Report Incident`, `Statistics` and `Data Extraction`.

Implementation of the `Search Medication and Medical Devices` Pop-Up

The `Search Medication and Medical Devices` pop-up window allows the user to search for the possibly involved medication and/or medical devices. The user can switch between the medication and medical devices interface by pressing the respected buttons. In the implemented pop-up window it is possible to both search automatically using the medication's ATC code or name in the interface for the *Medication* window shown in figure 5.19, or by using medical device's code or name in the interface for the *Medical Device* window shown in figure 5.20. The search should be then performed in the corresponding database either `Medication Database` or `Device Database`, for the selected criteria and show the results in form of a table. The user then can select a medication or a medical device by either checking a check box, or by clicking on the respected table row. Then the user must press the *Add* button to add the selected medication and/or medical devices and press *OK* button to close the pop-up window. The user may as well press the *Reset* button to clear the added variables from the text fields. If the user resets variables in one of the windows, the selected variables are not erased

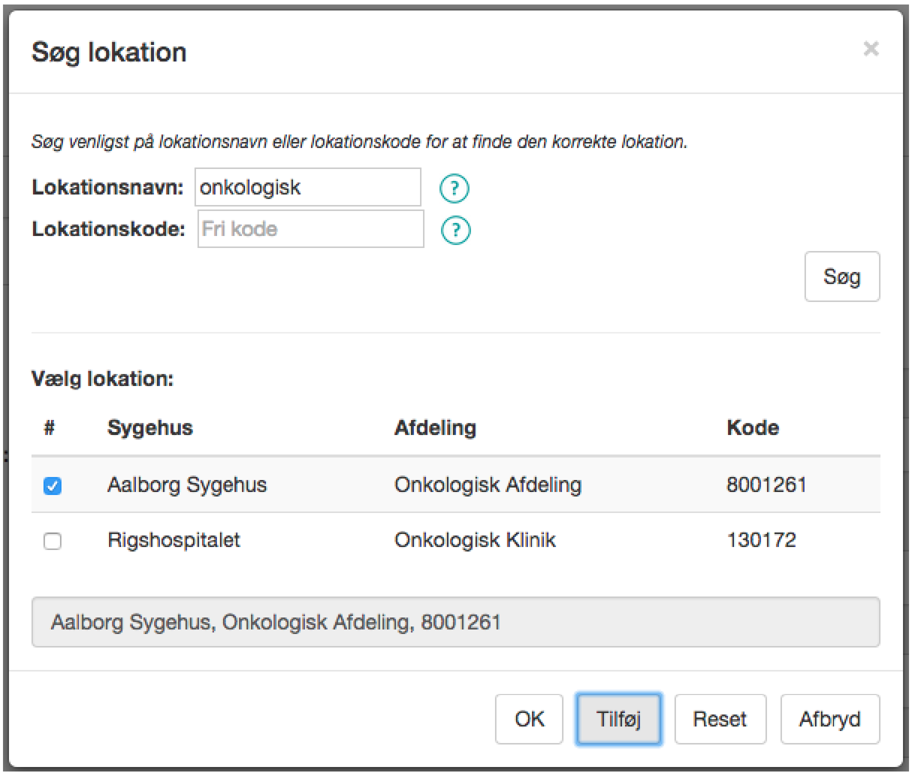


Figure 5.18. The implemented `Search Location` pop-up.

in the other one. Therefore, to reset all of the selected variables, the user must press the *Reset* button in each of the windows.

If the user cannot find the needed medication and/or medical device, it is possible to input the details about the medication and/or medical devices manually. The manual input is activated by choosing *Manual* button form either of the windows. If the user is in the *Medication* window, the *Manual* button will open form for medication input, as shown in figure 5.21. A corresponding situation is when the user is in the *Medical Devices* window, then by pressing the *Manual* button, a form for medical device input will open, as shown in figure 5.22. The user then must press *Add* button to add the inputted information and then press *OK* to close the window. The manual input information are based on the nowadays DPSD form for providing information about the involved medication and/or medical devices. However, in the implemented manual input of the medical devices, a text field for additional comments was removed, since it is believed that the necessary comments about the medical device will be provided in the incident description text field in the `Report Incident` interface.

The information obtained from the `Search Medication and Medical Devices` pop-up are used only in the `Report Incident` interface.

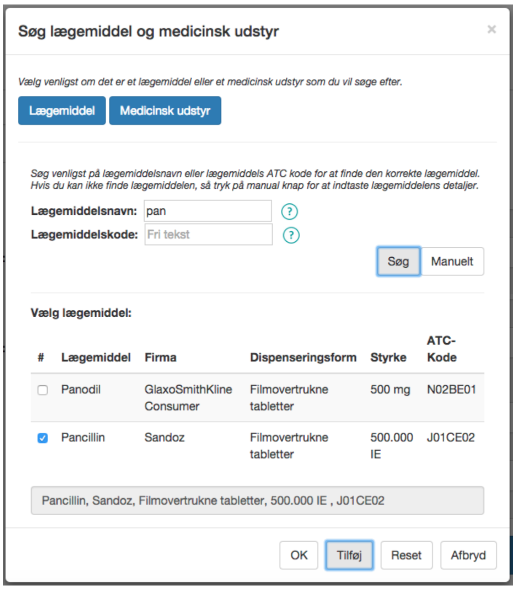


Figure 5.19. Implemented *Medication* window in the `Search Medication and Medical Devices` pop-up.

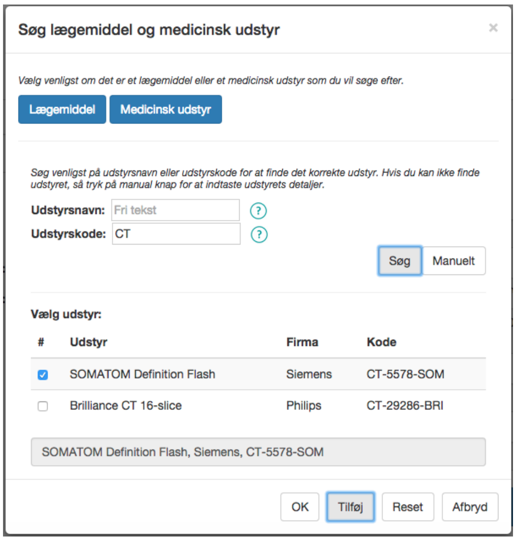


Figure 5.20. Implemented *Medical Device* window in the `Search Medication and Medical Devices` pop-up.

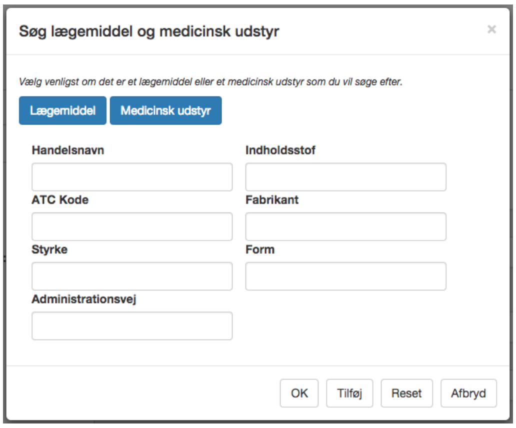


Figure 5.21. Implemented *Manual* button for a manual medication input.

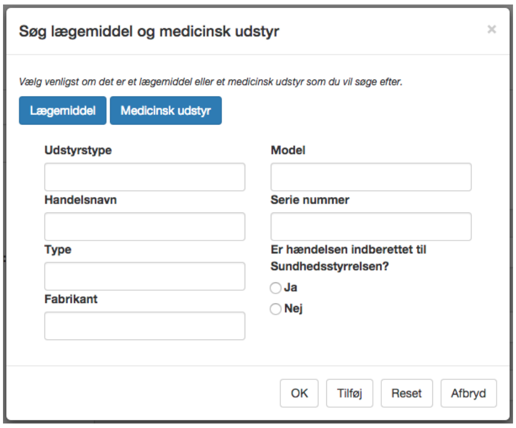


Figure 5.22. Implemented *Manual* button for a manual medical device input.

Implementation of the `Search Patient` Pop-Up

The `Search Patient` pop-up window is shown in figure 5.23 and is used for search of the patient involved in the reported incident. The user has a chance to search after patients CPR-number or name in the `Patient Database`. The results of the search are presented in a table form, where it is possible to select the patient by checking a checkbox or by clicking on the respected row. The user then presses *Add* button to add and confirm the patient selection, and then presses *OK* button to dismiss the `Search Patient` pop-up window. Furthermore, the user may press the *Reset* button to reset the added patients in the text field.

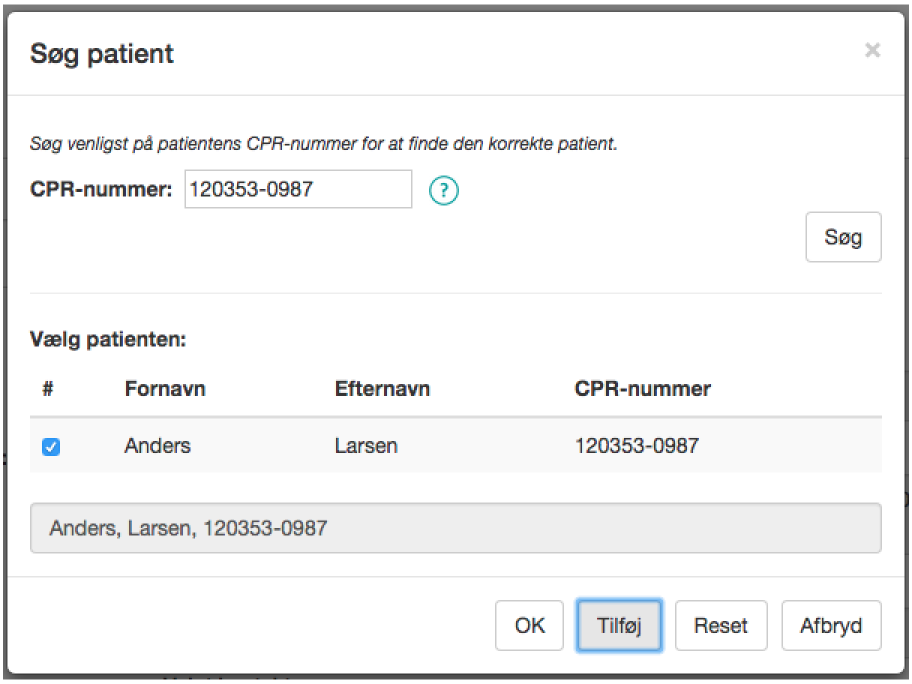


Figure 5.23. Implemented `Search Patient` pop-up.

The `Search Patient` pop-up window is only used in the `Report Incident` interface.

Implementation of the `Search Healthcare Professionals` Pop-Up

The `Search Healthcare Professionals` pop-up window is shown in figure 5.24 and allows a search for the reporter's contact details. The user can search after reporter's ID-number or name through the `Staff Database`. The result of the search is presented in a table form allowing the user to select the reporter's details by checking a checkbox or by clicking on the respected table row. The user then presses *Add* button to add the wanted contact selection and then *OK* button to close the pop-up window. The *Reset* button is as well implemented to aid the user to delete the added variables from the text field.

The `Search Healthcare Professionals` pop-up window is only used in the `Report Incident` interface.

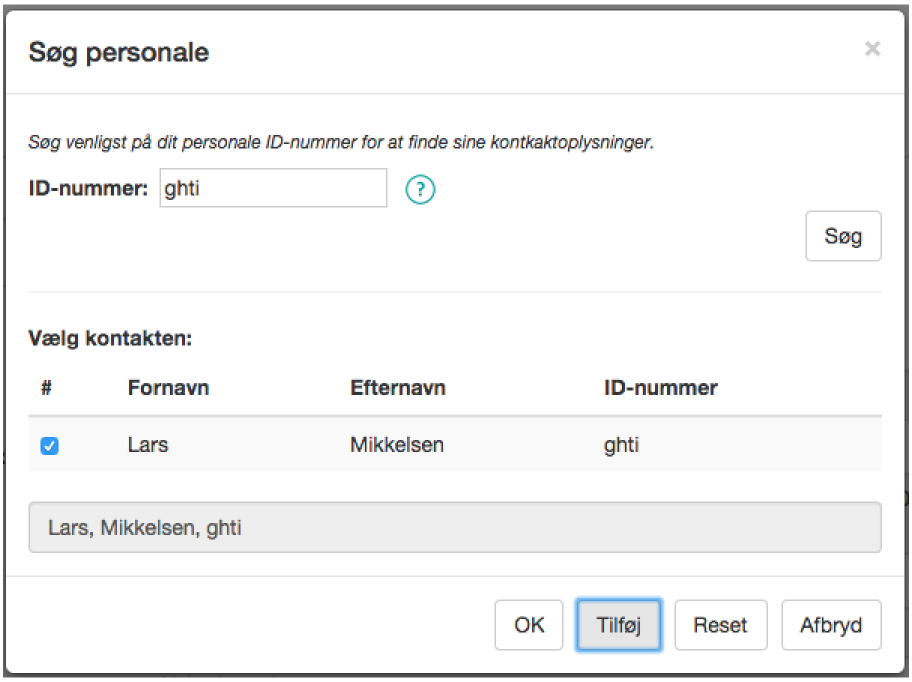


Figure 5.24. Implemented `Search Healthcare Professionals` pop-up.

Implementation of the `Statistics` Interface

The `Statistics` interface is implemented to allow the user search, data exploration and monitoring of the reported incidents. The implemented `Statistics` interface is shown in figure 5.25.

Selection of the time range from which the search should be performed is implemented with a calendar function, which allows a search from the selected date and maximum up to the current date. The other dates are disabled based on the user's selection. The implemented calendar function for both `Statistics` and `Data Extraction` interfaces is shown in figure 5.26.

The user can select none, some or all of the severity rates and/or incident types for the search. The different selections are allowed by use of check boxes. Selection of a graph type in which the data should be presented is implemented with check boxes to allow the user choice of one or more graph types. Lastly the user can select a location from which the incidents' statistic will be shown. The search for the location is performed as described in implementation of `Search Location` pop-up window.

By pressing the *Search* button, a search after the selected data in the DPSD database is performed. The collected data is then plotted in the chosen graph type. The figure an be saved by pressing *Save* button. To reset the form and figure, the user presses the *Reset* button. It is also possible to perform several different data searches for the statistical purposes.

However, since the `Statistics` interface contains sensitive data, a login function is

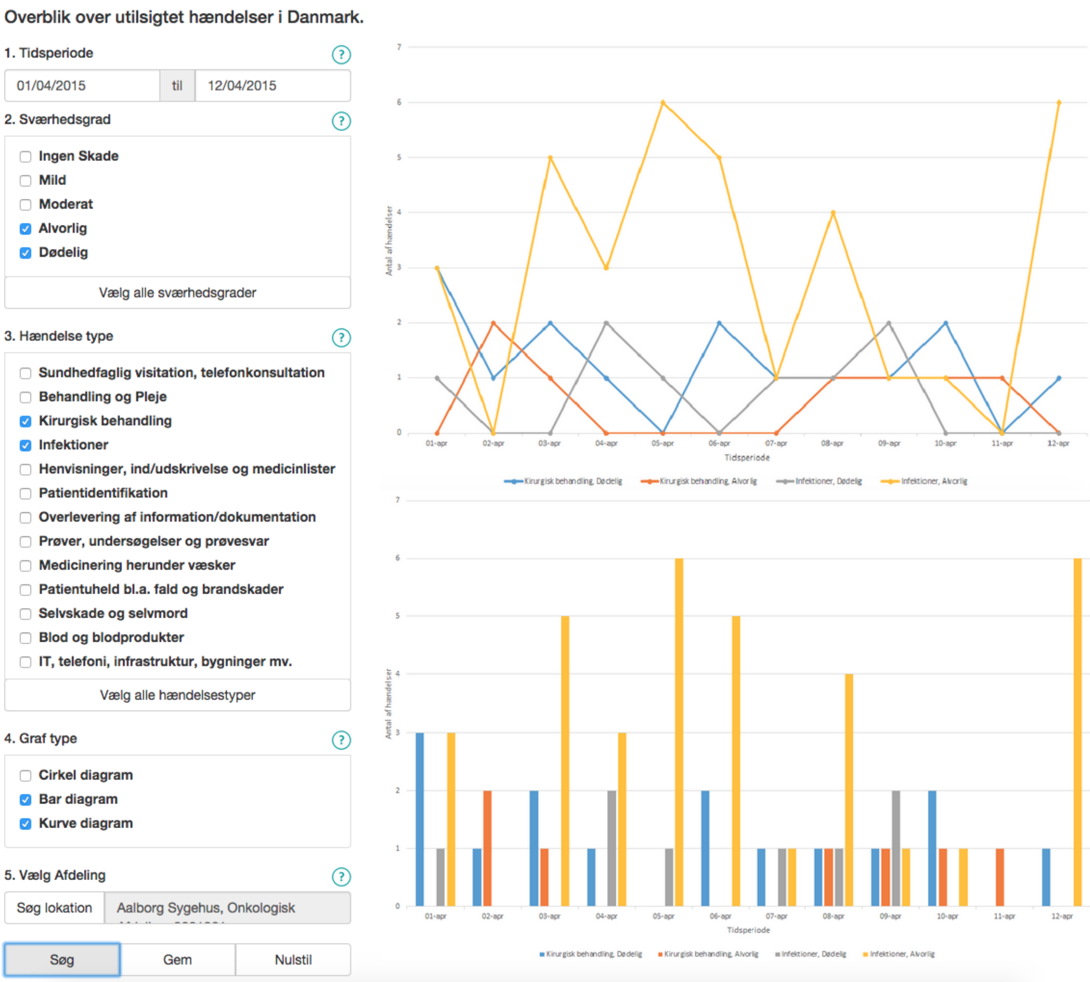


Figure 5.25. Implemented `Statistics` interface.

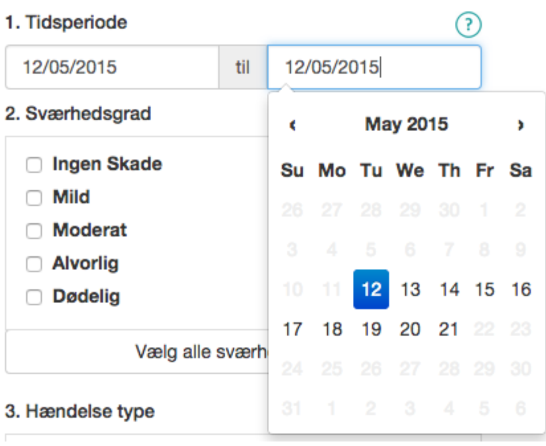


Figure 5.26. Implemented calendar function for both `Statistics` and `Data Extraction` interfaces

implemented to increase the security and privacy of the data. Additionally from the `Statistics` interface, user has possibility to access the `Report Incident` and `Data Extraction` interfaces.

Implementation of the `Data Extraction` Interface

The `Data Extraction` interface is shown in figure 5.27 and is implemented to optimize the analysis of the incident reports. Similarly to the `Statistics` interface, the user selects the wanted time range with the calendar function allowing the user to selects the dates before and up to the current date, as shown in figure 5.26. Then the user may select none, one or several severity ranges and/or incident types. Since the incident reports have different statuses depending on the progress of the analysis, it is also possible to select which statuses are of interest to be extracted from the DPSD database using checkboxes. Lastly the user can search after the location of the incidents as described in implementation of `Search Location` pop-up window.

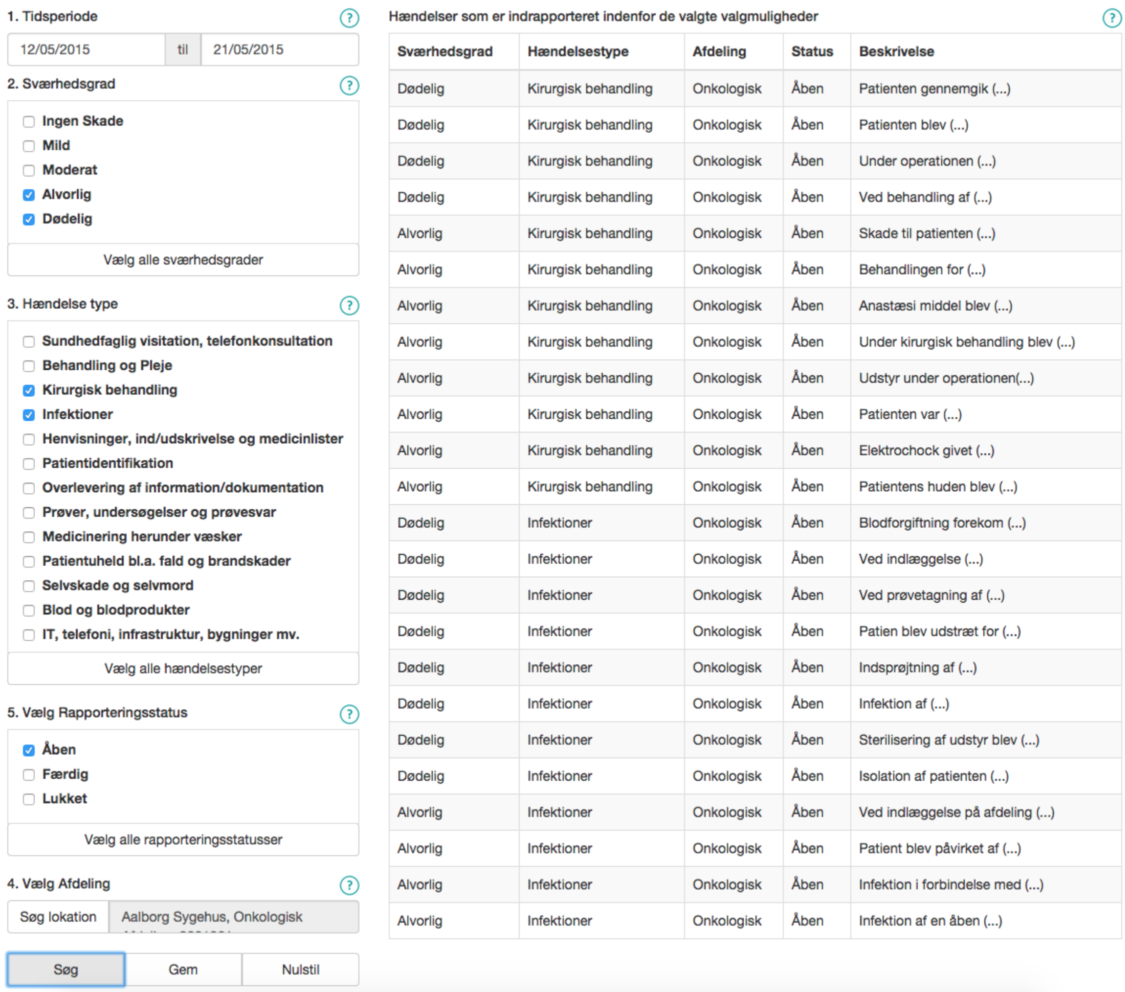


Figure 5.27. Implemented `Data Extraction` interface.

By pressing the *Search* button the selections are searched in the DPSD database and presented in a form of an interactive table. By pressing on the table rows it is possible to view the respected incident report. An example of the extracted report is shown in figure 5.28. Furthermore it is possible to save the search results by pressing the *Save* button, or reset the selections in the form and the table by pressing the *Reset* button.

Since the `Data Extraction` interface contains sensitive data, a login function is attached to the interface.

Figure 5.28. An example of the extracted incident report.

Implementation of the `Login` Pop-Up

The `Login` pop-up window is implemented in order to provide a login function for the `Statistics` and the `Data Extraction` interfaces. The pop-up window is presented in the figure 5.29.

The login function requires the user to provide a *username* and *password*, which are verified in the `Staff Database` when the user presses the *OK* button. If the login details match the database criteria, the user obtains access to the restricted area of the implemented system.

The `Forgotten Password` pop-up window described in the section 5.2.1 was not implemented. Since the IR system has no connection has no database connection, it was not possible to validate the user and therefore provide technical help in case the user has forgotten password or had troubles with the login function.

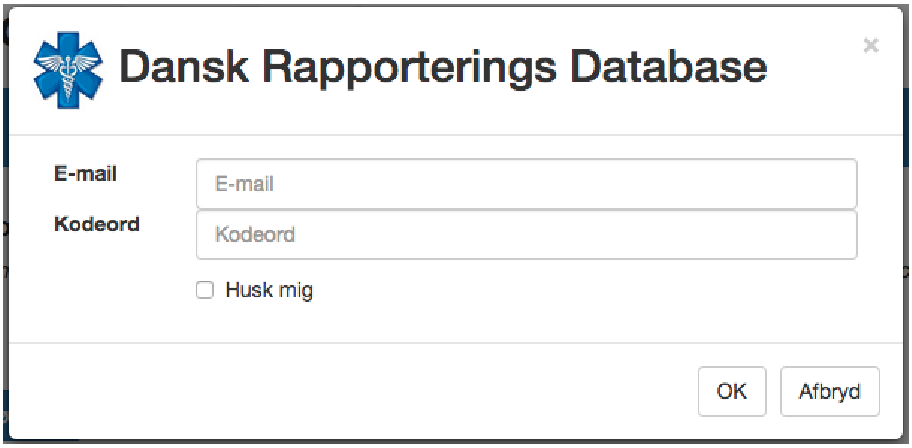


Figure 5.29. Implementation of the `Login` pop-up.

5.3.2 Planned Implementation of the IR Database

Due to lack of incident data from the Danish DPSD database, it has not been possible to implement the IR database. Therefore, a description is given of how the databases would have been implemented.

The implemented IR database structure should have been based on the design described in the section 5.2.2, where the six databases were to be created in MongoDB. The predefined information in five of the databases and the `Incident Database`, with the reported incident data, would be imported to the databases by converting the predefined and reported incident data into a `csv` or `JSON` file from the original Danish DPSD database.

Extraction of data from the IR database was performed by using a specific query. Based on the example shown in figure 5.14, the search might be when the user chooses to view all incident data - in this case surgical treatment. The selected values would then be used by a PHP file in a query to search all the documents in the `Incident Database` for the documents that fulfill the specific requirement. The query for finding information would be:

```
> db.Incident_DB.find ( { type : "surgical treatment" } )
```

Additionally, when the user has completed the incident form and presses *Submit* button, the report is saved in the `Incident Database` with a saving query:

```
> db.Incident_DB.insert( {"Severity": "deadly",
..."type": "Surgical Treatment",
..."description": "The patient was (....)",
..."Time": "04.03.2015",
..."Department Code": "8001261",
..."Department": "Department of Oncology",
..."CPR": "120353-0987",
..."personale_id": "ghti",
```

```
..."name": "Lars",
..."surname", "Mikkelsen"} )
```

The query content inside the brackets will vary according to what the user has filled into the form. The information that is stored in the `Incident Database` is the information that is used to extract data for visualization in the `Statistics` interface and extraction of the reports in the `Data Extraction` interface. Since a document does not have an unique key, the MongoDB will assign a key to the document, and therefore the document will look like the following in the `Incident Database`:

```
{"_id": ObjectId("4c1a86bb4076"), "Severity": "deadly", "type": "Surgical
Treatment", "description": "The patient was (....)", "Time": "04.03.2015",
"Department Code": "8001261", "Department": "Department of Oncology",
"CPR": "120353-0987", "personale_id": "ghti", "name": "Lars", "surname",
"Mikkelsen"}
```

In case where the user would like to have a data visualization in which the user would like to see all the data with a moderate severity level, with the incident type of infection, at the department of oncology and with a specific date another query would be formed. The query will search in the `Incident Database` for data where all the requirements are fulfilled. Afterward a JavaScript will prepare the data for viewing in a plot. The query for extracting the data would look like the following equation:

```
> Incident_DB.find( {"Severity": "moderate",
..."type": "infection",
..."Time": "03.05.2015",
..."Department": "Department of Oncology" } )
```

5.4 Test

The test section provides description of the functional tests performed on the IR system. Since the database was not fully implemented in the IR system, the description of the IR database test includes plan for the test and not the actual results of the test.

5.4.1 Functionality Test of the IR System

During the development process the functionality of the developed and implemented interfaces for the IR system was tested and adjusted iteratively to eliminate possible errors in the IR system's functionality. Therefore, within this section a final test of the IR system is performed and described to verify if the systems functionality fulfills the specified requirements in section 5.1. Furthermore, possible errors are as well identified.

The test of the IR system elements is divided according to the described use cases:

1. `UC1: Report Incident`
2. `UC2: Search`

3. UC3: Statistics
4. UC4: Extract Data
5. UC5: Verify User

Most of the functional requirements described in section 5.1 were fulfilled, i.e. possibility for incident reporting, adjustable content of the IR system, implementation of different search functions and login. Moreover, non-functional requirements related to the use of Danish language in the implemented interface, development of the design based on usability theory and working properly on different web-browsers were fulfilled as well. However, the implemented IR system is regarded as a proof of concept system, since the requirements related to precise data extraction and visualization of data were not fulfilled due to lack of provided incident data and therefore connection to the database. Therefore, within this section a functionality test of the implemented interfaces of the IR system is performed and functionality of the elements is evaluated. The test is divided according to the described use cases, since they are a good representation of the possible interactions with the system and its functionality.

Test of `UC1: Report Incident`

The performed test of the `Report Incident` use case is based on use case specification presented in table 5.1 and is described in the table 5.3. The use case main functionality was to allow incident reporting for the user.

Description	Test of possible interactions of the user with the `Report Incident` interface.
Approach	User opens the IR system and selects `Report Incident` from the navigation bar.
Input	1. User presses *Patient/Relatives* button 2. User presses *Healthcare Professionals* button 3. User moves mouse over the different radio button variables and green question mark 4. User chooses a severity rate 5. User chooses an incident type 6. User chooses medication including fluids from incident type 7. User chooses a related situation to the incident type 8. User inputs *"Test"* in description of the incident 9. User presses on the calendar icon 10. User presses on the text field next to calendar icon 11. User presses the *Submit* button 12. User presses the *Reset* button
Expected output	1. The pop-up window appears informing the user that the interface is not implemented 2. A reporting interface for healthcare professionals appears 3. A hover bubble including a corresponding help message appears 4. User can only choose one severity rate 5. User can only choose one incident type 6. A box with related situations to the incident appears 7. User can only choose one related situation 8. It is possible to input the text in the text area 9. Nothing happens 10. A window including a calendar opens 11. The user receives an informing message in a pop-up window and all of the inputted variables are shown in the console log 12. All of the filled and selected variables in the form are cleared
Result	All test results have the expected output.

Table 5.3. Test of the UC1: `Report Incident`.

Functionality of the *Print* button was not tested, since it was not implemented. Furthermore, the test of different search functions are described in test `UC2: Search`.

Test of `UC2: Search`

The performed tests of the `Search` use case are based on use case specifications presented in table C.1 and are described in tables 5.4, 5.5, 5.6 and 5.7. Main functionality of the use case allowed the user to search for location, medication and medical devices, patient and contact details for the healthcare professionals. Therefore, the different search functions are divided to test the functionality of the different pop-up windows in more details.

Description	The test focuses on the functionality of the `Search Location` pop-up window.
Approach	User selects `Report Incident` from the navigation bar and *Search Location* button.
Input	1. User presses *Search Location* button 2. User inputs name of the location in *Location Name* text field 3. User inputs SOR code of the location in *Location Code* text field 4. User presses *Search* button 5. User selects a row by pressing a row 6. User selects a row by checking a checkbox 7. User presses *OK* button 8. User presses *Add* button before the location search 9. User presses *Add* button after the location search 10. User presses *Reset* button 11. User presses *Cancel* button
Expected output	1. A pop-up window for location search appears 2. It is possible to input text in the text field 3. It is possible to input text in the text field 4. A table with search results appears 5. A variable is selected 6. A variable is selected 7. The pop-up window closes and the results are shown in `Report Incident` interface 8. The *Add* button is disabled 9. The *Add* button is enabled and variables added to a text field in the pop-up window 10. The variables from the text field in the pop-up window are erased 11. The pop-up window is dismissed and non of the variables appear in the `Report Incident` interface
Result	All test results have the expected output.

Table 5.4. Test of the UC2: `Search` for location.

Description	The test focuses on the functionality of the `Search Medication and Medical Devices` pop-up window.
Approach	User selects `Report Incident` from the navigation bar and *Search Medication Medical Devices* button.
Input	1. User presses *Search Medication Medical Devices* button 2. User selects *Medication* button 3. User selects *Medical Device* button 4. User selects *Manual* button from the *Medication* window 5. User selects *Manual* button from the *Medical Devices* window 6. User inputs name of the medication in *Medication Name* text field 7. User inputs ATC code of the medication in *Medication Code* text field 8. User inputs name of the medical device in *Device Name* text field 9. User inputs code of the medical in *Device Code* text field 10. User presses *Search* button 11. User selects a row by pressing a row 12. User selects a row by checking a checkbox 13. User presses *OK* button 14. User presses *Add* button before the location search 15. User presses *Add* button after the location search 16. User presses *Reset* button 17. User presses *Cancel* button
Expected output	1. A pop-up window for medication and medical devices search appears 2. A *Medication* window appears 3. A *Medical Devices* window appears 4. A window for manual input of medication appears 5. A window for manual input of medical devices appears 6. It is possible to input text in the text field 7. It is possible to input text in the text field 8. It is possible to input text in the text field 9. It is possible to input text in the text field 10. A table with search results appears 11. A variable is selected 12. A variable is selected 13. The pop-up window closes and the results are shown in `Report Incident` interface 14. The *Add* button is disabled 15. The *Add* button is enabled and variables added to a text field in the pop-up window 16. The variables from the text field in the pop-up window are erased 17. The pop-up window is dismissed and non of the variables appear in the `Report Incident` interface
Result	All test results have the expected output.

Table 5.5. Test of the UC2: `Search` for medication and medical devices.

Description	The test focuses on the functionality of the `Search Patient` pop-up window.
Approach	User selects `Report Incident` from the navigation bar and *Search Patient* button.
Input	1. User presses *Search Patient* button 2. User inputs 120353-0987 in *CPR-number* text field 3. User inputs 1203530987 in *CPR-number* text field 4. User inputs 120s53-b987 in *CPR-number* text field 5. User inputs 120353098745 in *CPR-number* text field 6. User presses *Search* button 7. User selects a row by pressing a row 8. User selects a row by checking a checkbox 9. User presses *OK* button 10. User presses *Add* button before the location search 11. User presses *Add* button after the location search 12. User presses *Reset* button 13. User presses *Cancel* button
Expected output	1. A pop-up window for patient search appears 2. The CPR-number stays in the expected form 3. The CPR-number is automatically adjusted to the expected form 4. It is not possible to input letters into the text field 5. The CPR-number stops at 1203530987 6. A table with search results appears 7. A variable is selected 8. A variable is selected 9. The pop-up window closes and the results are shown in `Report Incident` interface 10. The *Add* button is disabled 11. The *Add* button is enabled and variables added to a text field in the pop-up window 12. The variables from the text field in the pop-up window are erased 13. The pop-up window is dismissed and non of the variables appear in the `Report Incident` interface
Result	All test results have the expected output.

Table 5.6. Test of the UC2: `Search` for patient.

Description	The test focuses on the functionality of the `Search Healthcare Professionals` pop-up window.
Approach	User selects `Report Incident` from the navigation bar and *Search Person* button.
Input	1. User presses *Search Person* button 2. User inputs healthcare professional's ID number in *ID-number* text field 3. User presses *Search* button 4. User selects a row by pressing a row 5. User selects a row by checking a checkbox 6. User presses *OK* button 7. User presses *Add* button before the location search 8. User presses *Add* button after the location search 9. User presses *Reset* button 10. User presses *Cancel* button
Expected output	1. A pop-up window for healthcare professionals search appears 2. It is possible to input text in the text field 3. A table with search results appears 4. A variable is selected 5. A variable is selected 6. The pop-up window closes and the results are shown in `Report Incident` interface 7. The *Add* button is disabled 8. The *Add* button is enabled and variables added to a text field in the pop-up window 9. The variables from the text field in the pop-up window are erased 10. The pop-up window is dismissed and non of the variables appear in the `Report Incident` interface
Result	All test results have the expected output.

Table 5.7. Test of the UC2: `Search` for healthcare professionals.

Due to lack of connection to the IR database, the variables used for the test were predefined and attached to the functionality of the buttons in the pop-up window.

Test of `UC3: Statistics`

The performed test of the `Statistics` use case is based on use case specifications presented in table C.2 and is described in the table 5.8. The use case main functionality was to allow the user to search and visualization of the reported incident data.

Description	Test of possible interactions of the user with the `Statistics` interface.
Approach	User opens the IR system and selects `Statistics` from the navigation bar.
Input	1. User chooses a date from which the statistics will be shown 2. User chooses a date to which the statistics will be shown 3. User wants to choose as a date a day after tomorrow 4. User chooses Serious and Deadly as severity rates 5. User chooses Surgery and Infection as incident types 6. User chooses bar and line graph types 7. User searches for the wanted location 8. User presses the *Search* button 9. User presses the *Reset* button
Expected output	1. A calendar function appears 2. User cannot choose a date before the chosen *“from”* date 3. User cannot selects a date later then the current date 4. The chosen severity rates are selected 5. The chosen incident types are selected 6. The chosen graph types are selected 7. The searched location appears in the `Statistics` interface 8. Graphs of the selected data appear 9. Selection of the variables and plot are erased
Result	All test results have the expected output.

Table 5.8. Test of the UC3: `Statistics`.

Due to lack of data and connection to the IR database, it was not possible to fully test the possibilities of the use case. Therefore, an image of the plotted incident data was used in stead to illustrate the functionality of the IR system. Functionality of the *Save* button was not tested, since it was not implemented. The used search for location was described in the section 5.4.1.

Test of `UC4: Extract Data`

The performed test of the `Extract Data` use case is based on use case specifications presented in table C.3 and is described in the table 5.9. The use case main functionality was to allow the user to search, extraction and access to the incident reports.

Description	Test of possible interactions of the user with the `Extract Data` interface.
Approach	User opens the IR system and selects `Extract Data` from the navigation bar.
Input	1. User chooses a date from which the extracted reports will be shown 2. User chooses a date to which the extracted reports will be shown 3. User wants to choose as a date a day after tomorrow 4. User chooses Serious and Deadly as severity rates 5. User chooses Surgery and Infection as incident types 6. User chooses *Open* as a report status 7. User searches for the wanted location 8. User presses the *Search* button 9. User presses on one of the table rows 10. User presses the *Reset* button
Expected output	1. A calendar function appears 2. User cannot choose a date before the chosen *"from"* date 3. User cannot selects a date later then the current date 4. The chosen severity rates are selected 5. The chosen incident types are selected 6. The chosen report status is selected 7. The searched location appears in the `Extract Data` interface 8. A table of reports based on the selected data appears 9. A pop-up window with a full incident report appears 10. Selection of the variables and table are erased
Result	All test results have the expected output.

Table 5.9. Test of the UC4: `Extract Data`.

The use case was not tested fully due to lack of data and connection to the IR database. Therefore, a table with predefined data was created to illustrate the expected functionality of the use case. Additionally, the functionality of the *Save* button was not tested, since it was not implemented. The used search for location was described in the section 5.4.1.

Test of `UC5: Verify User`

The performed test of the `Verify User` use case is based on use case specifications presented in table C.4 and is described in the table 5.10. The use case main functionality was to verify the login details of the user. The verification aimed to ensure privacy and safety of the reported incident data.

Description	Test of possible interactions of the user with the `Login` pop-up window.
Approach	User opens the IR system and selects `Login` from the navigation bar.
Input	1. User inputs *"test"* as a username 2. User inputs *"test"* as a password 3. User presses *OK* button 4. User presses *Cancel* button
Expected output	1. It is possible to input the text in the *Username* text field 2. It is possible to input the text in the *Password* text field 3. User logs in to the system 4. The `Login` pop-up window is dismissed
Result	All test results have the expected output.

Table 5.10. Test of the UC5: `Verify User`.

The verification of the user was not fully possible due to the lack of connection to the IR database, where the entered details would be verified and processed when the user presses the *OK* button.

5.4.2 Planned IR Database Test

The absence of data from the Danish DPSD database has made it impossible to execute test on the IR database. Therefore a description is provided of how the test should have been conducted.

Based on the DPSD database problems provided in the section 2.4, the importance of testing the IR database against the problems with the current DPSD database is essential. Therefore, the tests should cover the following subjects:

- The time it takes from query to display/extract data
- Verifying outgoing data from the query
- Verifying incoming data
- All incident data are extracted from the IR database

The performance of the IR database should be tested accounting to how long it takes from an asked query until the data is provided to the user as either a plot or a table. The tested queries should both be complex and simple, because a complex query provides the user with a small amount of specified data, whereas the simple query would consist of a large amount of unspecified data. Therefore, to test the IR database's performance, a time recording should be performed.

The data which is being displayed for the user in the `Data Extraction` and `Statistics` interfaces must match the user requested variables. Lack of precise match may cause miscalculation and misunderstanding of the reported incidents. In order to perform the test, it is important to compare the data which should be displayed with the data being displayed.

The reported incident should be saved correctly, to make data extraction as smooth as possible. Therefore, when the user submits the reported incident or manually enters a new medicine/medical devices, the reported data and the new medicine/medical devices should be saved with the correct data structure and information in the document. To perform this test, it is essential to compare what should be stored to what is being stored in the `Incident Database`.

When a query is executed, it is essential that the requested data includes all of the data requested in the query, that is being extracted from the `Incident Database`. The rest should be performed to ensure that the query would not only find 115 documents if there are 125 documents stored in the `Incident Database`. Therefore, it is necessary to compare how many documents were extracted in relation to which documents were extracted.

CHAPTER 6

SYSTEM VALIDATION

Within this chapter, a description of the performed system validation and its results were presented. The validation was performed with a validation protocol and an interview with validation participants from Aalborg University Hospital and Aalborg University at the Department of Health Science and Technology. After the description of the performed validation of the IR system an introduction to the planned validation for the IR database was presented.

6.1 Validation of the IR System

The validation was performed to observe how well requirements for the IR system were implemented regarding the design principle and usability theory, The IR system's validation was only performed with focus on usability of the IR system for users with different professional backgrounds. The validation of the IR system with the selected group was performed in order to identify possible errors and changes, which need to be made before possible implementation of the IR system in the Danish healthcare system.

The validation of the system was conducted with 4 participants: two from Aalborg University Hospital and two from the department of Health Science and Technology at Aalborg University. The variation of the validation group was chosen to create an overview of the IR system from different healthcare perspectives, which thus can provide a broader impression of the IR system with many different angles.
The participants' professional backgrounds are:

- Two trained nurses with a Master in Clinical Science and Technology and currently PhD fellows at the Department of Health Science and Technology, Aalborg University.
- Two doctors from Aalborg University Hospital: a specialized cardiologist and a specialized anesthesiologist.

The validation of the IR system was performed at Aalborg University and in the doctors' offices to provide a familiar and comfortable environment, to ensure that the individuals would not be affected and distracted by unknown or new surroundings.

The participants performed a validation of the IR system by performing tasks described in a validation protocol and by participating in a short interview at the end of each validation. The validation protocol provided an overview of possibilities in the IR system functionality. Since the IR system was not connected to the IR database and thereby missing possible access to the incident data, a small amount of data was manually added to provide a visual representation of the IR system's functionality for the participants. The interview with the participants was based upon questions related to the theories described in B.3.3. The questions were regarding layout, structure, functionality and provided information on the `Report Incident`, `Statistics` and `Data Extraction` interfaces.

6.1.1 Summary of the IR System Validation

The performed validation form was the usability test described in section B.3.5. The participants were asked to perform reporting of a incident, search for statistic and extract data related to the reported incidents. During the validation, all conversations were recorded to ensure that all information provided by the participants would be analyzed and documented in the report.

The opinions expressed by the participants were transcribed to get an overview of the participants' comments to the implemented IR system.

To ensure privacy and anonymity of the participants, their comments were marked as:

- `[NURSE1]` - One of the trained nurses
- `[NURSE2]` - Second of the trained nurse
- `[DOCTOR1]` - One of the doctors
- `[DOCTOR2]` - Second of the doctors

`[NURSE1]:`
The IR system is logical, not complicated to navigate through, and it does not take long time to search for a location. Furthermore, the used color scheme fits well to the hospital environment.

The pop-up windows have a misleading functionality since the user does not expect that they should press *Add* button before pressing *OK* button. Therefore, the pop-up window should include a help text indicating that the user should remember to press *Add* button before *OK* button, or it should be possible for the user to only press *OK* button, to get the selected variables appearing in the gray text felt. Furthermore, the order of the buttons *Add* and *OK* are confusing and should therefore be rearrange. The *Cancel* button seems to be unnecessary, since is it possible to close the window with a cross in the top-right corner. Additionally the *OK* button should be inactive until the *Add* button is pressed and the *Search location* button should be renamed to either "search place'" or "search department".

To increase the usability it would be good to indicate in the `Report Incident` interface that the reporter may only choose one of the e.g. incident types. Furthermore some of the help hover bubbles in `Report Incident` should be explained further, e.g. in severity rate - *Deadly*.
The scroll bar should be more visible for the user and the calendar functions is misleading since it is not possible to click on the calendar icon. Therefore a consistency must be created throughout all of the interfaces.
Missing standardization of language in the buttons, since some of the buttons is written in English instead of Danish. Also in the plot type in the `Statistic` interface the user do not known what a bar plot is, so it should be change to histogram instead.

The pop-up windows are missing more guidance regarding what user should/can do in the different search functions and also indicate that the user may add more than one variable in the search functions.
More color could be added to the severity level e.g. red, yellow and green to indicate seriousness of the severity rate. A *Back* button or *Go to Home Page* to indicate a clear exit for the user is missing as well.
It could be good if in the table `Extract Data` to have a possibility to sort the reported incidents according to the user's needs e.g. to the incident type or severity rate.
A focus indication of the required fields would be nice for guiding the user through the interface `Report Incident` and thereby allow immediate writing and marking in the fields.

`[NURSE2]:`
The overall view is really good, simple and works well with great help all the way through the IR system. The design looks good, is easy to use and has a good text size. The possibility to choose between patient and healthcare professionals, and the use of the healthcare language is well implemented. The IR system is much simpler and more manageable for a new user, than the current incident reporting system. The IR system is very intuitive, easy to navigate, and it is good that the interfaces do not remember what is written earlier, since the user may quickly get to pick the wrong one.

There are some difficulties to know that the hover bubbles exist and it may be disturbing that the hover bubbles are flying around in the `Report Incident` interface. It would be better if there was a text description about the existence of hover bubbles, or when the user clicks on the option a short description will appear below. Additionally the hover bubbles should be centered to the left to make it more consistent.
The calendar function in `Report Incident` interface is misleading, since it is not possible to press the calendar icon but instead need to tap in the text field. The possibility of not being able to report incidents ahead is implemented well. However, the calendar function should be consistent in all the interfaces.
The healthcare professionals are very cautious regarding the information they provide, so it is good with the pop-up window in the last part of the incident form. By using the hyphen in the CPR number, possible doubt and inconsistency will be removed. Also possibility to search for department code is great. However the *Search* and *Cancel* button

should be the only visible buttons when the window shows for the first time and the *OK* button should be disable until the *Add* button has been pressed.
The table extraction is brilliant, but there is a slight doubt about how to extract data. It is however good that the user has the possibility to print single incident out. Furthermore, the scroll function should be larger in the interfaces and the language should be unidirectional.
More consistency is needed regarding what is a button in the IR system. The buttons should have a specific color, which indicate it is a button. However, it is good that the color changes when the mouse is over the buttons.

It would be great if there was an exit button so it is possible to go back to the home page. Also a functionality to see the completed data before it is submitted to the database would be beneficial to draw attention to possible mistake. The possibility to both write the date manually and by pressing a calendar icon would be good. Furthermore, there is missing indication with respect to that it is a reliable website in terms of e.g. the National Health Service logo and when the website was last updated. There is also a missing information that it is possible to choose more healthcare professional involved in the incident.

`[DOCTOR1]`
The IR system is very user-friendly, clear, logical, easy to read and easy to use by using the provided help functions. The help hover bubbles are very useful and good with the definitions and explanations in the severity rate and incident type, but also in the help provided through out the IR system. There are provided clear information and definitions in relation to how and where to search. The incident report form is very short and fast to report an incident due to all of the provided help. The visualization of the data is really good and the used text color works well because black and white are the best colors for the eyes.

The old website is missing overview and it is not possible to see old reports that could be used for statistics, training, conclusions, congresses and conferences. So it is great to have the retrospective overview for the incident at my department, like shown in the IR system as both view of curve and bar diagram, and with the table including the specific incidents. The incident report should be quick and fast because the nurses are very overloaded with their job tasks and responsibilities. The old website is not quick and fast, as the IR system seems to be.

`[DOCTOR2]`
The IR system is very clear, intuitive, easy and lot less complicated to use than the current website. It is very convenient, easy to use and intuitive so definitely it would be possible to fill out the incident form without any help.
It is useful, easy to learn and good that there are not so many text fields. Good with the very little amount of text in the IR system because there is a possibility that the text is not being read.

It takes not long time to report an incident in the IR system. It is very important that it is fast to report an incident, otherwise the staff begins to consider if it is really an incident or not preventing them from reporting the incident at the end.

Result Summary

The results are based on the collected validation results provided in section 6.1.1. All of the participants' statements from the IR system validation are gathered and divided into positive and negative feedback and missing features. The validation results are presented in the table 6.1.

Table 6.1. Result summary of the feedback from the IR system validation.

Feedback type	Result
Positive Feedback	Logical, simple, clear, convenient, uncomplicated and user-friendly
	Easy to use and learn
	Intuitive and easy navigation
	Correct color choices and overall good looking design
	Short and fast reporting time
	Easy and great search functions
	Very useful and good help hover bubbles
	Clear, helpful and good amount of information provide
	Really good visualization
	Great and brilliant with the retrospective overview
	Manageable for a new user
	Remove possible doubt in the search function with CPR-number
Negative Feedback	Restructure of the pop-up windows
	Rename *Search location* button and the options *bar diagram*
	Need more information on how to display the hover bubbles
	Need more information on how to choose *severity* rate and *incident type*
	More visible scroll bar
	Misleading and inconsistency in the calendar functions
	Standardization of language in the interfaces
	Consistency in button appearance
Missing Features	More guidance regarding search options in pop-up windows
	More guidance regarding how to extract data
	Exit button to go back to start page
	Possibility to sort the table in `Extract Data` according to the user choice
	More colors in the severity selection
	Indication of most important fill-in fields in `Report Incident`
	See the completed data before submission

Continued on next page

Table 6.1 – continued from previous page

Feedback type	Result
	Possibility to use both manual writing and calendar function Visible references to reliable healthcare websites Change hover bubbles to text appear below the click option

6.2 Planned IR Database Validation

As described previously, the missing data from the Danish DPSD database has made it impossible to perform a validation of the IR database. Therefore a description of how the validation should have been conducted is given.

Validation of the IR database could be performed by comparing the current Danish DPSD database with the proposed IR database by executing the same tasks as described in section 5.4.2. The validation results could be evaluated and the advantages and disadvantages of the two databases for the IR system could be identified and verified.

Part III

Synthesis

CHAPTER 7

DISCUSSION

Within the project, a possible solution to the problem statement in chapter 3 was researched and developed. The results of the validation, identified problems, used methods and limitations of the developed IR system and IR database were discussed within this chapter.

The IR system was developed as a proof of concept aiming to aid the healthcare professionals with reporting procedure, and provide them with means for statistical analysis and data extraction of the incident data.

The implemented IR system was validated to verify, if the focus points and specified requirements were met. Based on the study by Holzinger [2005], usability in the IR system was evaluated using usability test technique. Since it was important to test if the system would provide a more clear and intuitive reporting form, validation of the IR system with the possible end users was chosen. However, by using the usability test alone, the implementation of usability theory was not inspected. Therefore, based on the studies by Jeffries and Desurvire [1992] and Holzinger [2005], a heuristic analysis could be additionally used as an inspection method, where the usability principles would be evaluated by the usability experts. The heuristic analysis would be beneficial for the developed IR system, since it would verify correctness of the proposed design and ensure that the usability theory is correctly implemented for a system being developed for the healthcare.

The validation participants' job positions and age varied, to make evaluation of utilization of the system by people with different professional background and IT skills. The documented validation results of the IR system are believed to represent well the variety in healthcare professionals at a hospital. However, more administrative participants from e.g. Patientombuddet, or risk managers at the hospital and regional level were missing in the validation group. Therefore, the statistical and data extraction aspects were not evaluated fully, and the IR system requires additional usability tests performed with the administrative participants.

The suggested interface for the incident reporting was based on the comments gathered through the interviews and Patientombuddet [2013] report. The comments regarding possible changes in the incident form were combined with the usability theory and principles for a good design. The validation results related to the incident reporting and consisted mainly of positive feedback to the implemented `Report Incident` interface. The participants stated that the reporting form is clearer, simpler and easier to navigate than in the current DPSD system. Due to reduced typing requirements, the reporting procedure could be performed faster and more intuitively, than with the current reporting form requiring a lot of clicking and selections. Faster reporting procedure with the IR system could improve the reporting culture and make the reporting throughout the healthcare professionals more accessible and likely. Additionally, the design in the IR system could reduce the stress of the reporting procedure for the healthcare professionals, and would be easier to introduced and learned by the potential users. Some minor design changes related to pop-up windows were required, which should be implemented in the future. Furthermore, since the validation results documented a disagreement between the participants regarding the amount of provided text in the IR system, additional validations should be performed to find the correct balance in the required text.

The `Statistics` interface was developed to provide statistical means to the users, allowing them to visualize data, improve awareness of the incidents and be used for providing feedback to the healthcare professionals. A similar visualization of data was performed by Ratwani and Fong [2014], where dashboards for data exploration and monitoring of incident patterns were developed and provided positive feedback within the healthcare professionals. The data visualization in the IR system was documented in the validation results as a useful functionality, which allowed a quick overview of the incident data. Since it would be useful to present and evaluate the incident data during briefings and meetings e.g. at the hospital, a broader user access to the interface was suggested. However, the broader access would require additional adjustments regarding the data security and privacy at the hospital.

The data extraction and analysis part focused on extracting the incident reports and their analysis from the IR system. The validation results were positive regarding the `Extract Data` interface, where it was commented as being useful tool for a retrospective analysis of the incident reports. Enhancement of the data privacy to ensure that the patients' and reporter's information are erased, would be required if the access to the incident reports would be extended to other users. However, the interface would be useful for case studies, or performing an analysis for possible improvements of the procedures.

Requirements of DPSD database optimization was based on the performed interviews, which indicated that search for data takes time and the extracted data may not represent all of the existing reports, thus the learning from the incidents was reduced. Therefore, during the implementation a new solutions for database the existing relational database could be replaced with a non-relational database. As described by Leavitt [2010], the non-relational database have higher performance on large data sets, scalability and flexibility rate compared to the relational database, which could be beneficial for the healthcare system. The higher performance would potentially reduce time of the data extraction, while scalability and flexibility would be useful features for changing and growing incident data. However, the non-relational databases are a new concepts in contrast to well established relational databases. Therefore, implementation of the non-relational database in the healthcare system could potentially improve the IR system. However, the transition of the incident data from one type of database to another could be time consuming and problematic at the beginning, since it would require extraction of the data from the relational database.

The chosen non-relational database for the IR database optimization was a MongoDB, which was described by Nayak et al. [2013] as a document-based database with high performance, and well chosen for document storage and content management. Alternatively, a column-oriented database, could be used for the IR database. According to Nayak et al. [2013], the column-oriented database, such as the BigTable, is advised for data mining and analytic applications, since it provides high scalability in data storage. The data mining aspect of the column-oriented database could be useful for the IR database, especially for the incident data analysis and pattern identification. However, since the suggested design for the IR database was not implemented and tested, it was not possible to evaluate further which one of the databases would be more suitable for the IR system.

7.1 Limitations

The knowledge from the performed interviews on the hospital and regional levels, was based on a limited number of interviewees, and only focused and represented the problems in Region North Jutland. For deeper understanding of the problems related to the incident reporting, more interviews with other hospitals and regions should be performed. The further interviews could provide further knowledge regarding the use of the DPSD system, and understanding of organizational structure for processing of the incident reports at the hospitals.

Due to lack of provided incident data, it was not possible to fully understand the used management methods regarding the data in the DPSD database, and to fully develop and implement a new database solution in form of non-relational database. The lack of the IR database implementation limited the implementation, testing and validation possibilities of the developed proof of concept for IR system. Therefore, it was not possible to verify, if the proposed IR database design improved the current problems with the DPSD database.

Chapter 8

Conclusion

The performed interviews and literature search documented multiple problems related to reporting of incidents in the healthcare system. The identified issues in the problem analyze contributed to three main problems: reporting of incidents, feedback on the reported incidents, and data extraction and analysis of the reported incidents. Based on the gathered knowledge the following problem was stated:

> *How could a system aiding reporting procedure and supporting feedback to healthcare professionals for the Danish healthcare system be designed, developed and validated?*

To solve the stated problem, an optimization of the DPSD system was designed and developed using web-development tools, usability theory and principles for good interface design. The implemented IR system was designed to be a proof of concept system. The validation results proved that the IR system fulfills clear appearance, easy use and fast navigation requirements.
Means for data exploration, visualization and feedback from the reported incidents were developed as a proof of concept, tested and validated. The IR system provided the user with the opportunity to investigate and statistical analyze the reported incident, which during the validation proved to be a proper and adequate solution.
The DPSD database was optimized and restructured using a MongoDB design to enhance data extraction and feedback. The design of the IR database seems to be well carried out, since it provides possibility for higher performance on large data sets, scalability and flexibility rate.

Based on the preformed study the next step would be to connect the IR database with the IR system and transfer the Danish incident reporting data form the DPSD database to the IR database. Furthermore, to inspect the proposed solution further validations and inspections are recommended.

BIBLIOGRAPHY

Indre Brasaite, Marja Kaunonen, and Tarja Suominen. Healthcare professionals' knowledge, attitudes and skills regarding patient safety: a systematic literature review. Scandinavian journal of caring sciences, pages 1–21, 2014. doi: 10.1111/scs.12136.

John Tingle. The global problem of adverse patient safety incidents in health care. British Journal of Nursing, 20(10):642–643, 2011.

Stuart Emslie, Kirstine Knox, Martin Pickstone, and Great Britain. Improving Patient Safety: Insights from American, Australian & British Healthcare: Based on the Proceedings of a Joint ECRI and Department of Health Conference to Introduce the National Patient Safety Agency. ECRI, 2002.

Bill Runciman and Merrilyn Walton. Safety and ethics in healthcare: a guide to getting it right. Ashgate Publishing, Ltd., 2007.

Sundhedsstyrelsen. Vejledning om rapportering af utilsigtede hændelser i sygehusvæsenet. Technical report, Sundhedsstyrelsen - National Board of Health, 2003.

Patientombuddet. Årsberetning 2013, dansk patientsikkerheds database del 1. Technical report, Patientombuddet, 2014a.

World Health Organization. Data and statistics, 2015. URL `http://www.euro.who.int/en/health-topics/Health-systems/patient-safety/data-and-statistics`.

Region Nordjylland. Utilsigtede hændelser i region nordjylland - årsrapport 2013. Technical report, Region Nordjylland, 2013.

Paul Barach and Stephen D Small. Reporting and preventing medical mishaps: lessons from non-medical near miss reporting systems. Bmj, 320(7237):759–763, 2000.

Eduardo Ortiz, Gregg Meyer, and Helen Burstin. Clinical informatics and patient safety at the agency for healthcare research and quality. Journal of the American Medical Informatics Association, 9(Suppl 6):S2–S7, 2002.

Joan S Ash, Marc Berg, and Enrico Coiera. Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. Journal of the American Medical Informatics Association, 11 (2):104–112, 2004. doi: 10.1197/jamia.M1471.

RP Mahajan. Critical incident reporting and learning. British journal of anaesthesia, 105 (1):69–75, 2010. doi: 10.1093/bja/aeq133.

R Lawton and D Parker. Barriers to incident reporting in a healthcare system. Quality and Safety in Health Care, 11(1):15–18, 2002.

WHO. Who draft guidelines for adverse event reporting and learning systems. Technical report, World Health Organization, 2005.

Farah Magrabi, Mei-Sing Ong, William Runciman, and Enrico Coiera. An analysis of computer-related patient safety incidents to inform the development of a classification. Journal of the American Medical Informatics Association, 17(6): 663–670, 2010. doi: 10.1136/jamia.2009.002444.

Ka-Chun Cheung, Patricia MLA van den Bemt, Marcel L Bouvy, Michel Wensing, and Peter AGM De Smet. A nationwide medication incidents reporting system in the netherlands. Journal of the American Medical Informatics Association, 18(6): 799–804, 2011. doi: 10.1136/amiajnl-2011-000191.

Raj M Ratwani and Allan Fong. Connecting the dots: leveraging visual analytics to make sense of patient safety event reports. Journal of the American Medical Informatics Association, pages amiajnl–2014, 2014.

Kevin EK Chai, Stephen Anthony, Enrico Coiera, and Farah Magrabi. Using statistical text classification to identify health information technology incidents. Journal of the American Medical Informatics Association, 20:980–985, 2013. doi: 10.1136/amiajnl-2012-001409.

Sue M Evans, JG Berry, BJ Smith, A Esterman, P Selim, J O'Shaughnessy, and M DeWit. Attitudes and barriers to incident reporting: a collaborative hospital study. Quality and Safety in Health Care, 15(1):39–43, 2006.

Catherine M Tighe, Maria Woloshynowych, Ruth Brown, Bob Wears, and Charles Vincent. Incident reporting in one uk accident and emergency department. Accident and emergency nursing, 14(1):27–37, 2006. doi: 10.1016/j.aaen.2005.10.001.

DPSD Dansk Patientsikkerhedsdatabase. Temarapport 2009: Arbejdsmiljœ og utilsigtede hændelser i sygehusvæsenet. Technical report, Sundhedsstyrelsen, 2009.

Jag Ahluwalia and Lin Marriott. Critical incident reporting systems. In Seminars in Fetal and Neonatal Medicine, volume 10, pages 31–37. Elsevier, 2005.

Marieke Kessels-Habraken, Jan De Jonge, Tjerk Van der Schaaf, and Christel Rutte. Prospective risk analysis prior to retrospective incident reporting and analysis as a means to enhance incident reporting behaviour: A quasi-experimental field study. Social Science & Medicine, 70(9):1309–1316, 2010.

Patientombuddet and Ministeriet for Sundhed og Forebyggelse. Rapport om serviceeftersyn af rapporteringssystemet for utilsigtede hændelser. Technical report, Patientombuddet and Ministeriet for Sundhed og Forebyggelse, 2014.

Sundhedsstyrelsen. Vejledning om rapportering af utilsigtede hændelser i sygehusvæsenet m.v. Technical report, Sundhedsstyrelsen, 2011.

Kvalitetskontoret. Program for håndtering af utilsigtede hændelser i region nordjylland. Technical report, Region Nordjylland, 2011.

Aalborg Universitetshospital. Oversigt over klinikker og specialer, 2014. URL `http://www.aalborguh.rn.dk/Genveje/Om-Aalborg-Universitetshospital/Oversigt-over-klinikker-og-specialer`.

Kaveh G Shojania, Bradford W Duncan, Kathryn M McDonald, Robert M Wachter, and Amy J Markowitz. Making health care safer: a critical analysis of patient safety practices. Agency for Healthcare Research and Quality Rockville, MD, 2001.

Patientombuddet. Årsberetning 2013. Technical report, Patientombuddet, 2014b.

Linda Drupsteen, Jop Groeneweg, and GIJM Zwetsloot. Critical steps in learning from incidents: using learning potential in the process from reporting an incident to accident prevention. International Journal of Occupational Safety and Ergonomics, 19 (1):63–77, 2013. URL `http://www.ncbi.nlm.nih.gov/pubmed/23498711`.

J Benn, M Koutantji, L Wallace, Peter Spurgeon, M Rejman, A Healey, and C Vincent. Feedback from incident reporting: information and action to improve patient safety. Quality and Safety in Health Care, 18(1):11–21, 2009. doi: 10.1136/qshc.2007.024166.

EH Bradley, ES Holmboe, JA Mattera, SA Roumanis, MJ Radford, and HM Krumholz. Data feedback efforts in quality improvement: lessons learned from us hospitals. Quality and Safety in Health Care, 13(1):26–31, 2004. doi: 10.1136/qshc.2002.4408.

Louise M Wallace, Peter Spurgeon, Jonathan Benn, Maria Koutantji, and Charles Vincent. Improving patient safety incident reporting systems by focusing upon feedback–lessons from english and welsh trusts. Health Services Management Research, 22(3):129–135, 2009. doi: 10.1258/hsmr.2008.008019.

Amanda Hall and Graham Walton. Information overload within the health care system: a literature review. Health Information & Libraries Journal, 21(2):102–108, 2004.

Patientombuddet. Analyse af spœrgeskemaundersœgelse, projekt fokuseret rapportering, dansk patientsikkerhedsdatabase. Technical report, Patientombuddet, 2013.

Patientombuddet. Evaluering af projekt fokuseret rapportering. Technical report, Patientombuddet, 2014c.

Abraham Silberschatz, Henry F Korth, and S Sudarshan. Database system concepts. McGraw-Hill New York, 6th edition, 2011. ISBN 978-0-07-352332-3.

ABM Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. arXiv preprint arXiv:1307.0191, 2013.

Andreas Holzinger. Usability engineering methods for software developers. Communications of the ACM, 48(1):71–74, 2005.

Robin Jeffries and Heather Desurvire. Usability testing vs. heuristic evaluation: was there a contest? ACM SIGCHI Bulletin, 24(4):39–41, 1992.

Neal Leavitt. Will nosql databases live up to their promise? Computer, 43(2):12–14, 2010.

Ameya Nayak, Anil Poriya, and Dikshay Poojary. Type of nosql databases and its comparison with relational databases. International Journal of Applied Information Systems, 5(4), 2013.

J. Ritchie, J. Lewis, P.S.P.J. Lewis, C.M.N. Nicholls, and R. Ormston. Qualitative Research Practice: A Guide for Social Science Students and Researchers. SAGE Publications, 2013. ISBN 9781446296202. URL https://www.google.pl/books?id=EQSIAwAAQBAJ.

S. Kvale and S. Brinkmann. InterViews: Learning the Craft of Qualitative Research Interviewing. SAGE Publications, 2008. ISBN 9780761925415.

Jim Arlow and Ila Neustadt. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. Pearson Education, 2 nd. edition, 2005.

Wilbert O Galitz. The essential guide to user interface design: an introduction to GUI design principles and techniques. John Wiley & Sons, 2007.

W3C. Level 1 document object model specification, 1998. URL http://www.w3.org/TR/WD-DOM/.

W3C. Html & css, 2014. URL http://www.w3.org/standards/webdesign/htmlcss.

MDN. Javascript overview, 2014. URL https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/JavaScript_Overview.

The jQuery Foundation. What is jquery?, 2014. URL http://jquery.com/.

Joseph S Dumas and Janice Redish. A practical guide to usability testing. Intellect Books, 1999.

David Hows, Eelco Plugge, Peter Membrey, and Tim Hawkins. The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB. Apress, 2013.

Katarina Grolinger, Wilson A Higashino, Abhinav Tiwari, and Miriam AM Capretz. Data management in cloud environments: Nosql and newsql data stores. Journal of Cloud Computing: Advances, Systems and Applications, 2(1):22, 2013.

Kristina Chodorow. MongoDB: the definitive guide. " O'Reilly Media, Inc.", 2013.

Part IV

Appendix

Appendix A

The Outcome from Interviews in the Danish Healthcare System

To gain insight knowledge behind reporting system for processing and analyzing incidents, qualitative interviews have therefore been conducted with various perspective and on different levels of the Danish healthcare system.

A.1 Meaning Condensation of the Interviews with Risk Managers

Interviews with the risk managers were performed between 11.02.2015 and 19.03.2015. The described meaning condensation was based on interviews with four risk managers: one from Patientombuddet and three from Region North Jutland. The three risk managers from Region North Jutland were responsible for different organizational levels in the healthcare system:

- Risk manager for all hospitals in Region North Jutland.
- Risk manager for the AalborgUH.
- Risk manager for the Oncology Department at AalborgUH.

References the different interviews were marked as follow:

- `[PAT]` - Patientombuddet
- `[RMRN]` - Risk manager for all hospitals in Region North Jutland.
- `[RMAUH]` - Risk manager for the AalborgUH.
- `[RMOD]` - Risk manager for the Oncology Department at AalborgUH.

A.1.1 Understanding of the Reporting Procedure

Reporter fills the reporting form on DPSD website and describes the incident. After the incident type is classified, a specific sub-group or problem connected to the incident is chosen together with the severity rate. The reporter chooses location where the incident occurred, where afterward the report is send to, e.g. hospital, clinics, [RMRN]. There are 10 possible receivers of the incident reports at the AalborgUH: risk managers of the AalborgUH and eight clinics. Each of the clinics has its own quality coordinator and financial assistant, [RMAUH]. If the reporter chooses not a specific hospital, then the incident is sent to the regional risk manager, whom sorts the incidents and sends them to their correct location for processing of the incident, [RMRN]. The staff responsible for analyzing the incident is found in all of the AalborgUH's departments, [RMAUH].

A.1.2 Obtaining Knowledge About the Scale of Incident Reporting

Every staff in the healthcare system reports a incident. There are many nurses reporting the incidents, as well as doctors and doctor's secretaries. However, there are not many reports from patients and relatives, possibly due to their lack on knowledge about the reporting possibility, [RMRN]. There are approximately 2.500 incidents reported yearly at AalborgUH, [RMAUH]. Medication incident types have the highest prevalence of all incidents, [RMRN].

A.1.3 Understanding of the Work Environment and Processes

The healthcare staff must report the incident within seven days from the occurrence date of the incident. It is valid for both adverse events and near-misses. Since there are no processing plans for all incidents, processing time can vary from one day to a longer period of time, [RMRN]. Most of the healthcare staff knows the reporting procedures, [RMAUH]. Therefore, courses about incident reporting are organized for newly hired healthcare staff during a central introduction, [RMRN]. It is risk managers' responsibility to teach the staff from the clinics about how the incidents should be processed, [RMAUH]. Furthermore there are key persons on each department, that can help with incident reporting, [RMRN]. However, no manual on how to report an incident exists, [RMOD].

During processing of the incidents, one has to be humble, in order to not discourage healthcare professionals from reporting, [RMOD]. Processing of the incidents is made by healthcare staff from the different clinics at AalborgUH, for whom the incident processing is an additional task to their job responsibilities. The meeting on which the incidents are processed and analyzed are usually interdisciplinary and consist of different work groups, e.g. doctors, nurses. The date of the meeting for processing of the incident depends on the amount of incidents needed to be analyzed. The results from the analysis have to be reported to the leaders of the groups, [RMAUH]. All leaders from the clinics and quality coordinators receive once per month a report describing all of the incidents from the month. There are yearly or quarterly meetings with special leadership, where the incidents are a part of the order for the meetings. A meeting across the departments are organized to discuss the incidents with each other, [RMOD].

Once a year, the risk managers at the AalborgUH go through all of the reported incidents in order to analyze them and search for patterns and tendencies of the incidents. It is done to see the whole picture of the incidents at the AalborgUH, since analysis of the individual departments may not provide the clear representation of the problem, [RMAUH].

A.1.4 How the Reporting System Works and is Used

The given title of the incident is used to identify to which group the incident belongs and who processes the incident. Within the DPSD program, incident types changes according to the chosen location. When the incident type is chosen, it is further possible to choose a situation, which fits the incident best. Under description of the incident, the incident is described with words. However, the shorter the description, the more probable it is that the incident would be dismissed due to lack of information. Each of the incident reports has a location code, allowing to know whom the report should be send for processing, [PAT]. It is possible to report the incident anonymously or provide contact details, which makes it feasible to contact the reporter for more details related to the incident. It is also possible to provide only their work position, [RMRN]. When the CPR of the patient is used, the program calculates age and gender of the patient. It is also possible to define the gender manually, e.g. by using “K” or “k” for a woman. Besides age and gender, all patient and reporter’s details are deleted from the database to protect safety and are not used in further analysis. There are also some who dismiss the incident report if patient and reporters details are not provided, because it is not possible to go further with processing of the incident, [PAT].

When a report has to be made on the incidents, a database search is performed in order to present statistics of the incidents in the DPSD system. During the database search, it is possible to search within free text areas. Furthermore, based on the reported incidents, Patientombuddet can analyze and group the incidents, [PAT].

A.1.5 How the Incident Reports are Processed in Practice

The analysis process of the incidents is decentralized for the whole hospital area. Processing methods of the incidents depend on who reports the incident, its severity rate and the reporting culture in the different reporting sectors. Processing time of the incident reports varies and depends on analysis organization and incident types, [RMRN]. Additionally, length of the incident processing depends on how fast the processing group will meet again. Typically, the processing takes a few months since the incident is reported until it is processed. The processing groups consist of staff from the clinics, [RMAUH]. The individual processing groups can consist of a chief physician, doctor, secretary, nurse, chief nurse and an administrative nurse, [RMOD].

The meeting for processing of the incidents are organized typically once a month, however the meeting point depends on the amount of incidents needed to be analyzed, [RMAUH]. The initial receiver of the incidents reads the incident reports and checks both selected level of severity rate and classification of the incident, [RMOD]. Only the reporter knows precisely about what happened during the incident, therefore other documents are often needed for the processing group to better understand what happened during the incident. To each of the incident report, a description of the patient's hospitalization, prescriptions and notes are printed out, where the relevant information for the incident are highlighted and presented during the meetings of the processing group, [RMOD]. The processing of the incident can vary in type and be local (department, hospital region), or across the sectors. Incidents with highest severity rate are processed using root cause analysis (RCA) and with an additional help from e.g. hospital's risk managers. The RCA requires more resources, than other types of analysis, [RMRN]. The incidents are presented on a big screen, where the processing group reads the reports and sometimes checks information from Clinical Suite and OPUS. Discovered problems from the incident reports are described according to the identified causes, [RMOD]. A decision is made on how the problem related to the incident should be solved. When the incident analysis is done in DPSD, the incident is marked as closed and send to Patientombuddet, [RMAUH]. With many incidents of the same type, Patientombuddet can work on database to process the incidents together. During analysis of the incident, free text written by the reporter is analyzed with what have happened, how it happened, how to avoid its recurrence and what were the causes, [RMRN].

Many times, the reporter describes the incident with too little information, [RMAUH]. An interview with the involved staff in the incident is therefore used during analysis of the incident report, [RMRN]. The interview is performed to gather more information about the incident, if it is not described well enough, [RMAUH]. When the incident processing is finished, a chosen person from the processing group must report to the leader of the department about what happened during the incident. The meeting with the leader is hold every 14 days to follow up on the situation. Other members of the processing group are also required to talk with other staff members from their profession about the incidents, [RMOD].

A.1.6 How the New Procedures are made and Introduced at the Hospitals from the Incident Reports

The incidents are mainly used to adjust the process rather than to make new procedures. The AalborgUH risk managers and IT gather the incidents and make suggestions on how safety could be improved, [RMOD]. If there are many incidents with the same type, one can start an analysis of why the incidents happened, [RMRN].

A.1.7 Used Feedback Methods

There is always a possibility for better feedback method. In the past, it was a required that the results of the incident processing should be reported back to the reporter. However, the requirement was hard to fulfill, [RMAUH]. Nowadays there is not always a direct feedback to the reporter, [RMRN]. Since it is not expected to receive the feedback, some of the reporters receive feedback on the incident and some not, [RMAUH].

It is the staff's responsibility to provide feedback on the incident reports, e.g. incidents from one's department can be discussed personally, [RMRN]. It is however possible that the reporter misses the staff meeting where the incidents are evaluated and therefore miss the feedback on the incidents, [RMAUH].

Nowadays a bulletin for the staff is published four times a year, which describes the incidents. Furthermore, it shows to the reporter, that their reports made a difference, [RMAUH]. The bulletins provide information about an incident group, if there were many incidents of the same type, and what changes connected to the incidents were made. To provide an overview of the incidents and severity rates, the bulletins includes a statistical overview of the incidents for the staff, [RMOD]. However, the incidents across sectors require another form of feedback. Reporting patients and relatives can be told that even though they do not hear a feedback, does not mean that the incident is not processed, [RMRN].

A.1.8 Identification of the Possible Problems with the Incident Reporting System

There are some challenges with the analysis part of the incidents for the AalborgUH risk managers. It is possible that the reported incidents can become out of date (over 90 days old), which represents that organization was not efficient enough and results in challenges during processing of the incidents. The situation is different for Patientombuddet, which has to process a large amount of incident data, [RMAUH]. However, there are too many incident reports compared to the given time to process the incident, e.g. 25 reports needed to be processed during 2,5 hours once a month. Therefore a preparations of the incident reports to the meetings are needed, [RMOD].

A Canadian system is used for DPSD, which is developed for one hospital. However, since it was bought for the whole Denmark it has to cope with several units and not only one organization, which is challenging. Some parts of the DPSD are serviced by CSC, but if it requires further work, it has to be send to Canada, [RMAUH]. Work on database framework is in progress in order to improve the DPSD database, [RMRN]. Statistics and extraction of incidents' information could be improved, especially the extraction of the location where the incident occurred using SOR code, [RMAUH]. Data search can cause performance problems and take too much time. The causes of the prolong extraction are unknown, but it is estimated that it has something to do with the database structure. Furthermore, the system cannot process a search with big amount of data, e.g. a search for incidents from four years ago can take almost 20 minutes per search, [PAT]. It is also difficult to extract specific sub-items from the database. Besides a big manual explaining how to extract data from database, sometimes the incident data is not correctly extracted, [RMOD].

Instead of focusing on getting more reports, it is more important to focus on the quality of the reports. The focus should be on more reports from which it is possible to learn and create prevention culture within the healthcare staff, where they are thinking about why the incident occurred and what should be done if it occurs again, [PAT]. However, there are no commercials or campaigns, telling the patients that it is possible also for them to report an incident. Therefore, most of the reports from patients or relatives are the ones, which have heard about the incidents before. If the patients or relatives have a need for help with reporting, they can call Patientkontor/Patientdialog, where a patient supervisor can help them with reporting the incident, [RMAUH].

The severity rate should be reported based on the actual result of the incident on the patient. However, there are still some reports describing a potential outcome of the incident, [RMAUH]. It can happen that the reporter does not know the consequences of the incident during reporting of the incident, [RMRN]. It is hoped that instead of providing a severity classification, the severity rate of an incident could be based on a questionnaire that the reporter would fill out, [PAT].

It takes time to report the incident, so the reporting procedure needs to be simplified to make the staff willing to report the incidents. Moreover, the nurses have a tendency to report the incidents that they already heard about and forget about the incident types, which they did not heard about, [RMOD].

All the information given to the doctors and nurses has to be concise, since they are overloaded with different information daily and are limited to how much they can learn from the provided information. Therefore, a balance between knowledge and time has to be ensured and the information should be short and precise, [RMOD].

Section in the DPSD program, where it is possible to describe the involved medication is important. However, it can be optimized and be shown only when it resulted in harm of the patient. Moreover if in the incident are different medication forms included, then the search results present the incident individually for each type of the medication, [PAT].

A.2 Meaning Condensation of the Interview with Healthcare Staff

An interview with the healthcare staff was performed on 09.03.2015. The described meaning condensation within this chapter is based on interview with a leading nurse for the department nurses and is a member of the incident processing group as a representative of the nurses.

A.2.1 Working processes at AalborgUH

It is different within the staff on how often they report an incident. There are some who are very good in reporting the incidents, but there are also some who never reported an incident. Mostly it is the department nurses that fill the incident reports, since they are the ones that the personnel goes to in case more persons are involved in the incident. If a nurse creates the incident, then it is the nurse that reports it.

A.2.2 Factors on how the Incident Occurred

There are not enough resources to find solutions to prevent the incidents. Within the last few years, amount of the patients increased resulting in a very high tempo of working, which often results in incidents.

A.2.3 Staff's point of view on Reporting System

A incident is in relation to a patient. However, not all of the incidents are reported due to lack of time, time it takes to fill out the report and doubt if it is really an incident or other accident related to the working environment. Reporting procedure of the incident takes a long time, i.e. around 20 minutes if one goes into details or is inexperienced with the reporting procedure. The more simple report, the better, but it is also difficult to include all necessary information. Reporter has to know a lot about the incident to be able to report the incident to the reporting system. Therefore, it is not possible to report on behalf of another person, since the reporting form is very specific and requires a deep knowledge about the incident and the reporter must know details about the incident. It is mostly administrative nurse, who teaches the department staff about the incidents, e.g. what is important? where the focus should be?

Most reporters write their contact information, however there are still some who report anonymous. Sometimes anonymity of the reporter is a result of the time pressure.

Reporting is helpful, however it is not always possible to see the results from the incident reporting. Many times the results of the incident analysis are long term preventing problems.

A feedback to the incident is generally good and given during the meetings and as a bulletin. However mostly bulletins from the processing groups are read by the nurses, instead of bulletins from Patientombuddet or AalborgUH risk manager.

There is a need for more positive association to the reporting procedure and identification of the errors.

A.3 Meaning Condensation of the Interview with Patient

An interview with the patient was performed on 19.03.2015. The described meaning condensation within this chapter is based on interview with a cancer patient.

A.3.1 Patient's Experience with Incidents

Patient had to stay at a hospital many times, due to cancer treatment. First the patient experienced lack of information about the spread of metastases to lungs and lymph nodes. Afterward a surgery was performed on the patient, which was wrongly completed. It resulted in complication in form of bacterial outbreak and bleeding from the surgical wound. Lastly, the nurse measured the blood sugar on patient by mistake, because patient's identity was not verified.

The patient knew about possibility to make a complaint, but had no previous knowledge about a possibility of reporting the incidents to the DPSD system. The website was unknown to the patient.

A.4 Meaning Condensation of the Written Interview with RL Solutions

A written interview was performed on 5.05.2015 with a product manager from Canadian company RL Solutions regarding DPSD database structure. The RL Solutions is the company which developed the Danish DPSD system. The questions were developed by the project group and send to the interviewee, whom provided written answer back to the project group.

The main aim of the interview was to learn and understand the database structure and aspects of the DPSD system.

A.4.1 Structure of the DPSD Database

The DPSD application was first developed in September 2010, but was first used on a large scale early in 2011. The terminology of the used taxonomy is Danish: names of tables, columns, and values in pick lists. None of this is hard-coded in the application. The taxonomy was developed largely by the end users, to suit their information requirements. It is quite different from the taxonomy used by the majority of RL Solutions' clients.

The DPSD database is a relational database managed by Microsoft SQL Server. The structure of the database is divided into two main parts:

- Application framework - contains tables required to support the application and configuration. Mainly is invisible to the application users.
- Information taxonomy - contains information entered by the users and the administrators of the application may create, update and delete taxonomy tables and columns. The information taxonomy is maintained through the application.

A set of framework meta-data tables contains many extended properties of the information taxonomy, business rules, pick lists and pick trees etc.. The application administrator maintains the meta-data tables through the application user interface.

The information taxonomy is conceptualized as a tree whose root element is a `file` or `case`. The logical data element types of the information taxonomy are:

- `BooleanElement`
- `DateTimeElement`

- `DecimalElement`
- `IdentityElement`
- `Integer64Element`
- `IntegerElement`
- `TextElement`

The logical layer separates the application from the physical tables and columns where the data is stored called "meta-base" layer. The meta-base is a collection of logical tables and fields that correspond to physical tables, columns, and relationships; but the meta-base contains many attributes relating to tables and fields that do not exist in the physical data mode. The meta-base defines the logical names of fields; the types of user interface (UI) controls that are used to expose them to users; lists of values for pick-list fields, and trees of values for trees; business rules expressing conditions that control whether fields are visible, mandatory, or read-only at run-time; and many other attributes. The meta-base also defines the hierarchical structure of the case-file, and the logical associations between tables. The meta-base itself is a collection of tables stored in the physical database and an information diagram of the DPSD database is showed in the figure A.1.

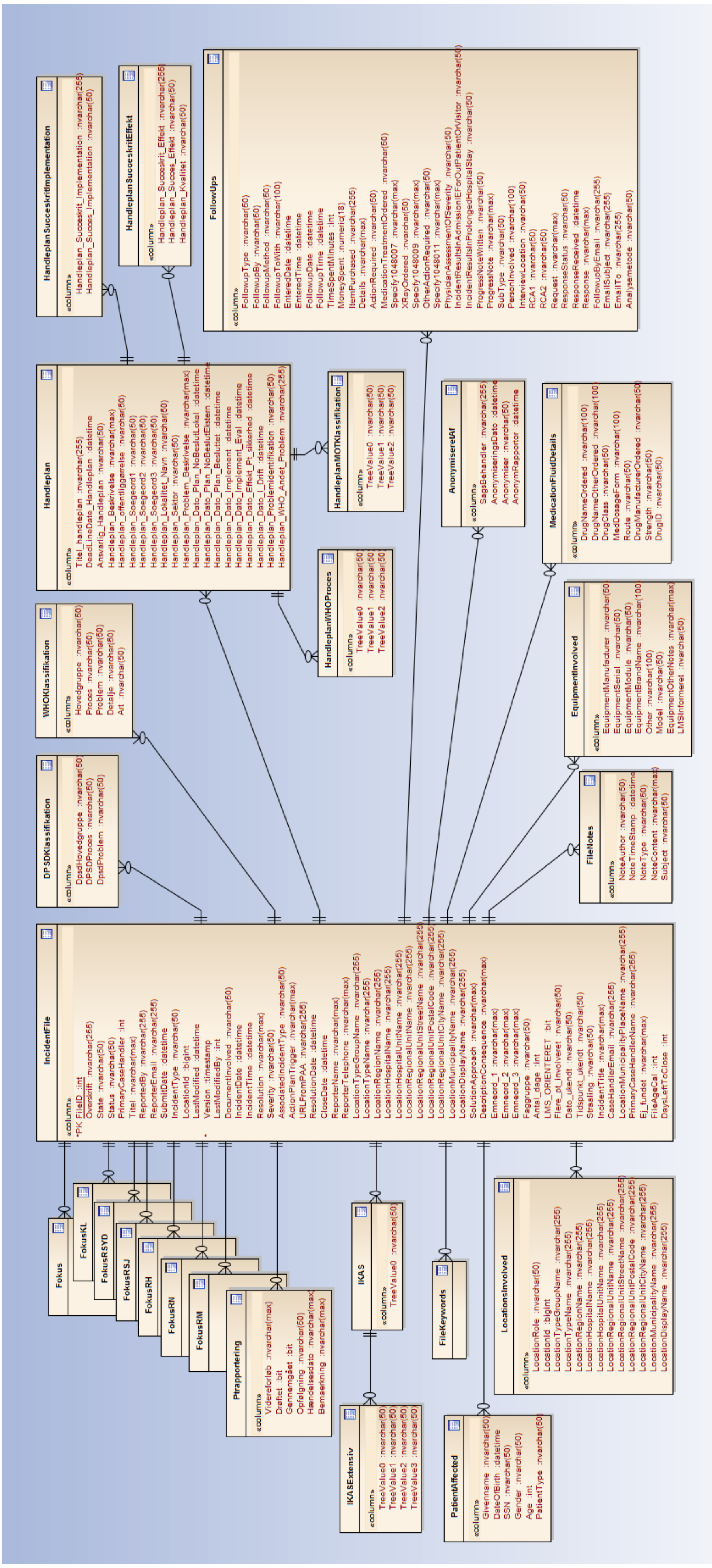


Figure A.1. Information diagram of the DPSD database

Primary and foreign keys in the DPSD database are automatically created by the application, within the information taxonomy. Furthermore, the values of primary keys are automatically generated. Keys and key values are internal to the application, and not visible to the user, with one exception: the primary identifier of a `file/case`. This identifier is exposed as the `File/Case Number`.

The main purpose of the application is to provide tools for the submission, management, and analysis of `files/cases`. The content of a `file/case` is customizable through the metabase. Furthermore, the application provides the user with a number of ways of viewing lists of `files`, and searching for `files`, based on their information content. The reporting system gives the user the ability to produce documents and charts listing and summarizing the information in `files`. It is also possible to export raw data using the reporting system.

Data integrity in the DPSD database is maintained by the physical data model, and by the use of the data access layer to perform all data operations on `files`.

Access to the information in files is controlled by a security system. Users may only view and report on files associated with the specific set of locations they have been granted access to. Accessibility of reporting and data exporting tools is also controlled by the security system.

Moreover, the application's data access layer handles reading and writing of `file/case` information. The data access layer uses the meta-data to provide the user with a logical view of the information in the database. It also uses the meta-data to navigate the physical structure and relationships of tables in the database, so that the user does not need to understand them. The data access layer is responsible for transaction processing, and for audit logging.

The key advantage of the DPSD database is the flexibility of its information taxonomy, and the tools provided to business users for maintaining it. However, it is also a disadvantage, because it makes the business user responsible for developing and maintaining a data model, without much constraint, which can be quite difficult to master.

Appendix B

Methods

The following chapter describes the methods used to solve the problem statement for the report.

B.1 Literature Search

A literature search was performed to gather the necessary knowledge and information about the incidents in both Denmark and worldwide. The search was helpful for understanding of the subject, identification of strengths and weaknesses, obtaining an overview of the problem and finding in specific area for the research.

The criteria for the literature search used for writing of the problem analysis were that the scientific publications should be as new as possible and not older than from year 2000, since it was first after this year, that the systems for reporting and gathering of incidents started, [Mahajan, 2010]. Search engines such as PubMed, Google Scholar and Aalborg University Library were used during the literature search. To gather information about the incidents in Denmark, publications from DPSD, Patientombuddet and Region North Jutland were used during the problem analysis.

B.2 Interview

In order to gain deeper understanding of the incidents, possible causes, working conditions and practical applications of the incident reports, a qualitative study method was used for this report. The qualitative study was done to understand and explore meaning of individuals by learning about their experiences, histories and perceptions regarding the incidents at the hospital sector, [Ritchie et al., 2013]. Therefore, the gathered information from the qualitative study were supplying the knowledge from the literature search. The chosen qualitative method was an interview, due to possibility to learn about the personal approach to the incidents and their reporting. An observational study was considered, however it was not chosen since it lacks the possibility of discussion of the underlying problems related to identification with the individuals, their reporting and processing of the incidents. Furthermore, a qualitative research interview can produce a new knowledge and help to understand the researched world, [Kvale and Brinkmann, 2008].

B.2.1 Seven Stages of the Interview

To ensure the quality of the gathered knowledge, the performed interviews were designed according to seven stages described by Kvale and Brinkmann [2008] presented in table B.1.

Stage	Description
Thematizing	Purpose and concepts of the investigation have to be formulated before the start of the interview. The questions of *why* and *what* have be clarified before deciding on *how* method. The *why* question clarifies the purpose of the study, *what* requires obtaining of a knowledge about the research subject and *how* focuses one possible interviewing techniques and methods are relevant to obtain needed knowledge.
Designing	Design of the study has to be planed and all seven stages of interview have to be taken into consideration to obtain required knowledge and moral implications during the study.
Interviewing	Interviews have to be conducted with a reflective approach to the sought knowledge and without a personal relation to the interviewed subject.
Transcribing	Material for an analysis has to be prepared from the interview. Mainly the material includes a transcription from speech to text.
Analyzing	Analysis method of the interview is chosen. The decision should be based on the purpose, topic of the investigation and on nature of the interview material.
Verifying	Validity, reliability and generalization of the interview findings have to be ensured. Validity checks if the interview investigates what was intended to be investigated, while reliability described how consistent the results of the interview is.
Reporting	The finding of the interview have to be communicated in a form fulfilling the scientific criteria, considering the ethical aspects of the investigation and results in a readable product.

Table B.1. Seven stages of the interview described by Kvale and Brinkmann [2008].

B.3 IR System Development

Methods used during solution part of the report are presented and described within this section. The solution part focuses on system description, documentation and development.

B.3.1 Unified Modeling Language and Unified Process

To ensure the quality of the system documentation, the elements of a Unified Modeling Language (UML) and Unified Process (UP) were used to described different development stages. The UML was chosen for the visual representation of the development, since it aids a correct implementation and describes interactions between the objects in the system. The different interactions within the system are described using a visual representation of system's architecture and by modeling interactions between the programmed objects, [Arlow and Neustadt, 2005]. Therefore, the UML is adapted to document the proposed structure and dependencies between the system's elements in a broadly understandable way.

Documentation of different development's stages was performed using UP, which is an iterative methodology. The UP divides the development process into smaller elements allowing identification and specification of the developed system's characteristics. The elements used by the UP methodology are: requirements, analysis, design, implementation and test and are described in table B.2, [Arlow and Neustadt, 2005].

UP Element	Characteristics
Requirements	Provides information and description of the system's functionality.
Analysis and Design	Defined requirements are structured and translated into software elements of the system's architecture.
Implementation	Presents the final, programmed version of the system.
Test	Describes performed tests for the developed system.

Table B.2. The UP elements theory described by Arlow and Neustadt [2005].

B.3.2 Requirements

System requirements reflect the features, which are needed to be implemented in the system, [Arlow and Neustadt, 2005]. Therefore, the chosen requirements are divided into functional and non-functional requirements. The functional requirements state the expected behavior of the system, while the non-functional requirements describe specific system's properties and constrains, [Arlow and Neustadt, 2005]. Chosen requirements are further specified using a MoSCoW theory. The theory helps with prioritization of the required features, which are mandatory (*must have*), important that may not be implemented (*should have*), truly optional (*could have*) or can wait for future releases (*wanted to have*) of the system, [Arlow and Neustadt, 2005].

To capture the chosen functionality of the system and present them from the user's point of view, a use case diagram is created, [Arlow and Neustadt, 2005]. The use case is a possible action, that the external actor can directly perform with the system. Modeling of the use case diagram aids in finding system boundary in order to get a scope of the modeled system. The possible actions in the system are performed iteratively and commonly in parallel to each other, [Arlow and Neustadt, 2005]. Furthermore, a generalization can used in use cases to illustrate, that the behavior is common to the actors/use cases and that they inherit the behavior of the parent use case, [Arlow and Neustadt, 2005].

A relationship between the use cases is described using `«include»` and `«extend»` relations. The `«include»` describes a relationship, where a base use case includes the behavior from the inclusion use case. The behavior from the inclusion use case is supplied to the base use case, which is not completed without its all inclusion use cases, [Arlow and Neustadt, 2005]. The `«extend»` is a way of inserting a new behavior from the extension use case to the base use case. The base use case has a set of extension points where the behavior from the extension use case may be added. The extension use case delivers a set of insertion segments, which can be used by the base use case. The base use case does not know about the extension use case and can be completed without it, [Arlow and Neustadt, 2005].

The functionality of use cases is further specified. During the use case specifications several aspects of a specific use case are describes using, [Arlow and Neustadt, 2005]:

- Use case name - specified name of the use case.
- Use case ID - a unique identifier of the use case.
- Description - describes goals of the use case.
- Primary actor - presents the actor triggering the use case.
- Secondary actor - presents the actor interacting with the use case after it being triggered.
- Pre-conditions - status of the system affecting execution of the use case.
- Main flow - description of time-ordered steps describing the situations where everything goes perfectly and as desired during execution of the use case.
- Post-conditions - status of the system after execution of the use case.
- Alternative flow - description of the alternatives to the main flow.

B.3.3 Analysis and Design

Usability

Usability is a part of the development process used as a quality attribute assessing the easiness of the interface use for an intended user. Moreover, the usability concept includes methods, which aid with improvement of the interface's ease of use during the design process. The usability is defined by several quality components, [Galitz, 2007]:

- Learnability - easy to use and to learn for first time users

- Efficiency - Efficient to use for experienced users
- Memorability - Easy to remember after a period of time
- Errors - Few and easily recovered errors
- Satisfaction - Pleasant to work with for the user

Usability is an important quality attribute of an interface, since it ensures that the users will be willing to use the designed system. Additionally, a utility has an equal importance as usability in the design of the interface, and focuses on the functionality and features of the system. Both usability and utility requirements have to be fulfilled, since usability cannot exist without the utility. Therefore, if the interface is too difficult to use, the correct functionality of the system cannot possibly be accomplished, and if the interface has usability but not utility, it does not constitute any value for the user, [Galitz, 2007].

Principles for Good Interface Design

Researchers demonstrated, that by making screen design less-crowded the user will become 20% more productive. Furthermore, by reconstructing screens the user is using 25% less time completing the transactions with 25% fewer errors, and the decision-making time is reduced by 40%, [Galitz, 2007].

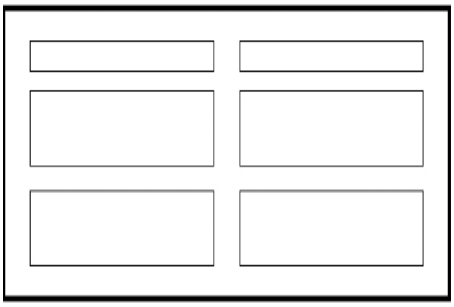
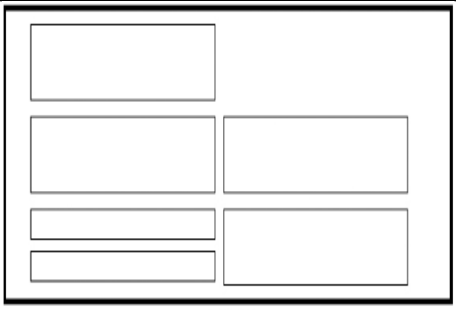
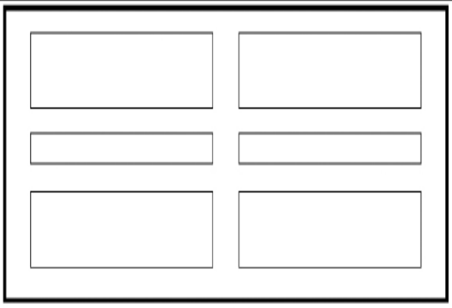
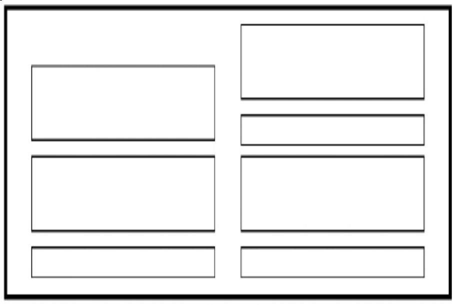
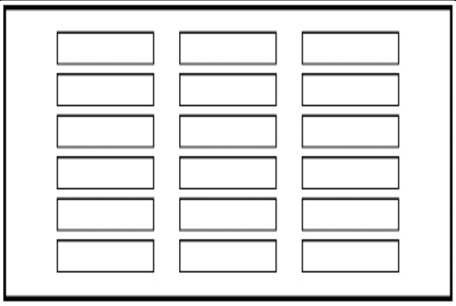
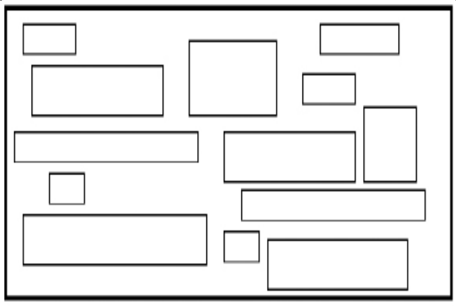
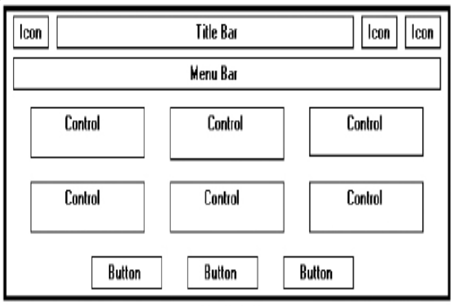
To create a good web design various factors have to be considered. Some of the good web design advantages include, [Galitz, 2007]:

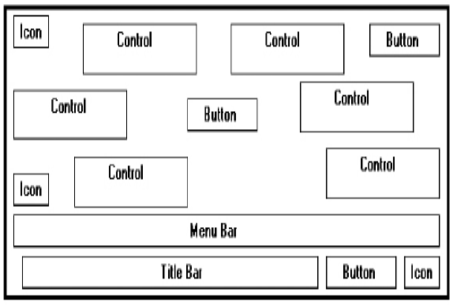
- The user recognizes symbols faster than text
- A more successful and faster problem solving achieved by visual representation of information
- Displayed objects are visible
- Fewer errors
- Increased feeling of control
- Immediate feedback
- Easily comprehended design and control reduce anxiety in use
- Equivalent in words take up more space than icons
- Low typing requirements

The principles of good screen and interface design are affected by multiple factors including amount of presented information, organization of the information, communication language, the clarity of the components, use of aesthetics, and page or screens consistency, [Galitz, 2007].

To achieve visual clarity the displayed elements on the screen are should be presented and organized in recognizable, meaningful and understandable ways. Creating a clean and clear organization screen makes it easy to ignore secondary information and the recognition of the screen essential elements are easier. The table B.3 represents a visualization of possible good and poor design choices, [Galitz, 2007].

Table B.3. Principles for good screen design and interface, [Galitz, 2007]

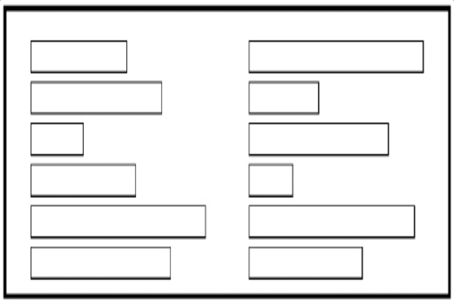
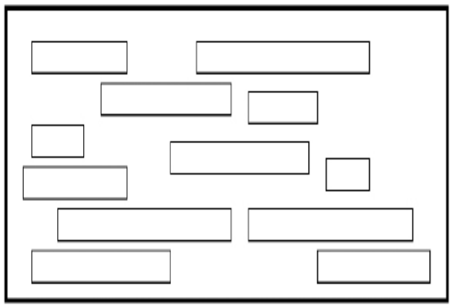
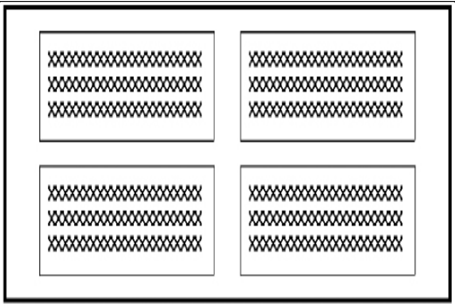
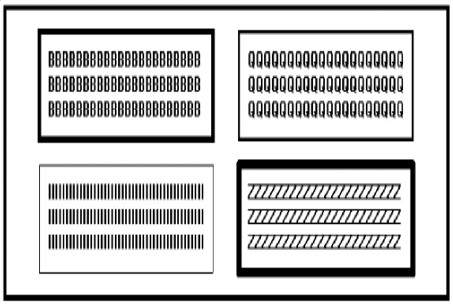
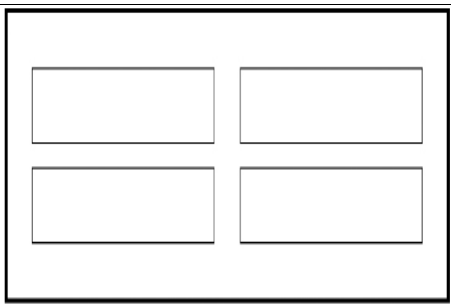
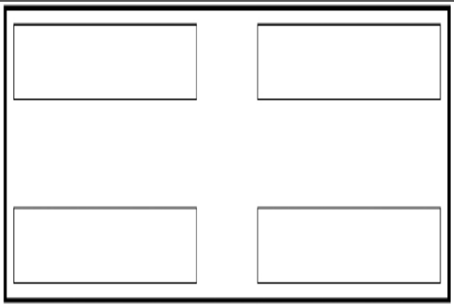
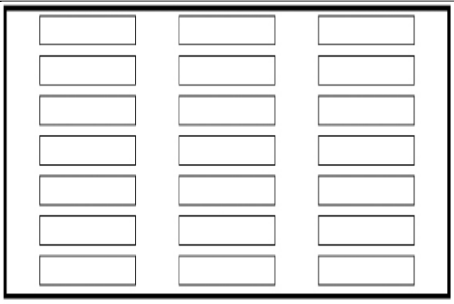
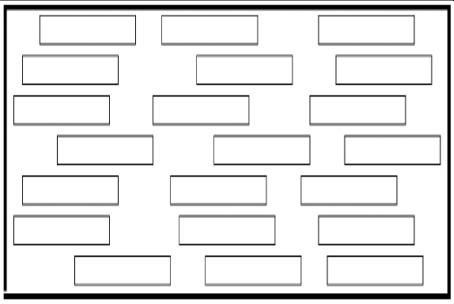
<table>
<tr><th>Good</th><th>Poor</th></tr>
<tr><td>
Balance</td><td>
Instability</td></tr>
<tr><td>
Symmetry</td><td>
Asymmetry</td></tr>
<tr><td>
Regularity</td><td>
Irregularity</td></tr>
<tr><td>

Predictability</td><td>

Spontaneity</td></tr>
</table>

Continued on next page

Table B.3 – continued from previous page

Good	Poor
 Sequentiality	 Randomness
 Economy	 Intricacy
 Unity	 Fragmentation
 Simplicity	 Complexity

Furthermore, it is important to create an atmosphere on the screen with the design and to establish reading order of the elements for the user. To avoid confusion and inconsistency in the design, mixed typographic should be avoided. The text provided on the interfaces should have a sans serif font format with the minimum font size of 10 points. However, the preferred font size to be used in the design ranges from 12 to 14 points, [Galitz, 2007].

From the user's point of view, a good screen design is represented by, [Galitz, 2007]:

- Clutter-free, orderly and clean appearance
- The indication of what is displayed, and what should happen with the web page is obvious
- The information is located where it is expected
- The indication of what relates to what is clear with use of headings, options and data
- Simple and plain language
- Simple to use and execution of the system functionality
- If an operation can result in a permanent change in the system or data, there must be a clear indication about it

Web pages and screens direction should be toward understandability, simplicity and clarity, [Galitz, 2007]. Therefore, all choices should be carefully considered in relation to good interface and screen design, but also in terms of usability as described in section B.3.3.

B.3.4 Implementation

Implementation of the IR system was performed using web-development tools, since the chosen form for the system is a web-based interface. A Document Object Model (DOM) is used to build, modify and define how a logical structure of an website can be accessed and modified with a HyperText Markup Language (HTML), [W3C, 1998]. The HTML is used to define a structure of website using HTML elements. Furthermore, the HTML allows to publish, design forms and retrieve information from user's interactions with the website, [W3C, 2014].

To define and describe a presentation layer of the web-interface, Cascading Style Sheets (CSS) language is used. The CSS allows definition of HTML elements' features, such as margins, layout, color and size. Furthermore, since CSS is independent from the HTML, maintenance and sharing of the style sheets in the web-interface is easier, [W3C, 2014].

A JavaScript is a lightweight, cross-platform and object-oriented scripting language used for control of the HTML objects, [MDN, 2014]. Since older browsers may have problems with supporting the newest versions of the JavaScript, a jQuery can be used in the system development to provide an easier access to the web-interface from different browsers, [MDN, 2014; jQuery Foundation, 2014]. The jQuery is a fast, feature rich JavaScript library allowing an easier manipulation of HTML objects and events, [jQuery Foundation, 2014].

B.3.5 Validation

To validate the implemented solutions of the IR system, a usability test was performed. The usability test was chosen, due to possibility of evaluation of the IR system by the end users - how do they use it and which problems may they have with each of the interfaces, [Holzinger, 2005]. The usability testing is based on five characteristics, [Dumas and Redish, 1999]:

1. Aims to improve product's usability
2. Participants in the test represent real users of the system
3. The participants perform a real task in the system
4. An observer observe and records what the participants do and say
5. The observer analyze the collected data, diagnose problems and recommends changes to fix those problems

Based on the usability test, it is possible to evaluate and improve the product by considering goals, such as easy navigation, ease of use for novice users and acceptance from the old users, or how easily the nontechnical staff at the end user will be able to operate the product, [Dumas and Redish, 1999].

To inspect the usability of the system, usability experts may be used as well by performing a heuristic evaluation. However, the usual way in which the end users use the system may exceed the experts' imagination. Identification of the possible problems by the end users is beneficial, since it can help the engineers to identify not obvious problems, and evaluate the actual impact of the problems on the user during the test, [Jeffries and Desurvire, 1992]. Therefore, the usability test was chosen as a validation method of the IR system.

B.4 Database Design

The investigated problems described in section 2.4 presented several aspects of the DPSD database related to performance and time consuming of search and processing of the reported incidents. Therefore, to optimize and evaluate the existing database, it is required to explore and understand the database design to improve the database model for processing of the reported incidents.

A database-management system (DBMS) is a collection of interrelated data containing a set of information, also referred to as a database, and a group of programs for accessing the data, [Silberschatz et al., 2011]. The main aim of the DBMS is to deliver a method for storing and retrieving the information from a database in an efficient and convenient way. The DBMS are adapted to manage large amounts of information, by defining storage's structures and providing tools for data manipulation. Furthermore, safety of the stored data in DBMS is required. It should ensure that the database will work properly against system crashes and accessing attempts by an unauthorised subjects, [Silberschatz et al., 2011].

A data model defines the underlying structure of the database and contains *"a collection of conceptual tools for describing data, data relationships, data semantics and consistency constrains"*, [Silberschatz et al., 2011]. Two chosen data models for the project is relational and non-relational models, which are described in the following sections.

B.4.1 Relational Model

The relational model is a collection of tables (relations) with multiple columns (attributes), representing both data and the relationships between the data. Each of the attribute in a relation has a set of unique name, [Silberschatz et al., 2011]. The relational model is the main, mature and traditional data model used commercially nowadays due to its simplicity and broad application, [Leavitt, 2010; Silberschatz et al., 2011].

A database using the relational model, stores the data for future analysis and access. Data used with the relational database is structured, such as set of sales figures, and fits well in the predefined tables' layout, fixed names and types of attributes. However, unstructured data, such as word-processing documents or images, causes problems with the relational databases. When data does not fit in the predefined tables correctly, the database becomes more complex, difficult and slower to work with, [Leavitt, 2010].

Furthermore, with increasing volume of data, the performance of the relational database decreases, [Nayak et al., 2013]. Scaling of the data in the relational database requires use of more powerful computer and distribution across many servers, however their work complexity in such distributed manner is increased since it requires joining of tables from the system. Partitioning of the data is also not possible with the relational database, [Leavitt, 2010].

The relational databases are updated and queried using a structured query language (SQL). The users can access a specific data in ways, that do not require reorganization of the tables in the database. The data is also more reliable since it is restrained with ACID (atomicity, consistency, isolation, durability), [Leavitt, 2010]. The ACID restrains describe, [Leavitt, 2010]:

- Atomicity - an update is performed completely or not
- Consistency - performed transactions cannot break database's rules
- Isolation - each application performs transaction independently of other application operating at the same time
- Durability - performed transactions are preserved

B.4.2 Non-Relational Model

In response to limitations of the relational database model, a new model for further effective utilization of a database was developed and is gaining popularity, [Leavitt, 2010; Nayak et al., 2013]. One of such models is a non-relational data model- the NoSQL database, which is utilized by many of the database users and developers, [Leavitt, 2010]. The NoSQL databases are not relational and can handle unstructured data with good performance on large data sets, which is faster than with the relational database. The NoSQL data models are more simple, flexible and scalable as the database grows, [Leavitt, 2010; Nayak et al., 2013]. Many of the available NoSQL databases are open-source, [Leavitt, 2010]. Several types of NoSQL databases are described in table B.4.

Table B.4. The NoSQL types based from descriptions by Leavitt [2010] and Nayak et al. [2013].

NoSQL type	Database examples	Characteristics
Key-Value stories	SimpleDB Amos II Scalaris DynamoDB RIAK	Stores values indexed for retrieval by keys, creating a key-value relationship. The key-value relationships provide a fast look-up and an option for mass storage. However, it is not possible to create custom views of the data due to lacking a schema. The key-value stories can be used for both structured and unstructured data and are accessed by a simple application programming interface. The key-value stories can be used for storing of user's session, websites for online shopping and forums.
Column-Oriented databases	Cassandra Hbase Big Table	Contains one extendable column for related data, where a key is assigned to one or more columns (attributes). The data is stored in a widely distributed architecture and is quickly combined with reduced input/output activity. The sorting of the data depends on the ones assigned to specific attributes and the structure of the database offers high scalability in data storage. The column-oriented database works well for data mining and analytic applications.
Document-Based stories	MongoDB CouchDB	Stores and manages data as a collections of documents, which are addresses using unique keys. The keys may be in form of a string, URI or path. It offers a high performance and options for horizontal scalability. The stored documents uses different formats, such as XML, PDF or JSON. The document-base stories are more complex compared to key-value stories as they allow to attach a key-document (key-value) pairs. It works well for applications requiring storing of documents having special characters, such as content management system or blog software.

Continued on next page

Table B.4 – continued from previous page

NoSQL type	Database examples	Characteristics
Graph Databases	Neo4j	Stores data as a graph consisting of nodes (objects) and edges (relationships between the nodes). Every node contains a direct pointer to the next node. It provides an effective way of storing semi-structured data and their queries are made using traversals resulting in higher performance that in relational databases. The graph databases are scalar and support ACID. They can be used for social networking applications, bioinformatics, content management, security and access control, or network and cloud management.
Object Oriented Databases	db4o	Stores data as an object, which has a unique identifier. The objects are directly accessed using pointers. This type of database allows features of Object Oriented Programming, e.g. encapsulation, polymorphism and inheritance. The structure of such database is comparable to the relational database. However, this type of database should not be used with simple data and relationships. It is tied to a programming language and difficult to scale. The object oriented database can be used for applications with complex object relationships, scientific researchers or telecommunication.

For some developers not familiar with SQL, the NoSQL is an easier alternative to be used, however it then requires a manual query which can be time-consuming for more complex tasks. The established manual queries differ across the different NoSQL databases, making change of the chosen database difficult for the user, since it would be required to change the query language as well, [Nayak et al., 2013]. The NoSQL database are also less reliable than relational databases, since they do not use ACID restrains. By not applying the ACID restrains, the database performs better and is scalable, but is not consistent without further manual support, [Leavitt, 2010]. However, the NoSQL database fulfills the BASE (Basically Available, Soft state, Eventual consistency) properties, which are consistent with CAP (Consistency, Availability, Partition tolerance) theorem, [Nayak et al., 2013].

APPENDIX C

USE CASE SPECIFICATIONS AND ACTIVITY DIAGRAMS

Within this chapter, the rest of the use cases from figure 5.1 are described and presented. Use case specifications are provided to each of the use case, together with corresponding activity diagrams to illustrate functionality of each of the use cases in the system.

Use case `Search`

The `Search` use case is used for search for a specific information associated with reporting of the incident. It is possible to search after medication and medical devices, patient, location and healthcare professionals' details. The medication and medical devices search allows to input medication and/or medical devices involved in the incident. The patient search is used to input details of the patient involved in the incident. The location search refers in this report to hospitals and their departments. However, in the real world it can contain details about the locations in different sectors. The healthcare professionals search allows to search for details of healthcare staff used as contact during processing of the incident report. The use case specifications are provided in table C.1.

Use case name	`Search`
Use case ID	UC2
Description	Allows the actor to search and select relevant data during incident reporting.
Primary actor	Healthcare Professionals
Secondary actor	Device Database, Medication Database, Patient Database, Location Database, Staff Database
Pre-conditions	The actor chooses to search for information in either the `Report Incident`, `Statistics` or `Extract Data` use cases. Connection to the Internet and the required database is established.
Main flow	1) The actor chooses to search for an information in the system. 2) The actor enters the data for the search in the relevant pop-up window. 3) The actor chooses from the presented data. 4) The actor presses *OK* button to confirm the selected location. 5) The system searches after the entered data.
Post-conditions	The search is submitted and appears in the system.
Alternative flow	1.a) The actor can search for `Medication and Medical Devices` details. i) The actor can choose to enter the medication and/or medical devices details manually. 1.b) The actor can search for `Patient` details. 1.c) The actor can search for `Location` details. 1.d) The actor can search for `Healthcare Professionals` details. 2.a) The actor selects *Free text* field in a pop-up. i) Search is performed after the variable in a database using free text. 2.b) The actor selects *Free code* field in a pop-up. i) Search is performed after the variable in a database using a predefined code. 3.a) The actor can add more variables from the search by selecting *Add* button. 5.a) The variable cannot be found. i) The actor needs to check the entered variable details. ii) The actor needs to verify, if the entered variable exists. iii) The actor needs to contact a technical support. 5.b) The actor cannot access to the required database. i) The actor needs to check, if the actor has rights to access the database. ii) The actor needs to contact a technical support.

Table C.1. Use case specification for `Search`.

The activity diagram of the `Search` use case is shown in figure C.1. The user chooses to search for different relevant information which can support the incident.

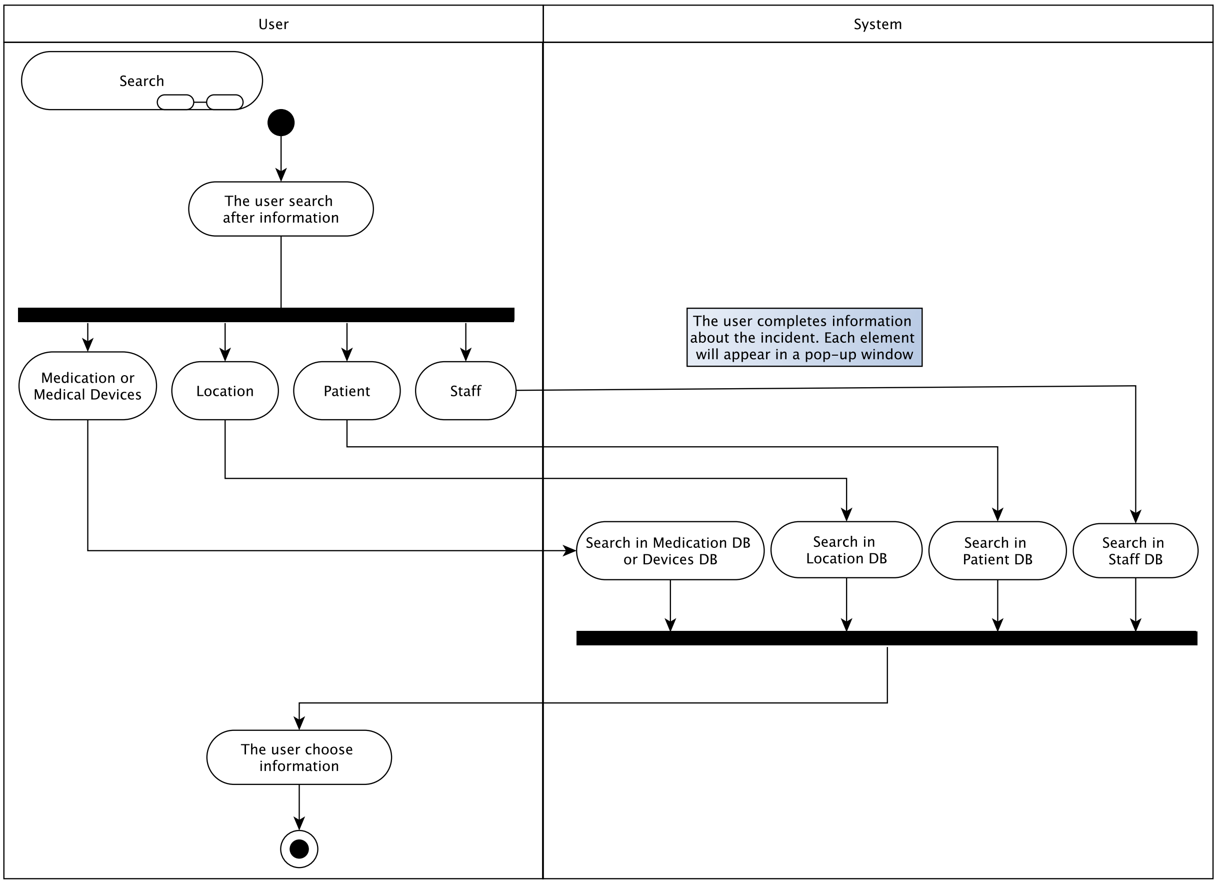


Figure C.1.

Use case `Statistics`

The `Statistics` use case allows an authorized actor to access the incident data in order to visualize the selected data. The visualization aids the actor in analysis of the patterns and trends of the incidents. The use case specification of the use case is described in table C.2.

Use case name	`Statistics`
Use case ID	UC3
Description	Allows the actor to create and view a visualization of the reported incidents based on the selected criteria.
Primary actor	Healthcare Professionals
Secondary actor	Incident Database, Location Database
Pre-conditions	The *Statistic* tab in the system is open. Connection to the Internet and the databases is established. The user is logged in to the system.
Main flow	1) The actor chooses a time range from which the data from the reported incidents will be shown. 2) The actor selects wanted incident type. 3) The actor selects wanted severity rate. 4) The actor selects a graph type for data visualization. 5) The actor selects a location from which the data from the reported incidents will be shown. 6) The actor presses *OK* button to confirm the selected data. 7) The system searches after the selected data in database.
Post-conditions	The selected reported incident data is plotted as a chosen graph type.
Alternative flow	2.a) The actor does not select any incident type. 2.b) The actor selects multiple incident types. 3.a) The actor does not select any severity rate. 3.b) The actor selects multiple severity rates. 5.a) The use case `Search` is initiated. 5.b) The actor does not select any location of the incidents. 7.a) The system cannot find selected incident data. i) Check the Internet connection. ii) Check, if the selection of the data is correct. iii) Contact the technical support. 8.a) The actor selects to save the plot of data. i) The file is saved in a chosen image format. 9.a) The actor selects to reset the data selection. i) All selections are cleared.

Table C.2. Use case specification for `Statistics`.

The described flows of the `Statistics` use case are shown in figure C.2. For the user to access the `Statistics` use case a login is required. The system checks if the user is signed in and identified, if the user is accepted they will get access to the `Statistics` use case and thereby has the possibility to graphically display in different combinations and type of plots. Otherwise the `Verify User` will appear.

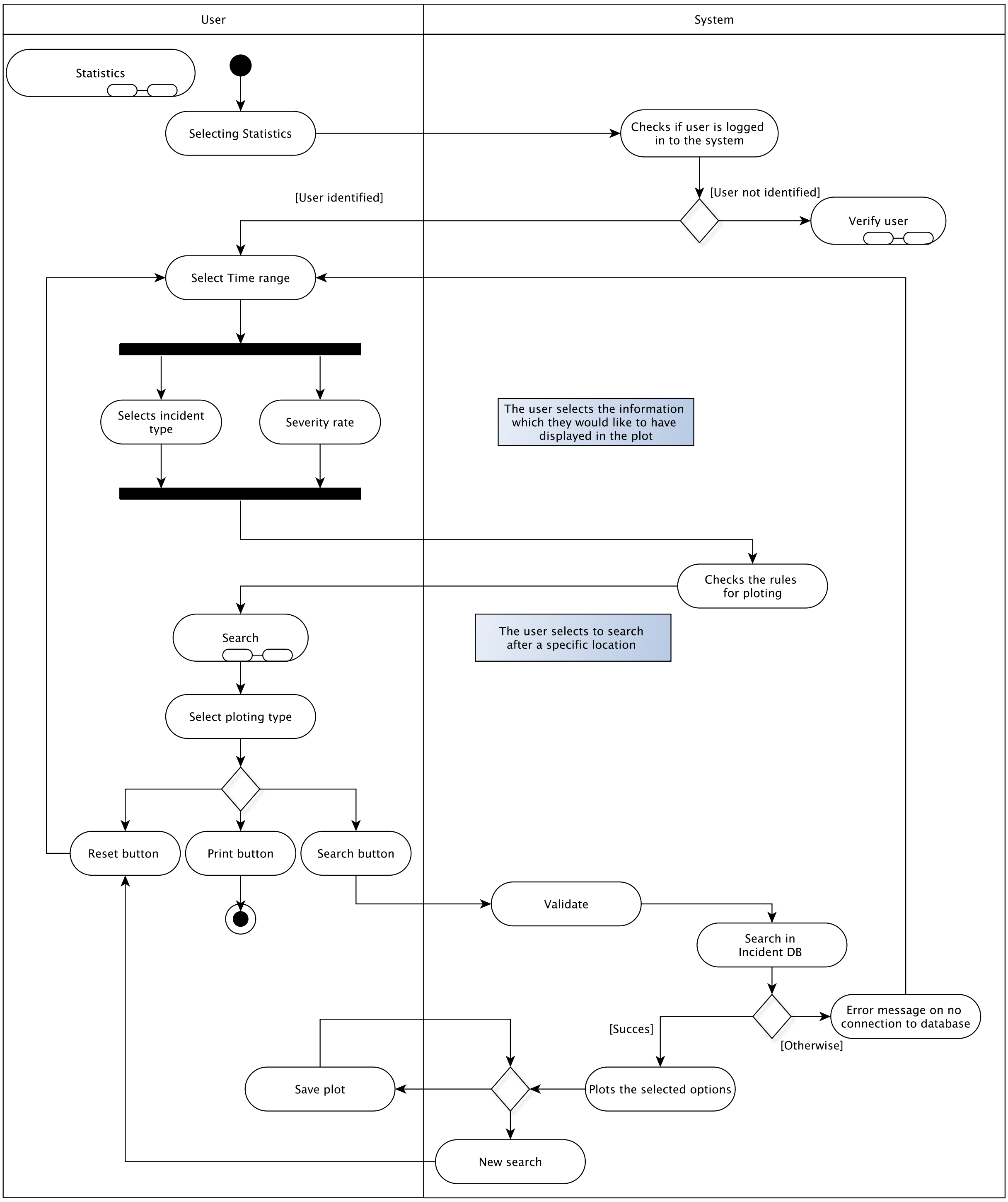


Figure C.2.

Use case `Extract Data`

The `Extract Data` use case allows an authorized actor to access the incident reports and analyze the individual reports. A data extraction of selected data from a database is provided to the actor. The use case specification to the use case is provided in table C.3.

Use case name	`Extract Data`
Use case ID	UC4
Description	Allows the actor to extract the data of the reported incidents based on the selected criteria.
Primary actor	Healthcare Professionals
Secondary actor	Incident Database, Location Database
Pre-conditions	The *Data Extraction* tab in the system is open. Connection to the Internet and the databases is established. The user is logged in to the system.
Main flow	1) The actor selects wanted severity rate. 2) The actor selects wanted incident type. 3) The actor selects a location from which the data from the reported incidents will be shown. 4) The actor selects status of the incident report. 5) The actor presses *OK* button to confirm the selected data. 6) The system searches after the selected data in database.
Post-conditions	The selected reported incident data is presented in a table from which the actor can see further details about the reported incidents by double click on the incidents.
Alternative flow	1.a) The actor does not select any severity rate. 1.b) The actor selects multiple severity rates. 2.a) The actor does not select any incident type. 2.b) The actor selects multiple incident types. 3.a) The use case `Search` is initiated. 3.b) The actor does not select any location. 4.a) The actor does not select any status of the incident report. 4.b) The actor selects multiple statuses of the incident reports. 6.a) The system cannot find selected incident data. i) Check the Internet connection. ii) Check, if the selection of the data is correct. iii) Contact the technical support.

Table C.3. Use case specification for `Extract Data`.

The activity diagram of the `Extract Data` use case is shown in figure C.3. When the user selects to open `Extract Data` the system first checks, if the user is sign into the system. The user has the opportunity to display different combinations of data extraction. Otherwise if the user is not identified, the `Verify User` will appear.

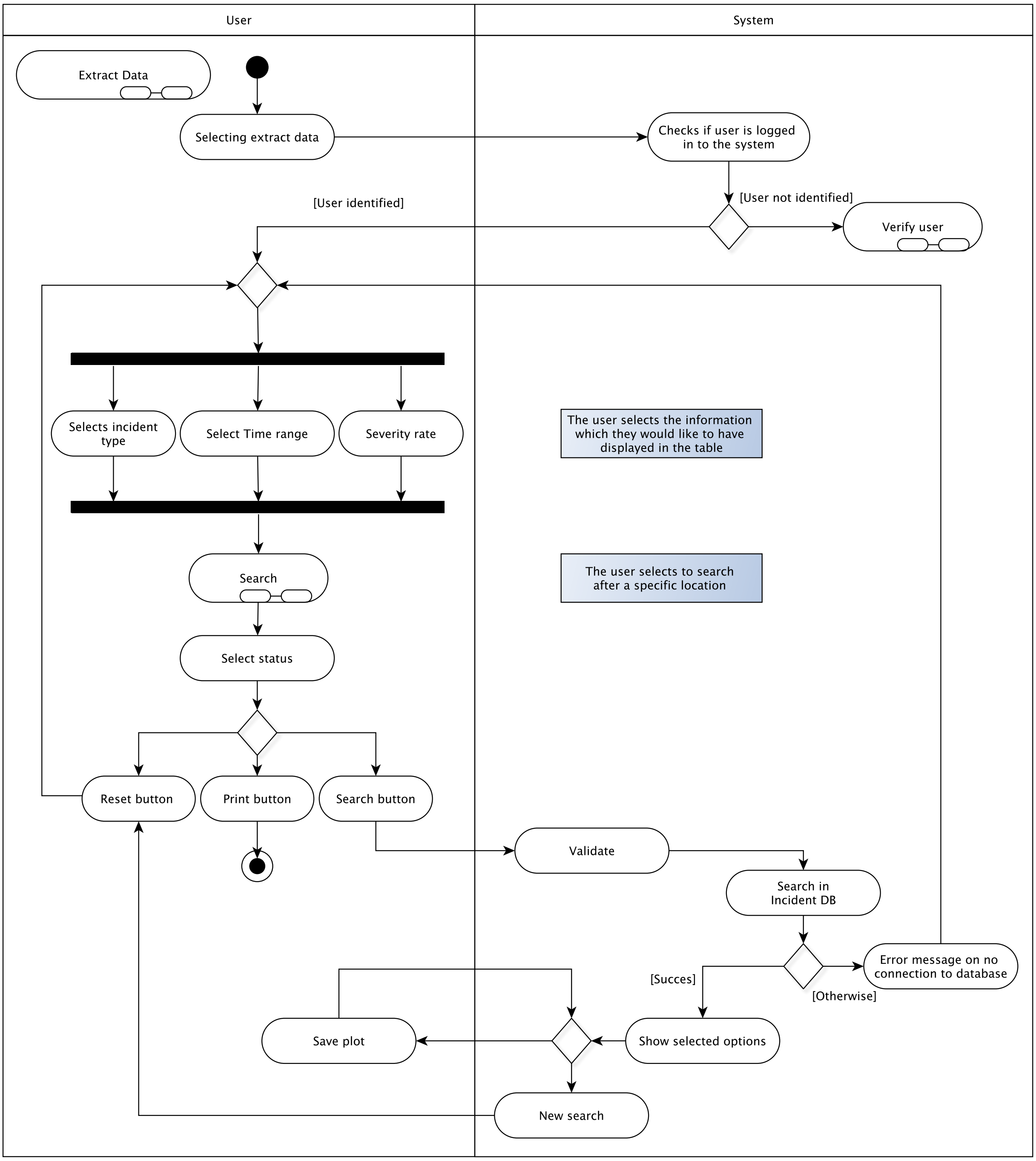


Figure C.3.

Use case `Verify User`

The use case `Verify User` described a login function, which is used for verification of the actors that wish to access `Statistics` and `Extract Data` use cases. The login function is created in order to ensure privacy and safety of the incident data, by protecting from access by an unauthorized actors. The use case specification of the use case is provided in table C.4.

Use case name	`Verify User`
Use case ID	UC5
Description	Verifies, if the actor is authorised to access *Data Extraction* and *Statistics* interfaces.
Primary actor	Healthcare Professionals
Secondary actor	Staff Database
Pre-conditions	Either the *Data Extraction* or *Statistics* tab in the system is selected and a pop-up window appears requiring login from the actor. Connection to the Internet and the database is established.
Main flow	1) The actor enters the correct username. 2) The actor enters the correct password. 3) The actor presses *OK* button to confirm the entered details. 4) The system verifies the entered data.
Post-conditions	The actor is logged in to the system and the selected tab is open.
Alternative flow	1.a) The inputted username is not correctly entered. i) Check the spelling of the username. 1.b) The inputted username does not exist. i) Contact the technical support. 2.a) The inputted password does not match the username. i) Check the entered password and username. 2.b) The inputted password is not correctly entered. i) Check the spelling of the password. 2.c) The inputted password match the username but the actor cannot log in to the system. i) Contact the technical support. 4.a) The system cannot verify the login details. i) Check the Internet connection. ii) Contact the technical support.

Table C.4. Use case specification for `Verify User`.

An activity diagram for the `Verify User` use case is shown in figure 5.7. The system checks when the login function is required and opens the `Verify User` use case. It is possible for the user to enter both username and password and by pressing the login button the system will validate the input text fields. When the login details are wrong the user will be present to a relevant error message.

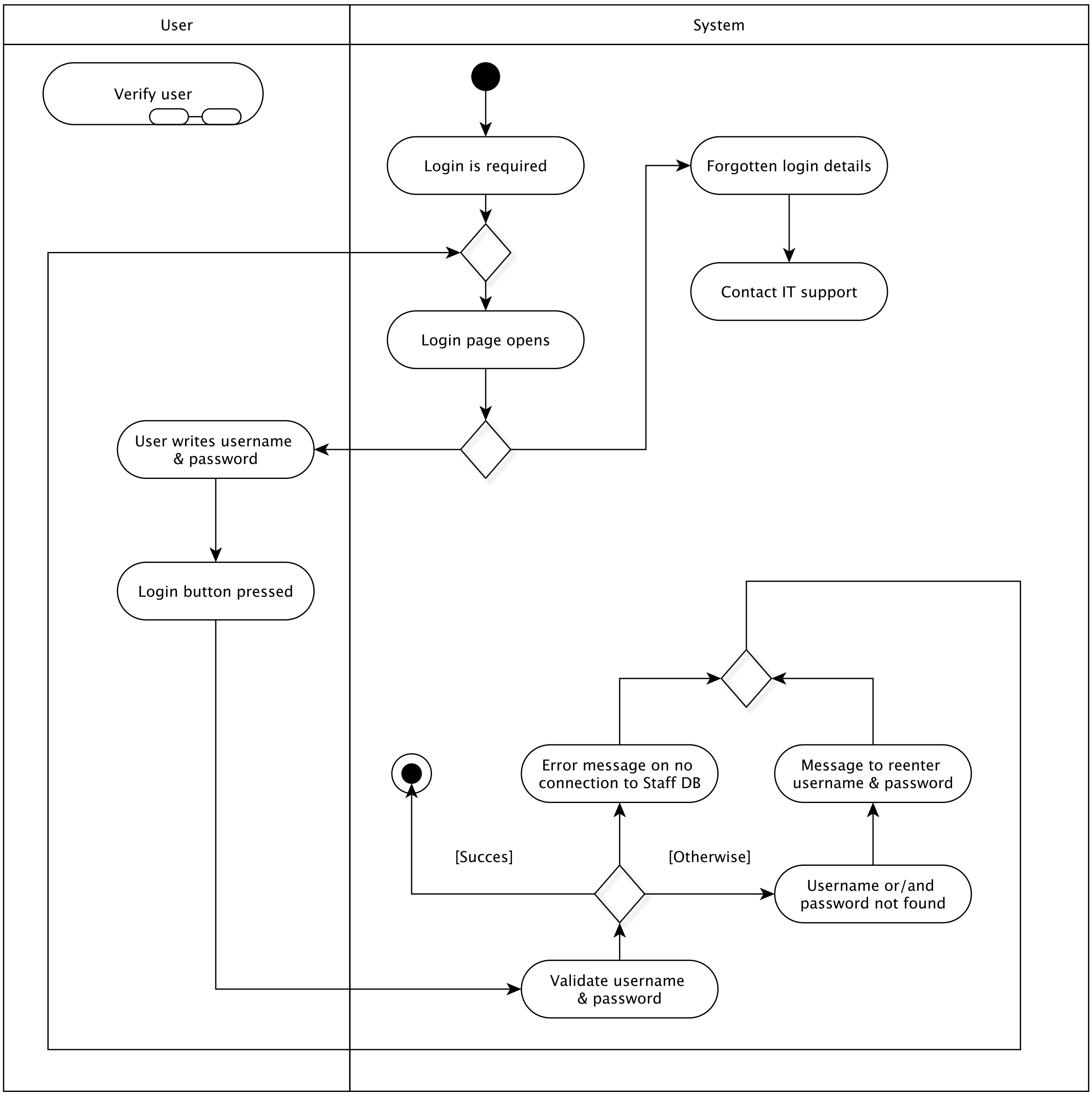

User
System
Verify user
Login is required
Forgotten login details
Contact IT support
Login page opens
User writes username & password
Login button pressed
Error message on no connection to Staff DB
Message to reenter username & password
[Succes]
[Otherwise]
Username or/and password not found
Validate username & password

Figure C.4.

Appendix D

MongoDB

The MongoDB (MD) is a document-based database as described in section B.4.2. The Relational Database (RD) has a highly structured architecture, which contains several tables consisting of individual items using JSON stored exchange data, whereas the MD contains several collections and uses only a single document to store the data using BSON, [Hows et al., 2013]. An illustration of the two models is shown in figures D.1 and D.2.

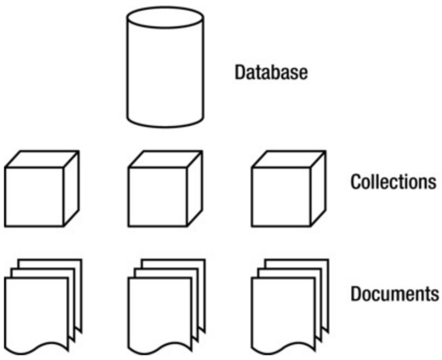


Figure D.1. MD model

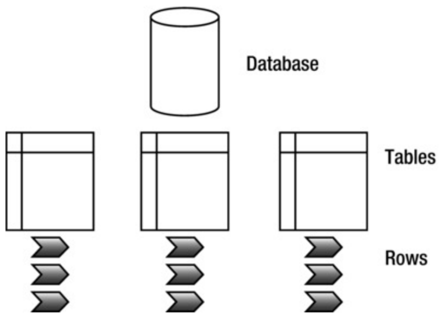


Figure D.2. RD model

Figure D.3. A comparison of the MD model and RD model structures, [Hows et al., 2013]

Furthermore, two examples of how the data is stored in the two different databases is presented in the figures D.5 and D.4.

```
|_media
    |_items
        |_<document>
```

Figure D.4. Data storage in MD model

```
|_media
    |_cds
        |_id, artist, title, genre, releasedate
    |_ cd_tracklists
        |_cd_id, songtitle, length
```

Figure D.5. Data storage in RD model

Figure D.6. A comparison between data storage in MD model and RD model, [Hows et al., 2013]

The BSON is a binary JSON, that provides a faster and easier way for computers to search and process the documents. Aside from reading, creating, deleting and updating data, the MD also makes it possible to add features like:

- Indexing - provides compound and unique capabilities and makes queries faster
- Aggregation - allows the database to optimize and construct complicated aggregations from simple items
- Special collection types - gives the possibility to time expire specific data with a time-to-live collection
- File storage - makes storing of large files easy by using an easy-to-use protocol

MD has the functionality of grouping documents into collections, see figure D.1, where a unique key is provided to the document inside each collections, [Grolinger et al., 2013]. Furthermore, some of the MD advantages are, [Chodorow, 2013]:

- Easier to scale out
- More flexible model by replacing "row" with "document"
- Complex hierarchical relationships are represented with a single record
- No predefined schemas
- Value, keys sizes, and types are not fixed
- Removal and adding of fields is easier because of no fixed schema

The MD has the opportunity to have encompass thousands of documents with different structure inside one collection. A document in MD contains data e.g an item, and is comparable to a row in RD. The presented example shows how a single collection called `"media"` could have two different document types.
The first document:

```
{ "Type": "CD", "Artist": "Nirvana", "Title": "Nevermind", "Genre":
"Grunge", "Releasedate": "1991.09.24", "Tracklist": [ { "Track" : "1",
"Title" : "Smells Like Teen Spirit", "Length" : "5:02" }, { "Track" :
"2", "Title" : "In Bloom", "Length" : "4:15" } ] }
```

The second document:

```
{ "type": "Book", "Title": "Definitive Guide to MongoDB: A complete guide
to dealing with Big Data using MongoDB 2nd, The", "ISBN": "987-1-4302-5821-6",
"Publisher": "Apress", "Author": [ "Hows, David" "Plugge, Eelco", "Membrey,
Peter", "Hawkins, Tim ] }
```

Based on the first document example, a single document contains all the information about the CD. Therefore, the MD requires one document and one collection whereas the RD requires minimum two tables for such set of data. Such functionality makes the MD well organized and incredibly efficient, [Hows et al., 2013]. MD requires a unique id number for each document. If the id is not specified for the document, the MD will automatically generate a document key, [Hows et al., 2013]. The MD is straightforward and flexible to insert data. There are several methods of inserting data to MD. One example of inserting data into a collection is demonstrated below. For the first document, the query would look like, [Hows et al., 2013]:

```
> db.media.insert( { "Type" :  "CD",
..."Artist" :  "Nirvana",
..."Title" :  "Nevermind",
...  "Tracklist" :  [
...  {
...  "Track" :  "1",
...  "Title" :  "Smells Like Teen Spirit",
...  "Length" :  "5:02"
...  },
...  {
...  "Track" :  "2",
...  "Title" :  "In Bloom",
...  "Length" :  "4:15"
...  }
...  ]
...}
...  )
```

The other method of inserting data into a collection is demonstrated below. For the second document the query would look like:

```
> document = ( { "Type" :  "Book", "Title" :  "Definitive Guide to MongoDB
2nd ed., The", "ISBN" :  "978-1-4302-5821-6", "Publisher" :  "Apress",
"Author":  [ "Hows, David", "Plugge, Eelco", "Membrey, Peter", "Hawkins,
Tim" ] } )
```

```
> db.media.insert(document)
```

The queries present all elements, which should be saved into the MB with the key and its value, where `"Type"` is the key and `"CD"` is the value. On the other hand, if the user wants to retrieve data from one collection, which contains several documents, the query would look like:

```
> db.media.find ( { Artist :  "Nirvana" } )
```

The query would search for all documents with an artist call `"Nirvana"`. Since the MD, in this case, contains only one document with this description, it will only display one document:

```
{ "_id" : ObjectId("4c1a86bb2955000000004076"), "Type" : "CD", "Artist"
: "Nirvana", "Title" : "Nevermind", "Tracklist" : [ { "Track" : "1",
"Title" : "Smells Like Teen Spirit", "Length" : "5:02" }, { "Track" :
"2", "Title" : "In Bloom", "Length" : "4:15" } ] }
```

In cases where user only wants to extract specific information out of the MD, and not all of the information in the document, the query may look as follows:

```
> db.media.find ( {Artist : "Nirvana"}, {Title: 1} )
```

The query would only search for artists with the value `"Nirvana"`, where only the title will be returned to the user.

```
{ "_id" : ObjectId("4c1a86bb2955000000004076"), "Title" : "Nevermind"
}
```