AALBORG UNIVERSITY
COPENHAGEN

Aalborg University Copenhagen
A.C. Meyers Vænge 15
2450 København SV

Semester Coordinator: Henning Olesen

Secretary: Maiken Keller

Semester: 4

Title: A gamification design framework recommender system

Project Period: 01/09/2018 – 06/06/2019

Semester Theme: Long Thesis

Supervisor(s): Iwona Maria Windekilde, Ph.D.

Project group no.: 34Bus L1

Members
(do not write CPR.nr.):
Attila Večerek

Pages: 103
Finished: 06/06/2019

Abstract:

Gamification is the use of game-design elements in non-game contexts. There are numerous gamification design frameworks that facilitate the process of designing gamified experiences providing methods and guidelines to the practitioners of gamification.
Less experienced practitioners might find it difficult to choose the most appropriate framework for a specific project. This thesis aims to find out how project requirements could be mapped to gamification design frameworks. The Design Science Research Methodology was adopted in order to develop a recommender system for a limited set of classified gamification design frameworks. The study recruited an interest group of gamification practitioners to validate the developed tool in terms of its usability and the relevancy of its recommendations. The study established the terms optimal and suboptimal frameworks and identified three main framework comparison factors: feature-completeness, domain specificity, and target specificity. It explored the possibility to merge different gamification design frameworks with the purpose of creating a single optimal framework.
The validation of the system revealed that additional research is needed in the area of gamification design framework classification to further improve the relevancy of the recommendations. The system was successfully deployed and is available to all gamification practitioners.

AALBORG UNIVERSITY IN COPENHAGEN

MASTER THESIS

A gamification design framework recommender system

Author:
Attila VEČEREK

Supervisor:
Iwona Maria WINDEKILDE,
Associate Professor

A thesis submitted in fulfillment of the requirements for the degree of Master in Computer Engineering

in the

Center for Communication, Media and Information Technologies

June 6, 2019

AALBORG UNIVERSITY IN COPENHAGEN

Abstract

Innovative Communication Technologies and Entrepreneurship
Center for Communication, Media and Information Technologies

Master in Computer Engineering

A gamification design framework recommender system

by Attila Večerek

Gamification is the use of game-design elements in non-game contexts. There are numerous gamification design frameworks that facilitate the process of designing gamified experiences providing methods and guidelines to the practitioners of gamification. Less experienced practitioners might find it difficult to choose the most appropriate framework for a specific project. This thesis aims to find out how project requirements could be mapped to gamification design frameworks. The Design Science Research Methodology was adopted in order to develop a recommender system for a limited set of classified gamification design frameworks. The study recruited an interest group of gamification practitioners to validate the developed tool in terms of its usability and the relevancy of its recommendations. The study established the terms optimal and suboptimal frameworks and identified three main framework comparison factors: feature-completeness, domain specificity, and target specificity. It explored the possibility to merge different gamification design frameworks with the purpose of creating a single optimal framework. The validation of the system revealed that additional research is needed in the area of gamification design framework classification to further improve the relevancy of the recommendations. The system was successfully deployed and is available to all gamification practitioners.

Acknowledgements

I would first like to thank my thesis supervisor Iwona Maria Windekilde, Associate Professor of the Center for Communication, Media and Information Technologies at Aalborg University. I always felt welcome at her office, whenever I ran into some issues or I had a question or two. She always provided honest and invaluable feedback. It was a pleasure to work with her.

I would also like to thank the colleagues from my workplace, who readily helped me with data collection and provided valuable feedback on the structure and layout of the questionnaires. I would like to especially thank Gorka Luis Cardona-Lauridsen who regularly checked up on my progress, helped me fine-tune the questionnaires and was the first beta tester of the developed tool.

Many thanks to Tim Allison who spent his valuable time reviewing my work from the perspective of design and user experience.

And last but not least, I would like to thank my manager, Liborio Cannici, for the enormous effort he put into the administrative processes to arrange the thesis collaboration with the company.

Contents

List of Figures

List of Tables

List of Abbreviations

API	**A**pplication **P**rogramming **I**nterface
BMC	**B**usiness **M**odel **C**anvas
CDN	**C**ontent **D**elivery **N**etwork
CLI	**C**ommand **L**ine **I**nterface
ES	**E**cma**S**cript
HCI	**H**uman-**C**omputer **I**nteraction
HTML	**H**yper**T**ext **M**arkup **L**anguage
HTTP	**H**yper**T**ext **T**ransfer **P**rotocol
JSON	**J**ava**S**cript **O**bject **N**otation
MDA	**M**echanics-**D**ynamics-**A**esthetics
NLP	**N**atural **L**anguage **P**rocessing
PCD	**P**layer-**C**entered-**D**esign
RPG	**R**ole-**P**laying **G**ame
RUP	**R**ational **U**nified **P**rocess
SDT	**S**elf-**D**etermination **T**heory
SEO	**S**earch **E**ngine **O**ptimization
SPA	**S**ingle-**P**age **A**pplication
SSL	**S**ecure **S**ockets **L**ayer
SVM	**S**upport **V**ector **M**achine
TAM	**T**echnology **A**cceptance **M**odel
TTR	**T**ime **T**o **R**esolution
UCD	**U**ser-**C**entered **D**esign
UI	**U**ser **I**nterface
URL	**U**niform **R**easource **L**ocator
UX	**U**ser **EX**perience

Chapter 1

Introduction

Gamification, the use of game design elements in non-game contexts [1], has drawn the attention of both academia and practitioners. It is no longer a question whether gamification works [2, 3]. Businesses use gamification to increase user engagement, customer loyalty, community activity, etc. [4]. However, in 2014, Gartner reported[1] that most of the companies adopting gamification in their products would fail to accomplish the outlined business goals due to poor design. A systematic survey of a selection of 12 theoretical and 30 implementation papers found that gamification brought positive-leaning but mixed results in most of the cases [3]. It argues that "early positive results may be subject to the phenomenon of regression to the mean due to the novelty factor associated with gamified systems". Many of the surveyed papers did not even define gamification as a starting point. It also found that "the majority of applied research on gamification is not grounded in theory and did not use gamification frameworks in the design".

Gamification design frameworks provide structure to the design process making gamification more effective. Some of the frameworks even focus on how to sustain the effects of gamification without the need to introduce new and fun features regularly [7].

The overall research regarding gamification and its practitioners suggest that it can provide value in several domains if approached correctly. Enterprises need sustainable solutions and consistent outcomes in order to adopt some technology or process. Defined business goals seem to be more achievable using gamification when systematic design processes are employed.

Today, there is a variety of gamification design frameworks available to designers, researchers, and practitioners of gamification in general. Different frameworks cover different needs. Several papers have been published in an attempt to classify them [5, 8, 6]. A recent study (2018) on gamification design frameworks explored 992 documents and selected 58 for critical comparative review [6]. However, there is no known rigorous method of translating specific project requirements into best fitting gamification design frameworks.

Even though there is this knowledge gap in the current literature about the use of gamification, it is an open question whether it is also perceived as a problem in practice. Experienced practitioners of gamification might have worked with a multitude of frameworks and created their own "universal framework". Others might have learned one specific framework and been tailoring it to their needs every time a new project introduces new requirements. Newcomers to the field or even less experienced practitioners might find it difficult to choose a framework from the wide

[1] The source is not available on Gartner's page anymore but several papers refer to the study [4, 5, 6]

selection of available ones. However, these examples only represent the practitioners who want to apply a framework. It is not uncommon to apply gamification in projects without an actual design framework [3].

1.1 Research questions

The aim of this thesis is to create a novel artifact, in form of an information system, that is able to translate project requirements into suggestions about what gamification design framework would be the most suitable to use. Such a tool could be used especially by the practitioners of gamification just entering the field. It could give them a starting point regarding what framework could be the best option to start with for their projects. In order to design and implement this tool, this project defines its main research questions and subquestions as follows:

RQ1. How to map project requirements to gamification design frameworks?

  RQ1.1. How could the correctness of the mapping be validated?

RQ2. How to implement a tool that takes project requirements as its input and returns recommended gamification design frameworks as its output?

  RQ2.1. Would such a tool provide utility to practitioners of gamification?

1.2 Scope and limitation

The scope of the project is to create a working prototype of a recommender system for a limited set of gamification design frameworks. It is not the aim of the thesis to collect and include all design frameworks in the tool. However, the system should be developed in a way that allows new frameworks to be easily added.

Furthermore, the project aims to test the assumption of whether the formulated problem also represents an issue in practice. The created artifact is set out to be validated with an interest group of gamification practitioners with a focus on the relevancy of the recommendations, the usability of the tool and the likelihood of the system being used in the future by the practitioners.

1.3 Expected outcome

This project expects to design and implement a working prototype of a gamification design framework recommender system. The produced recommendations should be relevant to practitioners of gamification and the created artifact should be easy to use.

Furthermore, it is expected to explore whether the knowledge gap regarding the transformation of project requirements into suggested gamification design frameworks also presents an issue in practice. If it proves to be an issue, the thesis expects to identify the user segment benefiting the most from the designed artifact.

Finally, the validation of the system should reveal whether the created artifact provides a sufficient amount of utility to its users to be adopted in practice.

1.4 Overview

Chapter 2 describes the adopted research paradigm, methodology and the architecture of this research. It also describes how the data used in this research were collected and analyzed.

Chapter 3 presents the reviewed literature relevant to this research. It briefly describes gamification in general, the different definitions found in the reviewed literature, the theoretical foundations of gamification as well as the literature regarding the classification of gamification design frameworks.

Chapter 4 analyzes the empirical data collected by the surveys described in Section 2.4.3. After that, it examines how project requirements could be mapped to design frameworks and explores the possibility to merge multiple frameworks with the purpose of creating a single optimal framework. It also delves into the different aspects that can be considered when comparing different frameworks. Finally, it infers the requirements for the design and implementation of the recommender system.

Chapter 5 describes the implemented recommender engine as well as the architecture of the whole system. It also presents the design of the user interface and describes the decisions made to provide a good user experience.

Chapter 6 evaluates whether the created artifact achieved its objectives defined by the requirements. It also examines the performance of the system. Then, it presents the results of the validation survey answered by an interest group of gamification practitioners. Finally, it presents the observations and suggestions of an independent expert in the field of design and user experience regarding the usability of the created artifact.

Chapter 7 summarizes the findings of this research, interprets them in relation to the defined research questions and discusses their implications for practice. Additionally, possible future improvements to the created artifact are also presented.

Chapter 2

Methodology

This chapter describes the means of designing and developing such an information system. Firstly, it describes the chosen research philosophy and the rationale behind this choice. Secondly, it describes the architecture of the research project which gives a basis for the employed data collection and data analysis techniques. Further on, it delves deeper into the data collection and analysis along with a discussion on ensuring data validity and reliability.

Finally, it presents the timeline of the research describing each stage of the project and the challenges faced during them.

2.1 Research paradigm

As suggested by Saunders et al. (2009) [9], it rarely happens that a particular research question would fall into one single philosophy domain. The nature of the project presented in this thesis is practical and choosing one specific philosophy is rather unrealistic. Hence, the researcher adopts the position of the *pragmatist* placing the research questions into the center of the methodology.

The previous chapter also claimed that the current state of the art in the field of gamification does not provide sufficient tools and conceptual frameworks to researchers and designers for choosing an optimal design framework for their gamification projects. This claim follows after an extensive literature review that did not yield any other recommender system or conceptual framework focused on this purpose. Starting from this claim, a deductive research approach seems to be suitable with the application of mixed method choices that provide more options to test it and validate the created artifact with a community of gamification design practitioners. These choices will be covered more in-depth within the next section.

2.2 Research methodology

This section describes in detail which research methodology has been chosen and the rationale behind it. Based on the claim and the research questions presented in the previous chapter, the following methodologies have been considered:

1. Action research

2. Design science research

Techniques such as questionnaires, interviews and focus groups could be used to ask the practitioners whether they really face the claimed issues. However, testing the claim is only of secondary importance to the thesis. An analogy could be made to the early 19th century car industry and Henry Ford's famous quote: *"If I* *had asked people what they wanted, they would have said faster horses"*. Similarly, some gamification designers might say that they would like to have more frameworks to choose from or more robust or universal frameworks. The researcher believes that more value could be provided by developing a tool that could help gamification practitioners better choose from the currently available frameworks instead. A recommender system could still provide utility to the community even if they were not aware of the problem formulated in the previous chapter.

The goal of *action research* (AR) is to iteratively solve a problem with a community of practice. The widely adopted definition of AR states the following: "Action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework" [10, p. 2]. It contributes to both practice and research at the same time and it assumes a concrete client being involved. Action research, in short, causes a planned intervention in practice and records what happens. Then, it compares the results with the predictions. Through this comparison, it creates knowledge which can also be used for further iterations of intervention and observation.

Design Science Research (DSR), on the other hand, does not seem to have a widely accepted definition [10]. This report adopts the definition of DSR by Iivari et al. (2009): "DSR is a research activity that invents or builds new, innovative artifacts for solving problems or achieving improvements" [10, p. 2]. A tight collaboration between researchers and a community of practice is not assumed by this methodology, although it aims at addressing a specific problem for a specific client. The major research contribution of DSR is to create new means of solving a certain problem. The creation of novel solutions is the main difference in DSR when compared to AR.

The solution addresses a general problem and is not developed in close collaboration with a specific client. However, the project aims to create a novel artifact to address the lack of a tool that allows gamification practitioners to select a design framework based on project requirements. Hence, Design Science Research has been chosen as the research methodology for this project.

Hevner identifies and presents three design science research cycles – depicted by Figure 2.1 – that exist in any design research project: (1) relevance cycle, (2) design cycle, and (3) rigor cycle.

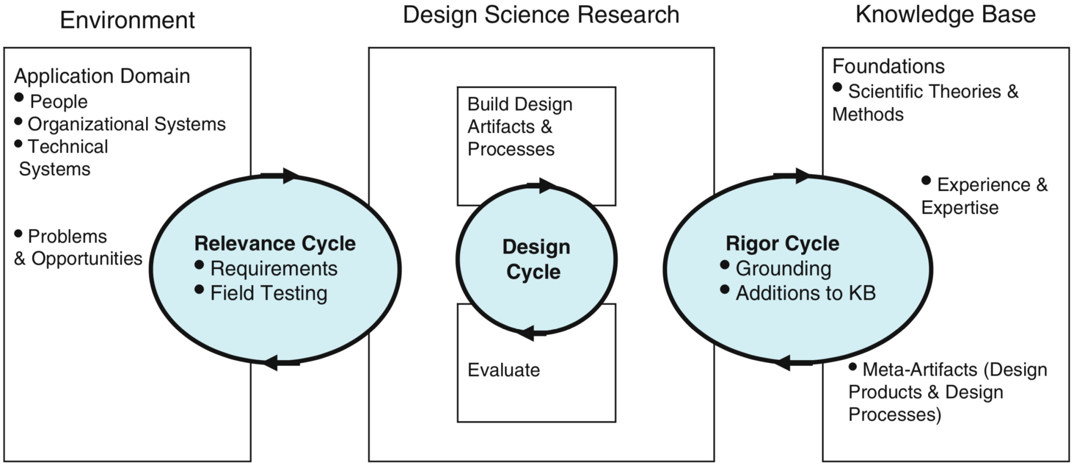
Relevance cycle is described as the one that initiates design science research with an application context providing the requirements for the research.

Rigor cycle "provides past knowledge to the research project to ensure its innovation" [11, p. 18].

Design cycle is described as the "heart of any design science research project" [11, p. 18]. In this cycle, multiple designs are evaluated against the requirements until a satisfactory design is achieved.

The thesis adopts the *Design Science Research Methodology* (DSRM) proposed by Peffers et al. [12] and described by A. Hevner and S. Chatterjee (2010) [11, p. 28]. This methodology was proposed and developed for the purpose of producing and presenting Design Science research in the field of Information Systems. By the words of Hevner et al. this methodology provides a "commonly accepted framework for successfully carrying out DS research and a mental model for its presentation" [11, p. 28].

FIGURE 2.1: Design Science Research Cycles [11, p. 16]



The described process in this framework constitutes of six major steps [11, p. 28-30]: (1) *problem identification*, (2) *definition of objectives and solution*, (3) *design and development*, (4) *demonstration*, (5) *evaluation* and (6) *communication*. The research strategy designed for this thesis is described in the following section.

2.3 Research design

This section describes the employed architecture of the research project. It sets up the basis for data collection and provides justification for the chosen techniques. It proceeds in regards to the six major steps of DSRM described in the previous section.

2.3.1 Problem identification

As the first step, a problem needs to be formulated. The problem definition needs to be conceptually atomic in order to encompass its complexity. The value of the solution needs to be justified in order to motivate the researcher and the reader to pursue the solution and accept the results. This step requires some knowledge regarding the state of the problem. It also helps to understand the importance of the solution. Hence, preliminary research has been determined to be needed. The preliminary literature review is further discussed in Section 2.4.1.

The problem formulation presented in Chapter 1 is based on the assumption that the amount of available gamification design frameworks makes the selection considerably more difficult, especially for the practitioners entering the field. The envisioned artifact could provide a solution to this problem. However, the assumption needs to be tested first. It could be achieved by surveying the practitioners to learn more about their use of gamification design frameworks. A questionnaire would also provide an opportunity to build a focus group of practitioners interested in the idea of a design framework recommender tool. Successful creation of a group of interested practitioners would prove the existence of a demand for the proposed solution and thus the presence of the claimed issue in practice. A more detailed description of the recruitment process is presented in Section 2.4.3.

2.3.2 Definition of objectives

The objectives of the solution are acquired through inference from the problem definition and knowledge about the state of the art in the field of gamification with a particular focus on gamification design frameworks and current solutions. To obtain this knowledge, an in-depth literature review is required. The process of literature review is thoroughly discussed in Section 2.4.2. The objectives are then rationally inferred from the problem specification and turned into *functional* and *non-functional requirements* prioritized using the *MoSCoW* method.

2.3.3 Design and development

In this step, the outcomes of the previous steps – knowledge of the problem domain acquired through literature review and the construction of functional and non-functional requirements – are used to design and develop an artifact in form of an information system. The design and development process will adhere to the values of extreme programming agile software development framework [13] and selected software development best practices. The aim of extreme programming is to produce higher quality software while keeping the developer team happy. It prioritizes developer efficiency over a long and detailed planning process involving numerous meetings with customers and other teams.

The main values of extreme programming are (1) communication, (2) simplicity, (3) feedback and (4) courage. Communication and respect are values mostly concerning a team of developers. The design and development have been produced by one person, the researcher himself. Hence, the adopted values of extreme programming are 2-4.

The meaning of *simplicity* lies in avoiding waste of time and resources and preferring solutions that are the simplest in their nature in order to facilitate ease of maintenance, support, and revision.

Through constant *feedback*, improvements can be identified and incorporated into the solution. It supports simplicity in the sense that simple design is required to be able to incorporate regular feedback and improvements.

Courage is required to raise issues, stop a development branch if it does not lead to improvement and incorporate feedback that is difficult to accept.

Extreme programming can be also perceived as an interconnected set of software development best practices. The following software development practices were adopted for the purpose of this thesis: *small releases*, *simple design*, *testing*, *refactoring* and *continuous integration*.

2.3.4 Demonstration

In this step, the solution needs to demonstrate the use of the artifact in order to solve one or more instances of the identified problems. It is achieved by the deployment of the project so that it can be publicly available. The reader is provided with access to the project so that he/she can try the solution in a real-life setting.

2.3.5 Evaluation

After the artifact is demonstrated, it can be evaluated based on how well it supports a solution to the problem. It is achieved by comparing the objectives of the solution – the functional and non-functional requirements – to the actual observed results from the use of the artifact in the demonstration. The evaluation will proceed in three steps:

1. The researcher assesses the performance of the artifact using different tools and frameworks further described in Section 6.1.2.
2. A group of gamification practitioners is asked for their feedback through a validation survey.
3. Additionally, an expert in the field of user experience (UX) is asked to review the artifact from the perspective of its usability.

The recruitment process of the group is described in Section 2.4.3 while the means of conducting the expert interview is presented in Section 2.4.5.

2.3.6 Communication

The last step was proposed by Archer (1984) and Hevner et al. (2004) to diffuse the resulting knowledge [11, p. 30]. Other than presenting the project in this thesis, a website is also developed to communicate the tool and its utility towards relevant audiences such as gamification designers or any other person interested in gamification design. Also, the project is intended to be open source and published on GitHub. In their own words, GitHub is a platform that "brings together the world's largest community of developers to discover, share, and build better software."[1]. It is probably the largest repository of open source projects in the world. At the time of writing, it listed more than 30 million public code repositories[2]. By hosting the code as a public repository on GitHub, other researchers or gamification practitioners with software engineering skills are able to contribute to the further development of the project. Feature requests and issue reporting is also available to anyone interested in the project. New frameworks can be suggested via a GitHub issue[3] to be added in the initial collection of recommendable gamification frameworks after a proper review.

2.4 Data collection

This section outlines the process of data collection. Firstly, it describes the preliminary literature review used to acquire knowledge in the problem domain. Then, it continues with the literature review that serves the purpose of acquiring knowledge leading to a solution to the identified problem. Section 2.4.3 describes the purpose, structure, and layout of the questionnaires used in the study.

One of the goals of the thesis was to build a focus group of gamification practitioners. However, due to the difficulties presented by the different geographical area and timezones of the respondents interested in the research a focus group could not be successfully created. Henceforth, the group of successfully recruited gamification practitioners will be referred to as the "interest group". The interest group was employed with the purpose of validating and evaluating the created solution. The recruitment process of this group is also described in this section.

Furthermore, the interview process with an expert in the field of design and user experience was also conducted and is described in Section 2.4.5.

[1] https://github.com/

[2] https://github.com/search?q=is:public

[3] https://github.com/vecerek/how-to-gamify/issues/new

Finally, the techniques used to ensure data validity and reliability throughout the entire process of data collection are summarized in Section 2.4.6.

Surveys were selected as one of the main data collection methods. The reason they were selected over expert interviews was that the research expected to collect larger amounts of data. Also, the nature of the collected data was more quantitative than qualitative. The drawback of the surveys was that the study did not gain very detailed insights from the practitioner into their day-to-day practice in the area of gamification design. It was also more difficult to keep in touch with the interest group and only the most motivated participants stayed active throughout the entire length of the study.

The usability study conducted as part of the validation study presented in Section 2.4.3 was complemented by an expert interview presented in Section 2.4.5. An expert in the field of design and user experience provided very detailed feedback regarding the UI and UX design of the created artifact. Data with such detail could hardly be collected only through the surveys.

2.4.1 Preliminary literature review

The purpose of the preliminary literature review was to gain general knowledge in the area of gamification. The timeframe for the preliminary literature review lasted from October 2018 to January 2019. The most important points of interest of the general search were: definitions, terminology and taxonomy, underlying theories of gamification, design guidelines and case studies. Publicly available databases and/or search engines such as *Google* and *Goodle Scholar* were used along with the literature suggested by the project supervisor. The keywords were the following: *gamification*, *gamification best practices*, *gamification design framework*, *gamification of enterprise software*, *gamification failures*, *gamification successes* and alike. As part of the preliminary review, the following literature was reviewed: 2 books about gamification, 9 gamification design frameworks, 4 motivational theories used in gamification and 3 papers regarding player taxonomy. The reviewed literature are listed in Table 2.1.

TABLE 2.1: Preliminary literature review

Category	**References**
Books	[14], [4]
Gamification design frameworks	[15], [16], [7], [17], [18], [19], [20], [21], [22]
Motivational theories	[23], [24], [25], [26, Chapter 1 and 2]
Player taxonomies	[27], [28], [29]

The preliminary literature review led to the discovery of the gamification design framework classification paper by Mora et al. (2015) [5] which in turn led to the problem identification and further literature review described in the next section.

2.4.2 Literature review

The goal of the literature review was to learn more about gamification design frameworks, their attributes, and possible classification. This section describes the process behind the literature search, filtration, and review. In order to stay focused on answering the research questions presented in Section 1.1, it was required to establish criteria guiding the literature review process.

Search The timeframe for this literature review lasted from January 2019 to February 2019. During this period, a rigorous search of the academic literature was undertaken in the subject area using *Primo*, *Scopus*, *Science Direct (Elsevier)*, and *IEEEXplore*. This search was primarily focused on the research question *RQ1*. Hence the selected keywords were mostly interested in the academic work around the classification of gamification design frameworks. The same three queries were run for each database. Table 2.2 shows the number of results for each query per database. The queries used were the following: (SQ1) *gamification AND design AND frameworks*, (SQ2) *gamification AND classification AND frameworks*, and (SQ3) *gamification AND frameworks AND review*.

Database	**SQ1**	**SQ2**	**SQ3**
Primo	1 908	702	28 435
Scopus	363	31	89
Science Direct (Elsevier)	824	289	712
IEEEXplore	52	4	12

TABLE 2.2: Literature review search results

Filtration In some cases thousands of results were returned but only the first 100 were taken into consideration. It was observed that the relevancy of the results would deteriorate after the first 50 items very quickly in each database returning results for serious games, game design frameworks, etc. The search results went through a process of filtering which happened in two steps. The first filtration step *(F1)* examined the title of the results. Only titles related to gamification design framework studies or design guidelines passed through the filter. Results related to serious games, game design, specific gamification frameworks, systematic map and review studies of the application of gamification were all rejected since they were irrelevant to this phase of the project.

The second step of filtration *(F2)* examined the abstract of the result. The focus of this filtration was to distinguish generic approaches from ones that focused on a specific domain. For example, a systematic review of frameworks in the context of healthcare, education, etc. was rejected since the goal of this project is to create a generic solution. Duplicates were also eliminated in this step. That is if the paper had already been reviewed it would not be selected for another review. Table 2.3 shows the number of results passed through each filter and the number of results selected for review.

Database	**F1**	**F2**
Primo	17	3
Scopus	7	2
Science Direct (Elsevier)	11	4
IEEEXplore	8	0
Total	43	9

TABLE 2.3: Literature review filtration

Review In total, nine research papers related to gamification design framework studies were reviewed. The number of relevant papers found indicates that the field of gamification is still very young and analyzing different design frameworks and guidelines is a niche within the field.

Out of the nine reviewed papers, only three were identified as generic gamification framework classification studies. Two out of the three classification studies included a direct mapping of gamification design frameworks to classification features [5, 8]. One only provided aggregated and statistical data about frameworks [6]. The authors of the paper were contacted and asked to provide the detailed results of their work. Section 3.3 describes these papers in detail.

The rest of the reviewed works were either limited to certain journals and conferences in a particular field or were not identified as classification studies.

2.4.3 Questionnaires

During the project, a total of three questionnaires have been employed. The first questionnaire was designed with the three main goals: (1) learn about the gamification design practice, (2) test the claim presented in the problem formulation and (3) recruit gamification practitioners to join the interest group. The goals of the second questionnaire were to learn about the project requirements of the created interest group as well as identify the preferences regarding gamification design frameworks. Lastly, the third questionnaire was a validation survey regarding the created artifact. The following subsections will describe the design of each questionnaire.

Questionnaire 1: Testing and recruiting

The objectives of this thesis evolved over time. Initially, the study was focused on the enterprise context of gamification. Hence, the wording of the questions presented in Appendix A.1 is centered around the business use of gamification. However, around 50% of the respondents who signed up for follow-up surveys came from the field of education and could interpret the questions in their own context. That allowed the research to retrofit the purpose of the questionnaire when the objectives of the thesis changed to provide a generic solution.

The questionnaire was internet-mediated and self-administered. Its layout and structure were designed in a way that allows the participants to be routed to different paths so that the amount of questions that need to be answered is minimal and the questions are always relevant. The project defined three participant segments and constructed a particular path for each: (1) non-practitioners of gamification, (2) casual practitioners and (3) advanced practitioners. *Non-practitioner* is a person who has never used gamification in a product or artifact. A *casual practitioner* is a person who may have used gamification but without utilizing a gamification design framework. An *advanced practitioner* is someone who knows what a gamification design framework stands for and has experience utilizing one.

To achieve a higher response rate, the survey contained an introduction section on the purpose of the survey. It stated that the goal of the survey was to learn more about the practice of gamification practitioners. The focus of this survey is the casual and advanced practitioners of gamification.

The first two questions of the questionnaire serve the purpose of segmenting the participant. Both are list questions with a yes/no answer. Together, they determine which segment the participant belongs to.

1. Have you used gamification in your product(s) to drive customer engagement, loyalty, or experience?
2. Have you been using a particular gamification design framework to design your gamified experience?

Table 2.4 lists the questions a casual practitioner would be asked and provides further details such as type of question, possible answers, if applicable, and the rationale.

TABLE 2.4: Questionnaire 1: casual practitioner

Question	Type	Answers	Rationale
Have you read/heard of any specific gamification framework?	List	Yes/no	At this point, it is known that the respondent has not used a gamification design framework before. The question allows the study to see whether the practitioner is aware of any framework.
If yes, could you describe the framework you are the most inclined to adopt at your company?	Open	N/A	Through this question, the study can learn whether there is a preference for a certain design framework.
What are your reasons for the above choice (if applicable)?	Open	N/A	The rationale behind the preference, if there is one, can be learned. It may tell the study more about what frameworks are of interest to the practitioners to learn what frameworks to include in the recommender system in the future.

End of Table

Table 2.5 lists the questions specific to the advanced practitioner similarly to the previous table.

TABLE 2.5: Questionnaire 1: advanced practitioner

Question	Type	Answers	Rationale
Please, describe the design framework(s) you've adopted at your company.	Open	N/A	The study can learn which design frameworks have been used by the practitioner.

Continuation of Table 2.5

Question	Type	Answers	Rationale
How satisfied are you with the currently used gamification design framework(s)?	Rating	Very unsatisfied (1) - Very satisfied (5)	The purpose of the question is to learn whether there might be an incentive to change the framework in the future. Possibly using the developed solution.
Do you customize/-modify the design framework(s) to fit your specific needs?	List	Yes/No/ Sometimes	Practitioners might choose to customize or modify a framework to suit their needs rather than learn to use a new framework. Through this question, it can be identified how often it is the case.
Do you plan on moving to other framework(s)?	List	Yes/No/ Maybe	This question is sort of a control question to the satisfaction question. It could be tested whether there is a relationship between the satisfaction with a framework and the plan of changing the framework.

End of Table

After the last questions, the paths are joined to complete the last two objectives of the questionnaire: (1) test the claim and (2) recruit practitioners to be part of a focus group reviewing the created artifact. The participants were presented with the other aim of the questionnaire, which is to create a tool that facilitates the process of choosing the best fitting framework based on specific project requirements. Table 2.6 describes the questions asked in the last part of the survey.

TABLE 2.6: Questionnaire 1: testing and recruiting

Question	Type	Answers	Rationale
Would you like to be part of a focus group reviewing the created tool?	List	Yes/No	Answering yes to this question indicates that there might be a demand for the proposed solution and also serves the recruiting purposes of the study.
Name?	Open	N/A	Collects the participant's name in order to be more personal in the e-mail communications to follow.

Continuation of Table 2.6

Question	Type	Answers	Rationale
What is your role in your company?	Category	Designer/ Software engineer/HCI researcher/ Other	More information about the background of the participant can be revealed and also he/she may be matched to certain gamification frameworks targeting his/her background.
How many years of experience do you have designing gamified experiences?	Category	No experience/0-1 years/1-2 years/3-4 years/4-5years/5+ years	It allows the study to see whether certain preferences and use of gamification design frameworks depend on the amount of experience of the practitioner.

End of Table

The questionnaire also collected the e-mails of the participants who decided to join a focus group, further on referred to as the interest group. The design and structure of the questionnaire were consulted with the project supervisor. It was not pilot tested, although the first iteration of the data collection can be perceived as a test.

Sample size and population In its first iteration, the questionnaire was disseminated to organizers of UI and UX design Meetups. There are considerably more and larger communities of designers than in the niche area of gamification design. Thus, it was thought that more responses could be collected. However, it turned out to be a false presumption. The engagement and participation were very low and the responses revealed that not all designers, in general, share an interest in gamification.

To collect more and more relevant responses, the second iteration of data collection focused only on gamification practitioners. Organizers of Meetups related to gamification design were contacted and asked to share the questionnaire with their community. In total, 24 organizers of gamification Meetups all over the world were contacted. Also, several Facebook groups of gamification practitioners were directly contacted. The results of the second iteration led to substantially more responses and sign-ups for the interest group.

Questionnaire 2: Preferences and initial feedback

The second questionnaire was only targeted at the assembled interest group. Its objective was to gather data regarding the project requirements of gamification practitioners, their preferences in terms of different types of gamification design frameworks as well as their initial feedback on the developed artifact. The questionnaire can be conceptually divided into three main parts:

1. Project requirements
2. Framework preferences

3. Feedback

In the first part, the participant was asked the following question: "*What concepts should a gamification framework tackle in your opinion? Think of a project and its requirements and list them in the answer box below.*" The question is open-ended and asks the participant to brainstorm what concepts does he or she require from a design framework to tackle. Then, the participant is shown 21 features, as identified by Mora et al. (2017) [8], and asked whether the feature could be considered as a project requirement. The possible answers are: (1) *yes*, (2) *no* and (3) *I don't know/I'm not sure*. The questions ask for the opinion of the participant and only serve the purpose of seeing whether there is any pattern in the considered requirements. It is crucial to ask the participant to brainstorm first without biasing his/her ideation by showing the list of requirements prepared in this study. After answering all of these, the participant is asked to reflect on the seen requirements and answer whether he/she had any additional ideas after seeing the list. The rationale is that seeing the list helps the participant to better understand the concept of the design framework related requirements and might also help with additional brainstorming.

The second part of the survey focuses on exploring the preferences of the participants in regards to the aspects of gamification frameworks. The analysis in Section 4.3 identifies three aspects of gamification frameworks based on the research of Mora et al. (2017) [8]. The questions asked in this part should uncover which aspects are more important to the participants. There are three identified aspects of framework comparison described in Section 4.3. The answers would help determine the optimal default weights for the respective comparison aspects. These weights would be then applied in future iterations of the recommendation engine. The purpose and the exact use of these weights are explained in Section 5.1.2.

Before the participant proceeds with the questions, the three aspects of framework comparison are thoroughly explained:

Feature completeness How many of the features your project requires is actually covered by the framework. A gamification design framework is feature-complete when it tackles all the features you required and nothing more (no extra features present in the framework).

Application area/domain Some frameworks are specifically designed for certain application areas like business, education, health, etc. Others may be generic (all-purpose) frameworks.

Target Some frameworks are specifically written for a certain target audience: people with a specific skill set. For example frameworks for designers, teachers, software developers, etc. Others may be targeted at a general audience (no specific skill set required).

Table 2.7 lists the questions and the available answers to this part of the questionnaire.

TABLE 2.7: Questionnaire 2: framework preferences

Question	Available answers
How important do you find feature-completeness?	Not important (1) - Very important (5)

Continuation of Table 2.7

Question	Available answers
How important do you find the matching application domain?	Not important (1) - Very important (5)
How important do you find a matching target (the target audience of the framework)?	Not important (1) - Very important (5)
Would you favor a generic but feature complete framework or a specific one with 1-2 features not being covered?	Generic and feature complete framework / Domain specific, less feature complete framework
Would you favor a feature complete framework written for the general audience or one that is specifically targeted at your particular skill set but missing 1-2 features?	General and feature complete framework / Skills specific, less feature complete framework
Would you rather favor a generic framework targeted at your skills or a specific framework written for the general audience? (consider both feature complete)	Generic framework, matching my skill set / Specific framework, for the general audience

End of Table

The final part of the questionnaire asked the participants to imagine an arbitrary gamification project and use the created artifact to select a framework based on their project requirements. After completing the task, the participants were asked to leave feedback regarding the usability of the system. The researcher had hoped that the participants would leave detailed feedback with all their observations so that they could be used to further improve the artifact. However, only three such feedbacks were collected out of a total of 10 responses. The results led to the creation of the third and last questionnaire. The low response rate to the last question might have been due to the substantial length of the survey. The complete survey with its exact wording can be found in Appendix A.2.

Questionnaire 3: Validation survey

The feedback section in the previous questionnaire received a low number of responses. Hence, a third questionnaire had to be created and sent out to collect validation data regarding the use of the artifact by gamification practitioners. The participants were asked the same task as in the previous questionnaire and then answered four questions. This time, 6 responses were received in total. The complete survey with its exact wording can be found in Appendix A.3.

Instruments

The questionnaires were created and their responses gathered using *Google Forms* and *Google Spreadsheets*. The raw data of the collected responses are linked under the respective section of the questionnaires. The names and e-mails collected in the responses are purposefully redacted to respect the privacy of the respondents. The created questionnaires were disseminated to the relevant audiences via *Meetup.com* and *Facebook Groups*.

2.4.4 Interest group

An interest group was established using the first questionnaire, described in Section 2.4.3, in order to learn more about the different project requirements a gamification practitioner might face in practice. In total, 45 responses were collected and 27 respondents showed interest to participate in the interest group. Out of the 27 respondents, two did not provide a valid e-mail address. The members of the interest group have varied amount of experience in gamification design. The interest group has a cumulative experience of approximately *65.5 years* in gamification design.

Due to the challenges presented by differing timezones and geographical area of the individuals in the group, the researcher chose questionnaires as the main channel of communication. These challenges also contributed to the requirements set for the developed artifact as described in Section 4.2. A total of two questionnaires have been sent out to the participants described in Section 2.4.3.

2.4.5 Interviews

During the study, one interview has been arranged in order to review the created artifact from the perspective of usability with an expert in the field of user experience. The expert assessment took the form of a semi-structured interview. The artifact was presented to the expert who was then asked to imagine an arbitrary project to be gamified and perform a search for a gamification design framework based on the project's requirements. The expert was asked to think out loud and communicate his observations. The session was recorded and the observations and feedback summarized by the researcher. The summary was sent to the expert for confirmation and possible corrections ensuring the validity of the summary.

2.4.6 Data validity and reliability

This section discusses the threats to the reliability and validity of the study in the respective order. Data cannot be valid without being reliable and consistent measurements do not ensure validity. Saunders et al. [9, p. 156-157] describes four threats to reliability and six threats to validity. The threats to reliability are: *subject or participant error*, *subject or participant bias*, *observer error* and *observer bias*. The threats charted to validity are: *history*, *testing*, *instrumentation*, *mortality*, *maturation* and *ambiguity about casual direction*.

Reliability

Participant errors may occur when research subjects are giving inaccurate responses. The wording of the questions was chosen in a way to reduce the risk of misinterpretation. The questions were mostly related to the practice of the participants and their personal opinions or stance. If the questions are interpreted correctly, the risk of providing inaccurate answers is minimized.

The *participant bias* is a tendency of participants to consciously or subconsciously respond to questions in a way they think the researcher would like them to respond. The validation survey described in Section 2.4.3 is particularly prone to this type of bias. The researcher is aware of this and it led to the decision to also conduct an interview with an expert in the field of user experience to assess the usability of the created artifact. The bias was minimized in the other questionnaires by reducing the number of details given to the participants about the purpose of the study. For instance, the questionnaire testing the claim presented in the problem formulation only described the purpose of the study, which is to create a requirements based gamification design framework recommender system, without mentioning the underlying claim itself.

Observer errors are less likely to occur in this project because of the chosen research methodology. However, there is still room for certain errors regarding questionnaires.

For instance, *sample frame error*. As described in Section 2.4.3, the first iteration of the questionnaire revealed that the population it was targeted at was too generic. The response rate was low and the received answers were mostly not so relevant. Hence a second iteration was undertaken targeting only a population of gamification practitioners.

A *selection error* is also very likely to occur in situations when respondents self-select their participation as it happened in the case of the interest group. The respondents were asked if they wanted to join and only then would the e-mail addresses be collected. Hence, the researcher could not send a second survey request to those who did not want to participate. However, multiple survey requests had to be sent out in the case of the second and third questionnaire in order to receive more responses. The researcher is aware of the possibility of selection error.

Sampling errors are also likely to occur when the sample size is low. In the case of this study, it might have happened if the members of the interest group had a similar background. For example, they had worked for the same company, came from the same industry or had the same amount of experience. Fortunately, the collected demographic data show that the interest group is diverse in all the listed aspects. If it had not been the case, the researcher would have had to continue with the recruitment process until a fair distribution of practitioners would be achieved.

Validity

Validity is concerned about the findings and whether the measurements represent what they were meant to represent (internal validity) as well as the generalisability of the results (external validity).

The way of testing the claim in the problem formulation, as presented in Section 2.4.3, is open to *internal valididty* threats. It was assumed that participants joining the interest group translates to a demand for the proposed solution. However, people might have joined for different reasons such as altruism or curiosity. To really test the demand, a validation survey was sent to the members of the interest group. One of the questions asked how likely they were to use the tool in the future. The likelihood of the further use of the created artifact is a more valid indicator of the demand.

The *external validity* of the research is threatened when it takes place in a single organization or a small number of organizations. Even though the created interest group shows to have members with a diverse background within the field of gamification, the results of this study cannot be generalized due to the small number of participants.

To ensure the *face validity* of the questionnaires, they were tested by the project supervisor, friends, and colleagues from the field of customer success and design.

The first questionnaire, described in Section 2.4.3, was also pilot tested with the initial group of user experience designers.

2.5 Research timeline

This section presents the timeline of the research, depicted by Figure 2.2. The project started with the stage of preliminary literature review – as described in Section 2.4.1 – and lasted for $\tilde{9}$ weeks. The objectives of the research were not clearly defined yet at this point and they were constantly being shaped by the new findings in this phase of the data collection.

The main literature review, as described in Section 2.4.2, lasted for about 5 weeks and was more focused on the objectives of the research which were clearly defined at this point.

The analysis of the research took the longest time of all the stages. It was an ongoing process throughout the whole duration of the research and encompassed all the data collection activities described in Section 2.4: literature review, three surveys, and the expert interview.

The data collection phase of the first survey started during the second week of March 2019 and lasted for about 2 months. Locating the relevant audience for the survey proved to be a challenging task, as the questions were targeted at a very specific audience: gamification practitioners. The first Meetup and Facebook groups organized by gamification practitioners were contacted during the first week of April. Contacting the relevant audience was an ongoing activity until the second week of May. Most of the gamification groups were located at *Meetup.com* which presented another challenge as their system blocked the account of the researcher for about a week due to suspicious behavior triggered by the messages sent to the organizers of the groups. The account was reinstated after contacting the support service of the platform.

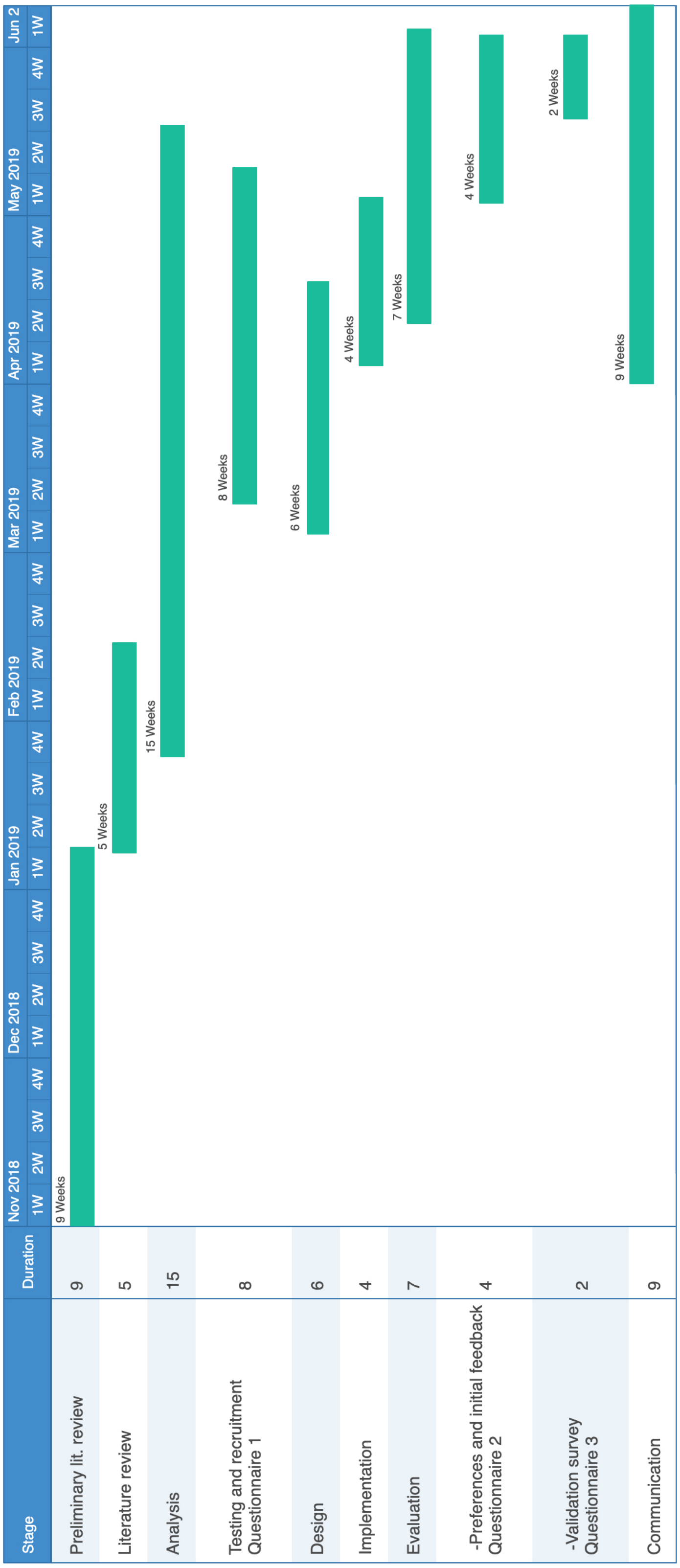
The design stage of the project started shortly after the end of the literature review and lasted for about 6 weeks. The design of the system was happening parallel to the analysis of the requirements presented in Section 4.4.

The implementation stage started shortly after and had an effect on the design stage. It lasted for approximately 4 weeks. The design and implementation affected the processes of each other and it is difficult to separate them completely. Hence they were executed in parallel.

The evaluation of the project started during the implementation. The observed performance of the system during the implementation stage also affected its design. This stage of the project lasted for about 7 weeks. Part of this process was also the evaluation of the implemented system by the assembled interest group, which started at the beginning of May as the second questionnaire was sent out. This stage of the research is described in Section 2.4.3. The evaluation continued with the validation survey described in Section 2.4.3. The data for these surveys were collected for about 4 and 2 weeks, respectively.

The last stage of the research is Communication and is described in Section 2.3.6. During this stage, an open source repository was set up for this project. This stage also included the work on the presentation of this thesis. This stage lasted for about 9 weeks.

FIGURE 2.2: Gantt chart: Research timeline



Chapter 3

Literature Review

This chapter presents both a general overview of gamification and the literature that has been reviewed based on the criteria presented in Section 2.4.2. It starts with a brief introduction to gamification and presents several definitions of gamification.

Section 3.2 introduces the theoretical foundations of gamification It introduces several motivational theories and behavior change models used as the basis of different gamification design frameworks.

Finally, Section 3.3 reviews the literature classifying and categorizing gamification design frameworks. The classifications facilitate the comparison of gamification design frameworks and might also take the study a step closer to answering the research question *RQ1*.

3.1 Definitions

Having a clear definition of gamification is considered by Mora et al. [8] as a prerequisite for proposing any gamification design framework. There is no unified and agreed upon definition of gamification. This section presents the definitions mentioned in the reviewed literature.

Deterding et al. (2011) Gamification is the use of game-design elements in non-game contexts [1].

Pelling (2011) Gamification is the application of game-like accelerated user interface design to make electronic transactions both enjoyable and fast [30].

Merino de Paz (2013) Gamification is the application of game elements and theories to non-game contexts with the intention of modifying behaviors, increasing fidelity or motivating and engaging users [31].

Zichermann and Linder (2010) Gamification is a tool for supplementing branding initiatives through the application of game elements and mechanics [32].

Huotari and Hamari (2012) Gamification refers to a process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation [33].

The most widely adopted definition seems to be the one from Deterding et al. (2011) [1] with 1586 citations at the time of writing based on the search result in Google Scholar[1]. Their definition suggests that gamification is centered around the

[1] `https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Gamification.+using+game-design+elements+in+non-gaming+contexts&btnG=`

application of game elements. Since these are difficult to define, the authors derived five levels of abstraction from the literature, shown in Table 3.1.

Design frameworks usually tackle all five levels of abstraction in varying levels of detail. Nonetheless, knowing these concepts might be helpful in discovering similarities between design frameworks which might be a key point in answering the research question *Q1.1*.

TABLE 3.1: Levels of Game Design Elements [3]

Level	**Description**	**Example**
Game interface design patterns	Common, successful interaction design components and design solutions for a known problem in a context, including prototypical implementations	Badge, leaderboard, level
Game design patterns and mechanics	Commonly reoccurring parts of the design of a game that concern gameplay	Time constraint, limited resources, turns
Game design principles and heuristic	Evaluative guidelines to approach a design problem or analyze a given design solution	Enduring play, clear goals, variety of game styles
Game models	Conceptual models of the components of games or game experience	MDA [34]; challenge, fantasy, curiosity; game design atoms; CEGE
Game design methods	Game design-specific practices and processes	Playtesting, playcentric design, value conscious game design

3.2 Theoretical foundation

Gamification is a very broad concept built on top of theories from various fields such as *HCI* and *psychology*. This section lists several of the concepts and theories that have been found during the literature review to broaden the general knowledge of the reader about gamification. These concepts were mentioned in classification papers reviewing gamification design frameworks using these concepts or other studies reviewing the literature regarding the publications in the field of gamification in general. Only some of these theories and models are presented in this section as detailed knowledge about them is not required to build a recommender system. The metadata regarding which gamification framework builds on top of which theories is sufficient to possibly provide relevant results.

Seaborn et al. analyzed several theoretical and implementation papers and compiled a list of theories the gamification frameworks relied upon [3]:

- Self-Determination Theory (SDT) [24]
- Intrinsic and Extrinsic Motivation [23]
- Situational Relevance [35]

- Situated Motivational Affordance [25]
- Universal Design for Learning [36]
- User-centered Design [37]
- Transtheoretical Model of Behavior Change [38]

As the list suggests, gamification has its foundation mostly in motivational and behavior change theories while placing the user in the center of the design process. Some frameworks take the idea of UCD further and refer to users as *players*. Several theories emerged around player typology, as well. In the following subsections, some of the above theories and models will be presented.

3.2.1 Self-Determination Theory

The SDT of Deci and Ryan is an approach to human motivation and personality [24]. It investigates people's growth tendencies and innate psychological needs. The theory identifies three such needs: (1) *competence*, (2) *relatedness*, and (3) *autonomy*.

Competence is the desire to be appreciated, respected, and recognized for something valued. Receiving unexpected positive feedback increases one's intrinsic motivation.

Relatedness is the desire for interpersonal attachments; the need to belong, interact, be connected, and experience caring for others.

Autonomy is the desire to be in control of one's own life while accompanied by the feeling of volition instead of independence, detachment or selfishness. Autonomy has a positive link to intrinsic motivation which increases with higher freedom of choice. It has also been found that extrinsic rewards for behaviors motivated intrinsically decrease the intrinsic motivation because of undermined autonomy.

3.2.2 Intrinsic and Extrinsic Motivation

This work loosely builds on top of SDT and was produced by the same authors [23]. It concludes that extrinsically motivated behaviors may become more self-determined, hence intrinsically motivated, through processes called *internalization* and *integration*.

Extrinsically motivated behaviors are the ones that are executed for some separable consequence, which is usually controlled by others (e.g.: superiors, teachers, parents).

On the other hand, *intrinsically motivated behaviors* are performed out of interest to satisfy the abovementioned innate psychological needs. Hence, they are the "prototype of self-determined behavior" [23].

Internalization is the process of "taking in a value or regulation" [23]. *Integration* takes this process one step further by integrating the accepted regulation into the self.

3.2.3 Situated Motivational Affordances

Deterding in his work argued that the models of video game motivation, at that time, were too general and were not linked to single specific game design patterns or elements [25]. He used the concept of *motivational affordances* to approach this gap. This concept builds on top of satisfaction theories of motivation, such as SDT. Motivation is afforded when the abilities of a subject meet the requirements of an objective. He uses the example of a Sudoku puzzle, which affords an opportunity to experience competency relatively to the person's skills and knowledge. However, this concept ignores the social situation or context of the subject.

"*Situated motivational affordances* describe the opportunities to satisfy motivational needs provided by the relation between the features of an artifact and the abilities of a subject *in a given situation*" [25]. This complex affordance is be broken down to (1) situational affordance and (2) artifactual affordance. These two affordances together determine whether an interaction is successful and the motivational needs are satisfied.

3.2.4 Flow Theory

Flow describes an optimal state of mind that allows a person to be engaged for long uniterrupted period of time as described by Csíkszentmihályi (1992) [26]. Such a deep level of focus and immersion requires a fine balance between the skills a person has and the challenges he/she faces. If the objective is too challenging relatively to the skills, the person might become too anxious and lose motivation to continue in the endeavor. On the other hand, if the skills of the person outweigh the challenge, the outcome might still be the same due to boredom. Flow Theory is also a common concept used in game design to create an immersive and lasting experience that keeps the player engaged for hours.

3.2.5 Bartle taxonomy of player types

Bartle (1996) classifies video game players based on their preffered actions within games [27]. This character theory divides players into four quadrants: (1) *Achievers*, (2) *Killers*, (3) *Socializers* and (4) *Explorers*.

Achievers These players are highly motivated by gaining points, levels, rare equipment, badges, etc. in order to belong to an elite group of high achievers elevating their social status among other players. Seeing clear progress drives them to take further actions to reach their goals.

Killers The players referred to as "Killers" are highly competitive who prefer player versus player game modes with the ambition of becoming the best. Climbing the leaderboard is highly satisfying for them because it clearly shows their progress compared to the other players.

Socializers This archetype describes players who enjoy building relationships since the very start of the game. They join guilds, clubs, factions, etc. to meet other people, make friends, chat and/or talk to them while exchanging experiences to learn more about the game and themselves.

Explorers These players find joy in the discovery of new areas, creating maps, finding hidden places, glitches or easter eggs. They also enjoy completing quests while paying attention to the lore while gaining experience points. Explorers tend to avoid fights and see them merely as something that hinders their exploration.

3.2.6 Yee's player motivation model

This model is often referred to as "Yee's player types" even though the paper itself does not categorize players [28]. Instead, it used a factor analytic approach to create an empirically grounded *player motivation model*. It revealed that Bartle's assumption of play motivations having a suppressing effect on each other is not valid. It means that a player can be a Killer, Achiever, Explorer, and a Socializer in varying degrees.

The model consists of three motivation components: (1) *Achievement*, (2) *Social*, and (3) *Immersion*. These three major components can be further divided into the following ten motivation subcomponents:

Achievement component Advancement, Mechanics, Competition.

Social component Sozializing, Relationship, Teamwork.

Immersion Discovery, Role-Playing, Customization, and Escapism.

3.2.7 Hexad User Types

Marczewski [29] proposes six user types in his Hexad model as personifications of people's intrinsic and extrinsic motivations. It is a different approach than the one taken by Bartle where the player types are derived from the observed player actions. Also, Marczewski argues that Bartle's and Yee's models focus on one specific genre of games, Massively Multiplayer Online Role-Playing Games (MMORPGs).

This character theory is based on the three types of intrinsic motivation from *SDT* – described in Section 3.2.1 – with the addition of *purpose*. The player types of this model are: (1) *Philanthropists*, (2) *Socializers*, (3) *Free Spirits*, (4) *Achievers*, (5) *Players* and (6) *Disruptors*.

Philanthropists are motivated by *purpose*. They are ready to help other players without expecting any reward.

Socializers are motivated by *relatedness*. They are the same as the Socializers in Bartle's taxonomy.

Free Spririts are motivated by *autonomy*. They are defined similarly to Bartle's explorers. Free Spirits like to create and explore.

Achievers are motivated by competence and match the definition of Bartle's Achievers.

Players are motivated by *extrinsic rewards*. They take actions regardless of their type as long as it takes them closer to earning rewards in the system.

Disruptors are motivated by the triggering of *change*. They like to be the disrupting force in the system. Often, they are players who try to uncover system bugs in order to either exploit it or improve it. They are similar to Bartle's Explorers.

3.3 Design framework classifications

The following section presents a review of the literature that passed the filtering criteria presented in Section 2.4.2. This section only reviews the classification papers. Even though some of the gamification design frameworks classified by these papers were reviewed, they are not presented in the literature review chapter.

The theories and processes presented in the design frameworks are not required for building a system that could recommend them. Hence, not all the referenced concepts were read and reviewed. The metadata provided by the classification papers is sufficient to build a recommender system.

Moreover, taking the design details of each gamification framework into consideration when building the system would have consequences on the engineering quality of the created tool. This decision would inhibit the creation of an easily extensible and adaptable system. It would not be possible to build a generic framework comparison algorithm as further described by the rationale behind requirement NFR2 in Section 4.4.

3.3.1 Mora et al. (2015)

This study presents the literature review of selected gamification design frameworks and their classification [5]. The work was published in 2015. However, it does not state the exact timeframe of the review. The study categorizes 17 gamification design frameworks and 1 taxonomy framework. It assesses them based on 10 out of the 19 identified features grouped into 5 main categories – summarized in Table 3.2.

TABLE 3.2: Feature categorization

Category	Feature	Description	Incl.?
Economic	Objectives	The specific performance goals.	✗
	Viability	A previous study, evaluation and analysis of the potential of applying gamification or refuse it.	✓
	Risk	A probability or threat of damage, injury, liability, loss, or any other negative occurrence.	✗
	ROI	The benefit to the investor resulting from running a gamified experience.	✗
	Stakeholders	A technique used to identify and keep in mind the people who have to interact with the design process.	✓
Logic	Loop	The game mechanics combined with reinforcement and feedback in order to engage the player in the key system actions.	✓
	End game	A pre-established end of game or glorious victory in the system, usually stretching players to the limits of their abilities.	✓
	On-boarding	The way of starting the new participants.	✓

Continuation of Table 3.2			
Category	Feature	Description	Incl.?
	Rules	The body of regulations prescribed by the designer.	✓
Measurement	Metrics	The standards of measurement by which efficiency, performance, progress, process or quality.	✓
	Analytics	The algorithms and data used to measure key performance indicators.	✗
Psychology	Fun	The enjoyment or playfulness.	✗
	Motivation	The behaviour which causes a person to want to repeat an action and vice-versa.	✗
	Social	The interaction between players.	✗
	Desired behaviors	The expected response of the players after the interaction.	✗
	Ethics	A branch of philosophy that involves systematizing, defending and recommending concepts of right and wrong conducts.	✓
Interaction	Narrative	The story and context created by designers.	✗
	UI/UX	Refers to everything designed into the gamified system which a player being may interact and the player's behaviors, attitudes, and emotions.	✓
	Technology	The use or need of a software component for development.	✓
End of Table			

The assessment of the frameworks assigns a value to each included feature. The possible values – as described in the authors' later work [8] – are the following:

Explicit (E) "the item is present in the framework description".

Implicit (I) "the item is not explicitly present in the framework description. It has been inferred by the authors, referring to other sources or clarified by means of contacting the authors".

Unreferenced (U) "the item is not present in the framework description".

The study also describes some of the reviewed frameworks in greater details discussing, for example, what other frameworks or theories are they based on. One of the observations is that most of the reviewed frameworks are based on so-called "Human-Focused Design principles". It also finds that *SDT* is a predominant motivational theory considered in the frameworks. Table 3.3 shows a compilation of dependencies on a per framework basis. The list has been extracted from the study by the author of this thesis.

TABLE 3.3: Framework dependencies

Design framework	Dependencies
Di Tomasso (2011) [16]	SDT (described in Section 3.2.1)
Werbach and Hunter (2012) [39]	[34]
Marczewski (2013) [40]	Hexad (Section 3.2.7), RAMP [41]
Marache-Francisco and Brangier [18]	UCD [37]
De Paz (2013) [31]	Werbach and Hunter (2012) [39]
Francisco-Aparacio et al. [22]	SDT (Section 3.2.1)
Versteeg (2013) [17]	Berdichevsky and Neuenschwander (1999) [42], Fogg (2002) [43]
Chou (2013) [44]	Human-Focused Design
Al Marshedi et al. (2015) [7]	Flow Theory [26] (Section 3.2.4), Pink (2011) [45], SDT (Section 3.2.1), UCD [37]
J. Kumar (2013) [19]	PCD
Gears (2013) [21]	Constanine and Lockwood (1999) [46], Reiss (2002) [47]
Jacobs (2013) [48]	Goal-Model Design
Julius and Salo (2013) [15]	Werbach and Hunter (2012) [39]
Li (2014) [49]	TAM (1989) [50], Delone and McLean (2003) [51]
N. Kumar (2013) [52]	∅
Jimenéz (2013) [53]	BMC, MDA [34]
Herzig (2014) [54]	Li (2014) [49], RUP
Robinson and Belotti (2013) [20]	∅

The study concluded that none of the analyzed frameworks is a "complete and generic framework" that would address all of the identified features. Hence, it considered a development of such a framework as future work.

3.3.2 Mora et al. (2017)

This framework classification paper [8] is the work of the same authors as the one discussed in the previous section. The timeframe of this paper is known and it spans until *01/10/2015*. The final number of analyzed frameworks was 27. The assessment followed the exact same methodology as the previous work. However, it considered more features. It included most of the previously identified (19) features – with some changes in the naming – and identified 2 more making the total number of included features *21* as opposed to the 10 features from the previous study. The full list of the included features compared to the previously identified features is shown in Table 3.4.

Another difference is that this time, the authors focused more on the applicability of the frameworks in higher education. Hence the introduction of the *domain* attribute. The classification also introduced two other attributes based on which the frameworks had been categorized: *Background* and *Target*.

TABLE 3.4: Feature comparison

Features (2015)	Features (2017)	Note
Objectives	Objectives	I

Viability	Feasability	A
Risk	Risk	I
ROI	Investment	A
Stakeholders	Stakeholders	U
Loop	Engagement Cycle	A
Endgame	Endgame	U
Onboarding	Onboarding	U
Rules	Rules	U
Metrics	Metrics	U
Analytics	Analytics	I
Ethics	Ethics	U
Fun	Fun	I
Motivation	Motivation	I
Social	Social	I
Desired behaviors	Desired behaviors	I
Narrative	Storytelling	A, I
UI/UX	User experience	A
Technology	Technology	U
	Profiling	E
	Taxonomy	E

I – Included
A – Altered
U – Unchanged
E – Extended

The study also provides a more detailed overview of the individual frameworks. It reports which definition of gamification is being used by the framework, provides a short description and also gives some more details about the nature of the framework in terms of its theoretical dependencies. Table 3.5 shows the dependencies extracted from the study grouped by framework. It also shows which design framework has been added to the study as opposed to the authors' previous work. The number of new frameworks in the study is 19 out of the 28 in total, which means that only half of the frameworks remained from the previous one.

TABLE 3.5: Framework dependencies and assumptions

Design framework	Dependencies	New?
Simões et al. (2012) [55]	– Social relevance design – No explicit reference to iterative design	✓
Nah et al. (2013) [56]	– No explicit reference to iterative design	✓

Continuation of Table 3.5		
Design framework	Dependencies	New?
Wongso et al. (2014) [57]	– Social relevance design – E-learning platform is a necessary environment – Explicitly stresses the relevance of an iterative process	✓
Mora et al. (2016) [58]	– MDA [34] (explicit) – Explicitly stresses the relevance of an iterative process	✓
Kortini and Tzelepi (2015) [59]	– No explicit reference to iterative design	✓
Klock and da Kunha (2015) [60]	– MDA [34] (explicit) – E-learning platform is a necessary environment – No explicit reference to iterative design – 6D [39] – Hexad (Section 3.2.7)	✓
Kappen and Nacke (2013) [61]	– MDA [34] (inspired) – No explicit reference to iterative design	✓
Marache-Francisco and Brangier (2013) [18]	∅	✗
Francisco-Aparacio et al. (2013) [22]	– Heavy reliance on psychological and motivational theories	✗
Marczewski (2013) [40]	– Hexad (Section 3.2.7)	✗

Continuation of Table 3.5

Design framework	Dependencies	New?
de Paz (2013) [31]	– UCD [37] (implicit) – 6D [39] – Technological relevance – Bartle's player types (Section 3.2.5)	✗
Chou (2015) [62]	– Heavy reliance on psychological and motivational theories – Bartle's player types (Section 3.2.5)	✓
Fitz-Walter (2015) [63]	– MDA [34] (inspired) – Sole focus on motivation and behavior change (neglects UX)	✓
Werbach and Hunter (2012) [39]	– Bartle's player types (Section 3.2.5)	✗
Kumar (2013) [19]	– UCD [37] – Technological relevance	✗
Gears and Braun (2013) [21]	– UCD [37] – Heavy reliance on psychological and motivational theories – Special emphasis on economic feasability – Only framework explicitly referring to "investment" – No explicit reference to iterative design	✗
Julius and Salo (2013) [15]	– 6D [39] – Bartle's player types (Section 3.2.5)	✗

Continuation of Table 3.5

Design framework	Dependencies	New?
Herzig (2014) [54]	– Technological relevance – Special emphasis on technological feasability	✗
Raftopoulos (2014) [64]	– UCD [37] – Technological relevance – No definition of gamification is explicitly included or cited	✓
Burke (2014) [4]	– UCD [37]	✓
Harms et al. (2014) [65]	– MDA [34] (inspired) – Technological relevance	✓
Neeli (2015) [66]	– MDA [34] (inspired) – Bartle's player types (Section 3.2.5) – Yee's player types (Section 3.2.6)	✓
Brito et al. (2015) [67]	– 6D [39] – Technological relevance – Focus on gamification in crowd-sourcing based systems	✓
Ruhi (2015) [68]	– MDA [34] (inspired) – Technological relevance	✓
Robson et al. (2015) [69]	– MDA [34] (inspired)	✓
Rojas et al. (2014) [70]	∅	✓
Charles and Mc-Donough (2015) [71]	– Hexad (Section 3.2.7) – Focus on gameful *rehabilitation* systems	✓

End of Table

3.3.3 Azouz and Lefdaoui (2018)

The study presented in this section is a systematic map and review of gamification design frameworks [6]. The timeframe of the study spans until the end of 2017. The difference between this study and the previous ones – as presented by the authors – can be summarized in the following 4 points:

1. *Objective*: the objective was to collect "all existing literature" on gamification design frameworks.

2. *Procedure*: it used an effective approach to literature reviews – the *Systematic Mapping*.

3. *Time period*: the period was "larger and newer".

4. *Multi-dimensional analysis*: The study assessed the frameowrks based on 5 attributes: (1) *period*, (2) *field of application*, (3) *lifecycle phase*, (4) *paper type* and (5) *patterns*.

The study found that the interest in concretizing the concept of gamification in conceptual frameworks has been steadily increasing since 2012. However, there are some aspects not very often covered by the frameworks. The coverage of these aspects is a recent development in the field. These aspects are: (1) *sustainability*, (2) *balancing* and (3) *ethics*. *Balancing* refers to the fine line between business objectives and user goals.

Additionally, the study identified a set of characteristics concluding the quality of a gamification design framework: (1) application of the UCD model, (2) business and player objectives and metrics, (3) business and player profiling, (4) balancing, (5) the presence of ethics, (6) meaningful orientation of gamification (i.e. added value in the intrinsic motivation) and (7) durability, a model based on constant improvement.

The application domains identified are: *Software engineering*, *Learning*, *Business*, *Health*, *Crowdsourcing*, *Education*, *Research activity*, *Social*, *Smart cities*, *Communication*, and *General*.

The type of research can have one of the following values: *opinion paper*, *solution proposal*, *validation research*, *philosophical paper* and *evaluation research*.

The product lifecycle phase may have one of the following values: *planification*, *design*, *integration and test & experimentation*, *requirements & analysis*, *implementation*.

Table 3.6 shows the patterns identified by this study in comparison to the features identified by Mora et al. (2017) presented in the previous section. Three of the identified patterns could not be matched to any of the features: (1) *sustainability*, (2) *balancing* and (3) *process/implementation*. One pattern could partly be matched to a feature.

TABLE 3.6: Identified patterns

Mora et al. (2017) [8]	Azouz and Lefdaoui [6]
N/A	game elements & mechanics
motivation	motivation
engagement cycle	engagement
unexpected	gamification meaningful
N/A	sustainability
objectives	goals
metrics	metrics

Continuation of Table 3.6

Mora et al. (2017)	Azouz and Lefdaoui (2018)
profiling	profiling
N/A	balancing
N/A	process (implementation)
N/A[2]	design model
N/A[3]	improvement
ethics[4]	security

End of Table

Unfortunately, the study only presents statistical results. E-mail communication was established with the authors of the study but they did not provide the requested details of their study. Hence, the results of the paper could not be applied in the design and implementation of the recommender system.

[2] In Mora et al. (2017) [8] all papers are required to be conceptual frameworks.

[3] All papers, except Gears and Braun (2013) [21], consider either explicitly or implicitly the use if iterative processes as a main design principle.

[4] partly matching

Chapter 4

Analysis

This chapter is logically divided into four parts. Section 4.1 presents the analysis of the collected empirical data through surveys. The combined duration of data collection spanned across several months, which means that the sections of this chapter are not necessarily chronologically ordered.

Secondly, Section 4.2 identifies how project requirements could be mapped to gamification design frameworks.

After the mapping is assessed, there still remains the question of whether it is possible to compare several frameworks in regards to the defined project requirements. An unambiguous process could be defined for determining which of two or more similar frameworks are more suitable for a specific set of requirements. Section 4.3 presents the findings in this area.

Once all the above are defined, Section 4.4 will proceed with extracting the system requirements for the recommender system that takes project requirements as its input and outputs one or more fitting gamification design frameworks.

4.1 Analysis of empirical data

The data collection produced multiple findings regarding the use of gamification design frameworks in practice, different ways project requirements could be translated to design framework recommendations and the means of designing and implementing a recommendation system for that purpose. This chapter presents these findings.

The first survey, presented in Section 4.1.1, found that the use of gamification design frameworks is more common than expected – 21 out of the 38 asked gamification practitioners reported that they had used a gamification design framework before. It has also been found that customizing a framework to suit it to specific needs is a fairly common practice – reported by two-thirds of the advanced practitioners. Several practitioners reported using a framework or a methodology developed by themselves when designing gamified experiences. The identified gamification practitioners were asked if they wanted to join an interest group for the purposes of this thesis. The conversion rate was higher than expected – a total of 27 respondents signed up, 2 of which provided an invalid e-mail address. In the end, only 10 of the signed up practitioners actively participated in further surveys. The relatively high rate of interest in the project and the common practice of customizing existing design frameworks for the specific needs of gamification projects indicate that the assumed problem of choosing the right gamification design framework for a particular project might also be an issue in practice. However, the analysis does not conclude whether the asked practitioners are aware of the assumed problem.

In the second survey, presented in Section 4.1.2, the created interest group was asked several questions regarding the requirements of their projects and their preferences regarding gamification design frameworks. Some practitioners require the frameworks to cover several high-level concepts such as objectives, target audience, motivation, and engagement cycle. Others expect the frameworks to include very specific theories and models such as the RAMP motivational theory [41], Hexad player taxonomy (described in Section 3.2.7) and 4K2F (4 keys to fun)[1] engagement model. Different practitioners prefer different frameworks. However, certain framework features are found to be less popular than others. When asked which features could be considered as project requirements, only 50% of the respondents answered "yes" regarding ethics and only one-third of the respondents considered "technology" as a relevant requirement.

In the last survey, presented in Section 4.1.3, the interest group was asked to complete a validation survey. They were asked how relevant they found the recommendations, how easy was it to use the system and the likelihood of its further use. The respondents mostly agreed that the results were relevant and the system was easy to use. Practitioners with 5 or fewer years of experience replied that they were likely to somewhat likely to further use the tool in the future. They also provided valuable feedback on how to improve the created tool.

4.1.1 Questionnaire 1: Testing and recruiting

The timeframe of the data collection for this questionnaire was between *15/03/2019* and *05/05/2019*. It received 45 responses. The raw data are available online in a spreadsheet[2] with view-only permissions. The responses are summarized for each segment separately – as described in Section 2.4.3.

Non-practitioners

Seven respondents have not used gamification in a business-oriented context and two of them were not interested in exploring gamification any further.

Casual practitioners

A *casual practitioner* is a person who may have used gamification but without utilizing a gamification design framework. 38 respondents said they had used gamification before in a business context and 17 out of them were identified as casual practitioners. 11 of the casual practitioners had not read nor heard about any specific gamification framework before. The rest of the practitioners were asked whether they were inclined to use a specific framework as well as their reason behind the choice. The answers are summarized in Table 4.1. One respondent would consider using different frameworks based on the project requirements. Two respondents would consider using the Octalysis framework because it suits their needs. One respondent would rather not use a framework because of the added overhead.

[1] http://xeodesign.com/xeodesign_whyweplaygames.pdf

[2] https://docs.google.com/spreadsheets/d/1RcbZLAEfvssnlf2L6v5N7p6uz57HNJqXXKwqTOrRGAM/edit?usp=sharing

TABLE 4.1: Frameworks being considered by casual practitioners

Framework	Rationale
It depends on the project	My approach is driven by users and their motivations. I always start with user research and design what is more engaging and motivating for them. I am trying not to stick to a framework but be flexible, because every project is very different.
I dont wanna, i makes things much more harder.	Too much hassle. I makes optimizations based on the revenue and current app usage.
Educational	I'm a teacher.
Octalysis framework	It perfectly embodies the way i have been designing my classroom game anyway, so it helps give more meaning
KOJAK, Octalysis	I've just heard about them, don't know them well.
Octalysis	It's all about motivation.

End of Table

Advanced practitioners

An *advanced practitioner* is someone who knows what a gamification design framework stands for and has experience utilizing one. 21 out of the 38 practitioners reported they had used a particular gamification framework. The adopted frameworks and the number of respondents adopting them is described in the following list:

- Self-developed framework (6)
- Octalysis (5) [44]
- A sort of role-playing game (2)
- Press Start To Begin by Scott Hebert[3] (2)
- Classcraft[4] (1)
- Mr. Matera's approach[5] (1)
- KOJAK[6] (1)

6 practitioners reported following their own framework or methodology and 2 reported to use methods based on RPG. The most frequently reported framework is the Octalysis framework.

[3] https://www.mrhebert.org/store/p14/TeacherGuide.html

[4] https://www.classcraft.com/gamification/

[5] https://www.mrmatera.com/

[6] https://kollektiva.eu/kojak-jatekositas-keretrendszer/

When asked about how satisfied the respondents are with the frameworks they currently use on a scale from 1 to 5, 9 practitioners answered “satisfied” (4), 9 practitioners answered “very satisfied” (5), and 3 practitioners answered “somewhat satisfied” (3). Two-thirds of the practitioners reported to customize and/or modify the design frameworks to fit their specific needs, 5 practitioners sometimes customize the frameworks, and 2 practitioners do not customize them. Surprisingly to the satisfaction, 9 practitioners reported they might consider moving to a different framework and 4 practitioners are already considering a change. 8 practitioners do not plan on moving to a different framework.

4.1.2 Questionnaire 2: Preferences and initial feedback

The second questionnaire was sent to the interest group who signed up to further participate in the research in the previous survey. The data collection for this survey started at the beginning of May and lasted for 4 weeks. 10 practitioners actively participated in this survey out of the 25 respondents who provided a valid e-mail address in the previous survey. The raw data are made available online in a spreadsheet[7] with view-only permissions.

As presented in Section 2.4.3, the questionnaire can be divided into three parts. The first part collected answers to the question regarding the framework features. The participants responded whether they consider the presented features as requirements on design frameworks. The precise wording of the question was: “Could the above feature be considered as a requirement for a design framework?”. Table 4.2 summarizes the distribution of each answer by features.

TABLE 4.2: Preferences of framework features

Feature	Yes	I don’t know	No
Objectives	100%	0%	0%
Feasability	40%	20%	40%
Risk	70%	10%	20%
Investment	90%	0%	10%
Stakeholders	90%	0%	10%
Engagement cycl	100%	0%	0%
Endgame	70%	20%	10%
Onboarding	90%	0%	10%
Rules	100%	0%	0%
Metrics	60%	10%	30%
Analytics	60%	10%	30%
Ethics	50%	20%	30%
Fun	80%	10%	10%
Motivation	90%	0%	10%
Social	80%	20%	0%
Desired behaviors	90%	0%	10%
Profiling	90%	0%	10%
Taxonomy	60%	30%	10%
Storytelling	80%	20%	0%
User experience	70%	10%	20%

[7] https://docs.google.com/spreadsheets/d/1UYzUJGnS4RTSHam0tVE4B9HCD81vYOAaupyXL2raYHo/edit?usp=sharing

Continuation of Table 4.2

Feature	Yes	I don't know	No
Technology	33.3%	22.2%	44.4%

End of Table

The participants were also asked to brainstorm regarding the requirements both before they were presented with the above list of features and after, as well. The reason behind it is to avoid the introduction of bias by presenting the list of features. The presented question in its exact wording was: "What concepts should a gamification framework tackle in your opinion? Think of a project and its requirements and list them in the answer box below.". The before and after responses are presented in Table 4.3.

TABLE 4.3: Framework feature brainstorm: before and after

Before	After
Behavioural, Motivation	its all covered
Collaboration and Cooperation	None
motivation for players; bonuses; improvements for players	not really
Not sure I understand correctly, but I think a framework should consider KPIs, personas (and player types), behavior design elements (including gamification), and a good process walkthrough.	Did I encounter any questions regarding incentives?
Objective, target audience, context, etc. Please refer to my Gamification Canvas for all the components.	I think you need to distinct framework, from concept from design document... It is all relevant and important, but watch out to put everything in one document... You will need a lot of sessions to come up with all the answers... Also: clients will not know the answer to all the questions when you start a new project... Hell I don't even know all the answers when I offer / quote a project ;-)
RAMP, HEXAD, 4K2F, SAPS -> For Rewards, Game Mechanics, Game Dynamics	∅
User retention	∅
motivation tools, reward (and punishment) system, milestones or levels, social contacts	∅

Continuation of Table 4.3

Before	After
Promote users that really use the platform. Make users create content. Make users log in everyday with small awards. Have a point system and badges.	∅
Engagement with the content, flexible understanding and critical thinking regarding the task at hand, an approach that can reapply the game's core to the real world.	This didn't seem to be covered, but a visual measure of progression that students are able to use to gauge themselves in the context of the game or system. Ideally something beyond simple points. Stats in an RPG, increasing strength, new armor or items, discovery of combos or other hidden mechanisms/skills, etc.

End of Table

The second part of the survey focused on the importance of feature comparison factors. The analysis presented in Section 4.3 identifies 3 framework comparison factors: (1) *feature-completeness*, (2) *domain specificity* and (3) *target specificity*. When the interest group was asked about their preferences, they mostly agreed that feature-completeness is an important factor – described by Figure 4.1. The answers regarding the domain and target specificity varied but the majority of the respondents assessed these factors as important as well – shown by Figures 4.2 and 4.3, respectively.

Each figure presents a histogram of the answers (1-5) representing the values ranging from "Not important" to "Very important". The histogram is overlaid with the normal distribution of the answers. Based on the figures, it can be concluded that all factors have been perceived as rather important.

These findings support the decision to design the system in such a way that it supports the customization of the weights regarding the importance of the framework comparison factors described in Section 5.1.2.

FIGURE 4.1: Feature-completeness

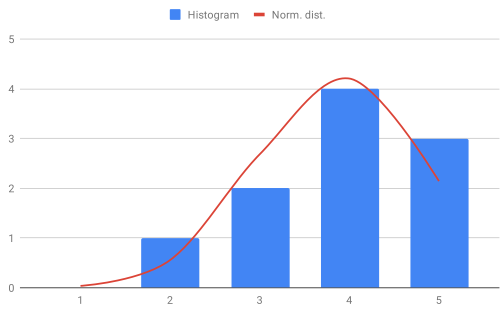
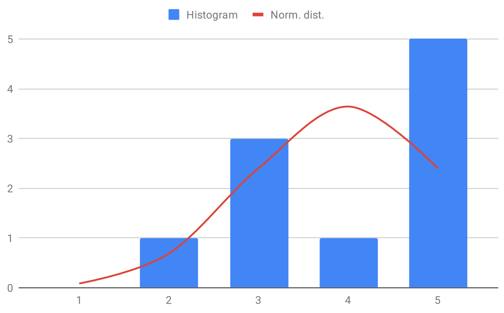
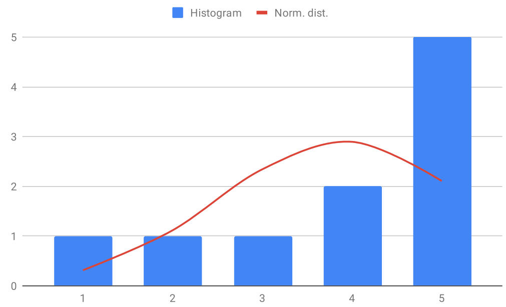


FIGURE 4.2: Domain specificity



To find out more about the preferences of the practitioners in the interest group, a series of questions were asked where the participants had to choose between two

FIGURE 4.3: Target specificity



specific types of frameworks. The results do not show a pattern and it can be concluded that the choice depends on personal preference:

- 60% of participants would choose a generic but feature-complete framework over a domain-specific and not feature-complete framework (Figure 4.4).
- 60% of participants would choose a feature-complete framework targeted at a general audience of practitioners over a skill-specific framework that is not feature-complete (Figure 4.5).
- 70% of participants would choose a generic framework matching their skill set over a domain-specific framework targeted at a general audience of practitioners assuming that both are feature-complete (Figure 4.6).

FIGURE 4.4: Generic and feature-complete framework vs. domain-specific and less feature-complete framework

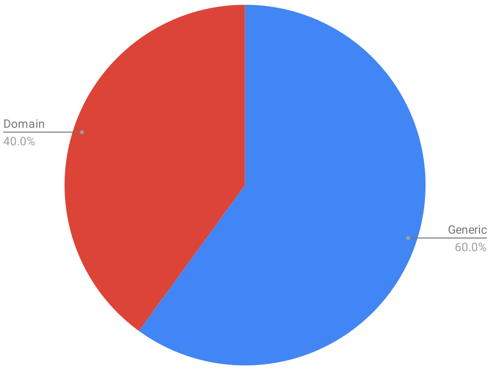
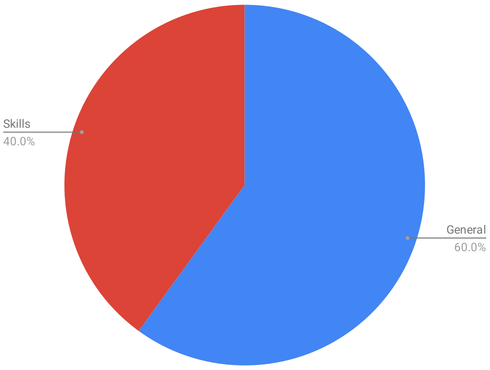


FIGURE 4.5: General and feature-complete framework vs. targeted and less feature-complete framework



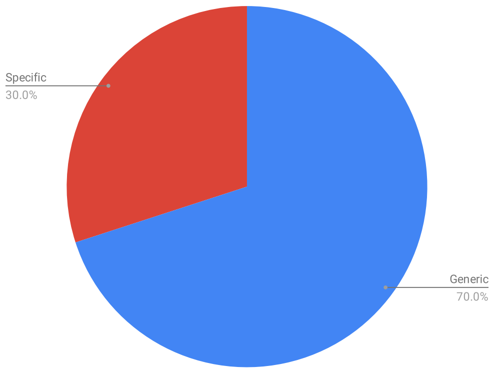
4.1.3 Questionnaire 3: Validation survey

The validation survey received 9 responses. The raw data along with the charts presented in this section are all available in Google Spreadsheets[8] with view-only permissions.

In this survey, the respondents were asked about the relevancy of the returned recommendations, the ease of use of the artifact as well as the likelihood of its further use. The final question of the survey asked the respondents to suggest future improvements to the tool.

[8] https://docs.google.com/spreadsheets/d/1uEtU782IzLhzuPUmyWORhJkmzuQY87sZ41Vv9BR22K4/edit?usp=sharing

FIGURE 4.6: Generic but targeted framework vs. domain-specific framework for general users

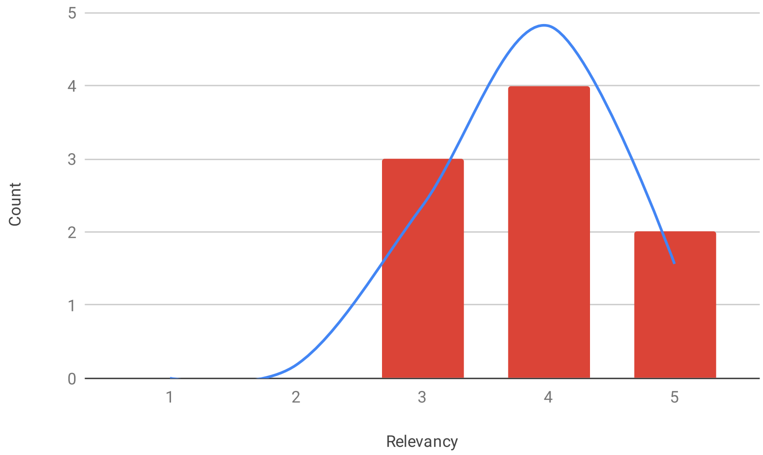


The results, due to the low number of responses, are not statistically significant enough to be generalized.

Relevancy of results

Figure 4.7 shows the histogram of relevancy overlaid with its computed normal distribution curve. The possible answers to the question "How relevant did you find the recommended frameworks in the results?" ranged from "Not at all relevant" (1) to "Completely relevant" (5). Based on the figure, it can be concluded that the respondents tended to find the recommendations relevant to very relevant.

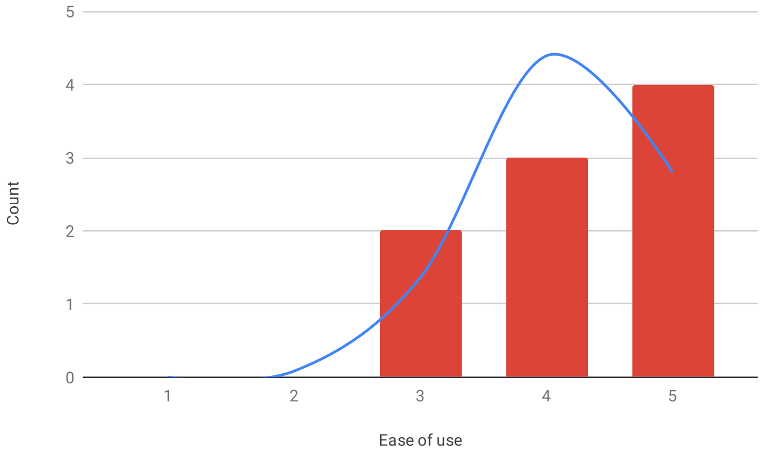
FIGURE 4.7: Relevancy of recommendations



Ease of use

Figure 4.8 depicts the histogram of ease of use overlaid with its computed normal distribution curve. The possible answers to the question "How easy was it to use the web application?" ranged from "Very difficult" (1) to "Very easy" (5). The respondents found the use of the web application to be easy to very easy – 4 out of 9 respondents answered 5, very easy.

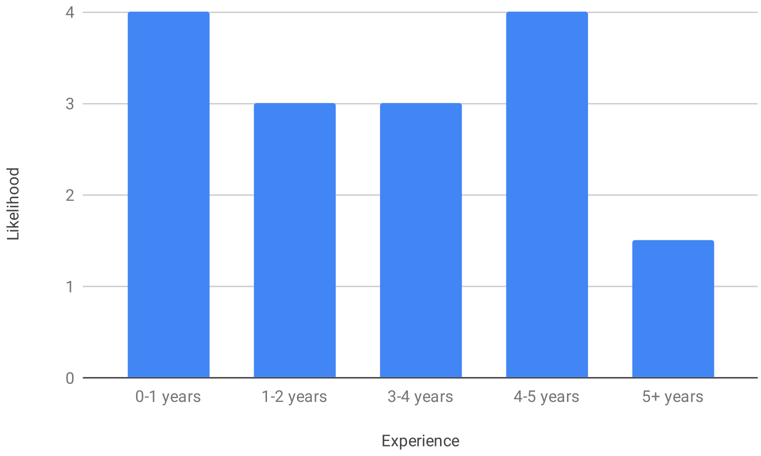
FIGURE 4.8: Recommender usability



Likelihood of future use

The possible answers to the question "How likely are you to use this application in the future?" ranged from "Not at all likely" (1) to "Completely likely" (5). The distribution of the answers was almost equal. In Chapter 1, it was assumed that the recommender system would probably prove to be more useful to gamification practitioners with less amount of experience. Hence, Figure 4.9 shows the likelihood of further use by the amount of experience of the respondents. The chart shows that the median likelihood of respondents with no more than 5 years of experience is 3 to 4, somewhat likely to likely. Respondents with 5 or more years of experience answered "not at all likely" (1) to "less likely" (2).

FIGURE 4.9: Likelihood of future use by experience



Suggestions for improvements

The suggestions related to the future improvement of the artifact are listed in Table 4.4.

S1 suggests that the recommender system might probably not be as useful for experienced practitioners favoring generic frameworks as learning new frameworks might be substantial overhead.

S2 provides several suggestions regarding the usability of the tool and how it could be improved. Furthermore, it points at the need to further distinguish frameworks based on their assumptions regarding the project. Frameworks in the domain of education might work with the assumption of an e-learning platform. This could be reflected in the recommendation engine.

S4 suggests that the tool might not be that easy to use for people, not that experienced in gamification. It says that the copies used to describe the feature are difficult to understand. It also points out that the match rate of the individual results is not emphasized enough.

S5 suggests to simplify the recommender system and *S6* would like to see more frameworks.

TABLE 4.4: Suggestions for future improvements

Id	Suggestion
S1	That's great that you collected so many methods and frameworks. Is it a validated problem to find the best suited framework each time I want to gamify? I tried 6 or 7 frameworks, and after those I assembled my own what I use in every project since then. I assume if I'm in a new project, use your system, and it suggests a totally new framework, it doesn't solve my problem: there will be a new method I need to learn to use properly. Is this occurs every time I get a new suggestion? (This is just boggling in my mind, because I really like general frameworks :)).
S2	I appreciate the format. As english is my second language, I would appreciate a little more description of all the features, maybe one two exemples. I had sometime troubles to see all the nuances between the different features. The choice of symbols (+ - nothing) are a bit confusing. at first, I though that "-" meant not important. So I changed my first two answers to, in my head "neutral" because I wanted to say "might come handy". I think we should be able to go back on the different pages. I could not find a place to return to the domain page. Even if I click Back, when I click on " getting started", it gives me the same thing I had. I canceled the domain to see if I could be able to chose something else and now I can't chose a domain anymore. The only way it works it to refresh the page, which reset everything. Most of the result seemed pretty good, except that I said technology is not important to me and the first 3 suggestions are for e-learning which is not exactly what I was looking for. The other results were more what I was looking for.
S3	As I've told you, reading the recommended framework results one question remained for me: which one is my style? But I couldn't really answer your questions, because I don't really know what information I'm expecting...

Continuation of Table 4.4

Id	Suggestion
S4	The tool is not clear in its purpose until the results are displayed. It is also unexpectedly long and the descriptions of each item are overwhelming. If someone needs a recommendation tool (as opposed to a person experienced with gamification), they may be unlikely to readily understand each concept presented to them. The percentage points are not immediately apparent, causing the results page to appear daunting, as though the tool is suggesting that I read every item on display.
S5	Simplify, but I know that's not what university students should do, they need / want to be thorough or complete. ;)
S6	Add my framework :(

End of Table

4.2 Mapping of project requirements to design frameworks

This section discusses how project requirements could be mapped onto gamification design frameworks. Project requirements, in general, can be defined in various ways. Two approaches can be immediately identified that could be taken in order to design a computer system for determining what framework to use based on the requirements.

One approach is to have a substantial amount of data about gamification design framework usage and the projects they have been used at. Also, it would require data regarding the success or failure and whether the project failed due to a poorly chosen design framework. Then, this data could be used as training data for a machine learning system. In lack of such data, this approach renders itself difficult to execute.

The second approach is to develop an algorithm based on the current knowledge in the field. In Section 3.3, the framework classification papers reveal substantial amount of knowledge about gamification design frameworks. This could be used to build an initial version of a system for determining the best fitting design frameworks. Mora et al. (2017) classify 27 different frameworks based on the presence of 21 different concepts like *motivation*, *desired behaviors*, *profiling*, etc. For simplicity, from now on it will be referred to these concepts as *"features"* or *"framework classification features"*.

The presence of a particular feature may directly be translated into a project requirement. In an enterprise context, designers might want to tackle features like *stakeholders*, *investment*, *feasability* and *risk*. In other application fields, like education, some of these concepts might not be the point of interest at all. Designers could pick a design framework based on their specific needs. In such case, an ideal design framework would be the one that holds all the required features without any additional overhead.

It is important to state that going forward with this approach, the system would not be able to tell which features should be used for a particular project. It would be the responsibility of the designers to know and define the required features for a project within the boundaries of the current gamification design framework classifications. The system could use this as an input and transform it to a set of recommended frameworks.

To overcome such limitations, the use of natural language processing (NLP) techniques would be required. That could identify the project requirements and translate them into the required framework classification features. This transformation could also be solved by machine learning algorithms. An example of such an algorithm could be a Support Vector Machine (SVM). However, this points to the aforementioned lack of data in this specific domain. The data could theoretically be also auto-generated and later on labelled by human experts. However, there are two issues with generating text resembling natural language. First of all, it is not a trivial task even though high quality algorithms already exist [72]. And second of all, generating un-biased and diverse set of training data to capture all possible types of projects that may occur in real life seems to be even less trivial if not impossible altogether.

Both of the discussed approaches display some disadvantages, as well. Since they are bound to the current knowledge in the field of gamification design framework classification, they may become outdated over time. Either by an updated classification or extended by more recent frameworks. Hence, an ongoing maintenance would be required. Substantially more maintenance in the second mentioned approach as the algorithm would need to be *"hard-coded"*.

Table 4.5 summarizes the attributes of the discussed approaches. This summary allows for their more apparent comparison.

TABLE 4.5: Comparison of potential solutions

Attribute	**ML Approach**	**Algorithmic Approach**
Usage data required	Yes	No
Domain knowledge required	Yes	Yes
Maintenance required	Yes	Yes (substantially more)
Ease of implmentation	Low	High

Both potential solutions require domain knowledge to a certain degree. The difference is in the requirement of usage data, the level of maintenance and ease of implementation. Since, the aim of the thesis is to create a proof of concept solution, it is reasonable to take the second approach for the design and implementation phase of the project. Once the solution is validated and the received feedback is evaluated, the first approach may be considered again if it proves to provide the desired benefits.

In summary, it has been decided that the current classification features of gamification design frameworks – taken from Mora et al. (2017) – may directly be used to map project requirements to the design frameworks with one assumption and one limitation. The assumption is that gamification designers or researches are able to determine what framework classification features they require from a design framework in context of their project. Hence, the solution would be limited in terms of recommending required classification features themselves.

4.3 Design framework comparison

The previous section concluded that the classification of gamification design frameworks created by Mora et al. can be directly used to map project requirements to the frameworks. However, a mapping of requirements is still not a sufficient solution in itself. Multiple frameworks might fit a particular project with full or partial coverage. In such a case, it is necessary to develop a process or mechanism through which those frameworks could be compared. Such a comparison would allow designers or researchers to pick the most suitable framework with less effort. This section describes three factors that could be taken into consideration for the comparison. The identified factors are as follows: (1) *feature completeness*, (2) *domain specificity* and (3) *target specificity*.

4.3.1 Feature completeness

Feature completeness is a metric that describes how many of the features required by a designer are part of a gamification design framework and to what extent. As described in Section 3.3.1, Mora et al. defined three possible values for their classification features: (1) E - explicit, (2) I - implicit and (3) U - unavailable. These values can be considered as a measure of the extent to which a design framework covers a particular feature. However, these values are not very clear in regards to the specification of project requirements. In order to provide more apparent means of specifying the extent of feature coverage, this metric could be translated into a concept that is easier to comprehend. The concept of importance fits particularly well to this purpose. A user of a gamification framework might be able to define how important a feature is to his/her project based on its requirements. The table 4.6 describes the mapping of feature coverage to its importance.

TABLE 4.6: Mapping feature coverage to the concept of importance

Feature coverage	Importance
Explicit	Very important
Implicit	Useful
Unavailable	Not important

Important features should be explicitly covered by the design frameworks. Features that are implicitly covered are still present in a framework but their importance is clearly lower. Otherwise, the feature would be explicitly covered. Features that are unavailable in a framework are most probably not considered important by its authors. Hence, they are unsuitable to users who do consider those features important.

A framework is considered *feature-complete* when it covers all the features required by the user to its exact extent. For instance, if the user considers *technology* an important feature, *motivation* as useful and *profiling* as not important, a framework that covers technology explicitly, motivation implicitly and profiling have the value of "unavailable" is considered feature-complete. A special case of feature-completeness is when all features are covered to their respective extent and no other feature is present in the framework. This case of feature-completeness is desired since the framework covers all features as requested without any additional overhead. Such a framework will be henceforth referred to as an *optimal framework*.

A *suboptimal framework* represents a state where the user requires one or more features to be explicitly covered by the framework but it only covers them implicitly.

Taking all the above into consideration, the comparison of the coverage of a single feature in two distinct frameworks may have the following values:

Match A framework matches in a feature to the user's requirements when the required extent of the feature equals to its coverage by the framework. For instance, a feature-complete framework matches the user's requirements in every feature.

Partial match A framework partially matches in a feature when the user requires it to be explicitly covered but the framework covers it only implicitly.

Mismatch If a framework does not fully or partially match in a feature to the user's requirements, it is a mismatch. For example, if a user does not consider a feature important, yet a framework has an explicit coverage for that particular feature, it is a mismatch.

4.3.2 Domain specificity

Gamification design frameworks can be created for a specific application area. Going forward, *application area*, *domain* and *business domain* will be used interchangeably describing the same concept. Domain specificity refers to whether the user's requirements meet the application area of the design framework. Mora et al. identify four different application areas: (1) generic, (2) business, (3) learning and (4) health.

Similarly to feature-completeness, the comparison of the required and provided domains can have the following three values:

Match A framework matches in its application area when the user's requirement equals to the application area of the framework.

Partial match A framework partially matches in its application area, if the required application area is specific (2, 3 or 4) but the application area of the framework is *generic* (1).

Mismatch If the framework does not fully or partially match the required application area, it is a mismatch.

4.3.3 Target specificity

Similarly to domains, a framework can be targeted as a specific type of user. The concept of the target should be understood as a particular skill set or knowledge a user possesses rather than a profession. Mora et al. identified five different target categories during their framework classification work: (1) general, (2) designer, (3) software developer, (4) researcher and (5) educator. Target specificity refers to whether the user's required target meets the target of the design framework.

The comparison of the required and provided domains is similar to the above and can have the following three values:

Match A framework matches in its target when the user's requirement equals to the target of the framework.

Partial match A framework partially matches in its target, if the required target is specific (2, 3, 4 or 5) but the target of the framework is *general* (1).

Mismatch If the framework does not fully or partially match the required target, it is a mismatch.

4.3.4 Importance of comparison factors

The three comparison factors described above can be used to evaluate whether a user's requirements match a particular framework and to what extent. Theoretically, it may still be possible to encounter several frameworks that match to the exact same extent but in different proportions. There might be frameworks that are feature-complete but created for a generic domain or for the general target. It is important to introduce a process or mechanism to determine which of the frameworks that match to the same extent are a better fit for the user.

To evaluate the importance of each aspect, the interest group has been asked several questions in a survey – presented in Section 2.4.3. For simpler reference in this section, the relevant questions have been shortened and given an ID:

SQ1. How important do you find feature-completeness? (on a scale of 1 to 5)

SQ2. How important do you find the matching application domain? (on a scale of 1 to 5)

SQ3. How important do you find a matching target? (on a scale of 1 to 5)

SQ4. Would you favor a generic but feature-complete framework or a specific one with 1-2 features not being covered?

SQ5. Would you favor a feature-complete framework written for the general audience or one that is specifically targeted at your particular skill set but missing 1-2 features?

SQ6. Would you rather favor a generic framework targeted at your skills or a specific framework written for the general audience considering both are feature-complete?

From the responses to these questions, there has been a general consensus that feature-completeness is important. A consensus was missing regarding the importance of domain and target specificity. However, most of the answers considered them as very important.

The answers to the questions SQ4 to SQ6 have also shown some variance. Two-thirds of the respondents would favor a generic and feature-complete framework over a domain specific and less complete framework. Also, two-thirds of the respondents would favor a general and feature-complete framework over target specific and less complete framework. Lastly, 70 percent of the respondents would favor a generic and target-specific framework over a domain-specific framework for the general target.

The respondents themselves come from different backgrounds with differing amount of experience in gamification. However, no pattern in the answers has been observed. Hence, it may be concluded that the preference regarding the importance of these comparison factors is completely personal. A generic solution might be to allow the users to configure their preference or filter the results based on it.

4.3.5 Merged frameworks

Two sub-optimal frameworks may be merged to create an optimal framework when they complement each other through their features in regards to the project requirements. However, two questions arise.

1. How could two complementing sub-optimal frameworks be merged?

2. What are the preconditions of merging two or more frameworks?

If there is no optimal framework based on the project requirements, a sub-optimal framework can be used as the basis for a merger. A second sub-optimal framework could be then found based on the mismatched and partially matched features of the base framework. If such a framework exists, the user could take the elements of the second framework and apply them together with the concepts and elements of the base framework, effectively creating a merged (and optimal) framework. This leads to the second question. There might be sub-optimal frameworks complementing each other that are incompatible with each other or more difficult to combine. Frameworks, in general, build on top of other theories. Table 3.3 in Section 3.3.1 lists the relationships of several frameworks to their underlying theories. Henceforth, these relationships will be referred to as *dependencies* or *framework dependencies*. Suppose the two frameworks are based on contradicting motivational theories. Misaligned dependencies might be the cause of incompatibility.

Another reason might be that a framework has embedded false assumptions about the project. For example, the design framework by Wongso et al. (2012) lists an e-learning platform as a necessary environment for gamification projects. Such a phenomenon will be referred to as a *requirement* or *framework requirement*. The framework from the example might be incompatible with other frameworks that do not meet its requirement.

Both mismatching dependencies and requirements might lead to incompatibilities and thus preventing frameworks from being merged. Such a mismatch between two frameworks suggests that they are different. Then, two similar frameworks might have a higher chance to be compatible with each other. To determine whether two frameworks are similar, a *similarity factor* needs to be constructed.

Similarity factor

Two frameworks are similar if they share similar dependencies and requirements. The similarity could be expressed in two ways.

1. With a single numerical value between 0 and 1.

2. With a vector.

Numerical value Similarity can be thought of as a ratio of shared attributes. For simplicity, two attributes such as the dependency and requirements will be considered. The similarity, in this case, can be then expressed as the average of the number of shared dependencies and shared requirements. Two edge cases can be identified.

1. The frameworks do not share any dependencies and requirements. In such a case, the value of similarity is 0 which means the compared frameworks are each other's opposite.

2. The frameworks share all dependencies and requirements. In this case, the value of similarity is 1 and means that the compared frameworks are identical within the analyzed scope.

Vector A more advanced representation of similarity could be achieved by using vectors. The numerical representation introduces a loss of information due to the application of an average function. On the other hand, a vectorial representation stores the similarity of each analyzed attribute in its own dimension. The length of the vector is then determined by the scope of the comparison, that is the number of framework attributes that are analyzed. These attributes may be authors, features, dependencies, requirements, application area and target. In this case, the similarity between the two frameworks is expressed by the Euclidean distance of their respective vectors.

4.4 System requirement extraction

It is crucial to distinguish between *project requirements*, *system requirements* and *framework requirements* to forego any possible confusion. Project requirements refer to the requirements specified by the user to retrieve a set of recommended design frameworks. System requirements, on the other hand, refer to the requirements set on the system that takes project requirements as an input and provides a set of recommended frameworks as its output. Framework requirements refer to the requirements posed by a specific gamification design framework or the assumptions it takes regarding gamification projects. For instance, some design frameworks were specifically designed for and they assume the use of e-learning platforms [57, 60]. This section describes the extraction of system requirements. The functional requirements are listed in Table 4.7 while the non-fuctional requirements in Table 4.8. Each requirement has a unique identifier and refers to the respective paragraph containing the rationale.

TABLE 4.7: Functional system requirements

ID	Reference	Description
FR1	Requirement specification	The user **MUST** be able to specify the project requirements.
FR2	Requirement modification	The user **SHOULD** be able to change the project requirements at any given moment in order to allow corrections in the input.
FR3	Detailed results	The user **MUST** be able to access detailed information about the match of the specified requirements to the shown results.
FR4	Framework access	The user **MUST** be granted a direct or indirect access to the frameworks presented in the results.

FR1: Requirement specification In order to recommend a gamification design framework, the user *must* be able to provide details about the framework being searched for. Hence, the users need to be able to input the requirements in a way that can be converted into a machine-readable format.

FR2: Requirement modification The user *must* also be able to modify the provided requirements. It allows making corrections in the input, for example, in case of a mistake.

FR3: Detailed results The user *must* see not only which framework is the most recommended one but also the rationale behind the choice. Showing only a match rate, e.g.: 85%, might raise the question of what does the remaining 15% stand for. Allowing the users to delve into the details of the recommendation results supports them in their decision making more than a single figure.

FR4: Framework access The user *must* be able to access the design frameworks presented in the result directly or indirectly. Direct access may be a URL pointing to the resource. Indirect access may be a URL pointing to a middleman who can provide direct access, e.g.: a web page of the publication provided its publisher.

TABLE 4.8: Non-functional system requirements

ID	Reference	Description
NFR1	Extensibility	The system **MUST** be effortlessly extensible in both dimensions: frameworks and features.
NFR2	Adaptability	The system **MUST** implement a generic framework comparison algorithm.
NFR3	Framework mergeability	The system **COULD** implement a merge recommendation feature to suggest frameworks to be combined with the selected framework for the purpose of creating an optimal framework.
NFR4	Iterability	The system **MUST** be architected for fast *development*, *testing* and *deployment*.
NFR5	Testability	The system **MUST** be designed with ease of testing in mind.
NFR6	Availability	The latency of the system **MUST** be approximately the same regardless of the geographic area of the user.
NFR7	Interoperability	The system **MUST** be implemented in a device-agnostic and operating system-agnostic manner.
NFR8	Usability	The results shown to the user **MUST NOT** cause information overload, effectively preventing him/her from taking a decision.

NFR1: Extensibility Section 3.3 described the three main classification frameworks of gamification design frameworks. However, the classification by Azouz and Lefdaoui (2018) [6] did not provide detailed results in terms of the classification itself as it could be seen, for instance, in the two papers by Mora et al. Azouz and Lefdaoui only provided high-level statistical data on different types of frameworks. Statistics alone are not sufficient to build a system that could recommend specific frameworks for specific projects. Not be tied to the work of a single group of researchers, the system could be built in an extensible way. Should any new design frameworks be classified or new classification features identified, the system *must* be effortlessly extensible in both dimensions.

NFR2: Adaptability In Section 4.3.5, it has been shown that there is a hypothetical probability of frameworks being incompatible with each other preventing them from being combined into a single optimal framework. Should the system support *extensibility*, the comparison of different frameworks cannot be based on attributes specific to any single framework. The comparison algorithm *must* be a generic implementation in order to be able to extend the system with currently unknown or unclassified gamification design frameworks.

NFR3: Framework mergeability The system *could* implement a feature that recommends one or more frameworks complementing the selected framework. This way, by merging the recommended framework(s) with the selected framework it might be possible to create an optimal framework. However, there are still many unknowns to merging different framework. Also, since the aim of the project is to create a proof of concept system that can test the assumption presented by the problem formulation in Chapter 1, this requirement has not been categorized as a *must*.

NFR4: Iterability Iterative development and prototyping require the system to be designed with flexibility in mind. This allows fast feedback-response cycles in development between the developer and the user. That, in turn, allows the validation of new ideas, features, and bugfixes in a shorter amount of time. The system *must* be architected for fast *development*, *testing* and *deployment*.

NFR5: Testability This requirement supports the requirement defined above. To achieve high iterability and maintainability of the project, it *must* utilize the testing best practices and it *should* ideally have a *100%* test coverage.

NFR6: Availability The members of the interest group described in Section 2.4.4 are geographically scattered around the globe. In order to receive unbiased feedback regarding the user experience, it is important to provide approximately the same latency. High network latency has been shown to negatively affect the user experience when browsing a website by several studies [73]. Thus, the latency of the system *must* be approximately the same regardless of the geographic area of the user.

NFR7: Interoperability In 2018, 52.2 percent of all website traffic worldwide was generated through mobile phones. Other devices such as tablets and desktops share the rest of the traffic. The user experience differs from device to device. In order to provide optimal user experience, the system *must* be implemented in a device-agnostic and operating-system-agnostic manner.

NFR8: Usability Several requirements can be inferred from the system usability perspective. The project requirements *should* be easy to define, yet specific enough for a computing system to comprehend them. The user *should* be able to change his/her project requirements at any given point in the process in case he or she introduces mistakes in the input. The user *must* be provided with enough information to be able to pick a framework from the set of recommended frameworks with confidence. On the other hand, the results shown to the user *must not* cause information overload, effectively preventing him/her from taking a decision.

Chapter 5

Implementation

The following chapter describes the architecture, design, and implementation of a gamification design framework recommender system while addressing the system requirements presented in Section 4.4.

5.1 Gamification design framework recommender engine

The following section describes the core of the solution and addresses the requirements NFR1-3 and NFR8.

NFR1: Extensibility The classification of frameworks based on the application area, target, and features covers 27 frameworks. More frameworks may be classified in the future and more features may be added as classification attributes.

The extensibility of the system can be achieved by separating the data from the logic. The data also needs to be structured. Some of the features and dependencies are shared between different frameworks. Hence, it is desirable to separate the definition of the frameworks from its features and dependencies. The features and dependencies could be then referenced from within the definition of the frameworks.

JSON has been chosen as the format in which the data will be stored. It is human readable, simple to extend and the draft JSON Reference Specification [74] allows to reference JSON objects from other files or web resources. Loose coupling is also achieved by separating the frameworks, features, and dependencies into their own JSON files. If a new framework or feature is introduced, they can be added in the respective files.

The solution is not concerned by the size of the files. There are only 27 frameworks and 21 features. In the future, as the amount of classified features grows, the data can be stored in a database. A graph database would be suitable for the purposes of the recommender system once the solution needs to be scaled due to a substantially increased amount of classified design frameworks. A framework may have several relationships: with other frameworks and its own dependencies. The dependencies have relationships with their own dependencies, etc. This way, the frameworks, and dependencies could be nodes in the graph database and the relationships between them the edges connecting the respective nodes. Graph databases such as Neo4j provide performant similarity algorithms[1] that could be utilized for calculating the similarity between the user-defined framework (shadow framework) and the classified frameworks.

[1] https://neo4j.com/docs/graph-algorithms/current/algorithms/similarity/

NFR2: Adaptability The research in the area of gamification design framework classification provides a lot of structured and unstructured data. In order to design a generic system, only data that is common to all classified frameworks may be used. The more exceptions are included in the comparison algorithm of the recommendation engine, the more complex the system grows. The recommendation criteria chosen for the engine are the following: (1) features, (2) application area, and (3) target. All classified frameworks include data points addressing all three criteria. In the future, the comparison algorithm can be extended by further criteria such as *framework requirements*, *theoretical dependencies*, etc.

NFR3: Framework mergeability The first iteration of the recommender engine will not recommend framework candidates for the purpose of merging two or more sub-optimal frameworks. However, the system needs to be designed to support the implementation of this feature at a later point. To achieve this, the data structure of the frameworks need to include a list of dependencies for each framework.

NFR8: Usability To provide enough context regarding the recommendation results to the user and thus facilitate the decision process, additional data points need to be presented along with the results. Most recommendation engines only provide a score or a match rate and expect the users to trust the algorithm. However, empowering the user by providing more details may have a positive effect on the decision making process. The `Comparison` object is designed in a way that allows detail extraction from the total score when comparing two frameworks. Multiple methods for detailed information retrieval have been implemented and are discussed further in detail in Section 5.1.2.

5.1.1 Data structures

It has been decided that the frameworks, features, and dependencies will be separately stored in their own JSON files. The structure of these entities is described along with the rationale for each data attribute.

Features

The `features.json` file should be a JSON object representing a collection of feature objects under their respective keys. It is expected that features will need to be looked up individually from the collection. Hence, an object representation is preferred over an array due to the performance gain for the lookup operation.

The object itself contains two attributes: *id* and *description*. The id uniquely identifies a single classification feature. The id should be generated from the name of the feature. Snake case formatting should be applied to ensure consistency. For example, the feature called *Desired behavior* should be converted to `desired_behavior`.

The description is used to describe the feature to the user. The name of the feature itself might not be enough to understand the purpose and meaning of the feature in the context of gamification design. For instance, the *Taxonomy* feature without its description might also refer to other concepts than the player typology.

```
// features.json
{
  "feature_id": {
    "id": "feature_id",
    "description": "Feature description"
  }
}
```

Dependencies

A dependency represents a theory or a publication like technical reports, article in a scientific journal or a book. In case of a theory, a publication in which the theory is described needs to be mapped as the dependency. All dependencies reside in a single JSON file – `dependencies.json` – as a collection of objects identically to features. However, the attributes of a dependency are different. It consists of the follwoing attributes: *id*, *title*, *authors*, *year*, *url* and *dependencies*.

The *id* is a string uniquely identifying a dependency. The id should be human-readable and derived from the title of the dependency. Deriving it from the title facilitates the reference to the dependency.

The *title* is a string representing the title of the theory or publication.

The *authors* attribute is a non-empty array of strings representing the names of the authors. The name of the author should follow the following format: it starts with the initials of all given names followed by the full last name. The format of the name is not important to the implementation. Its only reason is to introduce a convention for consistency reasons.

The *year* is an integer representing the year of publication.

The *url* is a string representing a URL that points directly to the publication or a website which provides more details on the publication. This attribute supports the requirement *FR4*.

The *dependencies* attribute is an array of dependencies representing the underlying theories used in the publication. For instance, the *HEXAD* player taxonomy, described in Section 3.2.7, is based on *Bartle's player typology* described in Section 3.2.5. Hence, HEXAD depends on Bartle's typology and it will be listed in its dependencies. The dependencies in this array are JSON pointers. The use of pointers allows the framework objects to refer to already defined dependencies and thus it saves storage space.

```
// dependencies.json
{
  "dependency_id": {
    "id": "dependency_id",
    "title": "title",
    "authors": [
      "author 1"
    ],
    "year": 1993,
    "url": "https://example.com/framework.pdf",
    "dependencies": [
      { "$ref": "#/other_dependency_id" }
    ]
  }
}
```

Frameworks

The `frameworks.json` file is a collection of frameworks following the previously set conventions. The file is an object containing the frameworks as keyed values. A framework, similarly to a dependency, represents a publication of a gamification design framework. It shares all the attributes of a dependency. The additional attributes are: *application_area*, *target*, *features* and optionally the *display_name* attribute.

The *id* attribute uniquely identifies the given framework. As a convention, the id should be generated from the last name of the first author of the *authors* array and the year of publication in a snake case format.

The *application_area* is an array of strings representing the different application areas the design framework has been designed for.

The *target* is an array of strings representing the different targets of the design framework.

The *features* attribute is an array of feature objects extended by a *value* attribute. The value attribute is a string and may have one of the following values: (1) `E` for explicit, (2) `I` for implicit, and (3) `U` for unavailable. These possible values are exactly the ones defined by Mora et al. (2015) [5].

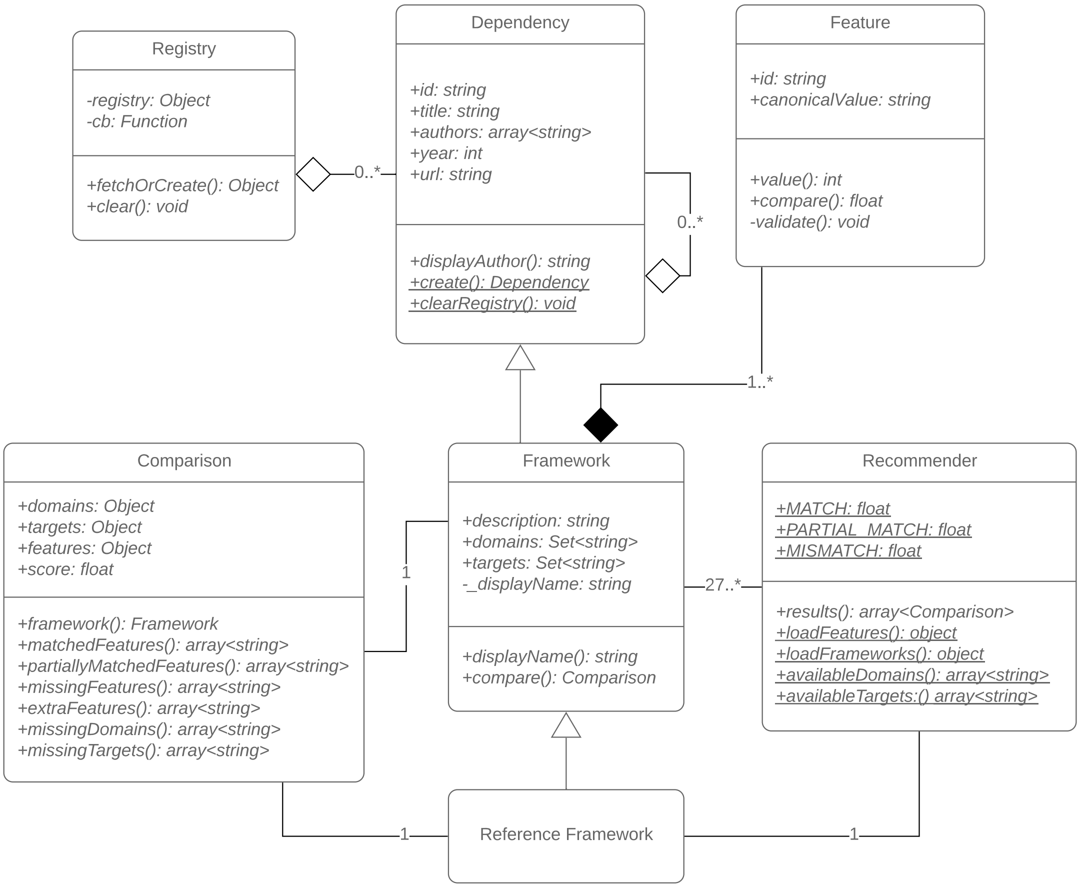
The *display_name* attribute is a string and is optional. Some frameworks are designed as part of a larger publication. These frameworks have a well-known name which usually does not match with the title of the publication. For instance, the well-known framework called *Octalysis* is part of the publication titled as *Actionable Gamification: Beyond Points, Badges, and Leaderboards* [44].

```
// frameworks.json
{
  "framework_id": {
    "id": "framework_id",
    "title": "title",
    "authors": [
      "author 1"
    ],
    "year": 2013,
    "url": "https://example.com/framework.pdf",
    "description": "description",
    "application_area": "domain",
    "target": [ "target" ],
    "dependencies": [],
    "features": [
      { "$ref": "features.json#/id", "value": "E" }
    ]
  }
}
```

5.1.2 Conceptual model

The data structures discussed in the previous section revealed several relations between the abstractions over features, dependencies, and frameworks. These can be used to design the conceptual model of the recommender engine presented in this section. Figure 5.1 shows the UML class diagram of the conceptual model. This section describes the classes of the system, their relationships, attributes and operations using a bottom-up approach.

FIGURE 5.1: Recommender engine class diagram



Feature

The *Feature* class represents a classification feature. The recommender engine uses the feature object of this class to compare two distinct frameworks. Hence, it is constructed from the feature object of a framework instead of the standalone feature object discussed in Section 5.1.1.

The `canonicalValue` attribute holds the value of the `value` attribute of the origin object. The class provides the `value()` object method to numericize the `canonicalValue`. A numerical representation of the value attribute is favored to achieve a simpler comparison algorithm. Table 5.1 shows the encoding of the respective canonical values as numbers.

TABLE 5.1: Encoding of canonical values as numbers

Canonical value	Numerical value
E	1
I	0.5
U	0

The other public method provided by this class is the compare() method. This method takes another feature object as the argument and compares the values of the two objects. Its return value is one of the three comparison values identified and presented in Section 4.3.1. These values are also encoded as numbers, as shown in Table 5.2.

TABLE 5.2: Encoding of comparison values as numbers

Comparison value	Numerical value
MATCH	1
PARTIAL MATCH	0.5
MISMATCH	0

The private method called validate() validates whether the canonical value holds one of the permitted values. Its main function is to guard against implementation errors that might occur in the future by raising a runtime error.

Dependency

The Dependency class is the respective class of the dependency data object discussed in the previous section. It is constructed from the dependencies of the framework object. It publicly exposes all the data attributes of the dependency object discussed in Section 5.1.1.

The displayAuthor() instance method returns a string represeantation of the authors array. This method is intended to be used outside of the recommender engine. Since the authors array is also publicly available, the system using the recommender engine is not tied to the representation returned by this method.

The class also exposes two static methods: create() and clearRegistry(). The constructor of the Dependency class is not intended to be used directly. To instantiate an object of this class, the static method create() will be used instead. These methods will be discussed in more detail in the section describing the Registry class.

The Dependency class has a zero-to-many relation with itself to represent that different theories and research papers build on top of each other. A dependency may have zero dependencies when a mapping of dependencies is missing.

Framework

Framework is a subclass of the Dependency class. It inherits its attributes and methods. The Framework class defines four addition attributes and two public methods.

The description attribute is a string holding the value of the description data attribute.

The domains attribute is a Set constructed from the application_area data attribute.

The targets attribute is a Set constructed from the target data attribute.

The _displayName private attribute holds the value of the display_name optional data attribute. The displayName() public method returns the value of this attribute if it is defined. Otherwise, it returns the title. However, the title attribute is publicly exposed, so the system using the recommender engine is not tied to the implementation of this method.

The compare method accepts another framework as its argument, compares the two and returns a Comparison object.

Reference Framework

The `ReferenceFramework` class is a subclass of the `Framework` class. It represents an imaginary gamification design framework, henceforth referred to as the *reference framework* or *shadow framework*. It is constructed from the project requirements defined by the user. Hence, it supports the requirement *FR1*. The recommender engine uses the shadow framework as its basis for the comparison across all frameworks. The class does not extend its superclass. It only overrides the declaration of the `id` attribute and sets it to a fixed value: `reference-framework`.

Comparison

An instance of a `Comparison` class is constructed using the shadow framework, a real framework, and weights.

The weights object contains a weight for each comparison factor described in Section 4.3.4. By default, the weights are balanced not to introduce any bias regarding the importance of the three factors, namely: (1) feature-completeness, (2) domain specificity, and (3) target specificity. By default, 21 features are included in the classification data. Table 5.3 shows the default weight distribution. Each feature has a weight of 1 and the weight of the other factors has the value of the number of features.

TABLE 5.3: Default weights

Comparison factor	**Weight**
Feature-completeness	1
Domain specificity	21
Target specificity	21

During the construction of the object, the `score` is calculated along other internal attributes such as: `domains`, `targets` and `features`. These are the respective scores for each comparison factor. Their value is used to calculate the value of the final `score`.

The main `score` is a float between 0 and 1 representing the percentual match rate of the two compared frameworks. The comparison algorithm is discussed in further details in Section 5.1.3.

To address the requirement NFR8 and provide enough context to the user about the recommendations, the following public methods are implemented:

framework() returns the object of the real framework that has been compared to the shadow framework.

matchedFeatures() returns an array of matching features between the compared frameworks.

partiallyMatchedFeatures() returns an array of partially matching features between the compared frameworks.

missingFeatures() returns an array of mismatched features between the compared frameworks.

extraFeatures() returns an array of features that are not required by the shadow framework but are explicitly covered by the real framework.

missingDomains() returns an array of domains that are required by the shadow framework but not covered by the real framework.

missingTargets() returns an array of targets that are required by the shadow framework but not covered by the real framework.

Registry

The `Registry` is responsible for holding references to the `Framework` and `Dependency` objects in order to prevent circular dependencies from being formed. Each framework dependency object is instantiated only once. Both classes have their own registry in order to separate concerns. The `Dependency` class implements a static method called `create` which calls the `fetchOrCreate` instance method of this class. The constructor of `Registry` accepts a callback function as its argument for the object creation process.

The `registry` attribute is private and stores the references to these objects as key-value pairs. The key is the `id` of the object and the value is the object itself. Initially, the `registry` is an empty object.

The `fetchOrCreate()` method accepts a framework or dependency data object – discussed in Section 5.1.1 – and first it looks into the `registry` whether the corresponding class instance had already been registered. If the object is not found in the `registry`, the method will use the callback function passed to its constructor to instantiate the corresponding class instance and store it in the `registry`. If the object was found, it will return the registered instance.

The `clear()` method resets the `registry` to a clean state. The only purpose of this method is to clean up after a unit test that has been run.

Recommender

The `Recommender` class is the main class of the recommender engine. It is instantiated using the shadow framework and an array of other frameworks.

The `result()` public method compares all frameworks to the shadow framework and returns an array `Comparison` objects sorted by their `score` attribute in descending order.

The class also exposes the three comparison values in for of static attributes: `MATCH`, `PARTIAL_MATCH` and `MISMATCH` for purely semantic reasons.

The class also implements additional four static methods: (1) `loadFeatures`, (2) `loadFrameworks`, (3) `availableDomains` and (4) `availableTargets`. The load methods return an array of data objects loaded from their respective JSON files as described in Section 5.1.1. The availability methods, on the other hand, return an array of possible values for *domains* and *targets*, respectively – as defined in Section 4.3.2 and Section 4.3.3.

5.1.3 Comparison algorithm

The comparison algorithm compares the user-generated shadow framework to a single framework. The comparison itself can be divided into four distinct steps:

1. Domain comparison
2. Target comparison
3. Feature comparison

4. Score calculation

Domain and target comparison

The user can define multiple acceptable domains as well as targets for the shadow framework. The comparison algorithm loops through the "shadow domains" and evaluates each in regards to the domains of the other framework. If the domain is present, it marks the domain as a `MATCH`. Otherwise, it marks it as a `MISMATCH`. In the end, the function returns an object of key-value pairs that effectively maps the shadow domains to the comparison values.

For a more explicit description of the algorithm see the pseudo-code below.

```
const compareDomains = (shadow, other) => {
  return shadow.domains.reduce(
    (acc, k) => Object.assign(
      acc,
      {
        [k]: other.domains.has(k)
          ? MATCH
          : MISMATCH,
      },
    ),
    {},
  );
};
```

Feature comparison

The shadow framework contains the features as defined by the user based on his/her project requirements. These features are then compared to the respective features of the other framework. The comparison is evaluated based on the rules presented in Table 5.4.

TABLE 5.4: Feature comparison rules

Shadow feature	Other feature	Comparison value
E	E	MATCH
E	I	MISMATCH
E	U	MISMATCH
I	E	PARTIAL_MATCH
I	I	MATCH
I	U	MISMATCH
U	E	MISMATCH
U	I	MISMATCH
U	U	MATCH

The comparison function then returns an object of key-value pairs which is a mapping of features to comparison values as presented in the pseudo-code below.

```
const compareFeatures = (shadow, other) =>
```

```
  Object.keys(shadow.features).reduce(
    (acc, id) => Object.assign(
      acc,
      { [id]: shadow.features[id].compare(other.features[id]) },
    ),
    {},
  );
```

Score calculation

The scores are calculated for each comparison factor, respectively, as described further on in detail.

Domain and target score The scoring formula takes the comparison object as its input as well as a partial match condition. If any of the respective attributes (domains or targets) matched, the score returns `MATCH`. However, if the framework does not include the respective user-defined attributes, the algorithm checks whether the partial match condition has been met. If it evaluates as true, `PARTIAL_MATCH` is returned. Otherwise, the function returns `MISMATCH`.

The partial condition for the domain comparison is that the application area of the framework includes "Generic". The partial condition for the target comparison is that the targets of the framework include "General".

The pseudo-code below shows a mock implementation of the domain scoring function as an example:

```
const scoreDomains = (domainComparisons, partialConditionMet) => {
  if(Object.values(domainComparisons).has(MATCH)) {
    return MATCH;
  }

  return partialConditionMet()
    ? PARTIAL_MATCH
    : MISMATCH;
};
```

Feature score The scoring function takes the feature comparison object as its input and returns the sum of individual feature scores, as shown in the pseudo-code.

```
const scoreFeatures = (featureComparisons) => {
  return sum(Object.values(featureComparisons));
};
```

Final score The final score represents a percentual match rate of a framework to the shadow framework. Its value must be between 0 and 1. First, the potential score is calculated. It holds the maximum potential score that could be achieved if all comparison factors return a complete match, that includes: matching application domain, targets and all features.

As discussed in Section 4.3.4, the comparison factors might have different importance based on the preferences of the individual user. To support the configuration of the importance of the respective comparison factors and the requirement *NFR2* for a generic solution, weights are introduced. Each factor has its own weight and a weighted score is calculated for each of them. The sum of these weighted scores is then divided by the potential score. The value of this division is always between 0 and 1 and represents the match rate of the framework to the shadow framework. A sensible set of default weights are introduced in Section 5.1.2 for convenience and ease of configuration on behalf of the user.

The pseudo-code below demonstrates the total score function.

```
const calculateScore = (comparisons, weights) => {
  const { domains, targets, features } = comparisons;
  const weightedDomainScore = weights.domains * domains.score;
  const weightedTargetScore = weights.targets * targets.score;
  const weightedFeatureScore = weights.features * features.score;


  const featureCnt = Object.keys(features).length;
  const potentialScore = weights.domains + weights.targets + weights.features * featureCnt;

  return (weightedDomainScore + weightedTargetScore + weightedFeatureScore) / potentialScore;
};
```

5.2 System architecture

The previous section described the core technology of the solution – the recommender engine. The focal point of this section is the system-wide architecture of the solution. It addresses the requirements NFR4-6, separately, in regards to *development*, *testing* and *deployment*.

5.2.1 Development

The requirements NFR4 and NFR7 directly affect the architectural and development phase of the solution. The solution can take many different forms. It could be a desktop application, a mobile application or a web application. The shape of this solution can be derived from requirement NFR7.

NFR7: Interoperability This requirement states that the solution needs to be device- and operating system-agnostic. Only web applications can achieve this from the abovementioned solutions. The other solutions would require the system to be implemented for different platforms independently. Devices – be it phones, tablets, smart TVs or desktops – may utilize different operating systems but each operating system is very likely to provide a web browser. Hence, a web application is able to satisfy the requirement.

NFR4: Iterability This requirement states that the system must be built with flexibility and maintainability in mind to support frequent code changes for prototyping. Several decisions can be drawn from this requirement.

Web technologies The system takes the form of a front-end single page application (SPA). The business logic of the recommender system is substantially simple to execute inside of a browser. The solution does not require network or database calls since the data are stored in JSON files. The prototype does not require any advanced functionality such as user authentication and authorization which would require sessions. All application state can be persisted inside the browser if needed.

The entire system can be bundled into static assets and a server can be used only to host those assets without any additional business logic placed on it. Taking these facts into consideration, the development of a SPA allows for faster iterations.

Frameworks, tools, and libraries The system is designed to be a SPA. Hence, the predominant language to be used for development is *JavaScript*. There are several popular and easy to use frameworks, tools, and libraries that can further improve the iterability of the system.

ReactJS is JavaScript library for building user interfaces. Other similarly popular libraries are *AngularJS* and *Vue.js*. The popularity of a library is important because it directly correlates to the extent of maintenance of the library. However, a stable and long supported library is not a requirement for a proof of concept prototype. Taking the author's experience into consideration, the chosen technology is ReactJS. Experience with a certain library or tool substantially increases the pace of iterations.

For similar reasons, *Redux* has been chosen as the state management library for the prototype. *React Router* has been chosen as the routing library. Currently, it is a "de facto" standard and is also compatible with Redux. A router allows creating the illusion of having multiple pages in a single page application. The tool synchronizes the UI of the web app with the URL presented in a browser. Hence, the developer has complete control over what content is being displayed under which URL.

In order to further increase the pace of iterations, a React component library has been chosen. Google's Material UI is a React component library that implements Google's *Material Design*. It provides a multitude of pre-made and highly customizable UI components. The use of these components is discussed in Section 5.3.

The use of React, however, puts a constraint on the project. It was designed with modularity in mind, whereas JavaScript did not. In order to have small hierarchical JavaScript components, the project will be required to use a bundler. A bundler is capable of traversing all JavaScript files through `require` or `import` statements given one or more entry points. This bundle is a single javascript file that can be included through a `<script />` tag in HTML code. One of the most popular and versatile bundlers for JavaScript projects is *Webpack*. Some of the other reasons for choosing Webpack are also mentioned in the next paragraph.

Project bootstrapping All the different libraries chosen to be used for the prototype implies a lot of configuration which could slow down the development in its initial phase. However, this initial configuration could be solved by bootstrapping a simple SPA skeleton for this project using the `create-react-app` tool developed by Facebook.

This tool creates and configures a project automatically to meet the latest development setup standards for single page applications. It sets up tools like Webpack and Babel for the developer providing the latest tools and technologies to be used since the very beginning of a project.

Babel allows the use of the latest JavaScript syntax and functions such as ES6, ES7, etc. `Create-react-app` configures Babel specifically for ReactJS projects for the developer. Thus, this tool eliminates the bottleneck caused by the previous choices of frameworks, tools, and libraries.

5.2.2 Testing

Writing unit tests is part of the TDD approach. Different libraries can be chosen for a single page application for this purpose. *AVA* has been chosen as the preferred test runner. This choice is justified by the requirement NFR5. AVA is a test runner for Node.js that embraces process isolation. It allows for more effective test writing and execution due to the possibility to run tests concurrently. Thus, reducing the development time in iterations even further.

Concurrent test execution and ease of configuration is the advantage over other test runners and frameworks such as: Jasmine[2], Karma[3] or Mocha[4].

5.2.3 Deployment

The requirement NFR6 – *uniform latency* – can be addressed by utilizing Content Delivery Networks (CDNs). However, before dealing with this topic, it is necessary to discuss the deployment of the project first.

The code of the project is hosted on GitHub[5] as an open source repository. It allows other researchers and test users to open issues or contribute to the project in the future. GitHub repositories can also host entire web apps in the form of GitHub Pages[6]. The benefit of hosting the project on GitHub is that the code is already hosted there and several tools exist that can deploy the projects to GitHub Pages with just a single command through a command-line interface (CLI). Utilizing these tools further shorten the iteration time – requirement NFR4.

To bring the content closer to the end-user and provide approximately the same latency regardless of the geographical area the user tries to access the website, a CDN service will be utilized. For this purpose, Cloudflare has been chosen as it provides a free tier CDN service. Apart from that, it also provides DNS, DDoS protection and an SSL certificate free of charge.

With a relatively small amount of configuration, it is possible to host the SPA on GitHub Pages behind Cloudflare as the CDN under a custom domain with an SSL certificate. The custom domain and SSL certificate adds credibility to the project and thus may earn the trust of the end-users in its validation phase. Additionally, the SSL certificate also provides SEO benefits [75]. For this purpose, the `howtogamify.tech` domain has been purchased. The domain has been pointed to the nameservers of

[2] https://github.com/jasmine/jasmine

[3] https://github.com/karma-runner/karma

[4] https://github.com/mochajs/mocha

[5] https://github.com/vecerek/how-to-gamify/

[6] https://pages.github.com/

Cloudflare where the A records were set to point to GitHub's designated IP addresses[7]. GitHub also requires the project to place a `CNAME` file in the root of the project with the custom domain as its content. The rationale behind placing that file is that GitHub can verify the origin of that domain.

These steps, however, are not complete for the solution. Due to the specific use of React Router in the implementation of this project, the routing logic is handled client-side but the server attempts to evaluate the URL before the client-side code could be loaded. According to the *create-react-app* documentation, GitHub Pages does not support routers that use the HTML5 `pushState` history API[8].

To overcome this limitation, a simple URL mapping solution created by Rafael Pedicini[9] will be utilized. A `404.html` page needs to be implemented with a piece of JavaScript code that transforms parts of the requested URL into a query param. Then, it redirects to the index page with the query param attached. The web app – with the client-side rendering logic being loaded – then parses the query param and handles the routing. This solution is semantically not correct since it returns a 404 HTTP status code for a page that does exist. However, better user experience can be achieved at the cost of semantic incorrectness for the purposes of this prototype.

5.3 User experience

This section describes the principles that were guiding the design of the user experience as well as the UI components that were chosen from the Material UI Component library for building the user interface.

Specifying project requirements in the context of gamification might be difficult for the users depending on the amount of experience they have in designing gamification solutions. For instance, the classification features of gamification design frameworks proposed by Mora et al. (2017) [8] might not be well-known in the gamification design community – especially among the less experienced designers. There are 21 different classification features, four different application domains and four different targets to choose from. The users at this point may not be fully aware of their preferences as well as the range of alternatives at their disposal. Also, letting the users configure all these items without guidance might have an overwhelming effect on them. However, the specification of these features and the rest of the attributes is essential to the implemented system. A more natural way of identifying the users' needs is by asking what is the project like and what aspects of the project are important to them. Hence, the implementation of a conversational recommender system is desired in this case.

A conversational recommender system supports an interactive process of querying and providing information between the user and the system. However, it is of critical importance how the dialogue is designed, what actions can the user and the system perform and what are the various stages of the interaction [76, p. 20].

The two main stages of interaction are: (1) *Requirement collection* and (2) *Presentation of results*. The stages of interaction are described in the following sections along with the actions a user or the system can take as well as the UI components chosen from the Material UI React component library.

[7] `https://help.github.com/en/articles/setting-up-an-apex-domain#configuring-a-records-with-your-dns-provider`

[8] `https://facebook.github.io/create-react-app/docs/deployment#notes-on-client-side-routing`

[9] `https://github.com/rafrex/spa-github-pages`

The implemented recommender system can be accessed through its website[10] and the reader is encouraged to try it out as he/she reads the following sections.

5.3.1 Requirement collection

The requirement collection stage fulfills the requirement *FR1*. It is depicted by the wireframe presented in Figure 5.2. This stage can be divided into three distinct steps:

1. Feature requirements
2. Domain requirements
3. Target requirements

Feature requirements

Through the collection of the above requirements, the user builds the *shadow framework* that is used by the recommender engine for evaluation and recommendation. These steps can be visualized using the `Stepper` component[11] of the Material UI library. *Steppers* visualize progress through a sequence of logical steps while giving transient feedback after each completed step.

As the first step, the user is asked about the project in terms of its requirements. The user is presented with 21 questions – one question per classification feature:

> *"How important do you find FEATURE NAME for your project?"*

The feature is shortly described under the question to give a broader context to the user. The user then chooses one answer out of the three presented options: (1) *Not important*, (2) *Might be useful*, and (3) *Very important*. The options are visualized using *Button* components[12]. The provided answer can be directly mapped to the value of the feature coverage as presented in the previous chapter in Table 4.6.

A progress indicator is also presented to visualize how many questions are left to be answered. Visualizing progress is important in order to keep the user motivated.

A sidebar is shown to the right of the *Stepper* where all the defined requirements are displayed and can be modified at any point during this stage of the interaction. The possibility to perform an amendment to the selected requirements in the sidebar directly addresses the requirement *FR2* defined in Section 4.2.

The selected feature coverage is visualized using a *Checkbox* component[13] as it is able to represent three states: (1) unchecked, (2) indeterminate, and (3) checked. This part of the UI can be seen in Figure B.3. These states map to the coverage values in the respectful order as follows: (1) Unavailable, (2) Implicit and (3) Explicit.

Domain and target requirements

In the second step of this stage, the user is asked to choose the domain specificity of the project:

> *"Which of the following domains are the closest to your project?"*

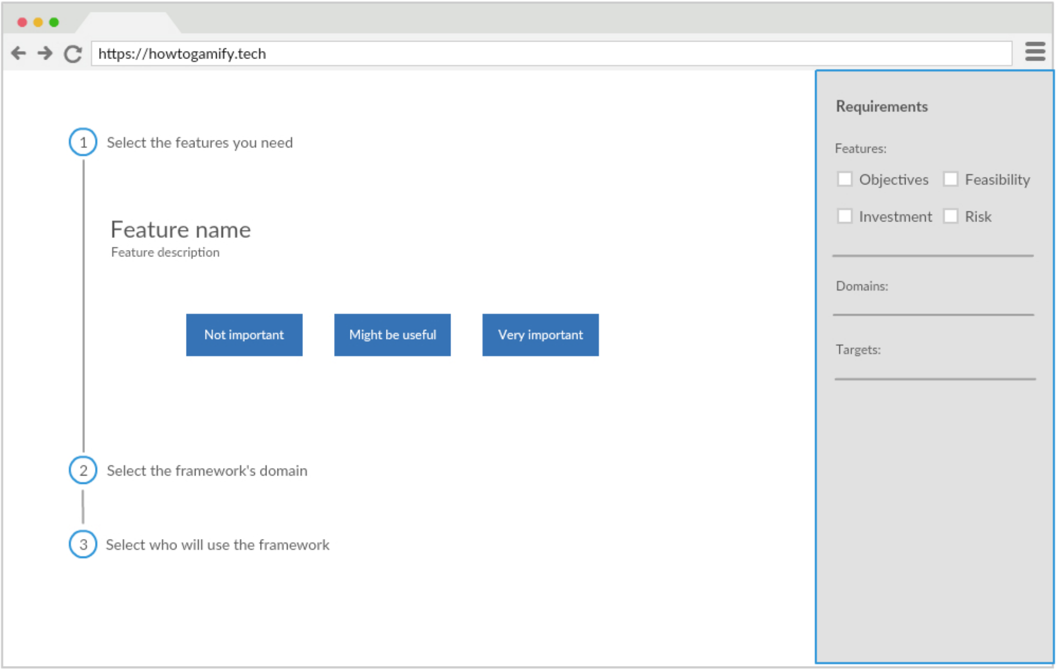
[10] `https://howtogamify.tech`

[11] `https://material-ui.com/demos/steppers/`

[12] `https://material-ui.com/demos/buttons/`

[13] `https://material-ui.com/demos/selection-controls/#checkboxes`

FIGURE 5.2: Wireframe: Requirement collection



The user is also reminded that multiple options may be chosen. Four options are presented as per the available domain values presented in Section 4.3.2. An option is represented by a *Checkbox* component placed inside of a *Paper* component[14].

After selecting the desired domains, the user clicks the "Next" button to proceed to the last step. The selected domains appear in the Sidebar as shown in Figure B.4. Each of selected domains is visualized using the *Chip* component[15].

The last step is analogical to the *Domain requirement* step. However, the user is asked about the target specificity of the project, instead:

> *"Who is the gamification designer of your project?"*

It is also hinted to the user that he/she should rather think of it in terms of skills and capabilities instead of a profession.

After selecting the desired targets and clicking the "Next" button, the user is asked to review the selected requirements in the sidebar. This presents an opportunity to make amendments before proceeding to the second and last stage of the interaction – displaying the results. After the user finished with the review, he or she may continue by clicking the "Finish" button.

5.3.2 Presentation of results

The most important part of any recommender system is probably the presentation of the results. They need to be relevant and prioritized to meet the user's expectations. Also, information overload needs to be avoided. Too much information may distract the user or even overwhelm him/her. Thus, the amount of information should be limited with an option to drill down into further details if the user decides to do so. The result page is depicted by the wireframe presented in Figure 5.3.

[14] `https://material-ui.com/demos/paper/`

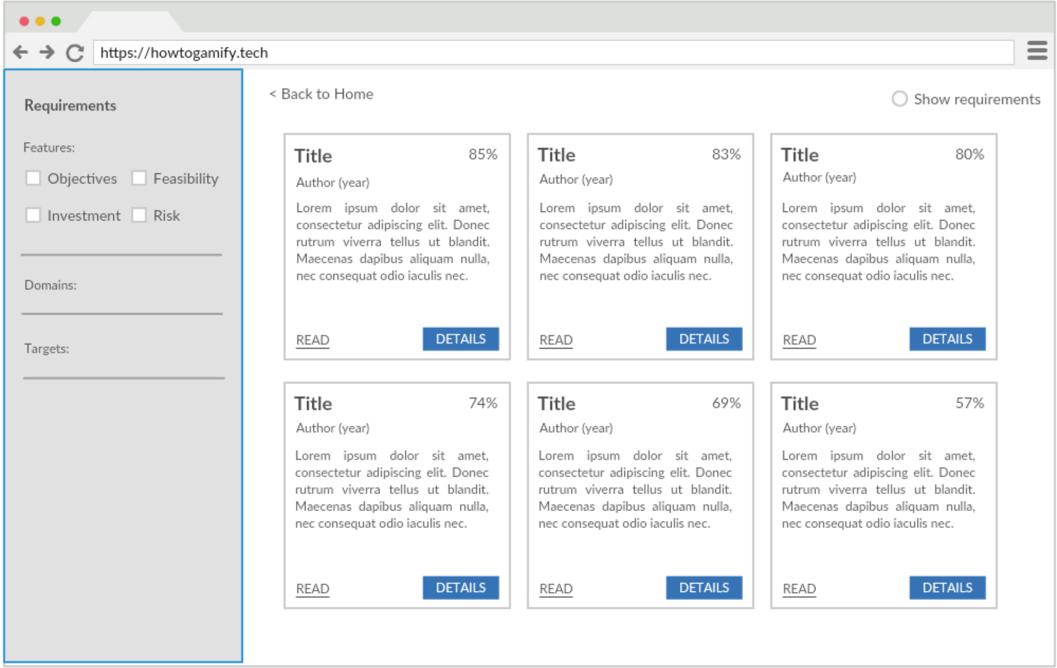
[15] `https://material-ui.com/demos/chips/`

Results

The results are visualized in a *Grid* layout[16] to achieve responsive design. The results are sorted based on the match rate in descending order. Since the number of currently classified frameworks in Mora et al. (2017) [8] is only 27, all results are displayed.

Each result is visualized using a *Card* component[17]. It contains several details about the result: match rate, title, authors, year of publication and a short description. The card also contains two action buttons: (1) read more link and (2) details button. The read more button opens the `url` of the framework in a new tab – fulfilling the requirement *FR4*. A new tab also allows the user to return back to the results for further browsing. The details button opens a side sheet on the bottom of the page and thus fulfills the requirement *FR3*. It is visualized by a *Drawer* component[18] and can be seen in Figure B.7.

FIGURE 5.3: Wireframe: Presentation of results



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum viverra tellus ut blandit. Maecenas dapibus aliquam nulla, nec consequat odio iaculis nec.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum viverra tellus ut blandit. Maecenas dapibus aliquam nulla, nec consequat odio iaculis nec.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum viverra tellus ut blandit. Maecenas dapibus aliquam nulla, nec consequat odio iaculis nec.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum viverra tellus ut blandit. Maecenas dapibus aliquam nulla, nec consequat odio iaculis nec.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum viverra tellus ut blandit. Maecenas dapibus aliquam nulla, nec consequat odio iaculis nec.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum viverra tellus ut blandit. Maecenas dapibus aliquam nulla, nec consequat odio iaculis nec.

Drawer

The drawer is divided into two parts: (1) overview and (2) feature coverage.

The overview contains more information about the framework, such as title, authors, year of publication, application area, and target. When the drawer appears, it overlays the results with a semi-transparent grey background indicating that the user should focus on the contents of the drawer. However, it poses the risk of losing the context of the selected framework. To prevent this from happening, the title, authors and year of publication are repeated in the overview, so that the user could connect the drawer to the selected framework. Each domain and target of the framework also has an indication of whether the shadow framework created by the user

[16] `https://material-ui.com/layout/grid/`

[17] `https://material-ui.com/demos/cards/`

[18] `https://material-ui.com/demos/drawers/`

matches the attributes of the selected framework. The indicator is visualized using colored icons: a match is represented by a green tick and a mismatch by a red cross.

The feature coverage part of the drawer list the following information about the selected framework in relation to the shadow framework:

1. Matching features
2. Partially matched features
3. Missing features
4. Extra features

Each feature is visualized using a *Chip* component together with a *Tooltip* component[19] depicted in Figure B.8. In the first stage of the interaction, a *Checkbox* component has been used to visualize a feature, so that its selected value could be altered. However, the drawer serves one purpose only: to display information. The Chip component has been chosen because of familiarity reasons. The user has already encountered this component during the first stage of interaction and serves well for the intended purpose.

When a user hovers over a feature, it displays a tooltip containing the description of the feature.

Sidebar

The results screen also contains a "Show requirements" switch control visualized by the *Switch* component[20]. The control has been placed on the right side of the screen. The left side contains a "Back to home" button navigating the user to the home page clearing the current session.

Once the requirements are switched, a sidebar appears on the left side of the screen as shown by Figure B.6. It is the same sidebar the user encountered during the first interaction stage. However, the editability of the fields is switched off. The implementation allows for switching this option back on, thus allowing the experience of changing the requirements and receive a real-time immediate response. This option would allow the users the better explore the relation between requirements and the results. However, this feature has not been considered as part of the scope of implementation.

The placement of the sidebar differs from the one during the first stage of interaction. If it was placed on the right side instead, it would push the switch control further away. By placing it on the left side, the user could easily switch between the states without a mouse movement.

[19] https://material-ui.com/demos/tooltips/

[20] https://material-ui.com/demos/selection-controls/#switches

Chapter 6

Evaluation

This chapter presents the evaluation of the implemented artifact based on the steps presented in Section 2.3.5. Firstly, the researcher discusses whether the objectives of the created artifact were met as well as its performance using different tools. Secondly, the results of the validation survey sent to the interest group will be presented. Lastly, the chapter presents a summary of the expert interview conducted with Tim Allison, Senior Design Manager, on the usability of the created artifact.

6.1 Objectives and performance

The objectives can be evaluated based on the fulfillment of the requirements presented in Section 4.4. Additionally, different page speed analyzing tools and the Chrome Dev Tools are used to evaluate the performance of the artifact.

6.1.1 Objectives

The functional requirements presented in Table 4.7 were all addressed in Section 5.3.

Users can specify their project requirements in the form of feature, domain, and target requirements fulfilling *FR1*.

The specified requirements can also be modified at any given point during the requirement collection stage by utilizing the *Sidebar* component fulfilling *FR2*.

FR3 and *FR4* are fulfilled by the implementation of the result presentation stage. The user can access the details of each result presenting the matched, partially matched, mismatched and extra features in the *Drawer* component. Whether the domain and target of the framework matches are also indicated. Each result also shows a link through which the respective framework can be accessed. In case of publicly not available frameworks, the link points to a resource providing more details about the publication.

The non-functional requirements presented in Table 4.8 were addressed as follows. The data are logically structured in a way that allows the recommender engine to be extended by new features as well as frameworks. The project is open source and new frameworks can be requested by opening an issue specifying the required details of the framework. With the above, the requirement *NFR1* is fulfilled.

The comparison algorithm only takes into consideration the features, domain and target requirements as they are considered to be generic comparison factors. All included frameworks have these attributes as specified by Mora et al. (2017) [8]. The recommender engine could be extended by other comparison factors in the future such as framework requirements and theoretical dependencies. Hence, the requirement *NFR2* is considered to be fulfilled.

The artifact does not recommend a merged optimal framework by suggesting framework combinations. However, each framework includes a list of its theoretical dependencies which allows the system to be extended in the future. *NFR3* is fulfilled.

The system was developed using modern and well-maintained web technologies leveraging popular and robust libraries. A new version of the artifact can be deployed to GitHub Pages using a single command. With the above, the requirement *NFR4* is fulfilled.

By using AVA as the test runner, unit tests are isolated on a process level and executed in parallel. With these, fast and reliable test execution is achieved which fulfills the requirement *NFR5*.

The created artifact was placed behind a CDN. Cloudflare's network is powered by data centers in over 180 cities[1] around the world. Utilizing their service, the requirement *NFR6* is considered fulfilled.

The artifact was developed as a single page web application. Any device with a modern web browser[2] is capable of interacting with the system. Hence, requirement *NFR7* is fulfilled.

The requirement *NFR8* is evaluated by the interest group and the expert in the field of user experience in the Sections 6.2 and 6.3, respectively.

6.1.2 Performance

To assess the performance of the artifact the following tools were utilized: PageSpeed Insights by Google[3], GTmetrix[4], Pingdom[5] and Chrome Dev Tools. The artifact is a single page web application that handles everything including routing. The routing has been implemented in a specific way to allow the app to be deployed through GitHub Pages – described in detail in Section 5.2.3. Some of the above page speed services do not follow redirects triggered by JavaScript code. Hence, they could not analyze the *https://howtogamify.tech/get-started* route where the main recommender logic is placed. Those tools analyzed the homepage instead[6]. Since the artifact is a SPA that does not implement *code splitting*[7], the requested page is the same using the same CSS and JS assets. The only difference is in the time used for scripting and rendering as the homepage is considerably simpler – rendering less and less complex components.

PageSpeed Insights

PageSpeed Insights analyzed the homepage[8]. It produced two reports: one for mobile devices and another one for desktop.

The artifact scored 89/100 for mobile devices and 98/100 for desktops[9]. The tool suggested eliminating render-blocking resources by deferring all non-critical JS files

[1] https://www.cloudflare.com/network/

[2] Internet Explorer by Microsoft is not considered as a modern browser as it does not support new web standards [77]

[3] https://developers.google.com/speed/pagespeed/insights/

[4] https://gtmetrix.com/

[5] https://tools.pingdom.com/

[6] https://howtogamify.tech/

[7] https://webpack.js.org/guides/code-splitting/

[8] https://developers.google.com/speed/pagespeed/insights/?url=https%3A%2F%2Fhowtogamify.tech%2F&tab=mobile

[9] https://developers.google.com/web/tools/lighthouse/v3/scoring

and CSS styles highlighting the styles regarding the "Roboto" font-family. It also suggested to leverage the "font-display" CSS feature to ensure the text is visible by the user while the web fonts are loading. It estimates that after these changes, 0.5 s of loading time could be potentially saved.

GTmetrix

GTmetrix provides two scores: one originating from PageSpeed discussed above and the other from YSlow[10]. It analyzed the homepage and reported[11] a page load time of 0.6 s with 8 requests and a total page size of 221 kB. YSlow identified 3 components that were not cookie-free and suggested to use a cookie-free domain dedicated to serving static assets. It also suggested to add an "Expires" header to the Roboto font-family CSS resource.

Pingdom Tools

Pingdom Tools was the only service out of the three that was able to execute the tests on the routed page. The report[12] says it took 723 ms to load the page, used 12 requests and had a size of 250.2 kB. Pingdom suggested using a cookie-free domain for static assets, too. Strangely, it suggested compressing the components with gzip. However, Cloudflare prefers Brotli as its content encoding method if multiple compression methods are supported by the client. Otherwise, it applies gzip. Pingdom reports the score generated by Google PageSpeed.

Chrome Dev Tools

The above three tools showed some metrics about the page load time, the number of requests and the combined size of the assets. Using Chrome Dev tools, it can be assessed how performant the recommendation engine itself is. Two metrics are considered: time to resolution (TTR) and memory usage.

Time to resolution, in case of the artifact, is the time required to process the requirements and show the results to the user. To remove the effects of the GPU, only the time spent with "scripting" was measured. The measurements have been performed on a desktop device with a 3.1 GHz Intel Core i7 processor. Three measurements were performed: (1) no CPU throttling, (2) CPU throttled by a factor of 4, and (3) CPU throttled by a factor of 6. The average scripting time per level of throttling is shown in Table 6.1.

TABLE 6.1: Recommender Engine: Time to Resolution

Throttling	**Avg. TTR**
No throttling	366.837 ms
4x slowdown	391.353 ms
6x slowdown	369.056 ms

To analyze the memory usage of the artifact, two heap snapshots were taken – one per each JavaScript VM instance. A heap snapshot profile shows the memory distribution among the JavaScript objects and related DOM nodes of the web application. The main instance (web application) showed a heap size of 7.9 MB while the

[10] http://yslow.org/ruleset-matrix/

[11] https://gtmetrix.com/reports/howtogamify.tech/5JrLpXIj

[12] https://tools.pingdom.com/#5ac5e45701400000

service worker had in use 1.3 MB of memory. The snapshots were taken immediately after the results had been displayed. It can be concluded that the memory usage of the artifact is low.

6.2 Interest group

A validation survey was sent to the interest group with the questions presented in Section 2.4.3. The analysis of the responses is presented in Section 4.1.3.

The respondents found the recommendations *mostly relevant* except a specific case where the recommended frameworks assumed the use of an e-learning system. There are two frameworks that assume the use of an e-learning platform [57, 60]. The current ranking algorithm of the recommender engine does not take these assumptions into account.

The respondents mostly agreed that the created artifact is *easy to use*. One respondent suggested in his feedback that the descriptions of the items, such as features, are too long and even overwhelming.

Based on the answers to the question regarding the further use of the system, a pattern may be identified confirming the assumption that the tool will probably provide more utility to less experienced gamification practitioners. Respondents with experience of up to 5 years answered that they are *somewhat likely* to *likely* to further use the tool. Two out of nine respondents said they were completely likely to use the tool in the future.

Regarding future improvements, the respondent suggested some corrections regarding the usability of the artifact. These suggestions are mostly about:

- descriptions being too long or difficult to understand,
- not possible to go back to the previous step,
- the presentation of the results does not emphasize the match rate well enough.

6.3 Interview

The member-checked summary of the interview is presented in Appendix A.4. Some of the observations are aligned with the suggestions of the interest group presented in the previous section. Several action points can be derived from the received feedback for the next iteration of development. Table 6.2 lists the derived action points.

TABLE 6.2: Suggestions for future improvements

Id	Action point
AP1	Change the primary color to color with higher contrast.
AP2	Create and use more apprehensible descriptions.
AP3	Create a visually clearer relationship between the buttons representing the importance of the feature and the representation of the feature in the sidebar.
AP4	Prevent the domain and target fields from being pushed below the fold once the feature requirements step is complete.
AP5	Design a consistent experience regarding the appearance of the items in the requirements sidebar.
AP6	Add a “Results” heading to the result page.

Continuation of Table 6.2	
Id	Action point
AP7	Fix the inconsistency regarding the sidebar placement in the result page.
AP8	Style the individual results to be equal height.
AP9	Put more emphasis on the match rate in the result component.
AP10	Display the top 6 results with a “Load more” option.
AP11	Enable requirement modification on the results page for real-time recommendation exploration.
End of Table	

Chapter 7

Discussion

This chapter reflects upon the chosen research methodology and data collection methods discuss the key findings of the research while addressing the research questions defined in Section 1.1. Furthermore, it presents the key contributions of this research as well as the implications of its findings and how they fit into practice. Finally, it discusses what possible future improvements could be done.

The structure of the selected research methodology – described in Section 2.3 – allowed the research to focus on completing its objectives and produce relevant results tested by a group of gamification practitioners. The survey strategy chosen as the main method of data collection brought mixed results. A higher response rate was expected for the first questionnaire, described in Section 2.4.3. Its main purpose was to recruit experts – gamification practitioners – for the purposes of evaluating the created artifact. Initially, the target audience was not well defined and resulted in a low response rate and interest in the research. Only after a more appropriate method of locating and contacting the target audience was found – gamification related Meetup and Facebook groups – the responses gained in relevance as well as the interest of the respondents peaked. Unfortunately, most of the organizers of Meetup groups have never replied and it resulted in a lower number of respondents to the initial survey than expected. However, the number of respondents interested in the research willing to participate in follow-up surveys was higher than expected – 60% of all respondents. Meetup groups are mostly organized around a specific profession, hobby or activity in order to share knowledge and experiences. The reason behind the high conversion rate might be the self-educative disposition of Meetup and Facebook Group organizers. Conversely, the mortality of the interest group also presented a challenge to the study. Only 40% of the interest group remained active in the 2 surveys following the initial survey. Despite the low response rate, the feedback provided in the last two surveys was valuable and allowed for the evaluation of the research presented in Chapter 6.

7.1 Summary of findings

The analysis of empirical data, presented in Section 4.1, provides the key findings regarding the use of gamification design frameworks in practice:

- Practitioners are often customing the frameworks in order to adapt them to the needs of their current projects.
- The expected feature coverage of gamification design frameworks differs from practitioner to practitioner. Surprisingly, only 50% of respondents could consider ethics as a gamification project requirement – despite the emphasis of the importance of ethics expressed by several researchers [17, 6].

- The preference of the frameworks in relation to the comparison factors identified in Section 4.3 also varies between individuals. Hence, it was a good decision to extract the implementation of a generic framework comparison algorithm as a requirement (NFR2) defined in Table 4.8.

The literature review revealed three papers – described in Section 3.3 – with the aim of systematically mapping and reviewing gamification design frameworks, out of which two papers originated from the same authors. The third paper [6] did not provide the detailed results of the performed classification. Hence, the designed artifact could only rely on the results of a single classification paper [8] which extends the previous work of the authors [5]. More research in this area is needed to further advance the developed recommender system which was specifically designed for extensibility. The system can be extended by both new frameworks and classification features in case the research in this area progresses. The recommender system maps the project requirements onto the three comparison factors: *feature-completeness*, *domain specificity* and *target specificity*, which in turn maps onto an ordered list of recommended frameworks based on their match rate – effectively answering the research question RQ1. Moreover, the research defined the terms *optimal* and *sub-optimal frameworks* in Section 4.3.1. It also explored the theoretical possibility of merging different frameworks in Section 4.3.5 with the purpose of creating a single optimal framework. This option is an interesting potential feature of the recommender system considering the limited amount of frameworks currently mapped by the classification studies of gamification design frameworks.

The created system was then tested by the interest group of gamification practitioners who were asked to imagine a project with its requirements and use the tool to find the best matching gamification design framework for the imagined purposes. The practitioners evaluated the returned recommendations as mostly relevant and revealed an edge case where the results could be improved. The identified case was an imaginary project in the domain of education not using an e-learning platform as its basis. The top ranking recommendations were reported to be frameworks assuming the use of e-learning platforms. To improve the returned results in such cases, another comparison factor would need to be introduced regarding the assumptions the frameworks work with. It is mostly the case of domain-specific frameworks as generic ones are designed to be universal in their use cases. The potential use of framework assumptions, or also called framework requirements, was predicted during the implementation phase of the project and mentioned as part of tackling the requirement *NFR4 - Adaptability* in Section 5.1. The validation of the created tool by the interest group proved to be a successful way of validating the mapping of project requirements to design frameworks as it uncovered the described edge case – answering the research question RQ1.1.

The overall analysis presented in Chapter 4 led to the extraction of several key requirements described in Section 4.4. The requirements provide an answer to the research question RQ2. They state that the implemented system requires to be extensible, adaptable, iterable and testable. These requirements and the specific implementation of the system allow it to improve over time as more research is conducted in the area of gamification design framework classification.

To address the research question RQ2.1, a validation survey was sent out to the interest group. They were asked about the likelihood that they would further use the tool in the future. The results show that the respondents with no more than 5 years of experience are positively leaning towards using the tool in the future as well. However, the researcher is aware of the possible participation bias and a better answer to the research question could be obtained over time by evaluating the ongoing usage data. For that, usage tracking needs to be implemented where every interaction of the users would be logged. A detailed analysis of the interaction data could reveal not only the utility provided by the tool but also the relevancy of the recommendations using CTR[1] as a metric following the example of major search engines.

7.2 Implications of findings

The created artifact provides value to gamification practitioners, especially to its less experienced segment. As one member of the interest group stated in his/her feedback, the tool is of *"great use as it helps to streamline articles related to the interest, rather than having to try and use a different search engine and read thousands of articles"*. The system has been designed to be extensible by both new frameworks and new comparison features. It is also open source and has followed a modular design and the recommendation engine can be extracted into its own NPM module[2], so that it can be reused in other projects. Other researchers and developers might further build on top of the created artifact. For instance, a machine learning-based recommender system could be implemented that extracts the framework requirements from text or speech using NLP and provides them as input to the recommender engine. This kind of augmentation of the created tool may provide an even better user experience following the trend in conversational UI design.

The project website[3] could also serve as a central repository of gamification design frameworks. Several papers communicate the importance of ethics and sustainable design in gamification [17, 6]. If extended in content, the website could also serve the purposes of raising awareness about the importance of these aspects.

7.3 Future improvements

Following the feedback received from the interest group and the expert in the field of design and user experience, a list of 11 action points was assembled and presented in Table 6.2. Addressing the raised issues this way would lead to a substantial improvement of the recommender system.

Another area for improvement was previously discussed in the summary of findings. Framework assumptions could be implemented as the fourth comparison factor for the recommendation engine. Also, an interesting way of presenting recommendations could be based on the theoretical foundations of the frameworks. The practitioners could select the theories, behavioral models, player taxonomies and other concepts they are familiar with and the system would recommend frameworks best fitting to their current skills and knowledge. This approach could be described as the *approach of least resistance* while the current approach taken by the recommendation system can be described as *best effort*. Nevertheless, both improvements would require further research to be conducted in the area of gamification design framework classification.

[1] Click-through rate

[2] https://www.npmjs.com/

[3] https://howtogamify.tech

Chapter 8

Conclusion

The thesis successfully met its objectives: a fully working gamification design framework recommender system has been developed and deployed to be used by any gamification practitioner. The created artifact is a novel solution that fills in the gap created by a large amount of gamification design frameworks and no method being present to determine which framework would be the most suitable to use for a praticular project.

The research questions of the thesis have also been answered. Through an extensive literature review, the main features of gamification design frameworks have been identified. The study established the terms *optimal* and *suboptimal frameworks* and identified three main framework comparison factors: (1) *feature-completeness*, (2) *domain specificity*, and (3) *target specificity*. It also found that project requirements can directly be translated into the identified framework features. This translation can be matched to the design frameworks through comparison to their feature coverage. Applying the other two comparison factors as well – (2) and (3) – relevant frameworks can be recommended.

The relevancy of the recommendations has been validated by an interest group of 10 gamification practitioners. The interest group succeeded in uncovering an edge case regarding the relevancy of the recommendations. It can be solved by the predicted fourth comparison factor: matching *framework requirements*. However, the implementation of this solution would require more research in the area of gamification design framework classification with a particular focus on the internal assumptions of the frameworks in regards to the projects they are intended to be used for.

The created artifact proved to be the most useful to practitioners with up to 5 years of experience. Time and more usage data could reveal if the presented conclusions of this study could also be generalized across the entire field of gamification design.

Some participants of the interest group asked for more frameworks to be covered by the tool. The recommender system has been implemented with extensibility and adaptability in mind. In case the research in the area of gamification design framework classification progresses, the system can be extended by new frameworks as well as new features. Furthermore, the developed software has a modular design that allows other researchers to re-use its core recommender engine and augment it by new functionality. For instance, a machine learning solution could extract the project requirements directly from natural language, speech or text in documents. Those requirements could be then forwarded to the implemented engine. Recommendations provided this way could improve the user experience of the system.

To improve the utility of the developed recommender system, the study proposes further research to be conducted in the following areas:

- More gamification design frameworks should be classified based on the approach developed by Mora et al. (2017) [8].

- New classification studies should analyze the frameworks based on their internal assumptions towards the projects they are intended to be used for as well as their dependencies – theories, models and related concepts they are based upon.

Progress in the research areas proposed above could substantially improve the depth to which project requirements can be specified which would lead to more relevant recommendations.

Appendix A

Data Collection

A.1 Questionnaire 1: Testing and recruiting

The first questionnaire was designed with three user segments in mind – as defined in Section 2.4.3 – each following a different path of questions. The first user segment were the non-practitioners and they were asked the questions presented in Table A.1. The second user segment were the casual practitioners; their questions are presented in Table A.2. The last segment were the advanced practitioners; their questions are presented in Table A.4.

TABLE A.1: Questionnaire 1: non-practitioner

ID	Question	Type	Answers
Q1	Have you used gamification in your product(s) to drive customer engagement, loyalty, or experience?	List	Yes/no
Q2	Would you be interested in exploring gamification and its use in your business?	List	Yes/no/-maybe
Q3	What are your reasons for the above choice (if applicable)?	Open	N/A
Q4	Could you shortly explain the reason you are not interested in further exploration of gamification?	Open	

End of Table

TABLE A.2: Questionnaire 1: casual practitioner

ID	Question	Type	Answers
Q1	Q1, Table A.1	List	yes/no
Q2	Have you been using a particular gamification design framework to design your gamified experience?	List	yes/no
Q3	Have you read/heard of any specific gamification framework?	List	yes/no
Q4	If yes, could you describe the framework you are the most inclined to adopt at your company?	Open	
Q5	What are your reasons for the above choice (if applicable)?	Open	

Continuation of Table A.2

ID	Question	Type	Answers
Q6	Select your company's main industry	List	Industry, Table A.3
Q7	What is the approximate size of your company?	List	Company size, Table A.3
Q8	Would you like to be part of a focus group reviewing the created tool?	List	yes/no
Q9	Name	Open	
Q10	What is your role in your company?	List	Roles. Table A.3
Q11	How many years of experience do you have designing gamified experiences?	List	Experience, Table A.3
Q12	If you'd like to keep in touch with us and/or learn more about the project, please provide your e-mail address.	Open	E-mail address

End of Table

TABLE A.3: Answer lists

Industry	Company size	Roles	Experience
Education	Less than 10 employees	Designer	No experience.
Energy	Between 10 and 50 employees	Software Engineer	0-1 years
Entertainment & Gaming	Between 50 and 250 employees	Human-Computer Interactions researcher	1-2 years
Financial & Insurance Services	More than 250 employees	Other	3-4 years
Government & Non-profit			4-5 years
Healthcare			5+ years
IT Services & Consultancy			
Manufacturing & Computer Hardware			
Marketing & Advertising			
Media & Telecommunications			
Professional & Business Support Services			
Real Estate			
Retail			
Social Media			
Software			

Continuation of Table A.3			
Industry	Company size	Roles	Experience
Travel, Hospitality, & Tourism Web Applications Web Hosting Other			
End of Table			

TABLE A.4: Questionnaire 1: advanced practitioner

ID	Question	Type	Answers
Q1	Q1, Table A.1	List	yes/no
Q2	Q2, Table A.1	List	yes/no
Q3	Please, describe the design framework(s) you've adopted at your company.	Open	
Q4	How satisfied are you with the currently used gamification design framework(s)?	Rating	Very unsatisfied (1) - Very satisfied (5)
Q5	Do you customize/modify the design framework(s) to fit your specific needs?	List	yes/ no/ sometimes
Q6	Do you plan on moving to other framework(s)?	List	yes/ no/ maybe
Q7	Q6, Table A.1	List	Industry, Table A.3
Q8	Q7, Table A.1	List	Company size, Table A.3
Q9	Q8, Table A.1	List	yes/no
Q10	Q9, Table A.1	Open	
Q11	Q10, Table A.1	List	Roles. Table A.3
Q12	Q11, Table A.1	List	Experience, Table A.3
Q13	Q12, Table A.1	Open	E-mail address
End of Table			

A.2 Questionnaire 2: Preferences and initial feedback

The second survey was only targeted at the assembled interest group through sign-ups from the previous survey. The questions asked the interest group are presented in Table A.5.

TABLE A.5: Questionnaire 2: Preferences and initial feedback

ID	Question	Type	Answers
Q1	What concepts should a gamification framework tackle in your opinion? Think of a project and its requirements and list them in the answer box below.	Open	
Q2	Could the above feature be considered as a requirement for a design framework? (for each feature with its description – Table A.6)	List	yes/ no/ I don't know/not sure
Q3	After reading about the previous features, do you have any additional idea about other requirements that could be covered by gamification design frameworks?	Open	
Q4	How important do you find feature-completeness?	Rating	Not important (1) - Very important (2)
Q5	How important do you find the matching application domain?	Rating	Not important (1) - Very important (2)
Q6	How important do you find a matching target (the target audience of the framework)?	Rating	Not important (1) - Very important (2)
Q7	Would you favor a generic but feature complete framework or a specific one with 1-2 features not being covered?	List	Generic and feature complete framework / Domain specific, less feature complete framework
Q8	Would you favor a feature complete framework written for the general audience or one that is specifically targeted at your particular skill set but missing 1-2 features?	List	General and feature complete framework / Skills specific, less feature complete framework
Q9	Would you rather favor a generic framework targeted at your skills or a specific framework written for the general audience? (consider both feature complete)	List	Generic framework, matching my skill set / Specific framework, for the general audience

End of Table

TABLE A.6: Features and their descriptions

Feature	Description
Objectives	The specific performance goals. Think of determining and designing what the main objectives should be of the gamified experience.

Continuation of Table A.6

Feature	Description
Feasibility	A previous study, evaluation, and analysis of the potential of applying gamification or refuse it. An analysis that reveals whether gamification is a correct approach to the problem you're trying to solve in a project.
Risk	A probability or threat of damage, injury, liability, loss, or any other negative occurrence. An analysis of what could go wrong, what the main risks are with the project from the gamification point of view.
Investment	The benefit to the investor resulting from running a gamified experience. An analysis of how much investment is required and what are the returns.
Stakeholders	A technique used to identify and keep in mind the people who have to interact with the design process. A design framework that implements this feature describes how to identify the key stakeholders and how to involve them in the design process.
Engagement cycle	The game mechanics combined with reinforcement and feedback in order to engage the player in the key system actions. A framework that implements this feature describes how to design a continuous feedback loop to increase the player's engagement.
Endgame	A pre-established end of game or glorious victory in the system, usually stretching players to the limits of their abilities. This feature describes the design process specifically with finite/infinite games in mind.
Onboarding	The way of starting the new participants. This feature describes the design process of introducing game mechanics and rules to the fresh players.
Rules	The body of regulations prescribed by the designer. This feature describes the process of designing the game economy and rules.
Metrics	The standards of measurement by which efficiency, performance, progress, process or quality. This feature describes what data and/or processes should be measured and how.
Analytics	The algorithms and data used to measure key performance indicators. This feature describes how to interpret and analyze the beforementioned metrics.
Ethics	A branch of philosophy that involves systematizing, defending and recommending concepts of right and wrong conducts. This feature describes what gamification design practices are ethical to use and to what extent. For example, what data is ethical to collect, how long the data should be retained, and where is the fine line between motivation and manipulation.
Fun	Enjoyment or playfulness. This feature describes how to design for fun in a project.

Continuation of Table A.6

Feature	Description
Motivation	The behavior which causes a person to want to repeat an action and vice-versa. This feature describes how to design for and trigger internal motivation. Probably also tackles how to combine internal and external motivation.
Social	The interaction between players. This feature describes how to design the social and cooperative aspects of gamified experiences.
Desired behaviors	The expected response of the players after the interaction. This feature describes how to design for a lasting change in behavior. Example: development of good habits like physical activities, reading, meditating, socializing, etc.
Profiling	The process of identifying players and player personas. This feature describes how to identify who your target audience is and what they are like.
Taxonomy	A classification/taxonomy of player types. This feature describes how the identified players (feature before) map to specific player types. It also reveals what gamification features might be more compelling to each player type.
Storytelling	The story and context created by designers. This feature describes how to design gamified experiences with narratives.
User experience	Refers to everything designed into the gamified system which a player being may interact and the player's behaviors, attitudes, and emotions. This feature describes the entire process of designing compelling interactions with the gamified system.
Technology	The use or need of a software component for development. This feature describes the architectural and implementation strategies for creating gamified systems/features.

End of Table

The thank you page presented the following text and asked the participants to leave a feedback in an answer box below this text:

“Thank you very much for spending your precious time answering the above questions! I would like to ask you to take a look at a working prototype of my gamification design framework recommender system.

https://howtogamify.tech/

The website is still work in progress (Read more link does not work, yet - only "Get started"). The optimization of mobile layouts is still under active development. For the best experience, consider using a desktop device or a table positioned horizontally.

I would like to ask you to visit the website and try to get a recommendation for an imaginary gamification project. After that, please, leave your feedback on the usability of the system and mention anything you particularly liked or disliked about it :-) Also, any other feedback is welcome and very much appreciated :-)

You may submit the survey without completing this step and return and edit your responses later, whenever you have time :-) Thank you very much."

A.3 Questionnaire 3: Validation survey

The third and last questionnaire was also sent only to the respondents participating in the interest group. The questions asked are described in Table A.7.

TABLE A.7: Questionnaire 3: Validation survey

ID	Question	Type	Answers
Q1	How relevant did you find the recommended frameworks in the results?	Ranking	Not at all relevant (1) - Completely relevant (5)
Q2	How easy was it to use the web application?	Ranking	Very difficult (1) - Very easy (5)
Q3	How likely are you to use this application in the future?	Ranking	Not at all likely (1) - Completely likely (5)
Q4	What improvements or corrections would you suggest me to make?	Open	

End of Table

A.4 Interview

This section presents a summary of the interview conducted with Tim Allison, Senior Design Manager. He was asked to imagine a gamification project with its requirements and, using the created artifact, find the most suitable design framework. The summary of the interview went through a member check. The observations made during the interview and the provided suggestions can be summarized in the following points:

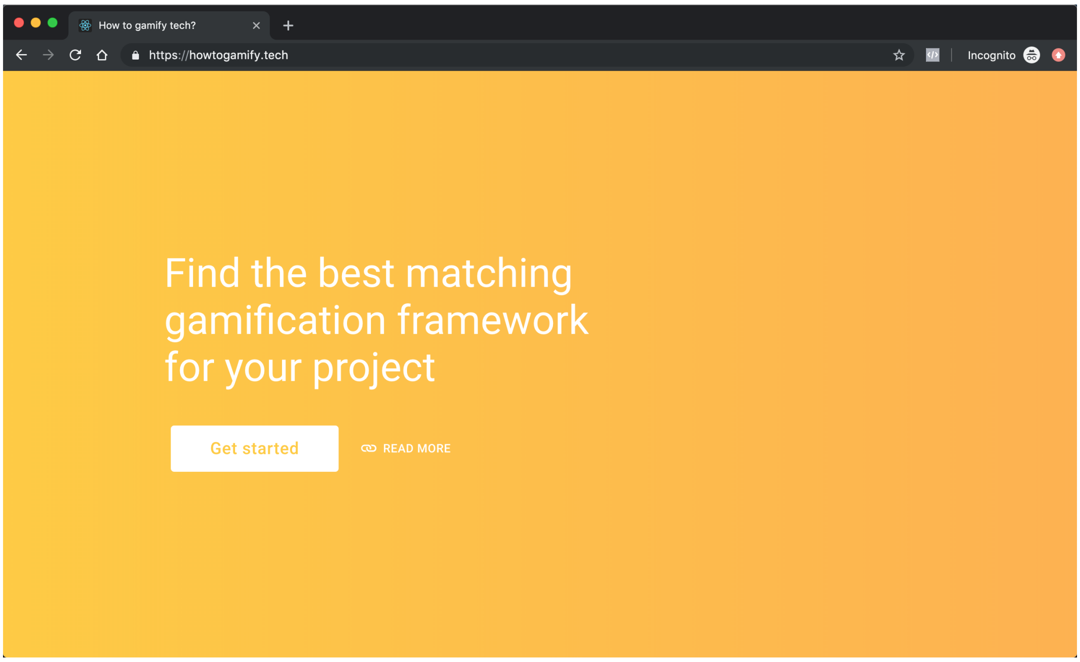
- The chosen primary color (yellow) does not have enough contrast.
- The descriptions of the items are not apprehensible enough. The items could be described with a shorter text and a tooltip could be used to provide further details.
- The relationship between the buttons representing the degree of importance and the resulting checkboxes in the sidebar is not clear enough.
- Completing the first step of the requirement collection results in the domain and target fields being pushed below the fold. The user needs to know that he/she needs to scroll in order to access or see those fields. The ability to scroll the sidebar is not particularly clear.
- There are inconsistencies regarding the sidebar and the way the items appear inside the component. The features appear immediately after selecting the importance, whereas the domain and target appear only after clicking the "Next" button.

- The result page is missing a headline.
- An inconsistency in the UI is introduced by placing the sidebar to the opposite side in comparison with the requirement collection stage of the interaction. The sidebar could be placed on the same side as previously if a button was introduced instead of the switch component and a close icon placed in the upper right corner of the sidebar.
- An inconsistency in the details component is introduced by visualizing the features using a pill (Chip component) instead of the previously used checkboxes. The pills might be used in the requirement collection stage of the interaction as well.
- The individual results differ in their height. It might give a false impression to the subconsciousness of the user that the larger results might also be more relevant. Displaying the results uniformly might help the user to focus only on the relevant differences between the results.
- The match rate should be visually more emphasized as it conveys more important information than e.g. the title.
- Displaying only a couple of results might help the user better focus on the task at hand. A load more button could be provided in case the recommendations are not satisfactory enough.
- It might be interesting as well as useful to have the requirements editable during the second stage of interaction as well – presentation of results. It would allow the user to explore the effects of the individual requirements on the recommendations. With the current implementation, the user would have to start the whole process all over again to see the change.

Appendix B

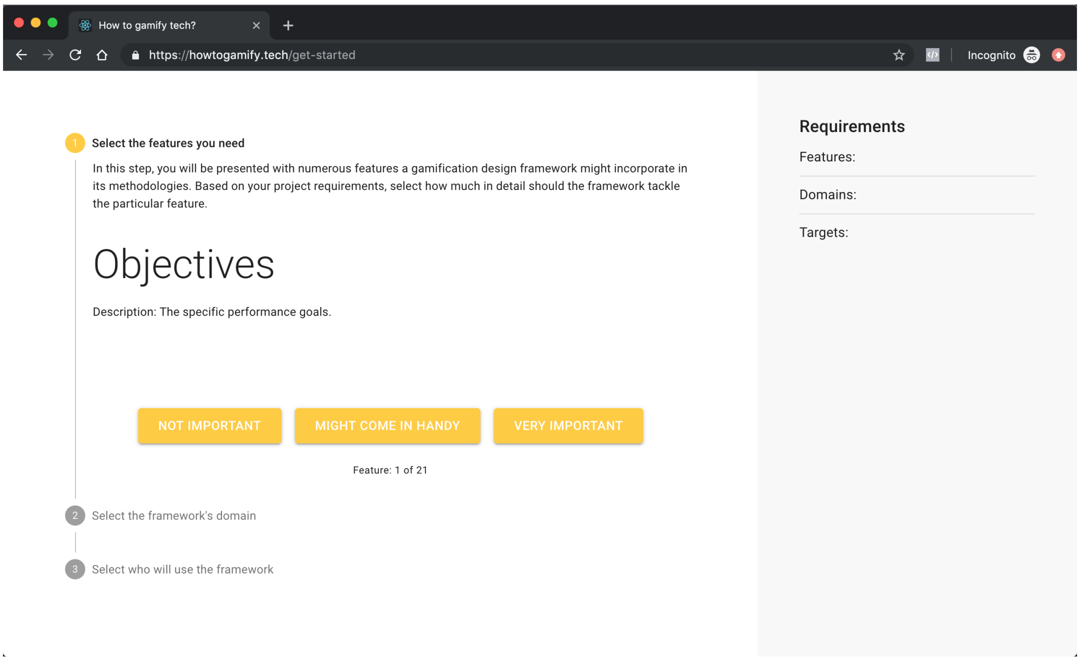
User interface

FIGURE B.1: User interface: Homepage



Find the best matching gamification framework for your project

FIGURE B.2: User interface: Recommender system - initial screen



1 Select the features you need

In this step, you will be presented with numerous features a gamification design framework might incorporate in its methodologies. Based on your project requirements, select how much in detail should the framework tackle the particular feature.

Objectives

Description: The specific performance goals.

Feature: 1 of 21

2 Select the framework's domain

3 Select who will use the framework

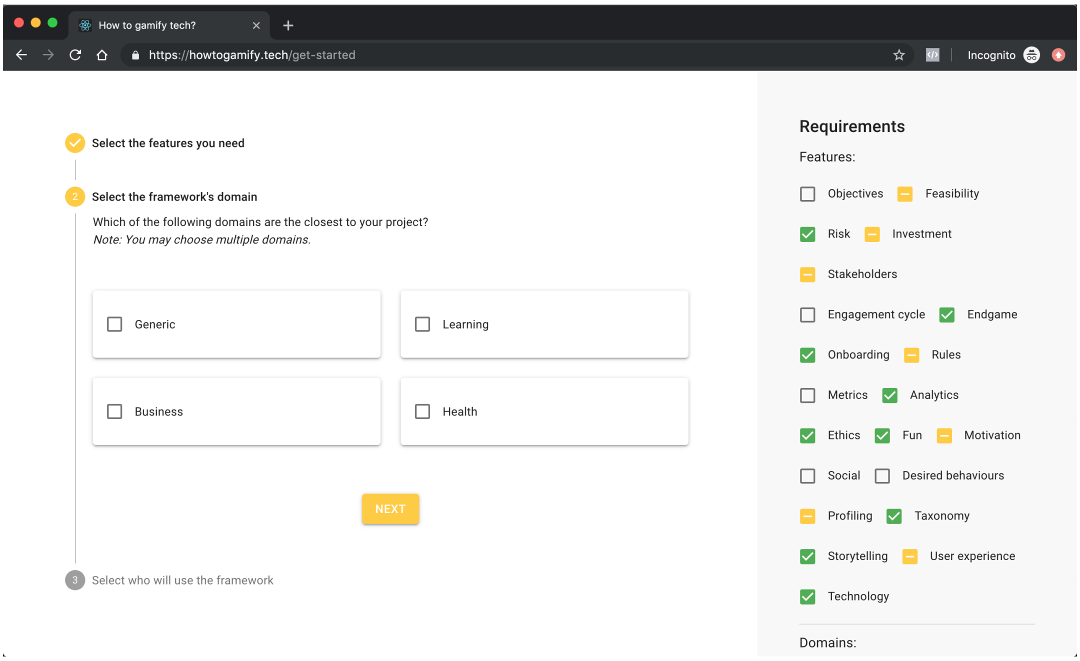
Requirements

Features:

Domains:

Targets:

FIGURE B.3: User interface: Requirements - importance to checkboxes



✓ Select the features you need

2 Select the framework's domain

Which of the following domains are the closest to your project?
Note: You may choose multiple domains.

□ Generic □ Learning

□ Business □ Health

3 Select who will use the framework

Requirements

Features:

□ Objectives ⊟ Feasibility

☑ Risk ⊟ Investment

⊟ Stakeholders

□ Engagement cycle ☑ Endgame

☑ Onboarding ⊟ Rules

□ Metrics ☑ Analytics

☑ Ethics ☑ Fun ⊟ Motivation

□ Social □ Desired behaviours

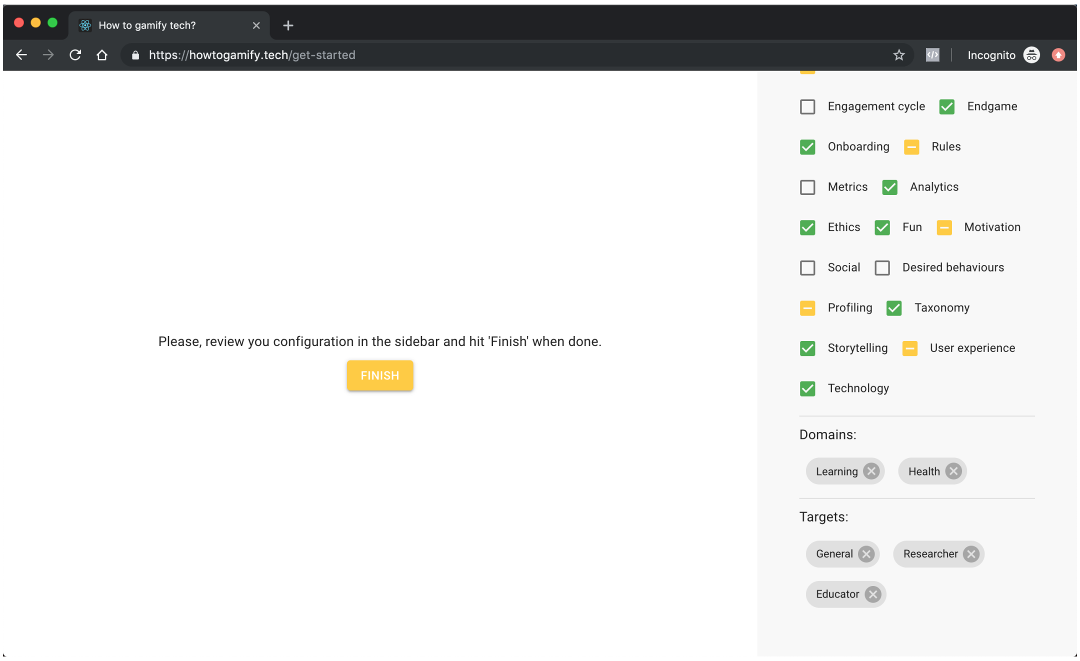
⊟ Profiling ☑ Taxonomy

☑ Storytelling ⊟ User experience

☑ Technology

Domains:

FIGURE B.4: User interface: Review



☐ Engagement cycle ☑ Endgame

☑ Onboarding ⊟ Rules

☐ Metrics ☑ Analytics

☑ Ethics ☑ Fun ⊟ Motivation

☐ Social ☐ Desired behaviours

⊟ Profiling ☑ Taxonomy

☑ Storytelling ⊟ User experience

☑ Technology

Domains:

Learning ⊗ Health ⊗

Targets:

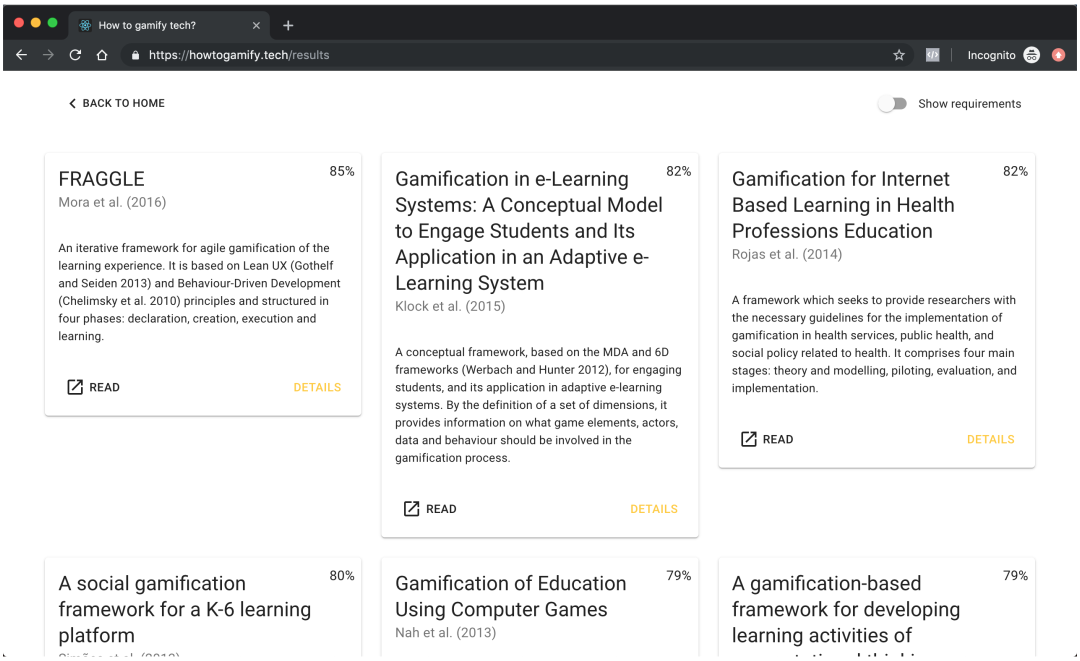
General ⊗ Researcher ⊗

Educator ⊗

Please, review you configuration in the sidebar and hit 'Finish' when done.

FINISH

FIGURE B.5: User interface: Results



< BACK TO HOME

Show requirements

FRAGGLE 85%
Mora et al. (2016)

An iterative framework for agile gamification of the learning experience. It is based on Lean UX (Gothelf and Seiden 2013) and Behaviour-Driven Development (Chelimsky et al. 2010) principles and structured in four phases: declaration, creation, execution and learning.

READ DETAILS

Gamification in e-Learning Systems: A Conceptual Model to Engage Students and Its Application in an Adaptive e-Learning System 82%
Klock et al. (2015)

A conceptual framework, based on the MDA and 6D frameworks (Werbach and Hunter 2012), for engaging students, and its application in adaptive e-learning systems. By the definition of a set of dimensions, it provides information on what game elements, actors, data and behaviour should be involved in the gamification process.

READ DETAILS

Gamification for Internet Based Learning in Health Professions Education 82%
Rojas et al. (2014)

A framework which seeks to provide researchers with the necessary guidelines for the implementation of gamification in health services, public health, and social policy related to health. It comprises four main stages: theory and modelling, piloting, evaluation, and implementation.

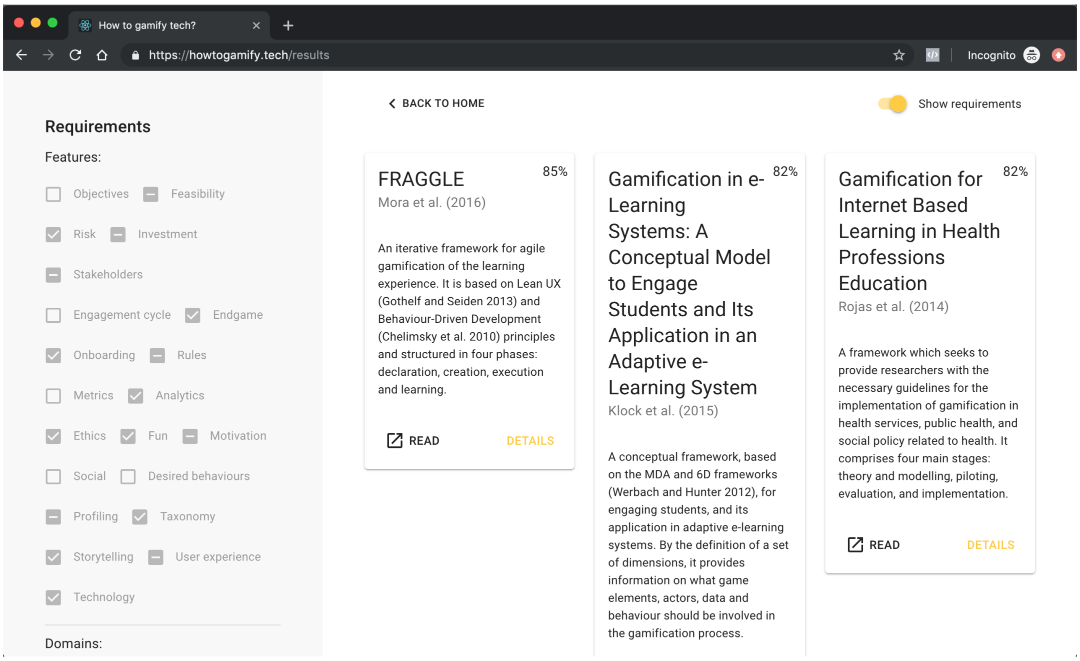
READ DETAILS

A social gamification framework for a K-6 learning platform 80%

Gamification of Education Using Computer Games 79%
Nah et al. (2013)

A gamification-based framework for developing learning activities of 79%

FIGURE B.6: User interface: Show requirements on results page



How to gamify tech?

https://howtogamify.tech/results

Incognito

Requirements

Features:

□ Objectives ⊟ Feasibility

☑ Risk ⊟ Investment

⊟ Stakeholders

□ Engagement cycle ☑ Endgame

☑ Onboarding ⊟ Rules

□ Metrics ☑ Analytics

☑ Ethics ☑ Fun ⊟ Motivation

□ Social □ Desired behaviours

⊟ Profiling ☑ Taxonomy

☑ Storytelling ⊟ User experience

☑ Technology

Domains:

< BACK TO HOME

Show requirements

FRAGGLE 85%
Mora et al. (2016)

An iterative framework for agile gamification of the learning experience. It is based on Lean UX (Gothelf and Seiden 2013) and Behaviour-Driven Development (Chelimsky et al. 2010) principles and structured in four phases: declaration, creation, execution and learning.

READ DETAILS

Gamification in e-Learning Systems: A Conceptual Model to Engage Students and Its Application in an Adaptive e-Learning System 82%
Klock et al. (2015)

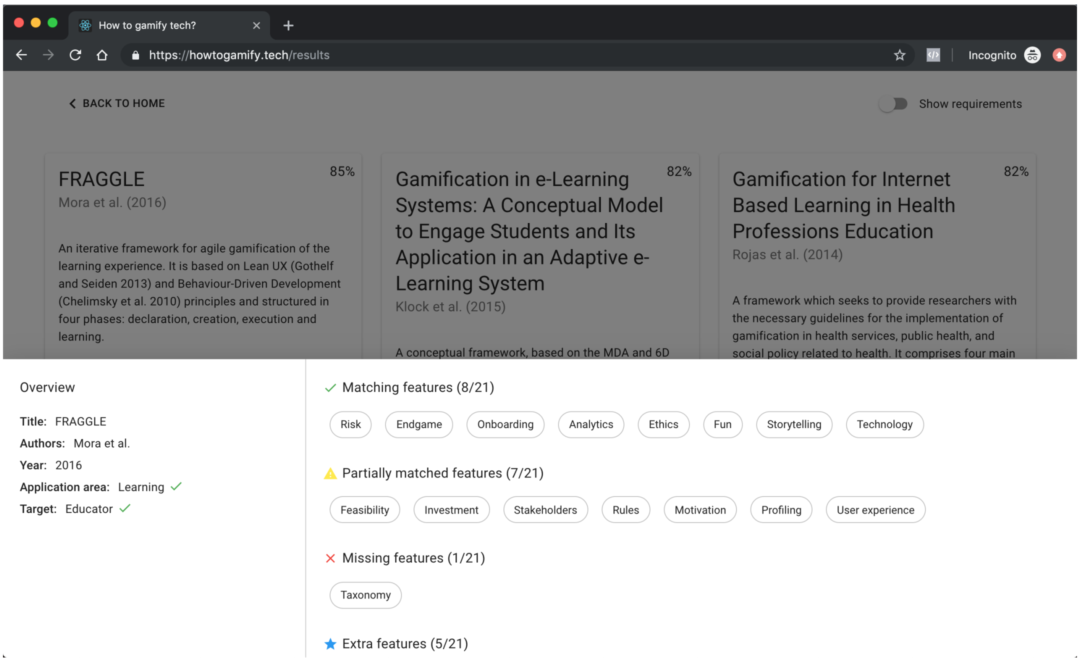
A conceptual framework, based on the MDA and 6D frameworks (Werbach and Hunter 2012), for engaging students, and its application in adaptive e-learning systems. By the definition of a set of dimensions, it provides information on what game elements, actors, data and behaviour should be involved in the gamification process.

Gamification for Internet Based Learning in Health Professions Education 82%
Rojas et al. (2014)

A framework which seeks to provide researchers with the necessary guidelines for the implementation of gamification in health services, public health, and social policy related to health. It comprises four main stages: theory and modelling, piloting, evaluation, and implementation.

READ DETAILS

FIGURE B.7: User interface: Details drawer



How to gamify tech?

https://howtogamify.tech/results

Incognito

< BACK TO HOME

Show requirements

FRAGGLE 85%
Mora et al. (2016)

An iterative framework for agile gamification of the learning experience. It is based on Lean UX (Gothelf and Seiden 2013) and Behaviour-Driven Development (Chelimsky et al. 2010) principles and structured in four phases: declaration, creation, execution and learning.

Gamification in e-Learning Systems: A Conceptual Model to Engage Students and Its Application in an Adaptive e-Learning System 82%
Klock et al. (2015)

A conceptual framework, based on the MDA and 6D

Gamification for Internet Based Learning in Health Professions Education 82%
Rojas et al. (2014)

A framework which seeks to provide researchers with the necessary guidelines for the implementation of gamification in health services, public health, and social policy related to health. It comprises four main

Overview

Title: FRAGGLE
Authors: Mora et al.
Year: 2016
Application area: Learning ✓
Target: Educator ✓

✓ Matching features (8/21)

Risk · Endgame · Onboarding · Analytics · Ethics · Fun · Storytelling · Technology

⚠ Partially matched features (7/21)

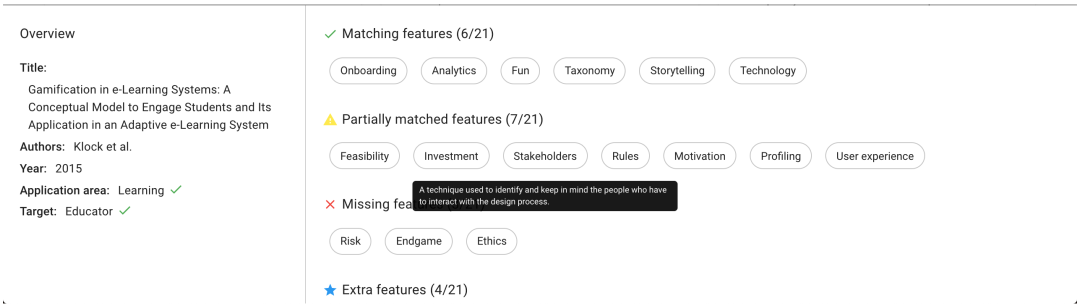
Feasibility · Investment · Stakeholders · Rules · Motivation · Profiling · User experience

✕ Missing features (1/21)

Taxonomy

★ Extra features (5/21)

FIGURE B.8: User interface: Tooltips



Overview

Title:
Gamification in e-Learning Systems: A Conceptual Model to Engage Students and Its Application in an Adaptive e-Learning System
Authors: Klock et al.
Year: 2015
Application area: Learning ✓
Target: Educator ✓

✓ Matching features (6/21)

Onboarding · Analytics · Fun · Taxonomy · Storytelling · Technology

⚠ Partially matched features (7/21)

Feasibility · Investment · Stakeholders · Rules · Motivation · Profiling · User experience

✕ Missing fea

A technique used to identify and keep in mind the people who have to interact with the design process.

Risk · Endgame · Ethics

★ Extra features (4/21)

References

[1] Sebastian Deterding et al. "Gamification. Using game-design elements in non-gaming contexts". In: *CHI'11 extended abstracts on human factors in computing systems*. ACM. 2011, pp. 2425–2428.

[2] Juho Hamari, Jonna Koivisto, and Harri Sarsa. "Does gamification work?–a literature review of empirical studies on gamification". In: *2014 47th Hawaii international conference on system sciences (HICSS)*. IEEE. 2014, pp. 3025–3034.

[3] Katie Seaborn and Deborah I Fels. "Gamification in theory and action: A survey". In: *International Journal of human-computer studies* 74 (2015), pp. 14–31.

[4] Brian Burke. *Gamify: How Gamification Motivates People to Do Extraordinary Things*. Bibliomotion, Incorporated, 2014. ISBN: 9781937134877.

[5] Alberto Mora et al. "A literature review of gamification design frameworks". In: *2015 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*. IEEE. 2015, pp. 1–8.

[6] Omar AZOUZ and Youssef LEFDAOUI. "Gamification design frameworks: a systematic mapping study". In: *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE. 2018, pp. 1–9.

[7] Alaa AlMarshedi, Vanissa Wanick Vieira, Ashokkumar Ranchhod, et al. "SGI: a framework for increasing the sustainability of gamification impact". In: *International Journal for Infonomics* 8.1/2 (2015), pp. 1044–1052.

[8] Alberto Mora et al. "Gamification: a systematic review of design frameworks". In: *Journal of Computing in Higher Education* 29.3 (2017), pp. 516–548.

[9] M Lewis Saunders and Philip Lewis. "P. & thornhill, a.(2009)". In: *Research methods for business students* 4 (2009).

[10] Juhani Iivari and John R Venable. "Action research and design science research-seemingly similar but decisively dissimilar". In: (2009). http://aisel.aisnet.org/ecis2009/73.

[11] Alan Hevner and Samir Chatterjee. *Design research in information systems: theory and practice*. Vol. 22. Springer Science & Business Media, 2010.

[12] Ken Peffers et al. "A design science research methodology for information systems research". In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[13] Kent Beck and Erich Gamma. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

[14] Stefan Stieglitz et al. *Gamification*. Springer, 2017.

[15] Kuutti Julius and J Salo. "Designing gamification". In: *Marketing* (2013).

[16] *Beyond gamification: Architect- ing engagement through game design thinking*. https://www.slideshare.net/DiTommaso/beyond-gamification-architecting-engagement-through-game-design-thinking. [Online; Accessed 10/11/2018]. Nov. 2011.

[17] MJJM Versteeg. “Ethics & Gamification design: a moral framework for taking responsibility”. MA thesis. 2013.

[18] Cathie Marache-Francisco and Eric Brangier. “Process of gamification”. In: *Proceedings of the 6th Centric* (2013), pp. 126–131.

[19] Janaki Kumar. “Gamification at work: Designing engaging business software”. In: *International conference of design, user experience, and usability*. Springer. 2013, pp. 528–537.

[20] David Robinson and Victoria Bellotti. “A preliminary taxonomy of gamification elements for varying anticipated commitment”. In: *Proceedings CHI 2013 Workshop on Designing Gamification: Creating Gameful and Playful Experiences. ACM. Recuperado de https://goo. gl/hElEnm*. 2013.

[21] Deborah Gears and Karen Braun. “Gamification in business: Designing motivating solutions to problem situations”. In: *Proceedings of the CHI 2013 Gamification Workshop*. 2013.

[22] Andrés Francisco-Aparicio et al. “Gamification: analysis and application”. In: *New trends in interaction, virtual reality and modeling*. Springer, 2013, pp. 113–126.

[23] Richard M Ryan and Edward L Deci. “Intrinsic and extrinsic motivations: Classic definitions and new directions”. In: *Contemporary educational psychology* 25.1 (2000), pp. 54–67.

[24] Richard M Ryan and Edward L Deci. “Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being.” In: *American psychologist* 55.1 (2000), p. 68.

[25] Sebastian Deterding. “Situated motivational affordances of game elements: A conceptual model”. In: *Gamification: Using game design elements in non-gaming contexts, a workshop at CHI*. 2011.

[26] Mihaly Csikszentmihalyi and Isabella Selega Csikszentmihalyi. *Optimal experience: Psychological studies of flow in consciousness*. Cambridge university press, 1992.

[27] Richard Bartle. “Hearts, clubs, diamonds, spades: Players who suit MUDs”. In: *Journal of MUD research* 1.1 (1996), p. 19.

[28] Nick Yee. “Motivations for play in online games”. In: *CyberPsychology & behavior* 9.6 (2006), pp. 772–775.

[29] Gustavo F Tondello et al. “The gamification user types hexad scale”. In: *Proceedings of the 2016 annual symposium on computer-human interaction in play*. ACM. 2016, pp. 229–243.

[30] Nick Pelling. “The (short) prehistory of gamification”. In: *Funding Startups (& other impossibilities)* 9 (Aug. 2011). `https://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/`.

[31] B Merino de Paz. “Gamification: A tool to improve sustainability efforts”. PhD thesis. Ph. D. dissertation, 2013.

[32] Gabe Zichermann and Joselin Linder. *Game-based marketing: inspire customer loyalty through rewards, challenges, and contests*. John Wiley & Sons, 2010.

[33] Kai Huotari and Juho Hamari. “Defining gamification: a service marketing perspective”. In: *Proceeding of the 16th international academic MindTrek conference*. ACM. 2012, pp. 17–22.

[34] Robin Hunicke, Marc LeBlanc, and Robert Zubek. “MDA: A formal approach to game design and game research”. In: *Proceedings of the AAAI Workshop on Challenges in Game AI*. Vol. 4. 1. 2004, p. 1722.

[35] Patrick Wilson. “Situational relevance”. In: *Information storage and retrieval* 9.8 (1973), pp. 457–471.

[36] David H Rose and Anne Meyer. *Teaching every student in the digital age: Universal design for learning*. ERIC, 2002.

[37] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[38] James O Prochaska and Bess H Marcus. “The transtheoretical model: Applications to exercise.” In: (1994), pp. 161–180.

[39] Kevin Werbach and Dan Hunter. *For the win: How game thinking can revolutionize your business*. Wharton Digital Press, 2012.

[40] Andrzej Marczewski. *Gamification: a simple introduction*. Andrzej Marczewski, 2013.

[41] Andrzej Marczewski. “The Intrinsic Motivation RAMP”. In: (2013).

[42] Daniel Berdichevsky and Erik Neuenschwander. “Toward an ethics of persuasive technology”. In: *Communications of the ACM* 42.5 (1999), pp. 51–58.

[43] Brian J Fogg. “Persuasive technology: using computers to change what we think and do”. In: *Ubiquity* 2002.December (2002), p. 5.

[44] Yu-Kai Chou. “Octalysis: Complete gamification framework”. In: *Yu-Kai Chou & Gamification* (2013). https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/.

[45] Daniel H Pink. *Drive: The surprising truth about what motivates us*. Penguin, 2011.

[46] Larry L Constantine and Lucy AD Lockwood. *Software for use: a practical guide to the models and methods of usage-centered design*. Pearson Education, 1999.

[47] Steven Reiss. *Who am I?: 16 basic desires that motivate our actions define our persona*. Penguin, 2002.

[48] H Jacobs. “Gamification: A framework for the workplace”. In: *University of Liverpool, Liverpool* (2013).

[49] Carol Li. “Evaluation of a Theoretical Model for Gamification in Workplace IS context”. PhD thesis. University of British Columbia, 2014.

[50] Fred D Davis, Richard P Bagozzi, and Paul R Warshaw. “User acceptance of computer technology: a comparison of two theoretical models”. In: *Management science* 35.8 (1989), pp. 982–1003.

[51] William H Delone and Ephraim R McLean. “The DeLone and McLean model of information systems success: a ten-year update”. In: *Journal of management information systems* 19.4 (2003), pp. 9–30.

[52] N Kumar. “A framework for designing gamification in the enterprise”. In: *Infosys Labs Briefings* 11.3 (2013), pp. 8–13.

[53] S Jimenez. *Gamification Model Canvas*. https://www.gamasutra.com/blogs/SergioJimenez/20131106/204134/Gamification_Model_Canvas.php. [Online; Accessed 11/01/2019]. 2013.

[54] P Herzig. “Gamification as a Service: Conceptualization of a Generic Enterprise Gamification Platform. Dresden, Technische Universität Dresden”. PhD thesis. Ph. D. Dissertation, 2014.

[55] Jorge Simões, Rebeca DíAz Redondo, and Ana FernáNdez Vilas. “A social gamification framework for a K-6 learning platform”. In: *Computers in Human Behavior* 29.2 (2013), pp. 345–353.

[56] Fiona Fui-Hoon Nah et al. “Gamification of education using computer games”. In: *International Conference on Human Interface and the Management of Information*. Springer. 2013, pp. 99–107.

[57] Oscar Wongso, Yusep Rosmansyah, and Yoanes Bandung. “Gamification framework model, based on social engagement in e-learning 2.0”. In: *2014 2nd International Conference on Technology, Informatics, Management, Engineering & Environment*. IEEE. 2014, pp. 10–14.

[58] Alberto Mora et al. “Fraggle: a framework for agile gamification of learning experiences”. In: *International Conference on Games and Learning Alliance*. Springer. 2015, pp. 530–539.

[59] Isabella Kotini and Sofia Tzelepi. “A gamification-based framework for developing learning activities of computational thinking”. In: *Gamification in Education and Business*. Springer, 2015, pp. 219–252.

[60] Ana Carolina Tomé Klock et al. “Gamification in e-learning systems: A conceptual model to engage students and its application in an adaptive e-learning system”. In: *International Conference on Learning and Collaboration Technologies*. Springer. 2015, pp. 595–607.

[61] Dennis L Kappen and Lennart E Nacke. “The kaleidoscope of effective gamification: deconstructing gamification in business applications”. In: *Proceedings of the First International Conference on Gameful Design, Research, and Applications*. ACM. 2013, pp. 119–122.

[62] Yu-kai Chou. *Actionable gamification: Beyond points, badges, and leaderboards*. Octalysis Group, 2015.

[63] Zachary J Fitz-Walter. “Achievement unlocked: Investigating the design of effective gamification experiences for mobile applications and devices”. PhD thesis. Queensland University of Technology, 2015.

[64] Marigo Raftopoulos. “Towards gamification transparency: A conceptual framework for the development of responsible gamified enterprise systems”. In: *Journal of Gaming & Virtual Worlds* 6.2 (2014), pp. 159–178.

[65] Johannes Harms et al. “Gamification of online surveys: conceptual foundations and a design process based on the MDA framework”. In: *Proceedings of the 8th Nordic conference on human-computer interaction: Fun, fast, foundational*. ACM. 2014, pp. 565–568.

[66] Basanth Kumar Neeli. “Gamification in the enterprise: Differences from consumer market, implications, and a method to manage them”. In: *Gamification in education and business*. Springer, 2015, pp. 489–511.

[67] Jailson Brito, Vaninha Vieira, and Adolfo Duran. “Towards a framework for gamification design on crowdsourcing systems: the GAME approach”. In: *2015 12th International Conference on Information Technology-New Generations*. IEEE. 2015, pp. 445–450.

[68] Umar Ruhi. "Level up your strategy: Towards a descriptive framework for meaningful enterprise gamification". In: (2015).

[69] Karen Robson et al. "Is it all a game? Understanding the principles of gamification". In: *Business Horizons* 58.4 (2015), pp. 411–420.

[70] David Rojas, Bill Kapralos, and Adam Dubrowski. "Gamification for internet based learning in health professions education". In: *2014 IEEE 14th International Conference on Advanced Learning Technologies*. IEEE. 2014, pp. 281–282.

[71] D Charles and Suzanne McDonough. "A participatory design framework for the gamification of rehabilitation systems". In: *Virtual Reality & Associated Technologies Gothenburg. Hauppauge: Nova Science Publishers* (2014), pp. 2–9.

[72] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI Blog* 1 (2019), p. 8.

[73] websiteoptimization.com. *The Psychology of Web Performance - how slow response times affect user psychology*. http://www.websiteoptimization.com/speed/tweak/psychology-web-performance/. [Online; Accessed 31/05/2019]. May 2008.

[74] P Bryan. *JSON Reference*. RFC draft-pbryan-zyp-json-ref-03. RFC Editor, Mar. 2013, pp. 1–5. URL: https://tools.ietf.org/html/draft-pbryan-zyp-json-ref-03.

[75] *Official Google Webmaster Central Blog: HTTPS as a ranking signal*. https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html. [Online; Accessed 20/05/2019]. Aug. 2014.

[76] Francesco Ricci, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook". In: *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[77] Chris Jackson. *The perils of using Internet Explorer as your default browser*. https://techcommunity.microsoft.com/t5/Windows-IT-Pro-Blog/The-perils-of-using-Internet-Explorer-as-your-default-browser/ba-p/331732. [Online; Accessed 02/06/2019]. June 2019.