SUMMARY

This paper researches how to design and develop an Zero UI-based open-source artifact utilising AI for controlling presentations through the use of gesture recognition. The need for touch-less approaches in order to mitigate the spread of disease, became a necessity during the COVID-19 pandemic. The risks of future pandemics, emphasise the continuous need for additional research into Zero UI.

We aim to answer three research questions:

- **RQ1:** How can open-source AI technology, specifically Mediapipe, be utilised to create a Zero UI-based sustainable gesture-based artifact for controlling presentations?
- **RQ2:** Which methods can be utilised to focus on a user-friendly approach, and how can they be used to establish the needed functionality and a set of user-friendly gestures for presentation control?
- **RQ3:** Which important software quality principles should be incorporated into a gesture-based presentation control artifact to ensure the development of a usable, efficient and effective application?

In order to answer these research questions, we utilise the Design Science Research Methodology, as the structure for the paper, in order to provide a focus on both the *utility* and *justified theory* through its six phases. We identify the problem, through the use of a focus group study, in order to determine the relevant functionality, based on the statements from five experts. Based on these findings and our findings from a previous study, we engineer a set of requirements based on the recommendations provided by Sommerville and the ISO 25010:2011 standard, more specifically the *Quality in Use* and *Product Quality* models. These are furthermore used in the design and development phase, to provide a set of software quality metrics utilised to develop a sustainable and usable artifact.

Moreover, the ISO criteria is used as metrics for evaluating the artifact, through usability and performance tests. In these test we utilise a Convolutional Neural Network, developed through transfer learning using the MediaPipe framework, to develop a model capable of performing pose recognition. Combined with the model, we employ a Finite State Machine, in order to perform gesture recognition.

We conducted a performance test, based on 1198 images across three different hardware setups. We found that the average time used to perform recognition ranged from 10.99 to 62.87 ms, which resulted in an average FPS between 15.91 and 90.99, depending on the hardware setup. Moreover we employed the *functional sustainability* criteria, and achieved an average accuracy of 99.68%, a recall score of 98.00%, a precision score of 98.08% and an F_1 score of 98.04%. Moreover, the test was used in order to evaluate the non-functional requirements, are based on the ISO 25010:2011 criteria.

In order to evaluate our requirements, we utilised two usability tests, based on the criteria presented in the ISO 25010:2011 *Quality in Use* model. It presents three requirements, *effectiveness* relating to the accuracy and completeness of the application, which help users achieve certain goals. This criteria was evaluated through the amount of tasks, which a user managed to complete, based on the recommendations made by Lazar. *Efficiency*, pertaining to the resources used in order to achieve the goals. We evaluate this criteria by timing each task conducted by the participants in the usability test. Lastly we measured the participants *satisfaction* through the use of a modified CSUQ questionnaire and semi-structured interviews. Furthermore we utilised the usability test as a means of evaluating our requirements. The tasks presented to the participants, were devised in a way, that required the user to employ the functionality described in the functional requirements, in order to complete the usability test.

Through iterative processes utilising the methods presented above, we procured an artifact. It is based on the *Distributed Data Client-Server* pattern, which ensures the *portability*, as it allows for utilising the application within any browser. Furthermore it helps to ensure the development of a sustainable artifact, as it requires a minimum of hardware, and can be utilised on already existing computers. The software-based approach furthermore support this sustainable approach, as it does not require additional peripherals to perform gesture recognition.

This leads to the conclusion that software-based models are capable of providing a sustainable and usable artifact, through the utilisation of the Design Science Research Methodology and the ISO 25010:2011 quality metrics.

Designing a Zero-UI Application Utilising AI for Gesture-Based Presentation Control

Casper Bisgaard Kristensen
Student, Aalborg University
cbkr19@student.aau.dk

Lasse Beck
Student, Aalborg University
lbeck18@student.aau.dk

***Abstract*—This paper presents the development of a Zero UI-based application[1] designed to control presentations through hand gestures. By utilizing the Design Science Research Methodology, we systematically structure the design and development phases. We leverage recent advancements in artificial intelligence, particularly Google's MediaPipe and a Convolutional Neural Network developed based on transfer learning, in order to recognise hand poses. The model obtains an average accuracy score of *99.68%*, a precision score of 98.08%, a recall of 98.00% and an F_1 score, of 98.04%. Additionally, we employ a Finite State Machine in order to perform gesture recognition. Our aim is to create a sustainable and usable artifact, guided by the software quality criteria outlined in the ISO 25010:2011 standard. We address the increasing demand for touch-less interfaces in the post-COVID-19 era, by creating a software-based prototype evaluated based on usability and performance testing. We evaluate it based on the Quality in Use and Product Quality models of the ISO 25010:2011 standard. The results underline the artifact's ability to providing an efficient, effective and usable approach to conduct touch-less presentations.**

***Index Terms*—ISO 25010:2011, Hand recognition, MediaPipe, Quality metrics, Zero UI, Accuracy, Recall, Precision, F_1 Score, CSUQ, Design Science Research, Usability testing, Focus Group Study**

I. Introduction

The field of Artificial Intelligence (AI) has undergone significant advancements, since the term was first established by McCarthy et al. in 1955 [1]. These advancements has led to an increasing integration of AI-based technology in society [2]. AI-based innovations like ChatGPT [3], Neurallink [4] and DeepMind [5] continuously emerge, leading to forecasts of market growth upwards of 2000% by 2030 [6]. One area of increasing interest, is Zero UI, which centers on interfacing with devices through input modalities such as gestures, voice commands and facial recognition [7]. Technologies such as Siri [8], Microsoft Kinect [9] and Face ID [10], have provided revolutionary user interfaces in this regard.

The COVID-19 pandemic shifted the Zero UI trend from a luxury to a necessity, as touch-less interfaces became vital in mitigating the spread of pathogens [7]. This accelerated the adoption of Zero UI-based interfaces, with companies like Coca Cola and KFC introducing gesture controlled vending machines [11], AirAsia [12] and AirEmirates [13] implemented Zero UI-based check-ins and schools started transitioning to touch-less smart boards [14]. Despite the end of the pandemic, these technologies remain of vital importance, as we are likely to see more pandemics in the future, due to increases in areas such as travel and urbanisation [15].

However, progressing towards a more Zero UI-based society, incurs *environmental* and *economic* challenges. In terms of the environmental impact, estimates from the World Health Organization show that the electronic waste (e-waste) production exceeded 53 tonnes in 2019, making it the fastest growing form of solid waste [16]. This provides a challenge, given the scale of the Zero UI transition. For instance, redesigning and maintaining the classrooms in the more than 100.000 primary and secondary schools in the United States alone [17], would require large amounts of materials to produce the necessary technology, and result in significant economic expenses. This emphasizes the necessity for a sustainable process, and furthermore emphasize the importance of conducting additional research into Zero UI-based technology.

This paper presents an AI-based open-source artifact [2], aimed at providing a more sustainable solution to the aforementioned challenges, while ensuring a seamless transition to a Zero UI-based educational system. Therefore we aim to explore three key aspects, *sustainability*, *usability* and *software quality principles*, through the development of an artifact for gesture-based control in presentations. In the following we present a brief overview of the division of the paper.

Following this introduction, section II presents an overview of the current state of the research within the area of gesture recognition for presentation control in subsection II-A. We furthermore present our previous findings, based on a comparative analysis of the existing open-source hand recognition models [19], evaluated using the software quality metrics from the ISO 25010:2011 SQuaRE standard [20]. These findings indicated that Google's MediaPipe framework [21] significantly outperformed the other models, and was utilized to develop an initial prototype. The prototype was employed in order to gather user feedback through a usability

[1] https://github.com/Xamalf/P10Project

[2] Artifact definition: *"Any designed object with an embedded solution to an understood research problem" [18]*

test, collecting both quantitative and qualitative data. These findings provide the baseline for the research conducted in this paper.

In section III, we present the principles, practices and procedures utilised as the methodical approach to develop our proposed artifact. To provide a structure for the process, we draw inspiration from the well-recognized Design Science Research Methodology (DSRM) [18], as it provides a framework for procuring artifacts [22]. It divides the process of procuring an artifact into six phases [18]. We divide our methodology and findings sections accordingly, inspired by Lapão et al. [23], Silva et al. [24], and Berkhout et al. [25]. We then utilize the ISO 25010 and other relevant methods, in order to ensure the utilization of software quality metrics throughout the DSRM phases. A visual illustration of the structure, is presented in Figure 1. Following these sections, we evaluate our contribution based on experiments conducted on prototypes of the artifact. We then discuss our findings, and compare them to the existing body of literature. Finally, we conclude the paper by summarising our findings, and evaluate on whether we have answered the following research questions, that will be used to guide our inquiry into the development of our artifact.

A. Research Questions

- **RQ1:** How can open-source AI technology, specifically Mediapipe, be utilised to create a Zero UI-based sustainable gesture-based artifact for controlling presentations?
- **RQ2:** Which methods can be utilised to focus on a user-friendly approach, and how can they be used to establish the needed functionality and a set of user-friendly gestures for presentation control?
- **RQ3:** Which important software quality principles should be incorporated into a gesture-based presentation control artifact to ensure the development of a usable, efficient and effective application?

Following we introduce the current state of the art, within the field of gesture based presentation control.

II. Background

This section presents an overview of the contributions within the field of gesture recognition, specifically focusing on its utilisation as a modality for controlling presentations. Furthermore we introduce our previous research within the field.

A. Related works

This section presents the contributions within the field of gesture recognition in regards to the topic of *presentation control*. The research conducted in this area is sparse, we present the series of queries that was utilised in order to discover the contents of this section subsection II-A.

Search query	Hits
hand AND gesture AND presentation AND control	104
gesture AND recognition AND presentation AND control	89
hand AND gesture AND powerpoint	12
touch AND less AND presentation AND control	16
mediapipe AND powerpoint	2
mediapipe AND presentation AND control	2

TABLE I
Search Query Hits on Scopus [26]

Based on these searches, we have inquired information from 39 papers. Initially, papers contributing to the field relied solely on hardware peripherals such as gloves and infrared cameras. However, due to the advancements within the area of artificial intelligence, newer researcher often apply purely software-based approaches. Therefore we categorise the area of research within presentations into two types, *software-based* and *peripheral-based* contributions. As technology within this field changes rapidly, we only present an overview of applications published since 2020. We start by evaluating the software-based solutions in the following section

1) Software-based papers: In the course of inquiry, we have discovered three tendencies within the literature. One group evaluate their papers solely on the *accuracy* metric. Another group utilises the *accuracy*, *recall*, *precision* and F_1 metrics in reporting their findings. The last group, presents no metrics at all, and has therefore not been included, since this there is no option of evaluating our artifacts against these. We present a selection of papers employing either *accuracy*, *recall*, *precision* and F_1 *score* in the following.

Islam et al. presents a solution capable of recognizing the number of stretched fingers, thereby providing an overall of 5 gestures. They use these finger counts in combination with the movement direction of the hand to activate actions like next slide, volume up or play. They achieve an overall accuracy of 97.8% [27]. The paper does not present any metrics depicting the resource consumption, in terms of CPU, memory or GPU.

Sathish et al. presents a model recognising 11 gestures, using OpenCV for image extraction and MediaPipe for hand landmark detection. They present accuracy for each gesture with results between 91-100% [28]. No metrics in regard to resource or time consumption are presented.

Idrees et al. presented a solution, utilizing a neural network, based on an LSTM approach. Their model is trained on 35373 videos from the 20BN-jester dataset, which they evaluate based on accuracy, and obtain an score between

98-100% depending on the gesture. Their application is based on a Python implementation, utilizing the OpenCV library [29]. The paper does not mention any resource or time consumption metrics.

Ahamed et al. presents their contribution to the field, through a solution which employs a transfer learning algorithm. Their application is implemented with Python using the OpenCV library. It is capable of recognizing five separate gestures, and obtains an accuracy between 80 and 100% depending on the gesture. Noteably, the model is only tested on 15 examples. Moreover, the contribution is absent in terms of information regarding the dataset utilised to train and test the model [30]. Furthermore, no information in regards to resource or time consumption is presented.

The other group of papers papers, which include accuracy, recall, precision and F_1 score are presented in the following. Osama et al. provides a CNN based on YOLO v3 and DarkNet-53, capable of recognizing 6 poses, trained on the Massey University and HUST-ASL datasets. It achieves an accuracy of 97.6%, a precision of 94.88%, a recall of 98.66% and an F_1 score of 96.70% [31]. The implementation is based on Python and Django, and according to the paper it requires an i5 / AMD Ryzen 5, 8 GB of RAM and 256 GB SSD to run the application. They furthermore run a usability test, with 3 participants, testing each gesture 20 times. No metrics in regard to time or resource consumption are presented in the paper.

Setiawan et al compares five different methods, KNN, SVM, Decision Trees, LDA and Random Forest in order to recognize 5 poses. They present accuracy, recall, precision and F_1 score [32]. The KNN achieves an accuracy of 100%, a precision between 99-100%, a recall of 98-100% and a F_1 score between 99-100%. Their SVM achieves 100% in all four categories, the same goes for both the decision tree and LDA. They mention having trained the model utilizing 10 second video, but there is no mention of which dataset that is applied, for training nor testing purposes. The paper does not provide metrics in terms of resource or time consumption.

Orovwode et al. presents a solution based on MediaPipe in order to achieve landmarks, and an LSTM capable of recognizing 15 gestures. The model is trained on the HaGRID dataset, using 552992 images, running 200 epochs. They achieve an *accuracy* of 90%, a *precision* of 92%, a *recall* of 90% and a F_1 *score* of 91%. They train the application using Python and TensorFlow [33]. No metrics in terms of the resource consumption or time used to perform the predictions are presented in the paper.

This concludes our *software-based* section.

2) Hardware-based solutions: Huo et al. presents a solution, which utilizes the LeapMotion controller [34]. They present a comparison of three different models, a *Support Vector Machine*, *K-Nearest Neighbor*, and a *Deep Neural Network*, trained on a dataset of 1600 customized images [35]. No information is presented, in regards to which dataset. The application is furthermore created with OpenCV in Python, and is capable of running on an Intel i5 CPU (1.6 GHz) and 16 GB RAM. For the KNN, they achieve an *accuracy* between 96.4-100%, the *recall* is between 97-100%, the *precision* is between 88.68-100% and the F_1 score between 91.26-100% [35]. For the SVM, they achieve an *accuracy* between 95.2-99.8%, the *recall* is between 87-100%, the *precision* is between 88.78-100% and the F_1 score between 87.88-99.50% [35]. For the DNN, they achieve an *accuracy* between 90.8-99.2%, the *recall* is between 89.75-99.75%, the *precision* is between 70.7-98.94% and the F_1 score between 71.95-98.02% [35].

Yang et al. presents a two-layer Bidirectional Recurrent Neural Network, which also utilizes a LeapMotion controller. They utilize the *ASL* and *Handicraft-Gesture* datasets, in order to obtain an *accuracy* of 95.238%, a *precision* of 95.546%, a *recall* of 95.238% and a F_1 *score* of 95.274% [36].

This concludes our review of the different solutions related to presentation control. The following selection presents findings from our previous paper.

3) Our previous research: In this section we present our findings from a previous study [19], based on a comparative analysis performed in order to obtain benchmarks of the open-source hand recognition frameworks. To develop the comparative analysis, we utilised a well-established software quality model, the ISO/IEC 25010:2011 SQuaRE standard [20], as presented in Figure 3. We furthermore used it as the guideline for developing evaluation metrics for performance and usability testing. We applied these for our initial prototype. The following provided the reached conclusions, which was utilised as the foundation for the development of our research questions, as mentioned in subsection I-A.

Our findings established MediaPipe as the best performing framework. Furthermore we extracted a series of issues based on the data gathered during the usability tests [19].

- Static poses: Users pointed out that static poses felt unnatural to perform, especially when doing tasks for longer periods of time. Therefore they requested dynamic gestures instead.
- Feedback: Users requested feedback from the system in regards to whether their hand as well as the pose they attempted to perform, was recognized by the application.
- Pointer: The pointer had an issue, where it would sometimes jump to a different set of coordinate on the screen, just for a split second and then return to its original location, which was pointed out by users.

We utilised these findings in order to form the baseline from which we generated our set of research questions, as presented in subsection I-A. In the following section, we present the methodology which lies the foundation for the structure of developing our artifact and conducting our tests as a means of evaluation.

III. Methodology

This section outlines the rationale behind the methodological approaches adopted throughout the process of procuring the artifact central to this paper. We commence by providing a more detailed explanation on the concept of artifacts. Subsequently, we present the Design Science Research Methodology (DSRM) [18], and how we utilize it as the structure for the overall process of developing an artifact. Furthermore, the ISO 25010:2011 standard [20] is introduced, alongside an elaboration on how it has been employed as an evaluation metric throughout the process.

A. What are artifacts?

In section I, we provided an initial definition of an artifact. According to Hevner et al. artifacts can be *software* such as *prototypes*, *formal logic*, *mathematics* or *natural language descriptions* [22]. They furthermore state that an artifact can be constructed of various parts [22]. These include *concepts*, the language used to describe the domain. The *models* used to represent real world concepts such as requirements. The *methods* employed to define the processes utilised to determine a solution. And lastly the *implementation* of these *concepts*, *models* and *methods* in a working system [22]. Due to the multifaceted nature of artifacts, their creation requires alternating between *building* the *implementation*, and *evaluating* it based on *models* and *methods* [22], providing an iterative cycle that seeks to develop efficient and effective artifacts [22]. After each of these iterative phases, the artifact is evaluated based on its ability to solve the given problem. The experiments and findings obtained through evaluating the *utility*, are used to devise new *justified theories*, that if needed, provide the baseline for a new iteration [22]. This allows for research to be *"evaluated in light of its practical implications"* [22], in an attempt to develop an artifact that satisfies user needs. In order to apply a similar approach, we draw upon the the well-acknowledged Design Science Research Methodology, and utilize it for structuring the process of procuring our artifact [18].

DSRM proposes to divide the process of developing an artifact into 6 activities, *Identify Problem & Motivate*, *Define Objectives of a Solution*, *Design & Development*, *Demonstration*, *Evaluation* and *Communication* [18]. We utilize the first 5 phases to determine the steps in our development process life cycle. As the *communication* phase focus on communicating the problem to *researchers and other practicing professionals*, which is done through this paper, therefore we do not focus on it throughout the rest of the paper. We build upon the examples of other studies [23], [24], [25] in this regard, and structure the rest of the methodology and the following findings section based on the DSRM phases. This provides us with the ability to apply the DSRM structure, and thereby focusing on both the *justified theory* and the *utility* as mentioned in section I. Furthermore, utilising the DSRM approach helps retaining an overview of each of these *building* and *development* phases individually [22].

Within each of the DSRM phases, we employ the ISO standard and its *Quality in Use* and *Product Quality* models [20]. The *Quality in Use* model focus on the *utility* paradigm, and presents criteria in regard to usability [20]. The *Product Quality* model presents a series of software quality criteria, which focus on the *justified theory* paradigm [22], by serving as design guidelines and evaluation metrics [20]. The ISO 25010:2011 are used as the overall guideline for determining how each of the 5 phases should ensure upholding relevant software quality criteria. To provide an overview of how the ISO 25010 criteria are utilised in the five DSRM activities, we present an outline of the project structure is presented in Figure 1.

The subsequent sections, provide an overview of the methods employed in each of the six phases illustrated in Figure 1. The outcome obtained from applying these methods, can be found in section IV.

B. Activity 1: Problem identification and motivation

This activity seeks to identify the research problem, and provide arguments justifying the impact provided by the suggested solution [18]. We build on the research baseline established in section I, by collecting additional qualitative user preferences data on topics such as *system usage*, *functionality* and *gesture preferences* [37, p. 121-151].

We use a focus group study in order to do determine the participants *functionality* and *system usage* preferences, due to its inherent ability to *identify problems*, *establish user needs* and *generate new design concepts* [38]. In order to have a rigorous process, we follow the recommendations set forth by Kontio et al [39]. They propose to divide the focus group study into a series of four phases, based on a survey conducted on the general approaches utilised in information systems literature [39]. We present the first 3 phases in the following. The fourth phase *Conduct the focus group session* is presented as part of our findings in section IV.

1) Definition of the research problem: Our primary interest is to obtain a better understanding of the application domain. More specifically, which functionality users would like to have within a presentation application, and their preferences in terms of the usage and functionality.

2) Partitipant Selection: We conducted the study on a group of participants capable of acting as both experts and users for our artifact. In order to capture the different participants viewpoints, we utilised a sample size of 5 [39]. The group of participants, consisted of lecturers from the

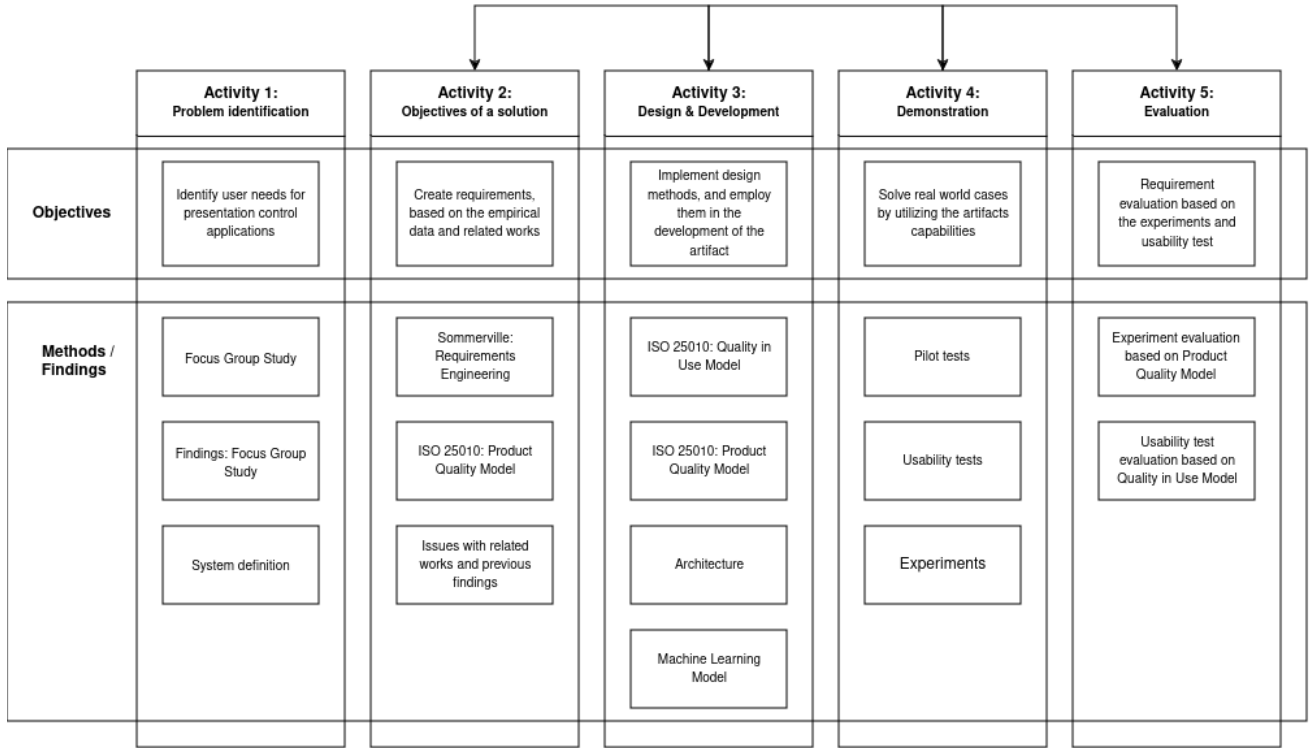

Fig. 1. The structure of our project based on the five DSRM phases

department of Human Centered Computering at Aalborg University alongside former high school teacher, with an additional degree in software engineering.

3) Plan the focus group event: The structure of our focus group event, is based on the recommendations of McDonagh et al [38]. The focus group session was conducted by a moderator who functioned as a facilitator, guiding the conversation and probing into relevant comments [38]. We furthermore have an assistant who notes down the key points during the session. The event is scheduled to last 2 hours and 30 minutes [39], in order to provide the basis for relevant topics to be discussed thoroughly.

We started with a warm-up session, in order to provide a context for the participants. We commenced by giving a short verbal introduction to the session, followed by a presentation of the presentation control prototype we previously devised. We continued with a series of warm up questions, inspired the *Day-in-the-Life* exercise [38]. Questions revolved around how the participants current habits and preferences when they teach. After the initial questions, we continued with an idea generating phase, to *"identify problems and establish user needs"* [38]. The participants were asked a series of questions related to their preferences in terms of functionality. The moderator initiated the session by asking questions, which were answered by each participant in sequence. This was followed by a group discussion in an attempt to create a collective *brainstorm*, among the participants, with the objective of having the participants build on each others ideas [38].

Afterwards we utilize another idea generation tool, a technique McDonagh et al describes as *drawing* [38]. The experts were presented with a piece of paper depicting an empty browser window, and were asked to draw a mock-up of their ideal presentation website. Afterwards, we engaged in a discussion, based on their mock-ups. Each user presented their mock-up, which lead to a discussion for each. This discussion on conflicting views, allowed the experts to discuss their arguments and exchange ideas, leading to a range of insights that might not have emerge during an interview. Lastly, we conducted an evaluation [38], discussing the features of each participant's mock-up, and explored the reasons behind their choices. This furthermore provided an opportunity to address potential challenges and limitations. We concluded the session by inviting the participants to share any final comments. This concludes the presentation of the phases employed to conduct the focus group study.

In order to obtain *gesture preferences*, we build upon the research conducted by Hosseini et al [40]. Their research attempts to achieve a consensus in terms of gestures, and provide a series of suggestions for the most preferred gestures for perform various tasks using gestures, such as *start*, *next*, and *undo* to mention a few [40]. We use these gesture recommendations, as a starting point for developing our initial set of gestures for the artifact. The gathered data in the aforementioned activities combined with our previous findings, are utilised in the following activity, in order to provide a set of requirements.

C. Activity 2: Defining objectives of a solution

This step is used to determine the objectives of the artifact. According to Peffers [18], the objectives can be both quantitative and qualitative. We provide these objectives through a set of requirements, as Hevner et al. describes an *"artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve"* [22].

In order to perform the requirements engineering in a rigorous way, we utilize the process recommended by Sommerville [41]. This process produce two types of requirements, *functional requirements*, which are detailed descriptions of the artifacts functionality, and *non-functional requirements*, these focus on the set of characteristics which the artifact as a whole should fulfill [41]. Sommerville presents three key activities for generating requirements, *discovery*, *specification* and *validation* [41].

1) Discovering requirements: This activity consists of two steps *elicitation* and *analysis*, which requires interaction with the stakeholders, followed by an analysis of the findings, in order to provide an initial set of unstructured requirements. We obtain the data from the aforementioned focus group study, alongside previous findings presented in section II [41].

2) Specifying requirements: The next step in engineering requirements, is determining how to create the requirements specification. This is done based on the requirements generated in the previous section. According to Sommerville, the most utilized approach is natural language, which can be described to its *expressiveness, intuitiveness and universality* [41].

3) Validating requirements: The last step within the requirements engineering, is used to validate whether the requirements satisfies the users needs. The selected methods for this evaluation, is elaborated further in the DSRM evaluation phase, presented in subsection IV-F.

This concludes our approach for requirements engineering, with it our attempt to definite the objectives for our artifact. In the following section we present the methods utilised in the design and development phase, which focus on a transformation of the requirements, into the *models* and *methods*, and the subsequent *implementation* hereof.

D. Activity 3: Design & development

This activity aims to *describe the artifact* [18], and introduce the methods used in the design and development phases. Here in the methodology section, we present the design phase, where as the development phase is presented in section IV.

According to Mathiassen et al. *"a good design balanced several criteria"* [37, p. 181]. Therefore we select a subset of the criteria presented in the ISO 25010 *Product Quality* and *Quality in Use* models [20]. In the following, we present our rationale, in regard to our choice of criteria. An overview of these criteria and a brief explanation of each, can be found in Figure 3. We employed the categorization tool provided by Mathiassen et al. [37, p. 187], in order to determine a ranking of the ISO 25010:2011 criteria, in relation to our artifact. The results from this activity, can be seen in Figure 2. The following sections elaborates on how we aim to measure the selected criteria, either through a quantitative or qualitative approach.

Criterion	**5**	**4**	**3**	**2**	**1**
Performance efficiency	X				
Functional completeness	X				
Usability	X				
Portability		X			
Compatibility		X			
Maintainability				X	
Security					X
Reliability					X

Fig. 2. Prioritization of criteria, where 5 is very important, 4 is important, 3 is less important, 2 is irrelevant, and 1 is easily fulfilled [37, p. 187]

1) Performance efficiency: This criteria is described as the *"performance relative to the amount of resources used under stated conditions"*, in the ISO 25010:2011 standard [20]. We evaluate this criteria based on two metrics:

a) Time behaviour: This is a quantitative metric, based on the measurement of the response time and processing speed for the application [20].

b) Resource utilization: We utilize this metric in order to measure the computational resources, more specifically the CPU and memory resources consumed, when utilizing the application [20].

We utilise these criteria, to measure the artifacts performance, to determine the hardware setup it is capable of running on, without compromising the *usability*, *efficiency* and *effectiveness*, thereby also providing data in order to evaluate the *portability* and *compatibility*. In the following section, we present the *functional suitability* criteria.

2) Functional suitability: This metric pertains to *"the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions"* [20]. The ISO 25010:2011 standard splits this into three sub-criteria:

a) Functional Completeness: This metric relates to the performance of the application under altering conditions, such as different light settings, hand sizes and backgrounds [20]. This criteria was evaluated in [19], and therefore we do not further elaborate on this criteria in this paper.

b) Functional Correctness: This metric is used to assess the accuracy of the model, such as the correctness and precision. In order to evaluate our model, we utilize an F1 score, as it is a key metric in the evaluation of machine learning models, according to Rahman et al [42]. and Goutte et al [43]. The F1 score consists of two sub-metrics, precision and recall. Precision is used to measure how well the model performs recognition, whereas recall determines whether a specific element exists or not. These two scores are then used to calculate the F1 score, as shown in Equation 4, which results in a harmonic mean, between the two metrics, in order to obtain balance between the metrics, to create a more precise depiction of the models accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The accuracy is calculated as in Equation 1, where TP is the number of correctly identified poses and TN is the poses correctly identified as not being the target pose. FP indicates the number of poses, incorrectly predicted to be the target pose, and FN are the number of poses that were not identified as the target pose, but should have been predicted as such.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Equation 2 shows the formula used in order to calculate the models precision. In this case the true positives (TP) would be the correct recognition of specific poses within our model. The false positives (FP) would be poses in which our model falsely recognizes something to be a hand, which is not.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The formula presented in Equation 3 presents the notation for the formula used in calculating the recall. The TP is referring to the number of correct poses found, whereas the false negatives (FN) refers to the number of poses it did not find.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

The notation displayed in Equation 4 show how to calculate the F_1 score, which provides the harmonic mean between the *recall* and *precision* metrics.

c) Functional Appropriateness: Pertains to how *"a product provides the necessary and sufficient steps to complete a task"* [20]. We evaluate this based on our requirements, and the degree to which these have been fulfilled through or usability tests, and therefore we do not further elaborate on this criteria. This concludes the *functional sustainability* section. In the following we present the portability criteria.

3) Portability: To ensure a portable artifact, we develop our application as a static website. This approach omits the backend, and instead focus on client-side rendering of the application. We utilize the the *Distributed Data Client-Server* pattern [37, p. 202]. According to Mathiassen et al. this allows for distribution where both the UI, Function Component and Model Component are located on the client side, and part of the Model Component, a static file server, makes up the server side [37, p. 202]. This allows users to utilise the artifact, on any type of device, as long as it supports a Chromium based browser, thereby increasing the accessibility. This concludes the portability section. The following section presents the *security* criteria.

4) Security: To enhance user security, we employ the *Distributed Data Client-Server* pattern [37, p. 202]. As previously mentioned, this enables the artifact to operate entirely in a browser, necessitating only a static file server to provide the source file upon initially visiting the website. This architecture ensures that the users camera feed and uploaded files are confined to local storage on their computer. Consequently, the security criteria are inherently built into the design, based on this architectural choices.

5) Reliability: As we procure a prototype as our artifact, and not a production ready application, we do not focus on the reliability, as it is outside the scope of the research questions for this paper.

6) Modularity: As we present a prototype, which primary focus is to function as an evaluation of our research questions and requirements, we do not focus on this criteria.

7) Usability: The Quality in Use model is used to evaluate the degree to which the solution fulfills the needs of the users. It specifically emphasizes *effectiveness*, *efficiency*, *freedom from risk*, *context coverage* and *satisfaction*. We refrain from utilising the *freedom from risk* and *context coverage* criteria, as these focus on the applications ability to be used outside of the specified context, which is out of the scope of this paper. We utilize the ISO standard criteria, as means of answering RQ3. The following sections will elaborate on how we have utilized the focus group study and the ISO 25010 standard, alongside the DSR method in order to produce our artifact.

This concludes the ISO 25010:2011 criteria. In the following, we present our theory in regard to developing our machine learning model used to perform gesture recognition.

8) Machine Learning Model: We trained a Convolutional Neural Network (CNN) through using transfer learning, thereby fine-tuning the existing MediaPipe model. We utilized the [44] and [45] and our own custom dataset, split it into

Metric	Description
Performance efficiency	Pertains to the use of resources and performance
Functional suitability	A metric pertaining to well the system meets its objectives
Portability	The software's ability to adapt to various environments
Usability	Pertains to the usability of the system
Reliability	The systems ability to perform without failing during a period of time
Maintainability	Pertains to the ease of being able to modify the program
Compatibility	Pertains to the software's interoperability with external systems
Security	Pertains to the systems ability to safeguard information and maintain data integrity

Fig. 3. The 8 criteria for software quality in ISO 25010:2011

9 categories, one for each of the hand gestures which the model should be able to recognize. The training process was run over 100 epochs. This produces a TensorFlow Lite model, which we utilize in the *gestureRecognizer* in order to recognize the poses. The models performance metrics, can be seen in Figure 13.

9) Component diagram: Component diagrams are used to provide an overview of the structural relationship between the systems components [46]. We present a component diagram of the artifact in Figure 9.

10) State chart diagrams: State chart diagrams are used in order to depict the behavior and collaboration of the different modules within the artifact, which is useful to illustrate the finite state machine which our application utilises for gesture recognition [47]. The state chart diagram is depicted in Figure 8. In the following section, we present the methodology behind the *Quality in Use model* which focus on a series of criteria, related to usability.

E. Activity 4: Demonstration

This phase focus on solving one or more of the problems, utilizing the artifact, through various methods, we utilize usability tests and simulations [18]. This section presents the methods used, in order to plan both of these.

1) Simulations: We ran a series of tests, in order to determine the performance of our artifact. The test was run on 1200 images from the [48], [49] and [50] datasets. The test was run on 3 different laptops, presented in subsubsection III-E7. We timed the each of the tests, and furthermore used *top* in order to observe the browser tab and the Chrome GPU tab, in order to gather quantitative data on the *CPU*, *memory* and *GPU* usage of the application. The results of these simulations are presented in subsection IV-E

2) Usability test: In order to answer our research questions and evaluate our requirements, we utilize usability tests, in order to ensure inclusion of user preferences. Initially, we outline the methodology employed in our usability test. Subsequently, we account for our approach in collecting data. Thereafter, we introduce our rationale for the measures employed to mitigate the learning effect. The section is concluded through a presentation of the tasks carried out during the usability test.

We utilize the methodology proposed by Lazar, as the outline for our usability test [51]. This involves utilizing a high fidelity prototype, through a summative approach. We refrain from formative testing, as a low fidelity prototype does not provide the suitable means for an effective evaluation of the capabilities of an application relying on hand recognition [51].

We selected a sample consisting of the same six users, whom also participated in testing our local prototype. This was done in order to achieve consistency, and to be able to obtain their viewpoints on the altercations made based on their feedback, alongside enabling the possibility of achieving a comparison between the two systems, in terms of their opinions. The users participating in the test, all held prior experience with PowerPoint presentations, through their roles as teachers, teaching assistants or experience related to their respective university studies. This ensured that the participants had an adequate understanding of the functionality of a presentation tool, allowing us to evaluate their performance from a perspective related to how well our gestures made sense.

In order to maintain a controlled environment, including factors such as lighting and distance, we conducted the usability tests in the same room. Moreover we employed the test using the same hardware, to avoid inconsistency. The hardware setup utilized during the test, is presented in subsubsection III-E7.

We structured the task list for the usability test as a scenario, as delineated in III-E6, based on Lazar's methodology [51]. We ensured that it was developed in alignment with the criteria set forth in the ISO 25010:2011 standard, more specifically the *Quality in use model*. It contains three criteria, *effectiveness*, *efficiency* and *satisfaction*, which are utilized in order to evaluate the extend to which an application meets the users needs and expectations [20]. We measure these criteria, through the use of their equivalent metrics, *time*, *completion rate* and *user satisfaction*, as presented by Lazar [51].

3) Effectiveness: We utilize completion rate, in order to assess the effectiveness. We measure the quantity, in terms of the number of tasks that the user either completed or abandoned [51].

4) Efficiency: We use the time as a metric to assess the users efficiency, by recording the time a user takes until completion or abandonment of a given task [51].

5) Satisfaction: We evaluate *user satisfaction* based on our own modification of the questions presented in the *computer system usability questionnaire* (CSUQ), which is frequently utilized [52]. These are shown in Figure 23. This approach employs the Likert scale, ranking a series of statements on a scale from 1 (Strongly disagree) to 7 (Strongly agree), in regard to their user experience. Following, we conducted a semi-structured interview in order to obtain additional qualitative data, as an opportunity for the participants to further reflect and clarify upon their quantitative answers provided through the questionnaire. This is presented in the appendix A.

6) Usability tasks: An fundamental aspect for conducting an effective usability tests, is how closely it resembles real world scenarios [53]. To ensure this, participants are provided with a PowerPoint slideshow, see appendix C, consisting of a set of tasks designed to simulate various situations which the user could encounter during a presentation.

a) Task 1: For the initial task, the user is asked to turn on the program, by using the hand gestures to navigate from the off to the on state. The slides used for this task, are shown in Figure 26

b) Task 2: For the second task, the participants were required to deliver a presentation based on a set of topics:

- A movie or TV show which they prefer
- Memories from a vacation that they particularly enjoyed
- A hobby or interest that they enjoy spending time on

Furthermore, we instructed the user to gesticulate to a suitable degree. This aimed to assess whether our model would respond to unintended gestures. Moreover, the user was asked to navigate to the following slide. Figure 27 illustrates the slide utilized for this task.

c) Task 3: Participants were instructed to navigate through a maze shown on the associated PowerPoint, as illustrated on Figure 28 and Figure 29. By providing this task, we established a means of evaluating the power pointer in terms of maneuverability and responsiveness.

d) Task 4: The participant is tasked with switching from slide mode to video mode, and asked to play the video for 10 seconds, and then stop it, and go back to slide mode. The slide can be seen in Figure 30

e) Task 5: For this task, the user is asked to go to video mode once again, and then navigate to a specific time point in the video, and once again play the video until it stops. The participant is the asked to navigate back to the slide show. The slide utilised for this task is illustrated in Figure 31

f) Task 6: The participant is asked to navigate to the last slide, and back again. This is done in order to evaluate the applications capability of handling identical consecutive gestures. The slides employed can be seen in Figure 32

This marks the conclusion of the methodology related to our usability test. In the subsequent section we present the configurations employed during both the performance and usability tests.

7) System setup:

a) Tests:

- Desktop:
  CPU: i9 11900k
  RAM: 32 GB DDR4
  GPU: GeForce RTX 3080
- Laptop 1:
  CPU: i5 1035G
  RAM: 16 GB DDR4
  GPU: Iris Plus G1 (IceLake)
- Laptop 2:
  CPU: i5 5300U
  RAM: 8 GB DDR3
  GPU: HDGraphics 5500

b) Usability tests:

- Laptop 3
  CPU: AMD Ryzen 5 5600H
  RAM: 16 GB DDR4
  GPU: GeForce RTX 3070 Laptop

F. Activity 5: Evaluation

This activity is used to determine the degree to which the artifact solves the problem. According to Peffers, this can be done through *any empirical evidence or logical proof.* Based hereon we evaluate our artifact through the requirements presented in subsection III-C, and the research questions presented in subsection I-A. We perform a comparison of these areas through usability tests and performance tests, in order to obtain measurement of the artifacts based on the *Product Quality* and *Quality in Use* models.

1) Requirements evaluation: In order to satisfy the previously mentioned quote set forth by Peffers et al., saying that *"a design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to* *solve"* [18], we evaluate our artifact based on the requirements. We utilize the last phase in Sommerville's approach to perform this evaluation [41]. He presents 5 types of checks in order to validate the requirements:

- Validity checks
  This entails checking that the users needs are reflected in the requirements [41].
- Consistency checks
  It is important to ensure, that the system functions specified in the requirements are not contradictory [41].
- Completeness checks
  Used to assess whether the requirements include all of the functionality and constraints which the user needs [41]
- Realism checks
  Used to check whether the system can be developed within the time frame and budget [41].
- Verifiability
  It must be possible to verify the requirements through a set of tests, in order to show that the system meets the requirements specification [41].

We refrain from using the *realism checks*, as this check mostly relates to applications developed by a company, and therefore not within the scope of this paper. The rest of the checks utilised in the following, in order to evaluate our requirements, presented in Figure 4.

We evaluate the system's ability to fulfill the functional requirements, through the usability test. The tests were devised in a manner, which required the users to utilize the artifact's functionality in order to complete the tasks. This ensures that we evaluate all of the requirements.

We evaluate the non-functional requirements of the system based on an set of tests. We run a test of the system, feeding it a dataset of 1200 images made from a sample of the *American Sign Language* [48], *American Sign Language Letters* [50] and *ASL Digits* [49] datasets. During the test, we monitor the CPU, memory and GPU usage of the application. Additionally we time the tests, in order to determine how quickly the application is able to handle the predictions.

Through this evaluation, we are able to focus on both performance through the *Product Quality model* and the usability through the *Quality in Use model*, which aligns with Pressman's view on these two metrics being *fundamental to software success* [54].

Given that our artifact is procured as a prototype, we review the requirements based on the feedback received from the focus study group and the usability tests, through utilizing a questionnaire, as shown in Figure 23. This concludes our methodology section. The following section presents our findings.

IV. Results

This section presents our findings, divided into sections based on the DSRM phases, in the same manner as in section III.

A. Activity 1: Identify problem & motivate

1) Focus Group Findings: This section presents the findings from our focus group study. This study found that some of the most desired features are *break countdown* for timing lecture breaks, *stable use* to ensure no unwanted actions, *no input lag* to always expect actions immediately, *feedback* to display whether the user's hand and gestures were recognized, and *functionality* in order to support actions like pausing at a specific time within a video.

Furthermore participants emphasized the importance of a flawless experience, as errors occurring during a presentation might distract the listeners. Participants also pointed out, that they were uncomfortable *uploading files to the web*. We also discovered, that participants rarely used the ability to draw on their slides, and did not request this as an important feature.

The *drawing* activity, produced a series of mock-ups. We analyzed these in order to determine the functionality desired by the participants. Based hereon we discovered, that *feedback* both regarding whether the hand was recognized and whether the gesture was recognized was important to the participants. Furthermore, *feedback* on the available gestures in a given state, was requested. Additionally, users were interested in functionality regarding the presentation, such as *lecture notes*, *current time*, *time until break* and similar information providing an overview for the lecturer. Furthermore several users were interested in a visual representation of the *current* and *next slide*.

The following section presents the next activity, which focus on determining the objectives that are required for the artifact to provide a solution to the problems presented in section I.

B. Activity 2: Defining objectives of a solution

We present our requirements, engineered based on the Sommerville approach presented in subsection III-C. Figure 4 presents the functional requirements, prioritized through utilization of the MOSCOW method, and Figure 5 presents our non-functional requirements, based on the selected ISO 25010:2011 criteria.

Non functional requirements
Usable
Performance efficient
Functionally suitable
Portable
Compatible

Fig. 5. The non functional requirements

C. Activity 3: Design & development

1) Artifact: This section presents our artifact, a sustainable Zero UI-based application capable of controlling presentations through hand gesture recognition. The application is build,

Functional requirements	
Must have	The artifact must be able to recognize the camera feed
	The artifact must be able to perform hand recognition
	The artifact must be able to recognize gestures
	Navigate to the next or previous slide
	Provide laser pointer functionality
	Navigating between slide- and video mode
	Provide functionality in order to start or stop a video
	Ability to start and stop the program
Should have	Navigate to a specific slide
	Navigate to a specific point in a video
	Should provide feedback on the last accepted gesture
Could have	Gesture instructions window
	Ability to go from presentation to black screen, and back
	UI: Ability to write own slide notes
	UI: Time per slide feature displayed
	UI: Time on current slide displayed
	UI: Time until break displayed
	UI: Current slide / next slide displayed
Won't have	Voice control
	Sound feedback
	Ability to draw on the slideshow

Fig. 4. MOSCOW for the functional requirements

based on the Distributed Data Client-Server pattern, as suggested by Mathiassen et al [37, p. 202]. We present an overview of the system architecture in Figure 9.

The application is created using Next.js and Typescript. Each of the modules presented in Figure 9 are implemented through function components. In order to perform pose recognition, we utilize Google's MediaPipe [21], implemented through TensorflowJS, to detect hand landmarks in images. We furthermore propose our own machine learning model for recognizing poses. It is trained using Mediapipe Model Maker [55], based on our own dataset, consisting of 7872 images combined with two other datasets. The *American Sign Language Digits Dataset* [48] consisting of 4995 images and the *American Sign Language Letters* [50] consisting of 3467 images. This adds up to a total of 16334 images used in the process of training the model. Our performance tests conducted shows an accuracy of 99.68% and an F_1 score of 98.04%.

In order to determine which gestures a user perform, we utilize a finite state machine, through the XState [56] framework. This provides the ability to recognize a series of static poses, thereby turning it into the recognition of a gesture. We provide the user feedback on which state they are currently in, as well as the the current recognized gesture, and a video feedback showing whether their hand has been recognized. As we applied an iterative approach, we underwent two iterations in the development of the artifact. We present each of the three stages, followed by the findings in the usability test conducted.

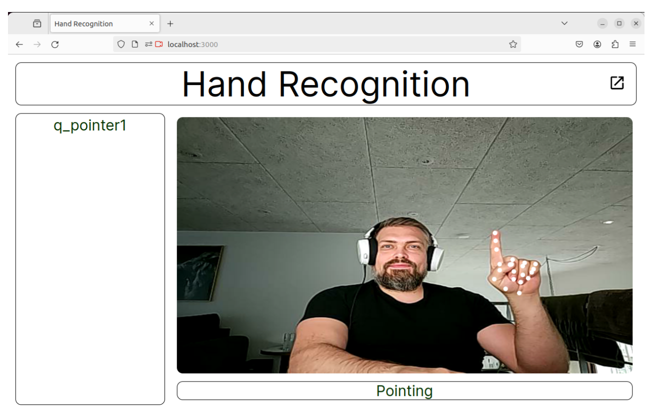

Fig. 6. The recognition page of our artifact

2) Artifact v1.0: The initial artifact, was based on the requirements presented in Figure 4 and Figure 5. The initial artifact contained 12 different poses, and 28 states. The model was trained based on a combination of our own dataset, consisting of 7481 images, and the 2515 images from the American Sign Language dataset [48], a total of 9996 images. This model provided an accuracy of 92% and an F_1 score of 94.7%. The state chart diagram depicting the v1.0 model, is displayed in Figure 7, in order to provide an overview.

However, based on the feedback received in subsubsection IV-D3, we performed a series of changes in the second

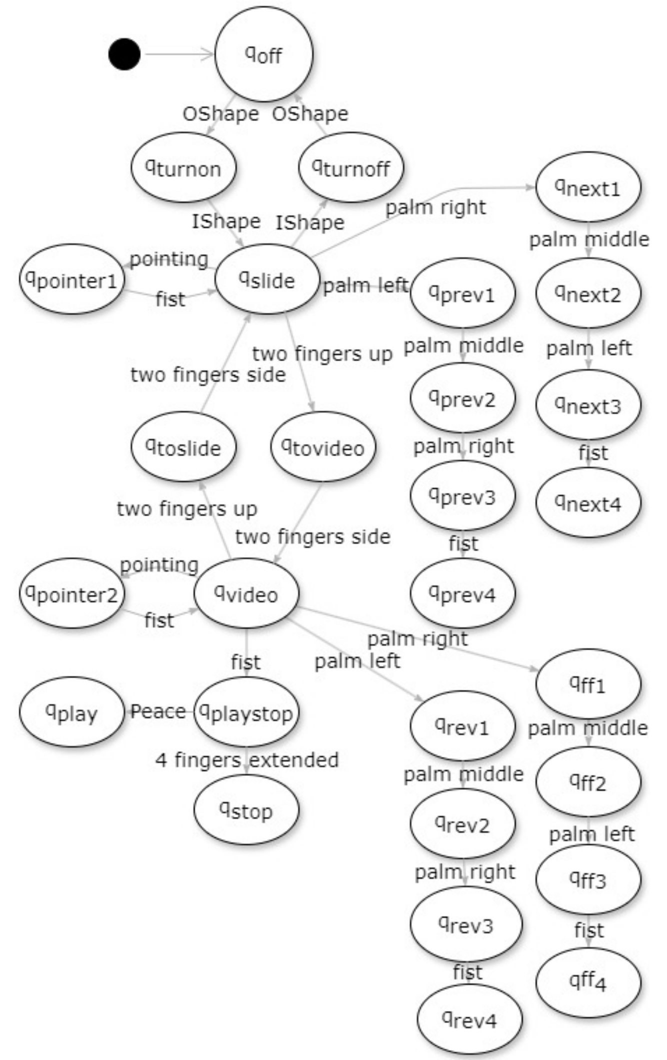

Fig. 7. The v1.0 Statechart diagram, The transitions back for some states like q_prev4 are left out for readability. In these cases the transition will go back to either q_slide or q_video depending on which state transitioned to the state.

iteration, in order to accommodate the user needs. We present these changes in the following section.

3) Artifact v2.0: The second iteration a reduction from 28 states down to 19. Based on an analysis of the feedback received during the first usability test, we found things that several aspects that users did not prefer.

- Improve the laser pointer
- Introduce more intuitive gestures
- Introduce a pinch functionality in order to navigate quickly through slides and video

83.3% of users complained about the laser pointer jumped , for which we introduced new functionality in order to provide a better user experience. We additionally accommodated the issue of having a steep learning curve, by reducing the number of gestures from 12 to 9. Furthermore, we made a change to the gestures, and utilized the gesture previously used to navigate between slide and video mode, as the new gesture for changing slides, due to users expressing excitement towards the way this gesture worked. Moreover, we introduced the fist as a state for canceling, as per recommendations from Jacob Nielsen's Heuristics [57].

Based on the various gestures, users are then able to utilize the functionality described in the requirements Figure 4.

For this iteration, we attempted to improve on the models accuracy. We applied a combination of the previous dataset, with signs from the the ASL Alphabet Dataset [45], which were equivalent to our poses. This resulted in a new dataset consisting of 38,501 images, distributed into 8 pose categories and a *none* category. The models metrics are presented in Figure 13.

4) Finite State Machine: The Finite State Machine were introduced to make the program's state more transparent for the users as requested in our previous research described in subsubsection II-A3. The mathematical notation for a finite state machine can be defined as a 5 tuple $(Q, \Sigma, \delta, q_0, F)$ [58, p. 35] where:

- Q is the finite set of states.
- Σ is the alphabet of the finite state machine.
- $\delta : Q \times \Sigma \to Q$ is the transition function.
- $q_0 \in Q$ is the initial state.
- $F \subseteq Q$ is the set of accept states.

Here we present the mathematical notation for our finite state machine:

- States:
  $Q = \{q_{off}, q_{turnon}, q_{turnoff}, q_{slide}, q_{pointer1}, q_{prev1}, q_{prev2}, q_{next1}, q_{next2}, q_{moveslider1}, q_{movedone1}, q_{toslide}, q_{tovideo}, q_{video}, q_{pointer2}, q_{moveslider2}, q_{movedone2}, q_{playstop}, q_{play}$
- Alphabet:
  $\Sigma = \{OShape, Pointing, Fist, Peace, Okay, 4FingersExt, TwoFingersSide, TwoFingersUp\}$
- Initial state:
  q_{off}
- Accept state:
  $\emptyset$

Transition function δ defined is shown in Figure 10.

There are no accept states in our FSM, as it continues to run until termination of the application, when the user leaves the website. The following section presents a demonstration of our artifact.

D. Activity 4: Demonstration

1) Pilot test findings: We prepared for our usability test through a pilot test. The pilot test was carried out based on the same tasks as presented in subsubsection III-E6 for the usability test, in order to ensure that the functionality worked as intended. We found no issues during the pilot test, which

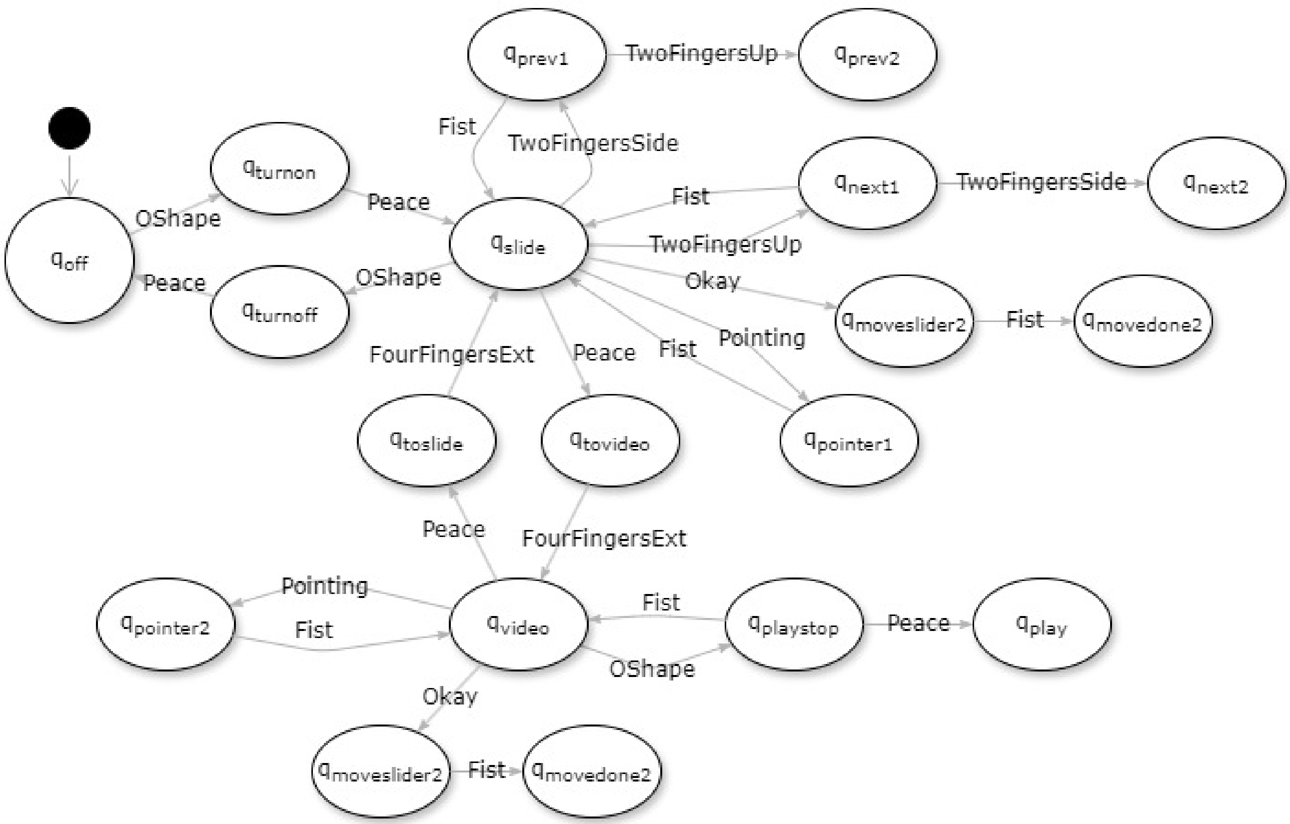

Fig. 8. The v2.0 Statechart diagram

enabled us to continue to the usability test, as presented in the following section.

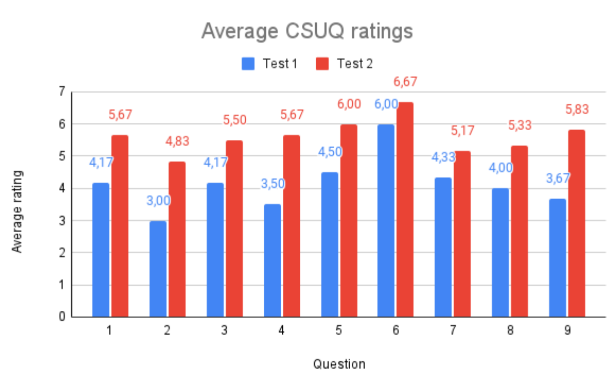

Fig. 11. Average participant ratings from the usability test

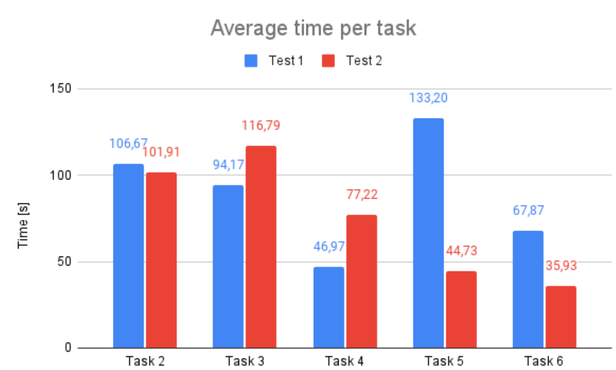

Fig. 12. Average time for usability tasks

2) Usability test findings: We conducted two usability tests, in order to determine whether the application satisfied the users needs. Both usability tests was conducted with 6 users, all proficient in the use of PowerPoint and other presentation tools.

Each user was measured based on the three metrics presented in subsection III-E. We evaluated their *effectiveness* through their ability to complete tasks. This evaluation showed, that 100% of the users were capable of solving each of the provided tasks, using the program, except for task

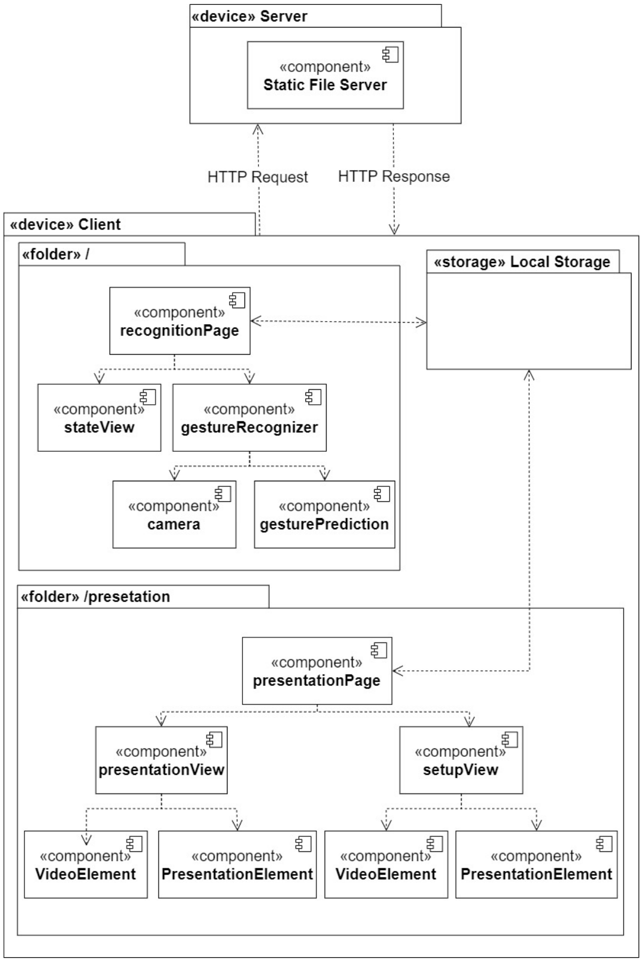

Fig. 9. Component Diagram: Distributed Data Client-Server Architecture

3 in the first usability test. The *efficiency* was measured as the time taken for each participant, to complete a specific task. Lastly we determined the users *satisfaction* through the questionnaire, the results of these can be seen in Figure 11. This showed that for the first usability test, the participants provided a rating between 3.0 and 6.0 for the first usability test, and a rating between 4.83 and 6.67 for the second one. From the overview it is evident that the users had a higher satisfaction during the second usability test, after the initial issues were fixed.

In the next sections, we elaborate the findings in regard to each of the tasks, which formed the basis for the program changes presented in the following iterations. The following sections presents our findings from the two usability test.

3) Usability test 1: Prior to the usability test, the participants were provided with an introduction to the different hand poses by the moderator, and following they were given 5 minutes to familiarize themselves with the system section C. This was done in an attempt to mitigate the learning effect, and ensure that each participant had an understanding of the gestures used in the application. After completing the usability test, the users were presented with a questionnaire, followed by a semi-structured interview, where the moderator asked follow-up questions. The subsequent sections presents the findings in each of the six tasks, supplemented by relevant comments from the participants.

a) Task 1: All participants completed the task. 66.7% of participants needed additional guiding, after having had the 5 minutes to play around with the program. And 50% of them commented that they found the gestures used for starting the program a bit counterintuitive.

b) Task 2: All participants completed the task. 33.3% of users experienced that the pointer was enabled as a result of their gesticulation. None of the users experienced any other sort of unintended actions taken, due to their gesticulation.

c) Task 3: 50% users were able to navigate through the maze, and thereby complete the task. 66.7% of users, commented on the pointer being difficult to control and 83.3% also commented on the pointer being jumpy.

d) Task 4: All participants managed to complete the task. During this task, 50% of users experienced issues with the thumb. The gesture used in this task, required that the thumb was not extended, which users did not find intuitive, when thinking of the other gestures.

e) Task 5: All users managed to complete the task. 66.7% of users experienced issues with remembering what the gesture was, and several had to navigate back to the initial slide to look it up.

f) Task 6: All users managed to complete the task. 66.7% of users could not remember the gesture for turning off the program.

4) Interview 1: Based on the feedback from the interview conducted after the first usability test, the following key points were emphasized.

- 66.7% of users expressed dissatisfaction with the gestures used to change the slides and navigate through the video mode.
- 50% of users disliked the gestures used to turn on the program, as they did not find it intuitive.
- 33.3% of users were dissatisfied the gesture used to enable and disable video mode.
- 83.3% of experienced issues with the laser pointer, both in terms of sensitivity and difficulties in controlling it.
- Users felt that the learning curve was steep, which 83.3% flagged as the main issue for them having issues with completing the tasks, alongside the pointer as mentioned in question 2.

δ	OShape	Pointing	Fist	Peace	Okay	4FE	2FS	2FU	10ms	800ms
q_{off}	q_{turnon}	–	–	–	–	–	–	–	–	–
q_{turnon}	–	–	–	q_{slide}	–	–	–	–	–	q_{off}
$q_{turnoff}$	–	–	–	q_{off}	–	–	–	–	–	q_{slide}
q_{slide}	$q_{turnoff}$	$q_{pointer1}$	–	$q_{tovideo}$	$q_{moveslider1}$	q_{next1}	q_{prev1}	q_{next1}	–	–
$q_{moveslider1}$	–	–	q_{slide}	–	–	–	–	–	–	–
$q_{movedone1}$	–	–	–	–	–	–	–	–	q_{slide}	–
$q_{pointer1}$	–	–	q_{slide}	–	–	–	–	–	–	–
q_{prev1}	–	–	q_{slide}	–	–	–	–	q_{prev2}	–	q_{slide}
q_{prev2}	–	–	–	–	–	–	–	–	q_{slide}	–
q_{next1}	–	–	q_{slide}	–	–	–	q_{next2}	–	–	q_{slide}
q_{next2}	–	–	–	–	–	–	–	–	q_{slide}	–
$q_{tovideo}$	–	–	–	–	–	q_{video}	–	–	–	q_{slide}
$q_{toslide}$	–	–	–	–	–	q_{slide}	–	–	–	q_{video}
q_{video}	$q_{playstop}$	$q_{pointer2}$	–	–	$q_{moveslider2}$	–	–	–	–	–
$q_{pointer2}$	–	–	q_{video}	–	–	–	–	–	–	–
$q_{playstop}$	–	–	q_{video}	–	–	–	–	–	–	q_{video}
$q_{playpause}$	–	–	–	–	–	–	–	–	–	–
$q_{moveslider2}$	–	–	q_{video}	–	–	–	–	–	–	–
$q_{movedone2}$	–	–	–	–	–	–	–	–	q_{video}	–

Fig. 10. The transition function for the state machine of Artifact v2. 4FE, 2FS, and 2FU is short for FourFingersExt, TwoFingersSide, and TwoFingersUp respectively. 10ms and 800ms indicates transitions taken after the specified time of respectively 10 and 800 milliseconds. This is used to ensure that if a user enters a state by mistake, the FSM will return to the previous state after these time limits, if the state is not met by an acceptable transition.

- 83.3% of participants mentioned that being able to use a pinch or pointer like functionality in order to rewind and forward the video would be preferable.
- 50% of participants would like to see a larger reuse of gestures, i.e. turn on and turn off would be based on the same gesture.

The above findings were used in order to make changes in the implementation. It resulted in the following changes:

- Fix pointer
- Change gestures
- Enable pinching gesture and apply it to the rewind / forward functionality

We implemented these changes in the artifact, and then conducted a second usability test, in order to obtain the users evaluation of the newly implemented features, to determine whether these better fulfilled their needs.

5) Usability test 2: This section presents the second usability test carried out with the same users as the first test. We utilized the same participants and assignments as last time, in order to be able to compare the two tests, based on the changes made from the findings of the last test. Users got an introduction to the new features, as we changed many of them, as per their feedback from the last usability test. Once again they had 5 minutes to familiarise themselves with the changes made to the prototype.

a) Task 1: All users managed to complete the task. No one had any issues or needed guidance.

b) Task 2: All participants managed to complete the task. We experienced that during gesticulation, two users accidentally activated the *pinch* gesture, causing the program to change to a different slide. Another user experienced issues with controlling the presentation using the left hand.

c) Task 3: All users managed to complete the task, navigating out of the maze within their first try. During this task, one user found the laser pointer to be mirrored compared to the other participants expectations.

d) Task 4: All participants completed the task. Two users could not remember the functionality of the video. With a little guidance they managed to complete the task together with the rest of the participants. One user activated the *pinch* functionality by mistake.

e) Task 5: All user completed the task without any issues. One user managed to activate *pinch* by mistake, leading to an unintentional fast forward in the video. However when the user became aware of it, they managed to mitigate the situation and complete the task.

f) Task 6: All users managed to complete the task. 50% of users navigated back to the initial slide, using the *navigate to previous slide* functionality. 4 users remembered the hand sign for turning off the program, the last 2 needed guidance.

6) Interview 2: All the participants commented positively on the improvements made to the pointer. Users generally provided positive feedback, they liked the changes that was made based on the previous usability test. Participants furthermore provided positive remarks regarding the improvements made to the *pointer*, the *pinch* functionality and the use of the flicking motion that was utilized for changing slides.

Our discussion with the users showed, that they felt that there was a learning curve that they needed overcome. They furthermore found that the pinch, the flicking motion for changing slides and the pointer had a more intuitive feel to them, compared to start video, changing between video and slide mode and turning the application on and off. However several mentioned, that it might be due to them having seen the other gestures in the previous test.

Several users still felt that the pointer was not responsive enough, it was discussed that having the ability to change the sensitivity could be a solution. Users in general expressed positive feelings towards the use of the fist as a cancel state. Furthermore a user suggested the change of the on / off screen to be a black screen instead.

E. Tests

This section provides the findings from the tests, described in subsubsection III-E1

To evaluate the hand pose recognition model, we calculate the values for Accuracy, Precision, Recall and F1 by using the formulas presented in paragraph III-D2b, the results are shown in Figure 13.

From the dataset, 1198 images were successfully evaluated, however MediaPipe did not find the landmarks in the remaining images and based hereon they were discarded. This was necessary, as we sought to evaluate our models ability to predict the correct pose, and not MediaPipe's abilitiy to recognize the hand.

The data is presented for each individual hand pose, to provide a more detailed evaluation of the model's performance. The hand poses "FourFingersExt", "Peace" and "TwoFingersUp" all received at score of 100% across all parameters, indicating no errors were observed for these hand poses during the testing phase. *Pinch*, *OShape*, *Pointing* and *None* demonstrated *accuracy*, *recall*, *precision* and F_1 scores all within the range of 95.16% to 99.94%. The "Fist" however, provided a less satisfactory score, as the *Precision* scored 86.09% and F1 scored 92.52%. The numbers can be seen in Figure 13.

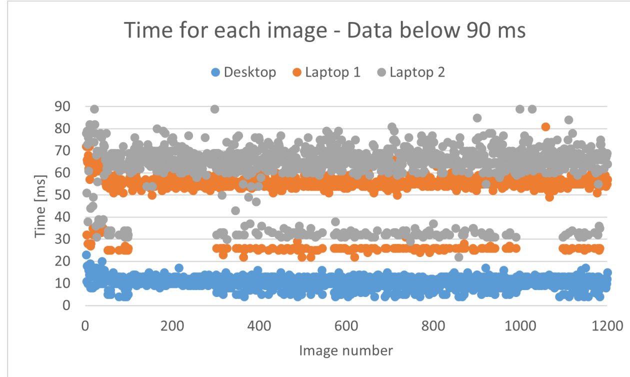

Fig. 14. The processing time for each image only showing data below 90 ms.

To compare the artifact's processing speeds achieved on different hardware, we timed the tests on 3 different devices. The timings for each image are plotted on Figure 14. The figure shows that the timings seem more or less consistent. However *Laptop 1* and *Laptop 2* display points below their average processing times. These points seems to occur when MediaPipe failed to detect the hand within the image.

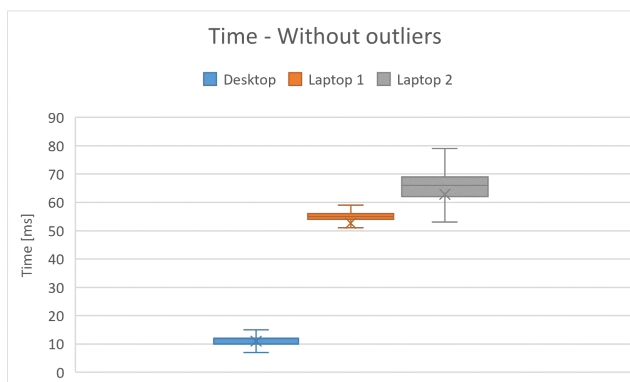

Fig. 15. This figure shows the three devices compared without the outliers.

To obtain a more precise understanding of the timing differences, a box plot for each device is shown in Figure 15. This comparison illustrates that *Laptop 2* has a higher degree of variability in the time taken, whereas the two other devices appear more consistent.

	none	Fist	FourFingersExt	Okay	OShape	Peace	Pointing	TwoFingersUp	Total
Images	565	99	99	99	79	98	60	99	1198
Accuracy	98,67%	99,12%	100,00%	99,94%	99,89%	100,00%	99,78%	100,00%	99,68%
Precision	99,63%	86,09%	100,00%	99,00%	98,73%	100,00%	95,16%	100,00%	98,08%
Recall	96,11%	100,00%	100,00%	100,00%	98,73%	100,00%	98,33%	100,00%	98,00%
F1	97,84%	92,52%	100,00%	99,50%	98,73%	100,00%	96,72%	100,00%	98,04%

Fig. 13. This tables shows the Accuracy, Precision, Recall, and F_1 for each hand pose and the total for the test.

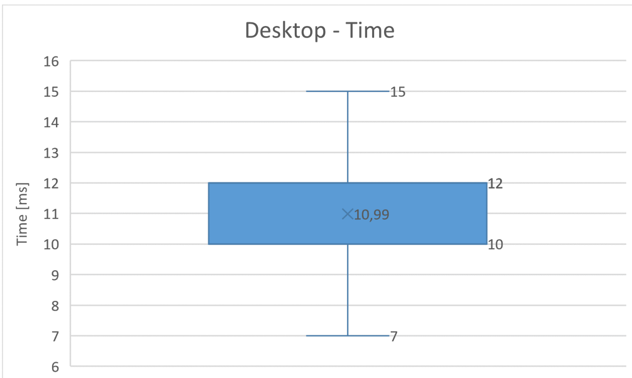

Fig. 16. Timings for Desktop, without outliers.

To better understand the actual timings of the *Desktop* computer Figure 16 shows a box plot for the timings without outliers. These lie between 7 - 15 ms, taking 10.99 ms on average for predicting a hand pose. This gives an average fps of ≈ 90.99.

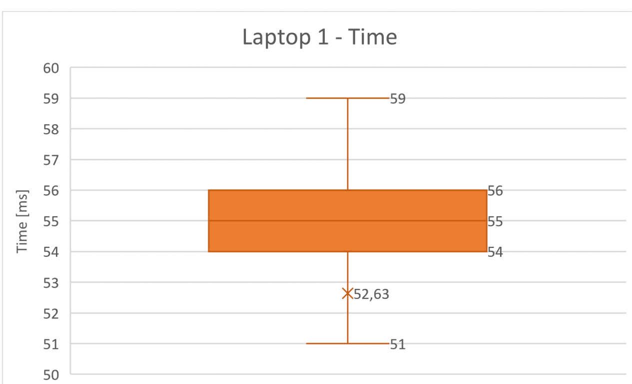

Fig. 17. Timings for Laptop 1, without outliers.

The timings for the *Laptop 1* is shown at Figure 17 as a box plot where the timings without outliers lies between 51 - 59 ms, taking 52.63 ms on average for predicting a hand pose. This gives an average fps of ≈ 19.00.

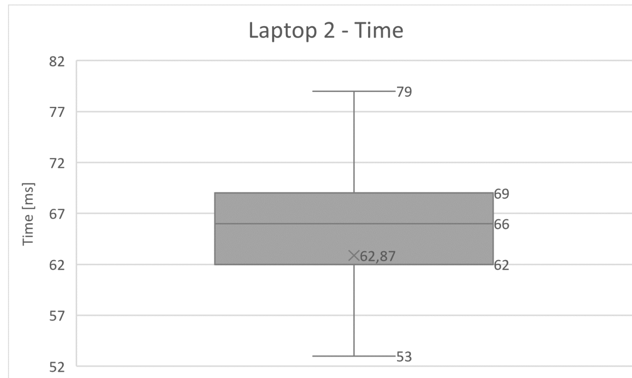

Fig. 18. Timings for Laptop 2, without outliers.

The timings for the *Laptop 2* is shown at Figure 18 as a box plot where the timings without outliers lies between 53 - 79 ms, taking 62.87 ms on average for predicting a hand pose. This gives an average fps of ≈ 15.91.

To evaluate the prototypes *resource utilization*, we collected the metrics for the browsers tab and its GPU process, both the users CPU, memory and GPU usage. These are presented in Figure 19, Figure 20, Figure 21, and Figure 22.

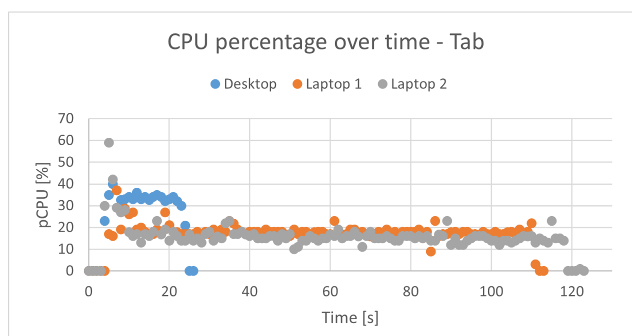

Fig. 19. The CPU percentage for the tab process for each device.

Figure 19 depicts the CPU percentage over time for the tab process. From this image it is evident that the *Desktop* was capable of completing operations faster than the other devices. *Laptop 1* was faster than *Laptop 2*, however not by a large margin. It furthermore shows that the desktop mostly used a CPU percentage just above 30%, whereas *Laptop 1* and *Laptop 2* used around 20%.

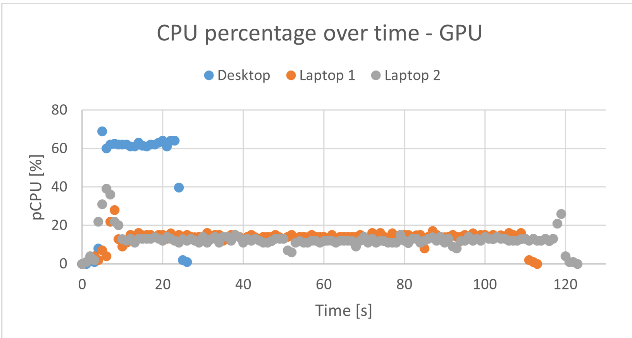

Fig. 20. The CPU percentage for the GPU process for each device.

The CPU percentage for the GPU process is shown in Figure 20. The figure show results similar to those depicted in Figure 19. The CPU usage for the *Desktop* consistently operated around 60%, where as *Laptop 1* and *Laptop 2* initially spiked to 30-40%, but dropped and stabilized around 15-20%.

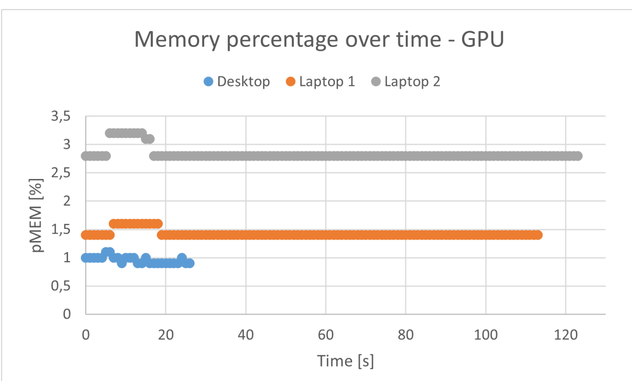

Fig. 21. The memory percentage for the GPU process for each device.

The memory usage during the execution of the test is shown in Figure 21. It reveals that the GPU process remained stable across all devices. *Laptop 2* used approximately 3% memory, *Laptop 1* around 1.5% and *Desktop* around 1%. It is noteworthy there is a difference in the amount of RAM, as *Desktop* has 32GB, *Laptop 1* has 16GB and *Laptop 2* has 8GB.

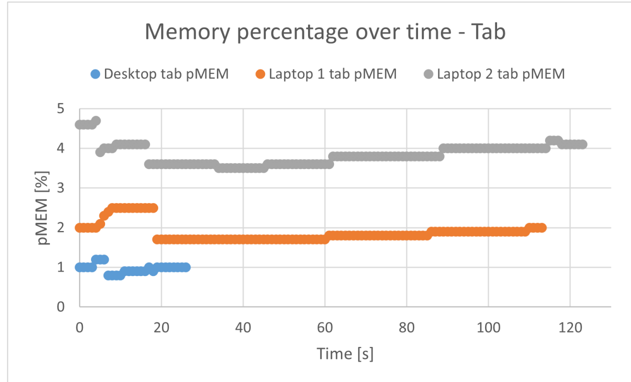

Fig. 22. The memory percentage for the tab process for each device.

For the tab process, displayed in Figure 22, the memory usage was less stable for all devices. *Laptop 2* exhibited around 4% memory usage, *Laptop 1* around 2% and *Desktop* around 1%.

F. Activity 5: Evaluation

In order to evaluate our requirements, we utilize the steps presented by Sommerville in section III.

1) Validity checks: The first step entails performing a validity check to ensure that the requirements reflect the users needs. As our requirements are based on previous findings, alongside the results from our focus group study, this put an emphasis on the users' needs in the requirements engineering phase, as our functionality is based on the analysis of the feedback received. Furthermore, during our usability tests, we had feedback regarding the changes of the functionality, for example users requested a *pinch* functionality, alongside the changes of gestures used in order to control the presentation, which we implemented in the second iteration.

2) Completeness checks: Completeness checks were done through our questionnaire and interviews of the participants at the end of each usability test. Question 7 in the questionnaire, asked users to evaluate whether the system had the functions and capabilities that they expected. Users provided an average score of 5.17. During the interview, we asked users which additional functionality that they would like to see. The primary requests were more intuitive gestures for switching to video mode, and the start and stop functionality. Additionally, some participants requested the ability to switch to a black screen, if the lecturer by mistake displayed information which was not meant to be displayed.

3) Verifiability checks: Verifiability checks were done, to verify that our artifact meets the requirements. We created a set of tasks for the usability test, which required the users to utilize the functional requirements, in order to complete the task. In this way we could ensure, that the requirements were fulfilled.

a) Functional: We start by evaluating our *must have* criteria. *The artifact should be able to recognize the camera feed*, as well as *perform hand and gesture recognition* based here on. This was done throughout all the tasks in the usability tests, as each of them required the user to utilise part of the functionality, which required the implementation of the first three requirements.

In order to be able to navigate through the tasks within the slideshow, the user was required to utilise the ability to switch slides. This fulfilled the requirement *Navigate to the next or previous slide*. The user was provided with feedback on their current state, alongside the currently recognized gesture, which fulfilled the requirement *Should provide feedback on the last accepted gesture*. An illustration of the gesture feedback can be seen in Figure 6.

The first task asked the user to turn on the application, and the last task asked them to turn it off again, which fulfilled the requirement *Ability to start and stop the program*. The third task asked users to test the laser pointer, and navigate through a maze presented in the slideshow. This fulfilled the requirement *Provide laser pointer functionality*.

The fourth and fifth tasks required the users to switch to the video functionality in order to complete the tasks, as well as switching back to the slide that provided the task. Therefore these tasks fulfilled the requirement *Navigating between slide- and video mode*. Task four additionally asked the user to start and stop the video, which fulfilled the requirement *Provide functionality in order to start or stop a video*. The fifth task utilized the *pinch* functionality, asking users to navigate to a specific point in a video, which fulfilled requirement *Navigate to a specific point in a video*.

The sixth task asked the users to navigate back to the initial slide, and then turn off the application. This fulfilled the *Navigate to a specific slide* requirement. Furthermore the usability test helped perform the *consistency check*, as contradictory requirements would have caused issues with completing the tasks provided. Thereby we conclude the evaluation of our functional requirements. Following, we present an evaluation of our non-functional requirements.

4) Non functional requirements:

a) The artifact should be usable: The general usability was evaluated based on three metrics based on the ISO 25010:2011 *Quality in Use Model* and its three criteria, as presented in subsection III-E. The *effectiveness* criteria was evaluated based on the time taken for the users to complete the tasks. The timings as presented in Figure 12 for task 2, 5, 6 were faster in the second usability test. This can indicate that the users were more comfortable solving the tasks. However the timings for task 3 and 4 were increased. For task 3 it could be due to the improvements on the pointer made it more possible to complete the path and this resulted in users being able to complete the task without any issues. In the first usability test, we experienced that 50% of users either gave up, or did not complete the task. For task 4 the increase might originate from the gesture change. Initially this task was assigned the gesture, that is now used for changing slide, however this gesture was reassigned to be used for navigating through slides, since it received positive feedback for being intuitive. Participants found it more difficult to remember the new gesture selected for this task, which could be the reason for the increase in the completion time.

The *satisfaction* criterion, was evaluated based on our CSUQ questionnaires. Analyzing the comparison of participant feedback, we discovered an increase in the *satisfaction* ratings on all parameters from the first to the second usability test, with some increases less than others. This variability might be due to the participants preferences for specific gestures. Some participants preferred the gestures used in the first artifact, while others found the changes in the second artifact to be better. To accommodate the different preferences the introduction of customisable gestures might provide an increased usability.

b) The artifact should be performance efficient: The performance efficiency was evaluated as part of ensuring the *utility* metric, as mentioned in subsection II-A. We utilise the two sub-characteristics presented in the ISO 25010:2011 *Quality in Use model*, as presented in section III. By evaluating the artifact's *performance efficiency*, we obtain an understanding of the required hardware for utilizing the application, as we need to ensure a seamless experience for the user. The *Desktop* provided a 90,99 average FPS, where as *Laptop 1* presented an 19 FPS on average, and *Laptop 2* had an average FPS of 15,91. The FPS indicates how many images the model is capable of predicting per second, and thereby indicates that each of the setups, would be viable solutions in presentation control. Thereby we can further conclude that the model is capable of running on most hardware, based on the specifications of *Laptop 2*, as presented in subsubsection III-E7.

c) Functionally suitable: The performance efficiency was evaluated based on our *accuracy*, *recall*, *precision* and F_1 *score*. As mentioned in Figure 13, we obtained an accuracy of 99.68% overall, a precision score of 98.08%, a recall of 98% and a F_1 score of 98.04%. Thereby we can compare our solution to the contributions within the area of hardware-based gesture recognition. Yang et al. [36] presented a solution utilizing the LeapMotion controller, they obtain an accuracy of 95.238%, a precision of 95.546%, a recall score of 95.238% and a F_1 score of 95.274%. Another paper, also utilizing the leap motion controller, is Huo et al. [35] who presents an accuracy between 95.2-99.8%, a precision score of 88,78%-100%, a recall score between 90% - 100% and an F_1 score between 87.88% - 99.5%.

Furthermore this allows us to compare our results to the software-based solutions, as presented in subsection II-A. Orovwode et al. [33] presents a solution with an *accuracy* of 90% *recall* of 90%, a *precision* of 92% and a F_1 *score* of 91%. Osama et al. [31] presents a contribution, utilizing 6 different hand poses, an *accuracy* of 97.6% *recall* of 98.66%, a *precision* of 94.88% and a F_1 *score* of 96.70%. Lastly, Setiawan et al. [32] presents a *accuracy* between 99-100%, a *recall* between 98-100%, a *precision* between 99-100% and a F_1 *score* of 99-100% %.

Thereby we can conclude that our MediaPipe based model performed equivalent to the state of the art presented in the subsection II-A, and that it can be utilized in order to implement a gesture-based presentation control artifact, which is capable of performing gesture recognition at a satisfactory level.

d) Portable: By utilizing the *Distributed Data Client-Server* architecture, we provide an application that can run on most platforms, such as *Windows*, *Linux* or *Android*, thereby ensuring its *adaptability*. The static website, furthermore ensures that users are not required to upload slides or any other material. This allows for the application to be available to a larger group of people, compared to solutions made specifically to run locally or via extensions, as presented in subsection II-A. As the application is supported in all chromium browsers, there is no need for installing it on a specific system, i.e. ensuring that the *installability* criteria, can be fulfilled on any device able to use chromium browsers.

e) Compatible: The findings in regards to *resource consumption*, are used to evaluate the *co-existence*, i.e. applications ability function alongside other processes in an environment. Our findings show, that the application uses approximately 10-50% of a single core during the test performed in Figure 19. Furthermore, it uses around 1-5% of memory Figure 22. This shows that it would not impact the ability for other applications to utilize resources.

V. Discussion

This section presents a discussion based on our research questions. It is divided into three sections, one related to each of the research questions presented in subsection I-A.

A. RQ1:

In section II we present two areas of research within the field of hand gesture recognition. One field focus on the utilization of *hardware-based* solutions, whereas the other focus on *software-based* solutions. By comparing these two solutions, we are able to provide an answer for RQ1.

Based on these findings, it becomes evident, that software based solutions are capable of providing similar results compared to their hardware-based counterparts. However no recent papers on the subject of utilizing hardware-based solutions on presentation control was found within the literature. Therefore it becomes relevant for more research into the area of hardware peripherals, such as the *LeapMotion* controller, in order to obtain a more concise comparison, as our solution and the solutions presented above are tested using different datasets. By comparing hardware-based and software-based solutions on the same datasets, this would provide more accurate metrics in terms of the differences between hardware and software-based gesture prediction within the topic of presentation control. However, the above metrics emphasize, that the two approaches are similar in their ability to recognize gestures, and thereby underlines that software-based solutions have their merit. Thereby it becomes possible to utilize these in order to provide a sustainable alternative, to the hardware-based solutions.

Turning our attention to the software-based solutions, we found two tendencies within the presentation control related literature, as mentioned in section II. One tendency is papers evaluating their model based on the *accuracy* metric, and the other tendency are papers utilizing the *recall*, *precision* and F_1 scores as well. However, according to Mathiassen et al. *"a good design balances several criteria"* [37, p. 181], as they provide metrics for quality evaluation through their focus on *experiments* and *reviews* [37, p. 180]. The lack of utilizing other software quality related criteria, underlines the necessity for additional research into the impact of employing these when developing applications for presentation control. By utilizing the ISO 25010:2011 standard, alongside the *checklist for prioritization* which we employed in Figure 2, we determined a set of criteria, which provided valuable in the evaluation of the application. We discuss each of the ISO 25010:2011 criteria that we selected, in subsection V-C. However, we start by presenting the discussion in regard to *Research Question 2* in the following section.

By utilizing the *Distributed Data Client-Server* pattern able to remove the need for a backend, and the HTTP traffic required for communication between the client and server. The need for less hardware, is suitable with our attempt to provide a sustainable solution. As we furthermore utilize a solely software-based solution, we are able to run the artifact on existing hardware, and do not require additional peripherals, such as smartboards and other devices needed to control the presentation, thereby further minimizing the need for resources consumption in order to utilize the application.

B. RQ2:

In this section, we attempt to answer Research Question 2, *Which methods can be utilised to focus on a user-friendly approach, and how can they be used to establish the needed functionality and a set of user-friendly gestures for presentation control?*.

1) Usability: Within the 39 papers that we surveyed, only two of these papers included a usability test. According to Jakob Nielsen, *"usability is a necessary condition for* *survival"* [53]. This underlines the importance of utilizing this criterion, during the design and development of an artifact. By employing usability tests, we discovered that users had different preferences in terms of gestures, compared to the ones presented by Hosseini et al. [40], which presented a series of normative gestures, used for different devices such as controlling televisions, computers etc. However, the process of determining the gestures employed when interfacing with the artifact, brings about a series of topics for discussion, in order to answer *RQ2*.

We employed *usability tests*, as presented in subsection III-E, in order to ensure that the gestures utilised in the artifact, did not affect the presentation if they share a certain level of similarity with hand poses often achieved during gesticulation. The research presented by [40], is focused on controlling devices as televisions and computers, which leaves out important parts of utilising gestures for presentation control. Presentation control has an extra element added, the gesticulation. This becomes an important but hard factor to account for. In our prototype we provide 8 distinct hand poses, alongside a none category. This means that everything that is not contained within the 8 categories, should be recognized as none. However, this furthermore renders many of the gestures used when controlling devices like a television obsolete, as it does not account for whether the motions resembles a gesture, and therefore could cause unintended behavior. We based our selection of hand gestured on the feedback received in the usability tests, in order to accommodate the users preferences in order to ensure the *usability* criteria presented by Lazar. However we did not find papers in the literature, providing any argumentation as to their choices of poses, despite the importance of including the users in the process of selecting the poses utilized for controlling presentations, which could potentially affect the usability experience when using the given application.

Furthermore, as presentations are held in front of large crowds, there is another aspect that comes into play, the social conventions and their limitation of the use of gestures. As the Zero UI-based transition is still in its infancy, standing in front of a crowd while utilizing hand gestures, could be a transgressing experience for some. Therefore selecting gestures, that are subtle, but still recognizable is preferable. This furthermore raises the question of determining which gestures that users would prefer. By comparing our findings from the questionnaires utilised during the two usability studies within this paper, we saw an increase in the users *satisfaction* scores, after the implementation of the user inspired gestures. It furthermore indicates that users preferred the gestures provided in the second iteration of our artifact. However, as our usability test was conducted on six participants, this calls for larger field-based studies, capable of providing an overview of which gestures that in general are found to be the most intuitive. We propose to build upon the approach made by Hosseini et al. [40], but with a specific focus on presentation control. By obtaining quantitative data, this allows for selecting the most preferred gestures, and thereby reaching the largest user base possible in order to ensure the *usability* criteria, presented in the *Quality in Use* model [20]. This emphasizes the need for additional research into the subject, as user preferences, is a subjective matter.

The above indicates, the importance of usability tests, as it provided invaluable feedback pointed out by the participants. This led to changes in the application and its functionality, which would not otherwise have been discovered and implemented.

Within the studies presented in subsection II-A, we did not encounter any argumentation as to the choices in terms of the functionality provided in the artifacts. By utilizing focus group studies, we obtained qualitative information in regard to the selection of functionality within the artifact, based on expert statements. This provided a baseline for our research questions and requirements, ensuring that our artifact implemented functionality which users requested. This underlines the importance of including the users in the design phase of the application, and throughout the development phase, in order to ensure the *effectiveness*, *usability* and *efficiency* criteria presented in the *Quality in Use* model, as presented in subsubsection III-E2.

C. RQ3:

This section aims to discuss *Research Question 3*, which asks *Which important software quality principles should be incorporated into a gesture-based presentation control artifact to ensure the development of a usable, efficient and effective application?*. The section is split into 5 sections, based on the selected ISO criteria, as shown in Figure 2.

1) Performance evaluation: The literature contains only a few examples of papers in regards to presentation control, whom utilizes performance evaluation criteria, such as their resource consumption and time taken to run the tests. However none of the papers published after 2020 contains metrics. We utilize this metric, as a means of determining the frame rate of our application, and whether it is capable of running on specific hardware, as mentioned in section IV. Utilizing this metric, was used as a means of proving that the application was capable of conducting the necessary calculations within the required time frame.

D. Functional suitability

1) Functional correctness: As mentioned in the introduction of this section, the two discourses using either *accuracy* or a combination of *accuracy* alongside *recall*, *precision* and F_1 score. This creates an indifference in the literature, complicating the process of comparing results. Furthermore, *accuracy* only provides overall information in terms of how the model performs. As the *accuracy* formula presented in paragraph III-D2b shows, *accuracy* determines the amount of correct predictions compared to the total number of predictions. However, this does not provide the ability to delve into further details in order to analyse the models performance.

In order to provide a more detailed perspective on the performance of the model, the *precision*, *recall* and F_1 scores are often utilised [43] [42], as a means of providing more in depth analysis of the models predictions. As previously mentioned, *precision* looks at the number of true positives, compared to the sum of true positives and false positives. This looks at the number of instances that were predicted to be correct (TP and FP), compared to the ones that were correct (TP), which is used in order to mitigate the effect of false positives. This is an important metric for using hand recognition, as it is important to determine whether the model is inclined to predict specific poses incorrectly. *Recall* is used as a metric to determine the models ability to predict all positive instances correctly. This gives important insights into the models ability to consistently recognise all instances of a hand pose. The F_1 score provides a harmonic mean of the precision and recall, resulting in a number which will make sure to indicate if one of the metrics are worse than the other.

Figure 13 shows that the Fist received the lowest score F_1. However it received a 100% score in recall, meaning every actual image of a fist, were identified, while the precision of 86.09% indicates that some images not containing a fist were falsely predicted to be a fist. But since the fist pose is used as a cancellation pose it is not seen as much of a problem that the cancellation is activated more often as it prevents a possible action.

E. Portability

The solutions presented in the literature predominantly rely on solutions running locally on the users computer. A few solutions are implemented as Chromium based browser extensions. These implementation details, would impose certain constraints on the use of the applications. The locally created solutions are primarily implemented using Python, as presented in subsection II-A. This impose certain limitations, and necessitate additional steps in order to ensure cross-platform compatibility. Similarly, despite Chromium based extensions being more accessible, there still exists limitations, as the user would be confined to Chromium based browsers. Furthermore, extensions could present challenges, for example in terms of communication with APIs, which would provide additional work as these can differ across platforms. Moreover, using extensions, provide a security challenge, as the application should be designed in order to ensure security, and must be regularly updated, in order to ensure that various security issues that emerge over time, are fixed. By utilizing the *Distributed Data Client-Server* architecture, we are able to provide a solution, capable of running on devices capable of utilising any type of browser that can render static websites. Furthermore we avoid the issues in terms of security, as presented in subsubsection III-D4. Additionally, this approach provides the ability for the user to utilise the website in different ways, one option is hosting a static file server either online or locally, as well as hosting the application locally one their own PC or utilising applications such as Electron [59] in order to run the website like a native desktop application.

F. Compatibility

By conducting the tests presented in subsection IV-E, we are able to provide an overview of the application's performance. This provides an idea of the minimum hardware specifications for the application. We tested using *Laptop 2*, which contains an 5th generation i5 5300U processor, 8 GB of DDR3 ram and an onboard HDGraphics 5500 onboard GPU, in order to provide an approximation of the minimum specifications required to run the application. We found that it is capable of predicting 15 frames per second, which we believe is sufficient in order to run the application. Thereby we are able to obtain an understanding of whether it seems viable that our application could be utilised within the educational sector. We believe that this is possible, based on the ability to run on *Laptop 2*.

Moreover, using the test provided in subsection IV-E, we are able to determine whether our application is capable of running on a system, without consuming so much resources, that it would hinder the ability to run other programs concurrently, as mentioned in paragraph IV-F4e. This ensures that we provide an experience that meets the criteria request in the *Quality in Use* model, i.e. *usability* as it ensures that the user gets a seamless experience with the application, due to it being *effective* allowing them to complete the task at hand and *efficient* through ensuring that the user is able to do it within an acceptable time frame, and are not limited by the lack of hardware resources.

G. Limitations

As we strive to procure a sustainable artifact, this has provided limitations in terms of the trained model, as we aim to reduce the usage of GPU. However, training a neural network to perform the gesture recognition part, instead of a finite state machine, could provide the ability to train the model based on a set of videos, and potentially allow for more edge cases to be recognized.

Another limitation is the fact that we use a normal camera. This creates a challenge when it comes to the lighting conditions, compared to a depth camera, which would be able to recognize the hand no matter the lighting conditions.

H. Future works

This section present an overview of topics related to the paper, that could be researched further. As we conducted our focus group study with a group of 5 participants, and the usability study with a group of 6 participants, further research utilising a larger population are relevant. This would provide more insight, in terms of which gestures that users would prefer, and thereby ensure that the selection of gestures, are based on the majority's preferences.

Furthermore our usability tests were only conducted in a single room, therefore there is a need for further testing. By providing various tests within real world settings, such as utilising the application during lectures with different light settings and people, could provide additional information as to the performance of the artifact.

VI. Conclusion

Restate the thesis This paper introduces a Zero UI-based artifact, designed using artificial intelligence in order to perform gesture recognition used for controlling presentations. We found that the six phases of Peffer's Design Science Research Methodology provides a useful structure, which focus on both the *utility* and *justified theory* of the design and development process.

In order to answer the first research question, we utilised MediaPipe in order to create a software-based artifact, capable of performing gesture recognition used to control presentations. We use Sommerville's recommendations for conducting requirements engineering, in order to fulfill the quote presented by Hevner, stating that an *"artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve"* [22]. By utilising a software-based approach, we remove the need for hardware based peripherals, such as smartboards, and can instead focus on employing existing hardware. Moreover we use the *Distributed Data Client-Server* architecture, in order to minimise the need for communication between client and server, thereby using less resources in order to conduct gesture recognition.

We answer the second research question, through utilising the *Quality in Use* model, presented in the ISO 25010:2011 standard, more specifically the *efficiency*, *effectiveness* and *usability* criteria. We conduct a focus group study in order to ensure that our functionality is based upon the user needs. Additionally, we conduct two usability tests in order to evaluate the *Quality in Use* model's criteria, based on the time, completion rate and satisfaction of the users, inspired by the recommendations set forth by Lazar. We started by utilising the research contributed by Hosseini et al. however, we discovered that users had varying preferences. Therefore, we used the feedback received from the first usability test, in order to produce a new set of gestures. We evaluated the satisfaction rating through a modified version of the CSUQ questionnaire, based on a 7 point Likert scale. The results showed that the satisfaction increased, from a ranking between 3.0 to 6.0 in the first usability test, to rankings between 4.83 and 6.67 in the second usability test. Moreover, we designed the usability tests in a way, that ensured that all the functional requirements were included as tasks, in order to ensure that the program fulfilled these.

The last research question, is answered through the use of the ISO 25010:2011 *Product Quality* model criteria. We used the prioritization model presented by Mathiassen et al. in order to rank the importance of these criteria. Based here on, we utilise the *usability*, *performance efficiency*, *functional sustainability*, *portability* and *compatibility* criteria, in order to provide a series of non-functional requirements. We evaluate these criteria, based on a performance test. The test utilises the sub-criteria of the *performance efficiency*, *resource utilization* and *time behaviour* in order to evaluate the performance of the artifact. Based on a test on 1198 images, we test three different hardware setups, we found that the average time taken to perform recognition is between 10.99 and 62.87 ms, providing an average FPS between 15.91 and 90.99, depending on the hardware setup. We furthermore utilised the *functional suitability* criteria, in order to conduct a series of tests on our own CNN model. We achieved an average *accuracy* across our poses of 99.68%, a *precision* of 98.08%, a *recall* of 98.00% and an F_1 score of 98.04%.

Thereby we can conclude that our artifact is capable of providing metrics at the level of the state of the art presented in the research, both the hardware- and software-based contributions.

References

[1] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955," *AI Magazine*, vol. 27, no. 4, p. 12, Dec. 2006. [Online]. Available: https://doi.org/10.1609/aimag.v27i4.1904

[2] Statista. (2022) Increase in touchless customer interactions during vs post covid-19, by country. Accessed: 14/06-2024. [Online]. Available: https://www.statista.com/statistics/1156020/increase-in-touchless-interactions-during-and-after-covid-19-by-country/

[3] OpenAI. Introducing chatgpt. Accessed: 14/06-2024. [Online]. Available: https://openai.com/index/chatgpt/

[4] Neurallink. Neurallink. Accessed: 14/06-2024. [Online]. Available: https://neuralink.com/

[5] Google. Deepmind. Accessed: 14/06-2024. [Online]. Available: https://deepmind.google/

[6] Statista. (2024) Artificial intelligence (ai) market size worldwide in 2021 with a forecast until 2030. [Online]. Available: https://www.statista.com/statistics/1365145/artificial-intelligence-market-size/

[7] M. Z. Iqbal and A. G. Campbell, "From luxury to necessity: Progress of touchless interaction technology," *Technology in Society*, vol. 67, p. 101796, 2021. [Online]. Available: https://doi.org/10.1016/j.techsoc.2021.101796

[8] Apple. (2024) Siri. Accessed: 14/06-2024. [Online]. Available: https://www.apple.com/siri/

[9] Microsoft. (2022) Kinect for windows. Accessed: 14/06-2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows

[10] Apple. (2024) Brug af face id på din iphone. Accessed: 14/06-2024. [Online]. Available: https://support.apple.com/da-dk/108411

[11] Ultraleap. Pepsico innovation: Touch less, smile more! Accessed: 14/06-2024. [Online]. Available: https://www.ultraleap.com/company/news/case-study/pepsico-touchless-ordering/

[12] F. T. Experience. Airasia introduces counter check-in fees to encourage use of contactless technology. Accessed: 14/06-2024. [Online]. Available: https://www.futuretravelexperience.com/2020/09/airasia-introduces-counter-check-in-fees-to-encourage-use-of-contactless-technology/

[13] E. Air. Emirates enhances smart contactless journey with touchless self check-in kiosks. Accessed: 14/06-2024. [Online]. Available: https://www.emirates.com/media-centre/emirates-enhances-smart-contactless-journey-with-touchless-self-check-in-kiosks

[14] S. Dhal, S. Samantaray, and S. C. Satapathy, "From chalk boards to smart boards: An integration of iot into educational environment during covid-19 pandemic," in *Intelligent Data Engineering and Analytics*, S. C. Satapathy, P. Peer, J. Tang, V. Bhateja, and A. Ghosh, Eds. Singapore: Springer Nature Singapore, 2022, pp. 301–309. [Online]. Available: https://doi.org/10.1007/978-981-16-6624-7_30

[15] M. Marani, G. G. Katul, W. K. Pan, and A. J. Parolari, "Intensity and frequency of extreme novel epidemics," *Proceedings of the National Academy of Sciences*, vol. 118, no. 35, p. e2105482118, 2021. [Online]. Available: https://doi.org/10.1073/pnas.2105482118

[16] W. H. Organization. Electronic waste (e-waste). [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/electronic-waste-(e-waste)

[17] Statista. Number of elementary and secondary schools in the united states in 2020/21, by school type. [Online]. Available: https://www.statista.com/statistics/238307/number-of-us-elementary-and-secondary-schools/

[18] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. [Online]. Available: https://doi.org/10.2753/MIS0742-1222240302

[19] L. Beck and C. B. Kristensen, "Evaluating hand recognition models for presentations based on iso 25010:2011 for performance and usability in a zero ui context," 2024. [Online]. Available: https://kbdk-aub.primo.exlibrisgroup.com/permalink/45KBDK_AUB/a7me0f/alma9921651320905762

[20] ISO/IEC. Systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models. [Online]. Available: https://www.iso.org/standard/35733.html

[21] Google. Mediapipe solutions guide. Accessed: 14/06-2024. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/guide

[22] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *Management Information Systems Quarterly*, vol. 28, no. 1, pp. 75–105, 03 2004. [Online]. Available: https://doi.org/10.2307/25148625

[23] L. V. Lapão, M. M. da Silva, and J. Gregório, "Implementing an online pharmaceutical service using design science research," *BMC medical informatics and decision making*, vol. 17, pp. 1–14, 2017. [Online]. Available: https://doi.org/10.1186/s12911-017-0428-2

[24] N. Silva, P. Sousa, and M. M. d. Silva, *A Conceptual Model for Expressing the Evolution of Enterprise Architecture Models*. Cham: Springer Nature Switzerland, 2023, pp. 265–277. [Online]. Available: https://doi.org/10.1007/978-3-031-30214-5_20

[25] M. Berkhout, K. Smit, and J. Versendaal, "Decision discovery using clinical decision support system decision log data for supporting the nurse decision-making process," *BMC Medical Informatics and Decision Making*, vol. 24, no. 1, p. 100, 2024. [Online]. Available: https://doi.org/10.1186/s12911-024-02486-3

[26] Scopus. Scopus search. Accessed: 14/06-2024. [Online]. Available: https://www.scopus.com/search

[27] S. Islam, A. Matin, and H. B. Kibria, "Hand gesture recognition based human computer interaction to control multiple applications," in *Intelligent Computing & Optimization: Proceedings of the 4th International Conference on Intelligent Computing and Optimization 2021 (ICO2021) 3*. Springer, 2022, pp. 397–406. [Online]. Available: https://doi.org/10.1007/978-3-030-93247-3_39

[28] K. Sathish, G. Renuka, M. Balachandra, and B. N. Lakshmi, "Gesture-controlled virtual mouse using media pipe," in *2023 1st International Conference on Optimization Techniques for Learning (ICOTL)*. IEEE, 2023, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICOTL59758.2023.10435188

[29] M. Idrees, A. Ahmad, M. A. Butt, and H. Danish, "Controlling power-point using hand gestures in python," *Webology*, vol. 18, pp. 1372–1388, 12 2021. [Online]. Available: https://www.researchgate.net/publication/358914004_Controlling_PowerPoint_Using_Hand_Gestures_in_Python

[30] S. F. Ahamed, P. Sandeep, P. Tushar, and S. Srithar, "Efficient gesture-based presentation controller using transfer learning algorithm," in *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 2023, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICCCI56745.2023.10128634

[31] N. Osama, Y. Ahmed, H. Mohamed, S. E. Hesham, Y. Ahmed, E. K. Elsayed, and D. Ezzat, "Virtual control system for presentations by real-time hand gesture recognition based on machine learning," in *International Conference on Advanced Intelligent Systems and Informatics*. Springer, 2023, pp. 327–335. [Online]. Available: https://doi.org/10.1007/978-3-031-43247-7_29

[32] C. A. Setiawan, G. N. Utomo, V. Pangesa, V. Wijayanti, V. F. Putra Basuki, Y. Hermanto, S. Hasana, and I. H. Permonangan, "Gestive: Evaluation of multi-class classification methods for gesture recognition to improve presentation experience," *Procedia Computer Science*, vol. 227, pp. 364–371, 2023, 8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023). [Online]. Available: https://doi.org/10.1016/j.procs.2023.10.535

[33] H. Orovwode, J. A. Abubakar, O. C. Gaius, and A. Abdullkareem, "The use of hand gestures as a tool for presentation," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, 2023. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2023.0141159

[34] UltraLeap. Leap motion controller 2. Accessed: 14/06-2024. [Online]. Available: https://leap2.ultraleap.com/products/leap-motion-controller-2/

[35] J. Huo, K. L. Keung, C. K. M. Lee, and H. Y. Ng, "Hand gesture recognition with augmented reality and leap motion controller," in *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2021, pp. 1015–1019. [Online]. Available: https://doi.org/10.1109/IEEM50564.2021.9672611

[36] L. Yang, J. Chen, and W. Zhu, "Dynamic hand gesture recognition based on a leap motion controller and two-layer bidirectional recurrent neural network," *Sensors*, vol. 20, no. 7, p. 2106, 2020. [Online]. Available: https://doi.org/10.3390/s20072106

[37] L. Mathiassen, A. Munk-Madsen, P. Nielsen, and J. Stage, *Object-Oriented Analysis and Design*. Marko, 2000.

[38] J. Langford and D. McDonagh, *Focus Group - Supporting Effective Product Development*. Taylor & Francis, 2003.

[39] J. Kontio, L. Lehtola, and J. Bragge, "Using the focus group method in software engineering: obtaining practitioner and user experiences," in *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE '04.*, 2004, pp. 271–280. [Online]. Available: https://doi.org/10.1109/ISESE.2004.1334914

[40] M. Hosseini, T. Ihmels, Z. Chen, M. Koelle, H. Müller, and S. Boll, "Towards a consensus gesture set: A survey of mid-air gestures in hci for maximized agreement across domains," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544548.3581420

[41] I. Sommerville, *Software Engineering*, 9th ed. Addison-Wesley, 2010.

[42] M. M. Rahman, A. Mohaimenul Islam, J. Miah, S. Ahmad, and M. Mamun, "sleepwell:stress level prediction through sleep data. are you stressed?" in *2023 IEEE World AI IoT Congress (AIIoT)*, 2023, pp. 0229–0235. [Online]. Available: https://doi.org/10.1109/AIIoT58121.2023.10174306

[43] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Advances in Information Retrieval*, D. E. Losada and J. M. Fernández-Luna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 345–359. [Online]. Available: https://doi.org/10.1007/978-3-540-31865-1_25

[44] S. M. Rayeed. American sign language digits dataset. [Online]. Available: https://www.kaggle.com/datasets/rayeed045/american-sign-language-digit-dataset

[45] Akash. Asl alphabet. [Online]. Available: https://www.kaggle.com/datasets/grassknoted/asl-alphabet

[46] IBM. The component diagram - ibm developer. Accessed: 14/06-2024. [Online]. Available: https://developer.ibm.com/articles/the-component-diagram/

[47] ——. Creating state machine diagrams - ibm developer. Accessed: 14/06-2024. [Online]. Available: https://www.ibm.com/docs/en/dmrt/9.5?topic=diagrams-creating-state

[48] A. Thakur. American sign language dataset. [Online]. Available: https://www.kaggle.com/datasets/ayuraj/asl-dataset

[49] V. Ofor. American sign language digits. [Online]. Available: https://www.kaggle.com/datasets/victoranthony/asl-digits-0-9

[50] D. Lee. American sign language letters. [Online]. Available: https://public.roboflow.com/object-detection/american-sign-language-letters

[51] J. Lazar, J. H. Feng, and H. Hochheiser, *Research Methods in Human Computer Interaction*, second edition ed. Morgan Kaufmann, 2017. [Online]. Available: https://doi.org/10.1016/B978-0-12-805390-4.00009-1
[52] J. R. Lewis, "Measuring perceived usability: The csuq, sus, and umux," vol. 34, no. 12. Taylor & Francis, 2018, pp. 1148–1156. [Online]. Available: https://doi.org/10.1080/10447318.2017.1418805
[53] J. Nielsen. Usability 101 - introduction to usability. Accessed: 14/06-2024. [Online]. Available: https://www.nngroup.com/articles/usability-101-introduction-to-usability/
[54] R. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. USA: McGraw-Hill, Inc., 2009.
[55] Google. Mediapipe model maker. Accessed: 14/06-2024. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/model_maker
[56] XState. Xstate - javascript state machines and statecharts. Accessed: 14/06-2024. [Online]. Available: https://xstate.js.org/
[57] J. Nielsen. 10 usability heuristics for user interface design. Accessed: 14/06-2024. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/
[58] M. Sipser, *Introduction to the Theory of Computation*. Cengage Learning, 2013.
[59] Electron. Build cross-platform desktop apps with javascript, html, and css. [Online]. Available: https://www.electronjs.org/

Appendix A
Questionnaire

This shows the CSUQ questionnaire that the users answered after each usability test.

		Strongly disagree								Strongly agree	NA
1	Overall, I am satisfied with how easy it is to use the hand poses		○	○	○	○	○	○	○		○
2	Overall, I am satisfied with how easy it was to use the laser pointer		○	○	○	○	○	○	○		○
3	I can effectively complete tasks using this model		○	○	○	○	○	○	○		○
4	I am able to complete the tasks quickly using this model		○	○	○	○	○	○	○		○
5	Using this model to control the PowerPoint was pleasant		○	○	○	○	○	○	○		○
6	The interface of the system is pleasant		○	○	○	○	○	○	○		○
7	The system had the functions and capabilities I expected it to have		○	○	○	○	○	○	○		○
8	The poses used to control the PowerPoint made sense		○	○	○	○	○	○	○		○
9	The poses used to control the PowerPoint worked as I expected		○	○	○	○	○	○	○		○
10	Overall, I am satisfied with the model		○	○	○	○	○	○	○		○

Fig. 23. The CSUQ questionnaire used for the usability tests, which were also used for our previous findings [19]

Appendix B
Time data with outliers

This appendix shows the time data with outliers.

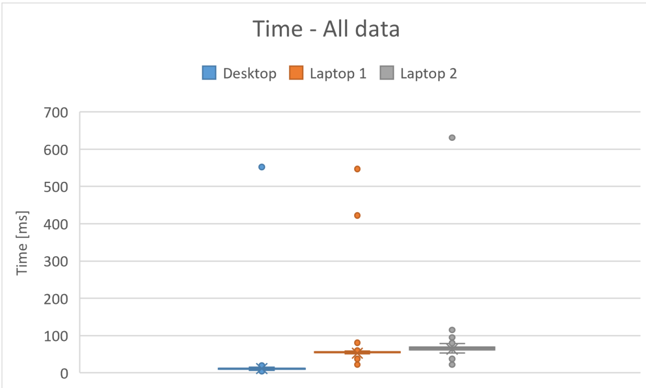

Fig. 24. Time data as box plots for each device with outliers.

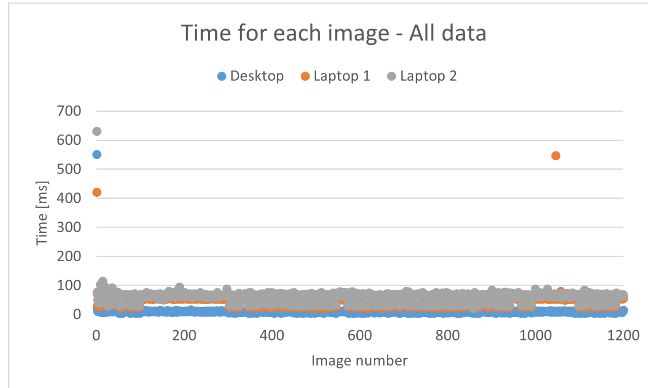

Fig. 25. Time data for each image with outliers.

Appendix C
PowerPoint Slides

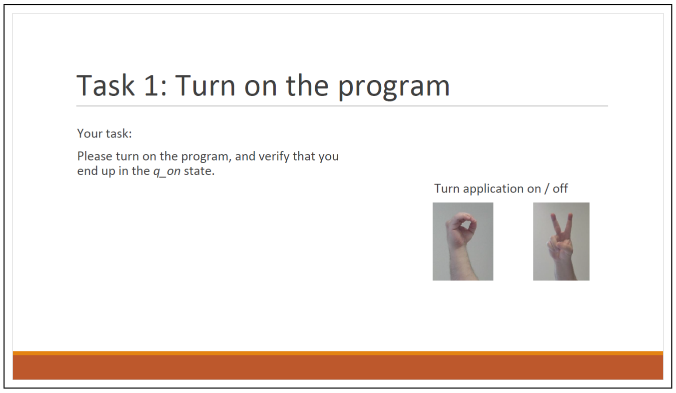

Fig. 26. The Slide presenting Task 1

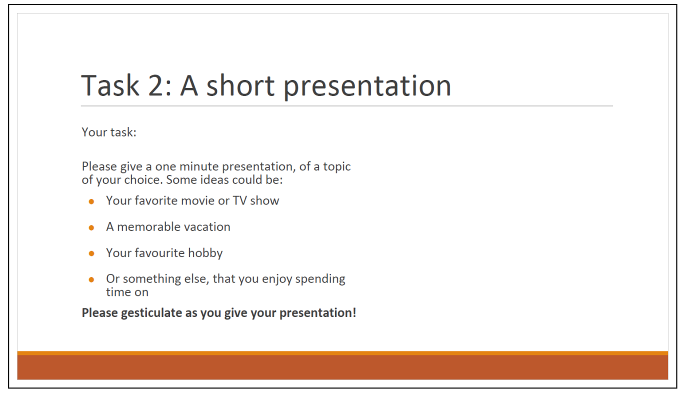

Fig. 27. The Slide presenting Task 2

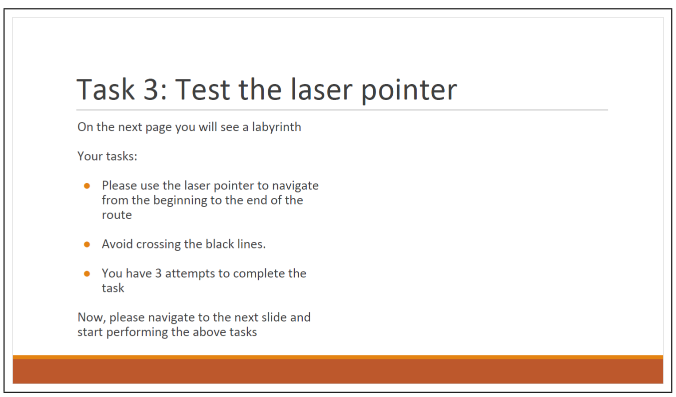

Fig. 28. The Slide presenting Task 3

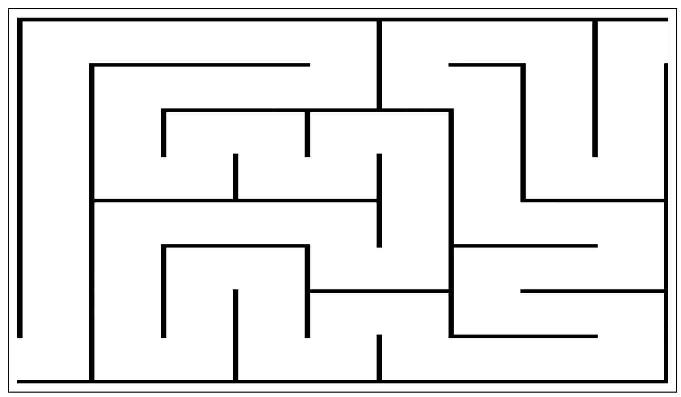

Fig. 29. The Slide showing the route for Task 3

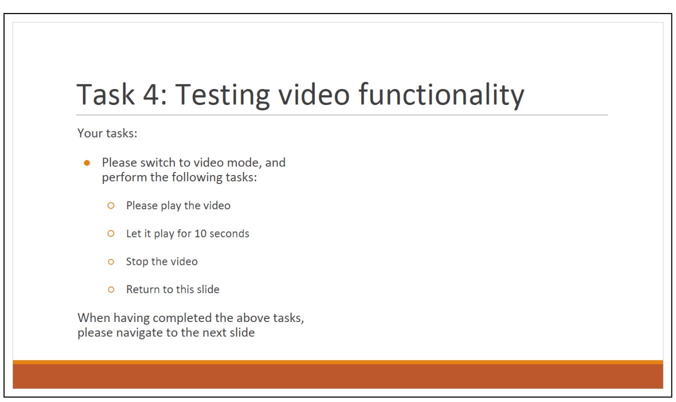

Fig. 30. The Slide presenting Task 4

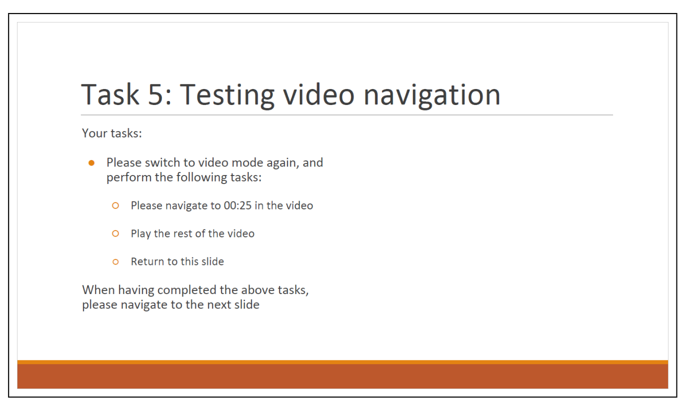

Fig. 31. The Slide presenting Task 5

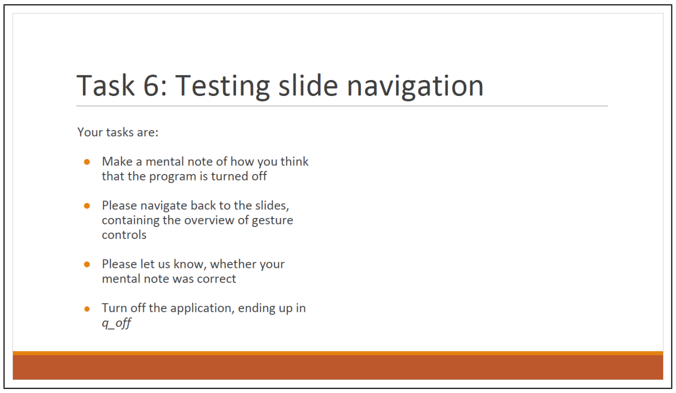

Fig. 32. The Slide presenting Task 6