NTNU

Norwegian University of
Science and Technology

Quiz Class

A Cloud-based Gaming Platform for Learning

Junjun Guo

Master of Science in Computer Science
Submission date: June 2017
Supervisor: Alf Inge Wang, IDI
Co-supervisor: Meng Zhu, IDI

Norwegian University of Science and Technology
Department of Computer Science

NTNU

Norwegian University of
Science and Technology

Quiz Class

A Cloud-based Gaming Platform for Learning

Junjun Guo

Master of Science in Computer Science

Submission date: June 2017

Supervisor: Alf Inge Wang, IDI

Co-Supervisor: Meng Zhu, IDI

Norwegian University of Science and Technology

Department of Computer and Information Science

Acknowledgements

First, with utmost heartfelt gratitude, I would like to take this opportunity to register my sincere appreciation and thanks to the staff of Norwegian University of Science and Technology, particularly to the staff in Department of Computer and Information Science.

Foremost, I would like to express my sincerely and heartily grateful to my supervisor Professor Alf Inge Wang, for the continuous support, constructive criticisms, professional, and inspirational guidance he showed me throughout my specialization project, master project, and my thesis writing, thank you for always willing to give time to guide me, read my work, give advance, and help me with testing. His enlightenment, attitude, and professionalism will inspire me lifelong.

Besides, I would like to extend my appreciation and earnest thankfulness to my co-supervisor Postdoctoral Meng Zhu, for his academic competence and his high level of professionalism with which he offered constructive suggestions, positive criticisms, and guidance throughout my specialization project, master project, and my thesis writing. His assistance, enlightenment, and advice have been boosted me truly.

During the testing process, I have met so many helpful people. The list of those who have assisted me is too long to name everyone. However, I am deeply indebted to all those who participated the Quiz Class testing, who took the time to speak to me, give feedback, suggestions, and answer the questionnaire, without them this study would not have been possible.

Finally, yet importantly, I wish to express my deepest and sincerest gratitude to my family members and friends, who has been so patient and understanding, have given so much help, support, and assistance. Especially, my deepest gratitude goes to my brother, for he took his time to contact teachers, visit schools in China, he did his best to strive for Quiz Class test opportunities. And most importantly, my heartfelt thanks go to my mother, for her never-ending belief, support and assistance, whose love and encouragement will abound forever.

It is a great pleasure to thank all people who have helped and inspired me during my study.

Trondheim, June 2017, Junjun Guo

Abstract

Digital games are very popular today and people spend a lot of time playing. Because it is fun and enjoyable to play, people get engaged and motivated while playing and would love to play more. The key to effective learning is to be engaged and motivated. But in traditional learning, it is usually hard to get learners engaged and motivated. What if we can introduce an engaging and innovative game concept into learning? What if some of the time people spend on playing digital games is used on learning? What if learning is just as engaging and motivating as playing?

The aim of this study is to investigate: how a game concept can be used in doing homework/assignments; how such an application can be designed, implemented with today's technology; how the game-based learning software (Quiz Class) can be easily accessed, and how Quiz Class can influence students learning motivation.

This project includes three parts. The first part studies the relevant research, applications, technologies, and develop a game-based learning concept, design a software architecture that combines gameplay, learning, and social interactions into an application. The second part implements a digital game-based learning application (Quiz Class), deploy Quiz Class in the cloud, test it on end-uses, and improve it. The third part finds real classes, test, collect data and investigate how Quiz Class can influence students' learning motivation.

Finally, a game-based learning concept and a software architecture have been designed, Quiz Class has been implemented and tested. Quiz Class had 173 registered participants from a preliminary test and final test. The preliminary test had participants as groups and individuals. The final test was taking place in the software architecture course at NTNU, with 116 participants and 78 valid questionnaire responses. Multiple sources including interviews, observations, survey were used to collect data.

The Quiz Class testing results showed that game-based learning application can be designed, implemented, and easily accessed. A game concept can be used in learning, and bringing a digital game into learning can boost learners' motivation, engagement, and make learning fun. In addition, between 29% to 41% of students did the assignments again to improve themselves. Some students even repeated the assignment 9 times to make progress, and yet all the assignments were optional. At last, Quiz Class analysis tools for teachers were viewed helpful to improve teaching efficiency.

Table of Content

List of Figures

List of Tables

Part I: Introduction and Research Design

Part I introduces the project and the research design.

1 Introduction

Every week, people spend 3 billion hours playing digital games (McGonigal 2010). What if some of this time and energy could be spent on education and learning.

The goal of this project is to design a game-based general purpose learning application (Quiz Class), which combines gameplay, learning, social interactions into a software application. The curricular content used for the game can be provided by the creator or teacher. The application is mainly designed for students to do homework/assignment (after lecture), but not limited to it. Quiz Class motivate students to start doing assignment early, doing the assignment as good as they could and encourages students to redo the assignment to make progress. Quiz Class also provides feedback for students to know how they are doing. Furthermore, Quiz Class provides a feedback and analysis tool for teachers to better understand their students, e.g. what student knows, understand, and are able to do.

1.1 Motivation

The traditional way of learning is not well integrated into the learning experience with today's technology (Moeller and Reitzes 2011). When students are doing their homework in a traditional way, they are often not very engaged and motivated with what they are doing, because it is not fun, not social, and they do not gain attentions from other students when they have done great with their homework. In addition, for those who do the homework, and want to know how they are doing, they cannot get feedback until teachers correct their homework. Feedback is one of the most powerful influences on learning and achievement. Studies indicate that immediately feedback is more effective than feedback is given in a few days, weeks or even longer time (Hattie and Timperley 2007).

Most teachers are still correcting tons of student homework by hand, which is a lot of work. Therefore, teachers are usually very busy. Furthermore, when they finish with correcting, more time is used to analyze the results, finding out which tasks students made more mistakes with. Moreover, for some of the student answers, even it was correct, but it is very difficult for teachers to know whether it was a student's lucky guess or was the student confident with the given answer. For teachers, knowing what their students know, understand, and are able to do, respond directly to more effective teaching and students better learning (Angelo and Cross 1993).

This project is addressing the problems described above which students and teachers are facing.

For students, this project first aims to integrate technology into their learning experience, which can equip students to independently organize their learning process. Instead of being passive recipients of information, students using technology become active users (Moeller and Reitzes 2011). Students can incrementally access learning content, and have the flexibility to direct their learning process individually. Second and most importantly, this project aims to bring game elements, competition, and social features into the application, which will make doing homework more fun, and enjoyable.

For teachers, firstly, this project can save teacher's time on correcting students' homework, which can be used to help students with more important things. Second and more importantly, it provides immediate statistical analysis of the homework results. Moreover, the application will provide information regarding how certain the student thinks about the answer they give. Thus, teachers know more about problems students have, how much, and how well their students learnt. Most importantly, teachers can use the information to refocus their teaching, and help students to study more efficiently.

1.2 Project Context

Digital devices are very common today, and they are used both in and outside the classroom. Most of the educational institutions allow students to bring their own digital device, and the devices are increasingly being integrated into the classroom and learning experience (Wang, Elvemo et al. 2014).

Students believe that digital devices are important for academic purposes, and digital devices being used for this purpose are growing; more importantly, student wishes to use more (Eden 2012). Educause Center for Applied Research did a survey on Mobile IT in higher education. Which states that students are driving the adoption of mobile computing devices. In higher education, 67% of surveyed students believe mobile devices are important to their academic activities and academic success. Substantial, but not surprisingly, smartphones are being used for academic purposes are growing from 37% (in 2011) to 67% (in 2012). More importantly, game-based learning is at the top of the list of what students wish their instructors use more. The study also points out that Academic success is underpinned by e-mail, face-to-face interaction, and using the course/learning management system (Eden 2012).

At last but not least, research shows that when a classroom application is well planned and well integrated, it will boost student engagement, learning, creativity, focus, attention, and social interaction (Wang, Elvemo et al. 2014).

More and more game-based software applications are used for academic purposes, which provides a better academic result. It is increasingly being used and is preferred by students. At Norwegian University of Science and Technology (NTNU) many learning purpose applications and studies like Kahoot (Wang and Lieberoth 2016), WordCloud, Post-It, Categories (Wang, Elvemo et al. 2014), Lecture Quiz (Wang, Øfsdahl et al. 2007), (Wu, Wang et al. 2011) have produced good and positive results. But most of these applications and research have focused on the classroom (during lectures), where everyone sits together and uses the application simultaneously. While this project is aiming to design a game-based application, which mainly focuses on after lectures usage, e.g. can be used for students to do homework. The end-users can use the application anywhere anytime, there are no time or location limitations.

1.3 The Outline of the Report

This section functions as a reader's guide, which represents a quick overview of the report. This report is organized into six parts: Part I Introduction and Research Design; Part II Prestudy; Part III Quiz Class; and Part IV Experiment; Part V Evaluation and Conclusion.

Part I – Introduction and Research Design

The first part introduces the project and the research design.

Chapter 1: An introduction of the project. Which covers the aims, motivations and the context of the study.

Chapter 2: Covers the research questions, overview of the research process, research method, and Ethical considerations.

Part II – Prestudy

The second part presents game and learning, previous studies, related work, and the chosen technologies for the project.

Chapter 3: Covers the growing of digital games, game theory, game design guidelines, most importantly the digital game-based learning.

Chapter 4: Covers related research has been done, related games has been developed and a summary with a comparison to this project.

Chapter 5: Covers the chosen front-end, back-end, and cloud technologies for this project.

Part III – Quiz Class

The third part presents Quiz Class's game concept, software requirements, UI Design, software architecture, and the software implementation.

Chapter 6: Covers the game concept of Quiz Class, and how it works.

Chapter 7: Covers the Use Cases, functional and quality requirements of Quiz Class.

Chapter 8: Covers the Quiz Class User Interface design, and how it works.

Chapter 9: Illustrates the software architecture in physical view, development view, process view, and logic view.

Chapter 10: Covers the front-end, back-end, client and server communication, and cloud implementation of Quiz Class.

Part IV – Experiment

The fourth part presents the experiment of the preliminary test, the final test, and the testing results.

Chapter 11: Covers the Quiz Class preliminary test.

Chapter 12: Covers the Quiz Class final test.

Chapter 13: Covers the Results of the testing.

Part V – Evaluation and Conclusion

The fifth part evaluates the project, presents the conclusion, limitations, challenges, and the further work of the study.

Chapter 14: Covers the evaluation of the project.

Chapter 15: Covers the answer of the research questions, and the conclusion of the study.

Chapter 16: Covers the discussion of the further work.

2 Research design

This chapter provides a detailed outline of how the investigation will take place, presents detailed explanations of how the whole processes of data collection involved in this study will be organized and carried out. The main themes covered in this chapter include research questions, the methods used for data collection, and the study process in addition to sampling, limitations, ethical considerations and data analysis.

This study is a continuation from TDT4501 (Computer Science, Specialization Project, Autumn 2016), where I developed a game concept for learning, designed software architecture for the game, and tested if I can implement it with today's technology. To make the study complete, most of the content including research questions from the previous project are included.

2.1 Research questions

The objectives of this study are:

I. develop a game concept for learning;
II. design, and implement a Digital Game-Based Learning[1] (DGBL) application (Quiz Class);
III. experiment and discover how Quiz Class influence the learning experience.

The research goal of this study is to design and develop an application (Quiz Class) to make learning fun; investigate how Quiz Class influences teachers' work and students' study. The goals can be decomposed into research questions as below:

RQ1. How can we make learning fun?

I will study the relevant research, theories, experiments to find out how to make learning fun, and how to use a game concept in learning.

[1] Digital game-based learning (DGBL) is a competitive activity in which students are set educational goals intended to promote knowledge acquisition. The games may either be designed to promote learning or the development of cognitive skills, or else take the form of simulations allowing learners to practice their skills in a virtual environment (Erhel, S. and E. Jamet (2013). "Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness." Computers & Education 67: 156-167).

RQ2. How can we design and implement a game-based learning application?

I will study the relevant works of literature, game-based learning applications, and technologies, develop a game concept, create software requirements, design UI and software architecture, implement a game-based learning application (Quiz Class).

RQ3. How can we make Quiz Class easy to access?

Accessibility is important for Quiz Class, as every student should be able to do their homework without limitations. I will study relevant technologies to find a good and affordable solution for Quiz Class, implement and test the solution.

RQ4. How do students being affected by using Quiz Class?

Quiz Class will be tested on real students. The data will be collected by observations, interviews, and participants' survey. The following will be mainly focused:

RQ4.1. Students react on game-based application being used for their study.

RQ4.2. Students interest and motivation in doing homework.

RQ4.3. Students doing homework again.

2.2 Overview of the research process

This study will be managed in an iterative process, as shown in Figure 1. I will start with a simple idea, and by iterative development, the research goal will be formulated. And the research questions will be developed iteratively. Literature will be reviewed, with a focus on related studies, research, and applications being developed. Technologies will also be investigated, to see what technologies may best fit Quiz Class application.

As the whole process is iterative, the ideate[2] process will be developing and updating all the time, some of the ideas will not be presented or implemented in Quiz Class, because of the time and resource limitations. The game concept will be developed after several iterates. Software architecture will be developed after software requirements are made. The software prototype will be developed together with testing. At the beginning of implementation, the unit test will be most used, and then component test, system test, performance test. When

[2] Ideate is the mode of the design process in which you concentrate on idea generation (DesignThinking (n.d.). "Wikipedia: Design thinking." Retrieved 25.05, 2017).

most of the important functionalities are implemented, the test will be moved from running server and client in local computer to host front-end and back-end on the cloud. And different cloud hosting will be used to test the performance and accessibility. Few end-user tests will be made to decide which cloud host to use. And after cloud host is decided, the preliminary test will be made, which will focus on usability, finding bugs, improve UI and functionalities. The questionnaire will be answered by users at the final test.

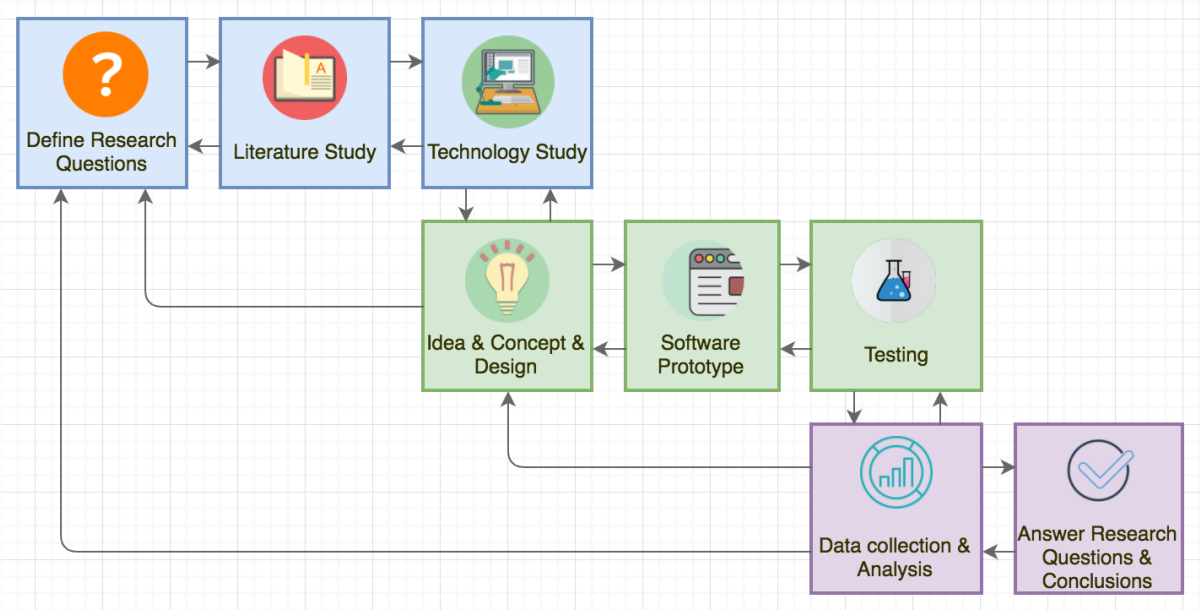


Figure 1 Research Process

Participant survey, participant observations, and informal interviews will be used to collect data. Quantitative and qualitative research methods will be used to analyze the data. Finally, I will answer the research questions and conclude the study.

2.3 Research method

The main method of this study is participants survey, using a structured questionnaire which contains predominantly forced-choice questions. Participant observations and informal interviews will be used to get extra data. Seeing that participant observations offer another angle of view, and informal interviews are more relaxed, personal, and flexible, which may provide auxiliary information to increase the reliability of the results. On the other hand, different sources of data can help researcher enrich data, and in addition, various sources of data can increase reliability. In addition, observations and informal interviews will be used in the preliminary test, which aimed to help improve Quiz Class design, usability, and functionalities. Notes will be taken after informal interviews and participant observations.

2.3.1 Participant Survey

To gain a more accurate sample from end-users, a participant survey will be performed for this research. Questionnaires with multiple choices will be provided to the users. The purpose of participant survey is to produce quantitative statistic descriptions, which will be used for quantitative data analysis of how end-users respond to Quiz Class.

2.3.2 Participant Observations

DeWalt & DeWalt defined participant observation as a method in which a researcher takes part in the daily activities, rituals, interactions, and events of a group of people as one of the means of learning the explicit and tacit aspects of their life routines and their culture (DeWalt and DeWalt 2002). As a participant observer in this study, I will follow the end-user, to observe how they use Quiz Class, and in what environments and routines. The participant observation may offer opportunities for the researcher to gather live data from live situations, and researcher can have opportunities to look at what is taking place in the situation rather than at second hand, to see things that might otherwise be unconsciously missed.

2.3.3 Interviews

The research interview is based on the conversation of daily life and is a professional conversation; it is an inter-view, where knowledge is constructed in the inter-action between the interviewer and the interviewee (Kvale and Brinkmann 2009).

> "I want to understand the world from your point of view. I want to know what you know in the way you know it. I want to understand the meaning of your experience, to walk in your shoes, to feel things as you feel them, to explain things as you explain them. Will you become my teacher and help me understand?" (Spradley 2016).

Spradley has summarized the stance qualitative researchers take in relation to their informants, and what the qualitative research interview attempts to do. Hatch has further broken interviews down into three basic types: informal, formal, and standardized (Hatch 2002).

Informal interviews will be used in this study. As stated by Fetterman, "informal interviews are the most common in ethnographic work. They seem to be casual conversations, but whereas structured interviews have an explicit agenda, informal interviews have a specific but implicit research agenda" (Fetterman 2009). In this study, informal interview questions will

be asked accordingly, no audio or video record will be used, but notes will be taking. As said by Hatch, informal interviews require interviewer to be a good listener and to create pertinent questions on the spot. Because informal interviews are often sidebars to the real action, they are usually not the place for taking out a tape recorder or trying to write verbatim notes (Hatch 2002). In this study, the informal interviews will be used in a more planned way, the purpose is to focus on the DGBL topic. And I will try to do informal interviews right after observations to collect as much information as possible.

In this study, interviews will be used to understand how end-user experiences, reacts and feels about Quiz Class, and how Quiz Class may influence the students' learning motivation. Which may be difficult to get from survey or observation. To achieve it, the interviewer needs to learn from the interviewee. To be better prepared, some interview questions are planned or predefined, but will not be used as guidance for informal interviews.

2.4 Ethical Considerations

In this study, there will be no potential sensitive issues. However, ethical considerations are needed through the whole research process. To get a deeper understanding of how Quiz Class application influences its users, the researcher needs to study the application's end-users. And participant interviews, observations, and surveys will be used. This process may become personal and ethical issues can be very important. When the dilemmas issues arise between the pursuit of objective truth and ethics, Walford, Tucker, et al. (2010) has highlighted the rules for the researcher to follow:

> "Ethics is a matter of principled sensitivity to the rights of others. Being ethical limits the choices we can make in the pursuit of truth. Ethics say that while truth is good, respect for human dignity is better, even if, in the extreme case, the respect of human dignity leaves one ignorant of human nature" (Walford, Tucker et al. 2010).

The data collection process will collect data that is necessary, and participants should not be identified. The survey only collect participants age and gender. The informal interview and observation notes do not record participants' real name or any identical information.

Part II: Prestudy

Part II presents game and learning, previous studies, related work, and the chosen technologies for the project.

3 Game and Learning

This chapter starts with the growths of digital games, then describes what makes a game fun to play, and most importantly, how to make things fun to learn. The primary goal of this chapter is to provide a set of guidelines for this project to design a game-based learning application, which can make learning fun and enjoyable. This chapter also discusses the theoretical basis for designing, organizing, and understanding the collected data.

3.1 The rising of digital games

To better understand the present of digital games, we need to know its past. This section briefly describes the rising of digital games historically, in the perspective of hardware, software, businesses and its players.

In 1958, nuclear physicist William Higginbotham developed the first educational interactive computer game, in order to instruct players about the effects of gravity (Overmars 2012). The digital games have changed ever since.

Hardware is changing all the time, e.g. more memory, more computing power, which had a huge effect on what is possible in digital games. Along with hardware updating, which enables game designers to make more various and fascinating games.

Software tools also changed, e.g. 50 years ago, game developers need to write every line of code themselves often in assembly language. Nowadays software frameworks, APIs, and extensive game engines and other middleware packages are available, which allows creating much more sophisticated games.

Game businesses changed over the years. Initially, games were often developed by individuals, who did game design and programming with minimal budgets. Nowadays, a game can be developed by teams with very various team members, and budgets may be up to tens of millions of dollars. Now the businesses are changing again: The big growth of mobile games for smartphones, tablets, and other mobile devices are taking the markets. The mobile games may be created by smaller teams with low budgets and use self-publishing systems like Google Play (GooglePlay n.d.). In the last few years the game industry has produced revenue greater than the movie industry, and the development rate has been one of the fastest growing in the United States economy (Ampatzoglou and Chatzigeorgiou 2007).

In the old days, young males are dominating players in digital games, which is almost entirely changed in recent years. At the present, the number of female players is near the number of male players, and the players' age ranges from 4 to 100 (Overmars 2012).

Today might be the best time for innovators to design and create digital games, especially Digital Game-Based Learning (DGBL). Since the hardware is powerful and easy to access, software is much easier to develop, and various business models give everyone opportunity to publish their creations, most importantly the game consumer is huge and covers all ages.

3.2 What makes game fun?

A central characteristic of games is that they are fun and a source of enjoyment. Enjoyment is a sense of achievement, that occurs, when one's skills are matched with the task's challenges (Garris, Ahlers et al. 2002), When a good computer game is being played, the player is often focused, engaged, and the player is so enjoyed that he or she may lose track of time, and may even forget he or she is playing a game.

Player enjoyment is the single most important goal for computer games. If players do not enjoy the game, they will not play the game (Sweetser and Wyeth 2005). Malone thinks that the essential characteristics of a good computer games and other intrinsically enjoyable situations can be organized into three categories: Challenge (providing a goal whose attainment is not guaranteed), Fantasy (images of physical objects and social situations and settings not found in the real world), and Curiosity (the motivation to learn independent of the attempts at reaching a specific goal or fulfilling a fantasy) (Malone 1980).

3.2.1 Challenge

Individuals desire an optimal level of challenge (Garris, Ahlers et al. 2002), goal must be provided in order to make computer game challenging (Malone 1980), one of the most robust findings in the literature on motivation is that clear, specific, and difficult goals lead to enhanced performance (Locke & Latham, 1990) cited in (Garris, Ahlers et al. 2002), clear goals should be provided at appropriate times (Sweetser and Wyeth 2005), multiple level goals makes a game even better. In addition, a performance feedback needs to be provided at appropriate times (Sweetser and Wyeth 2005), so the end-user can judge how well they are achieving their goals. The outcome of reaching the goal must be uncertain, which is not

predictable by users, a game would be boring if the player can predicate what will happen (Malone 1980).

Achieving success in challenging games and activities makes people feel better. Failure in challenging games has the opposite effect, which can decrease the desire to play the game, reinforces the idea of variable difficulty, better feedback can lessen negative effects on self-esteem (Malone 1980).

3.2.2 Fantasy

Fantasies often make computer games more interesting. Games that include fantasy show or evoke images of physical objects or social situations not actually present (Malone 1980). Some research indicates that instructional content that is embedded in fantasy contexts leads to greater student interest and increased learning (Cordova & Lepper, 1996; Parker & Lepper, 1992) cited in (Garris, Ahlers et al. 2002).

Malone suggests a relatively easy way to increase the fun of learning: take an existing curriculum and overlay it with a game in which the end-user progresses toward some fantasy goal, or avoids some fantasy catastrophe, depending only on whether the end user's answers are right or wrong (Malone 1980). Which is interchangeable across subjects. Materials may be learned more willingly when presented in an imagined context, that is of interest than presented in a generic or decontextualized form (Garris, Ahlers et al. 2002).

3.2.3 Curiosity

> "The important thing is not to stop questioning; never lose a holy curiosity". -- Albert Einstein

Curiosity is one of the primary factors that drive learning (Garris, Ahlers et al. 2002). Curiosity is important for engagement. When a player or learner is curious about a game or topic, they will pay more attention and play more or learn more. To provoke the end-user's curiosity, the activity must provide an environment that has an optimal level of informational complexity (Piaget, 1951) cited in (Ebner and Holzinger 2007). The environment being provided should be neither too complicated nor too simple with respect to the end-user's existing knowledge (Malone 1980).

Malone (Malone 1980) emphasizes that the way to engage end user's curiosity is to present enough information, where the end-user knows enough to be able to anticipate, but their

existing knowledge seems incomplete, inconsistent. And to engage a learner's curiosity, feedback should be surprising. Also, the feedback should be constructive, where feedback can help end-user to see, how to improve their knowledge, to become more complete, consistent, to be educational.

Make learning fun for the end-users is important for this project. Malone's (Malone 1980) challenge, fantasy, and curiosity are considered and carried out throughout the project design. In this project, challenges can be adjusted by the questions being asked, and the alternatives being provided. Fantasy is not very strong in this application, but social interaction may be strong enough for the end-user. Curiosity can be triggered while playing.

3.3 GameFlow

Enjoyment is the same, regardless of social class, age, or gender. A key concept that frequently emerges in the literature is that of 'flow', first discussed by Csikszentmihalyi. 'flow' is described as an experience "so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous" (Csikszentmihalyi 2002). How to design and evaluate a game that is fun and enjoyable? Based on Csikszentmihalyi's Flow, Sweetser and Wyeth (Sweetser and Wyeth 2005) has introduced GameFlow.

The GameFlow model consists of eight core elements: concentration, challenge, skills, control, clear goals, feedback, immersion, social interaction (Sweetser and Wyeth 2005). Each element includes a set of criteria for achieving enjoyment in the games. The GameFlow model can be used as a designing tool for game design, can also be used as an evaluation tool for reviewing games and identifying issues that could affect game enjoyment. The core elements of the GameFlow model are indicated by Sweetser and Wyeth (Sweetser and Wyeth 2005) and listed in Appendix: GameFlow.

The GameFlow is a great game design guideline and evaluation tool, it is considered throughout the design process of this project and will also be considered to evaluate the end-user enjoyment of this project.

3.4 Digital Game-Based Learning

Motivation is important for successful learning. Motivated learners are interested in and enjoy what they are doing, they try hard, and persist over time. Motivated learners are easy to

recognize, but hard to create (Skinner and Belmont 1993) cited in (Garris, Ahlers et al. 2002). In traditional learning, much of the content that needs to be learned is not directly motivating the learner, and the problem for teachers, trainers, and educators is to get the learners motivated (Prensky 2003). On the other hand, good computer games have the ability to keep people engaged, motivated, persist and even begging for more.

Nowadays, computer games have already become a dominating form of entertainment due to their higher level of attractiveness to game players (Magerkurth, Cheok et al. 2005). When a good digital game is being played, the player is often interested, competitive, cooperative, results-oriented, actively seeking information and solutions (Prensky 2003), those attitudes are opposite toward school for most of the players.

> "Today's students are no longer the people our educational system was designed to teach" (Prensky 2001).

There has been a major shift in the field of learning from a traditional, didactic model of instruction, where "learning by listening" to a learner-centered model that emphasizes a more active learner role (Garris, Ahlers et al. 2002).

Playing games is increasingly linked to learning. Several models have been developed that identify distinct learning outcomes which playing digital games can have (Connolly, Boyle et al. 2012). With modern technology, learning can be interesting, fun and enjoyable as well. And merge the content of learning with a game to motivate learners is the tending. Where young people's intrinsic motivation towards games contrasts with their often noted lack of interest in curricular contents (Prensky 2003), may be fixed or integrated with games, which can create motivated learners.

Games are effective not because of what they are, but because of what they embody and what learners are doing as they play a game (Eck 2006). Empirical evidence already showing that games can be effective tools for enhancing learning and understanding of a complex subject (Garris, Ahlers et al. 2002). However, the motivation of games could be combined with curricular contents into what Prensky (Prensky 2003) calls 'Digital Game-Based Learning' (DGBL).

Eck (Eck 2006) has pointed out three factors that the DGBL is interested in the widespread public:

- researchers have already published dozens of essays, articles, and mainstream books on the power of DGBL.
- “Net Generation”, or “digital natives” who have become disengaged with traditional instruction. They require multiple streams of information, prefer inductive reasoning, want frequent and quick interactions with content.
- The popularity of the games is increasing.

Modern theories of effective learning suggest that learning is most effective when it is active, experiential, situated, problem-based and provides immediate feedback (Boyle, Connolly et al. 2011). Games appeared to offer activities which have these features. Research have also pointed out that, play is a primary socialization and learning mechanism, common to all human cultures and many animal species (Eck 2006). In DGBL, the learning effect is much higher in the emotional and motivated situation, even players not willing to learn may do so indirectly (Ebner and Holzinger 2007).

4 Related Work

There are many studies that evaluate digital game-based learning, many products that aim to make learning fun. This chapter discusses the previous research and existing applications have been done in relation to digital game-based learning (DGBL). Comparison with existing works and discussion of the significance of this project is also presented.

4.1 Related research

E-learning has a bright future in both education and industry. Clark Aldrich in his book "Simulations and the Future of Learning: An Innovative (and Perhaps Revolutionary) Approach to e-Learning (Aldrich 2003)" contributed the information and perspective to understand, design, build, and deploy computer simulations, he also created a detailed case study of the creation and deployment of an e-learning simulation using a modern computer game.

Games are popular today, and children spend ever more time on games. Marc Prensky in his book "Don't Bother Me Mom—I'm Learning! (Prensky 2006)" argued that children "are almost certainly learning more positive, useful things for their future from their video and computer games than they learn in school!" (Prensky 2006). Prensky claimed that doctors who played video games earlier in their lives made almost forty percent fewer mistakes in surgery! Because of the controls of laparoscopic instruments resemble a video game controller, and the entire surgery is seen only on a computer monitor (Prensky 2006).

Ebner and Holzinger (Ebner and Holzinger 2007) did an experimentation to gain insight into whether and to what extent, online games have the potential to contribute to student learning in higher education. A user-centred game-based learning application was successfully implemented and tested in higher education in civil engineering. The calculation of the internal forces is necessary for every student of Civil Engineering. The purpose of the online game was to help learners in the field of static determinate systems by training them to distinguish carefully the correct internal forces from wrong solutions within extremely short time. The game was tested during the Structural Concrete course at the University of Technology of Graz from November 2003 to January 2004. Parallel to the students of Civil Engineering from other universities was invited to participate as well. A high score and time limits system were used to motivate students, which lead the learner to play again and again.

And they conclude that the experimental support the efficacy of game playing, students enjoyed the game-based e-learning (Ebner and Holzinger 2007).

Papastergiou (Papastergiou 2009) has pointed out that, even though games are believed to be motivational and educationally effective, but the empirical evidence to support this assumption is still limited, particularly regarding the effectiveness of games for concrete educational purposes. Prior studies have focused more on motivational aspects than on curricular content aspects and core academic benefits.

To be more curricular-relevant and academic beneficial, Papastergiou (Papastergiou 2009) did a digital game-based learning research on a computer game, used for learning computer memory concepts. The computer game was designed according to the curricular objectives, the subject matter of the Greek high school Computer Science (CS) curriculum. The result shows that within high school CS, educational computer games are effective and motivational learning environments for all students, regardless of genders.

The game used in Papastergiou's research was specified in Greek high school CS (Papastergiou 2009), which has very limited usage and cannot be used in e.g. other subjects or courses. A more general purpose Digital Game-Based Learning application Lecture Quiz (Wang, Øfsdahl et al. 2007) was created, and Lecture Quiz 2.0 (Wu, Wang et al. 2011) was implemented with improved functionalities. Lecture Quiz has a new game concept and can easily be integrated into lectures. Lecture Quiz can promote strong student participation, which enables variations in how lectures can be taught. The Lecture Quiz game runs on the teacher's PC and projected on a large screen, whereas the students will interact with the game using their own mobile phones or laptops. The game was easy to use and set up using existing equipment in the classroom, the game also increased learning, motivation, and student attending to the lecture (Wang, Øfsdahl et al. 2007).

Socrative shares many characteristics as Lecture Quiz. Socrative is a student response system (SRS) that enables teachers to engage their classrooms through a series of educational exercises and games via smartphones, laptops, and tablets (Socrative n.d.). Socrative provides a real-time formative assessment to students' answering results. A study has shown that the student response system (Socrative) help students to understand concepts, they facilitate the argumentation and the exchange of options, students turns out to be interested in the classroom, it motivated the students and increased opportunities for active learning (Méndez and Slisko 2013).

Wang (Wang 2015) has summarized three categories where education can be integrated into games:

1. Traditional exercises or tasks can be replaced by letting students play motivating games, which provide opportunities for teachers to monitor their students' progress in real time. e.g. Socrative (Méndez and Slisko 2013), this project.
2. Game development can be used to learn other subjects like design patterns, literacy, software architecture, computer science, mathematics, and physics. e.g. Papastergiou's (Papastergiou 2009) game for Greek high school CS.
3. Games can be integrated into a traditional classroom lecture to improve learning, motivation, and engagement (Wang 2015). e.g. Lecture Quiz (Wang, Øfsdahl et al. 2007), Lecture Quiz 2.0 (Wu, Wang et al. 2011), Kahoot! (Wang 2015).

Researchers have published dozens of essays, articles, and books to demonstrate the power of DGBL, e.g. Digital Game-based Learning (Prensky 2004), Digital game-based learning: Towards an experiential gaming model (Kiili 2005), Everything Bad is Good for You: How Today's Popular Culture is Actually Making Us Smarter (Johnson 2006), Games and simulations in online learning: research and development frameworks (Gibson, Aldrich et al. 2007). Research has shown that DGBL can enhance the learning interest of student (Malone 1980), are beneficial for academic achievement, motivation and classroom dynamics (Rosas, Nussbaum et al. 2003), students using DGBL learned more and were more motivated compared to students using a similar but non-game teaching approach (Papastergiou 2009). Integrate DGBL and hands-on activities could support student scientific understanding (Anderson and Barnett 2013). The student attendance is much higher when a DGBL is used compared to when it is not used, and 94% of the students want a DGBL to be used at least once a week (Wang 2015).

4.2 Related games

Buzz!: The Schools Quiz (Buzz! n.d.) is an educational game, based on the popular Buzz! series of games, released in January 2008 for PlayStation 2. The game was developed by Relentless Software[3] in association with the UK Government's Department for Education and Skills[4].

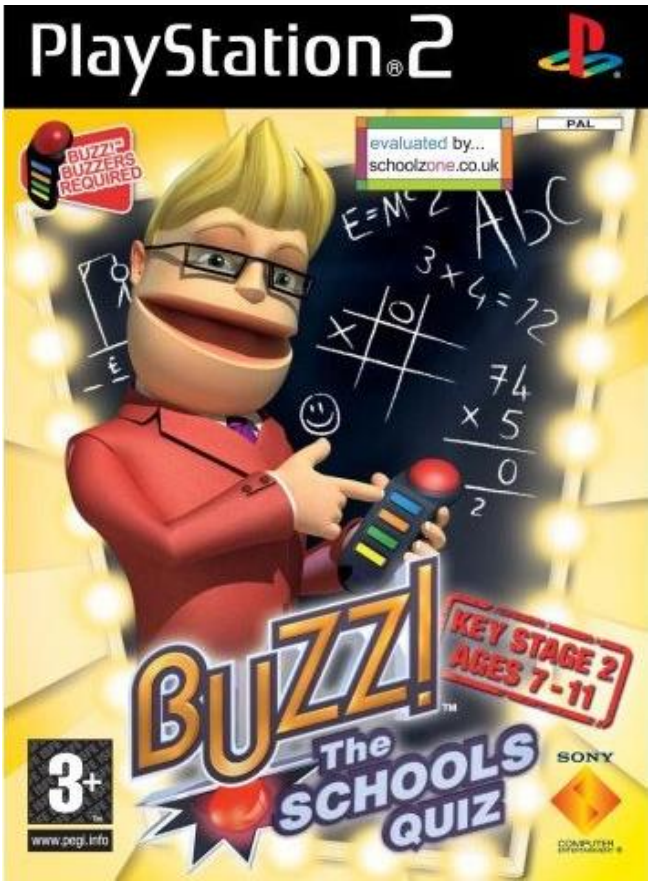


Figure 2 Buzz!: The Schools Quiz (Buzz! n.d.)

[3] Relentless Software was a British video game company formed in 2003 that was based in Brighton. The company was best known for developing 12 games in the BAFTA award winning and multi-million selling Buzz! franchise from 2005 to 2010 (Relentless Software, retrieved 15.06.2017, from https://en.wikipedia.org/wiki/Relentless_Software).

[4] The Department for Education and Skills (DfES) was a United Kingdom government department between 2001 and 2007, responsible for the education system (including higher education and adult learning) as well as children's services in England (The Department for Education and Skills, retrieved 15.06.2017, from https://en.wikipedia.org/wiki/Department_for_Education_and_Skills_(United_Kingdom)).

Figure 2 shows the cover of Buzz!:The Schools Quiz. The aim was to create a classroom application for Key Stage 2[5] pupils that would aid the teaching and testing of National Curriculum content, and be fun for kids to play. The game is featuring over 5000 questions based on the Key Stage 2 Curriculum and supports up to 8 players.

Buzz!:The Schools Quiz was viewed as a great way for both teachers and parents to make learning fun, and to become more involved in the learning process (Allan, Boxer et al. 2008).

ClassDojo (ClassDojo n.d.) is a classroom management platform for teachers, parents, and students. It was launched in 2011, which makes a classroom more fun, helps teachers save time, boosts classroom engagement, and improve student behavior.

Figure 3 ClassDojo (ClassDojo n.d.)

ClassDojo can help teachers to encourage specific classroom student behaviors, through real-time teacher-to-student feedback. The software can be used on any computer or mobile devices. It can generate behavior reports, that can be shared with parents and students. Behaviors tracked by the application include hard work, persistence, teamwork, creativity,

[5] Key Stage 2 is the legal term for the four years of schooling in maintained schools in England and Wales normally known as Year 3, Year 4, Year 5 and Year 6, when the pupils are aged between 7 and 11 (Key Stage 2, retrieved 15.06.2017, from https://en.wikipedia.org/wiki/Key_Stage_2).

and curiosity. A student can get Dojo points for different tasks. As showing in Figure 3 ClassDojo (ClassDojo n.d.), the application users colorful avatars, the behavior reports can track behavior and class trends. The analyzing tools for each student's performance are also provided for the teachers.

ClassDojo was perceived as a tool, which helps maintain student engagement and aids teachers in developing a positive and productive classroom-learning environment (Hammonds, Matherson et al. 2013).

Kahoot! (Kahoot! n.d.) is a free game-based learning platform that makes learning fun, launched in August 2013 from Norway, which temporarily transforms a classroom into a game show. The teacher is the game show host, and the students are the players. The teacher's computer is connected to a large screen, where the questions and possible answers are displayed. Students respond their answers as faster as possible by using their own digital devices, when the answer is correct, the less time a student use the more score the student get.

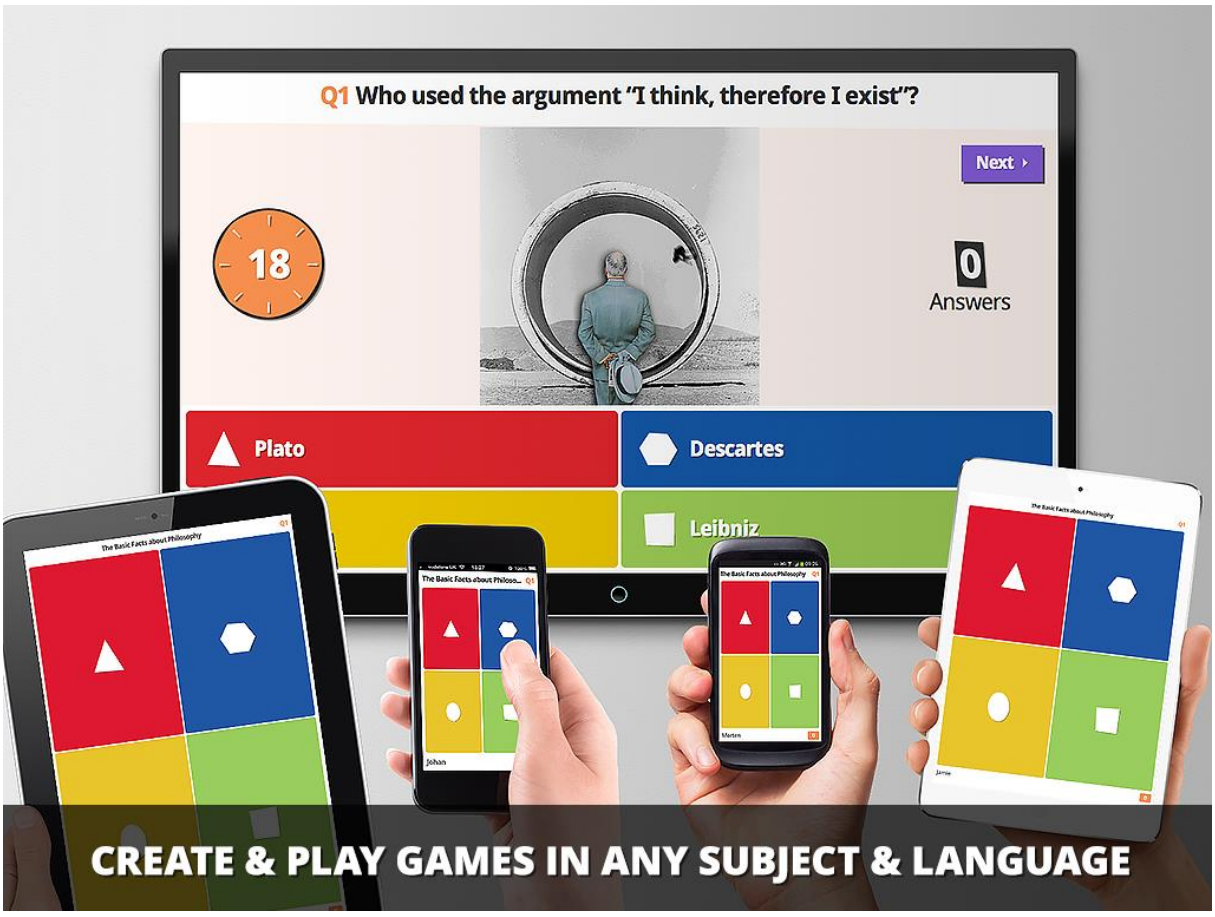


Figure 4 Kahoot! (Kahoot! n.d.)

Figure 4 shows how players interact with the game. After each question, a distribution chart of how the students have answered the question is shown. The distribution chart is useful for the teacher to get feedback, which can be used for assessment, to monitor students' progress towards learning objectives, identify strengths and weaknesses, and to identify areas where the students need help. Between each question, a scoreboard shows the nicknames and scores of the top students, and at the end of the game, a winner is announced.

Kahoot! uses playful and colorful graphics and audio to increase the engagement. And is designed to be accessible to classrooms and other learning environments worldwide, Kahoot!'s learning games ("kahoots") can be created by anyone, for any subject and for learners of any age. Kahoot! support multiple users, which can be played by an entire class simultaneously (Kahoot! n.d.).

Wang and Lieberoth did a research on the effect of points and audio in Kahoot!, which point out that the use of audio and score in Kahoot! have produced best positive effect on student concentration, engagement, enjoyment, and motivation (Wang and Lieberoth 2016). Wang did a research on "The wear out effect of a game-based student response system" using Kahoot!, which compares the results from students using Kahoot! for the first time in a single motivational lecture vs. using Kahoot! in every lecture in a class for five months. The results shows that the DGBL managed to boost students engagement, motivation, and learning in both groups (Wang 2015).

Quizlet is an online learning tool originally conceived in 2005 and released to the public in 2007. Quizlet trains students via flashcards and various games and tests. Every month, over 20 million active learners from 130 countries practice and master more than 140 million study sets of content on every conceivable subject and topic (Quizlet n.d.).

Quizlet is a memorization tool, which lets registered users create sets of terms customized for their own needs. These sets of terms can then be studied under several study modes. The Flash Cards mode is similar to paper flash cards. Users are shown a card for each term, and users can click to flip over the card, to see the definition of that term. In the Quizlet Learn model, a user can create a study set of material needed to practice and master, and set a deadline for it. Quizlet Learn will use its Learning Assistant Platform, which uses machine learning to make learning more effective and more efficient. The Quizlet Learn also provides an adaptive study plan and notifications to remind uses to study (Quizlet n.d.).

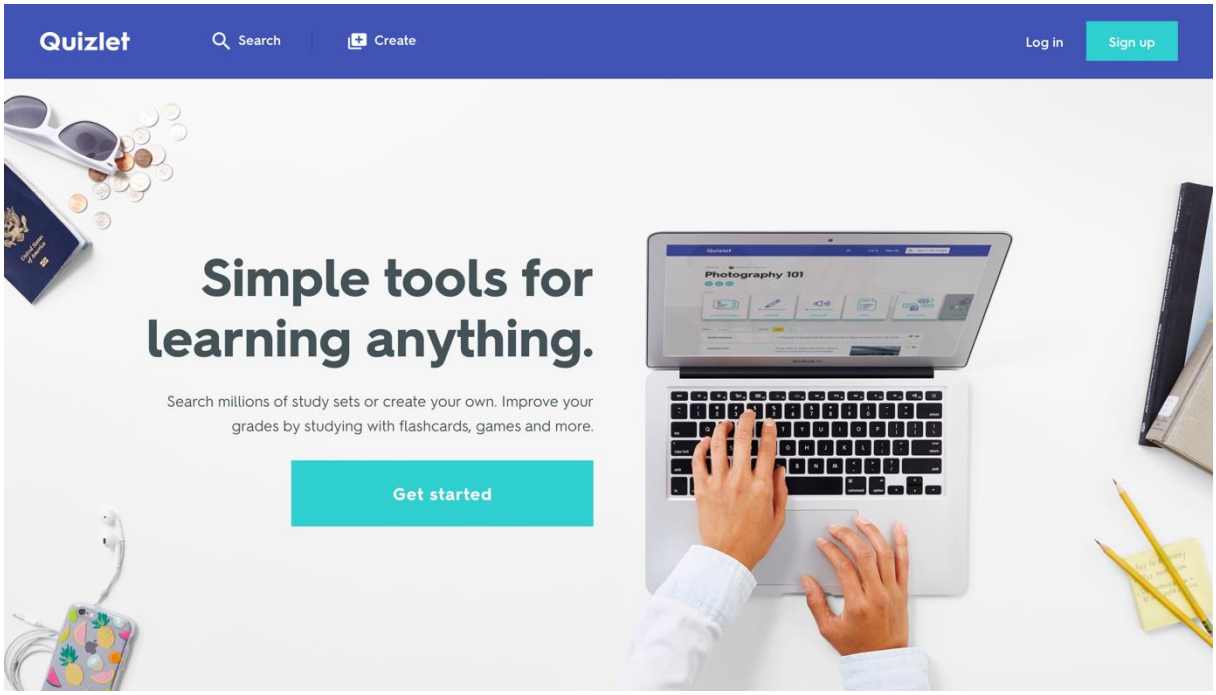


Figure 5 Quizlet (Quizlet n.d.)

Quizlet is a great tool to learn new words or memorialize contents. A study has indicated that use computer-aided instruction (Quizlet) yielded better results over teacher-student direct instruction (Adaptive Frayer Model[6]) (Zyda 2005, Vargas 2011).

4.3 Comparison with this project

This project is inspired by Buzz!: The Schools Quiz, ClassDojo, Kahoot! and many other applications and studies. Buzz! is fun, but you need a PlayStation to play it, ClassDojo is mostly a classroom management tool, and Quizlet is a memorization tool. If compare with those three, the Quiz Class is closer to Kahoot!, but combines some element from Buzz! and ClassDojo, with some other new elements, which any of those mentioned do not or can not offer.

Compare with Kahoot!, the Quiz Class is also a Digital Game-Based Learning platform, that anyone can create learning games, for any subject, and for learners of any age. The application supports multiple users to play simultaneously. The application can also be used

[6] The Frayer Model is a strategy that uses a graphic organizer for vocabulary building. This technique requires students to define the target vocabulary words or concepts, and apply this information by generating examples and non-examples. This information is placed on a chart that is divided into four sections to provide a visual representation for students (Frayer, D., et al. (1969). A schema for testing the level of cognitive mastery, Madison, WI: Wisconsin Center for Education Research).

for assessment, to monitor students' progress towards learning objectives, identify strengths and weaknesses, and to identify areas where the students need help.

This project is different from Kahoot! in many aspects. Kahoot! must be used simultaneously, e.g. during the class time, need a shared screen. While the Quiz Class will mainly focus on after class usage, the users can use the application simultaneously or asynchronously, there is no time or location limitations, namely, users can sit anywhere and gaming anytime they like. No shared screen will be needed for the Quiz Class. Kahoot! is similar to and developed from Lecture Quiz, according to Wang's (Wang 2015) three category of educational game, Kahoot is category 3, which integrated the game into a traditional classroom lecture to improve learning, motivation, and engagement. Whereas the Quiz Class was inspired by Lecture Quiz and Kahoot!, and mainly focus on Wang's (Wang 2015) first category of three, which is to replace traditional exercises or tasks by letting students play motivating games, and providing opportunities for teachers to monitor their students' progress. Kahoot! has a single score system, which is based on right answer and time used to give an answer. This Quiz Class will have a more complex score system, which also considers how certain the player is about the answer he/she give. Importantly, this project provides a much wider ranking system, the ranking list includes top score ranking, top progress ranking, earliest start and earliest finish ranking.

This project is unique, at the time of the project started, there is no other research has been done or applications have been developed using mechanics and systems used in Quiz Class.

Most learning applications are usually content based, being used mostly as contents distribution tools. e.g. ProProfs, a Knowledge Management Software for Quizzes, Test, Training, Flashcards, Knowledge Base (ProProfs n.d.).

Many games based learning Applications being too little connections with curricular. e.g. Buzz!: The School Quiz (Buzz! n.d.); duolingo (Duolingo n.d.), a language learning tool, earn points for correct answers, race against the clock, and games from Learning Games for Kids (LearningGamesForKids n.d.).

Many researchers and DGBL applications are focused on specific subjects. Papastergiou (Papastergiou 2009) showed improvement of DGBL in performance on computer memory concepts. Beale, Kato, Marin-Bowling, Guthrie, and Cole (Beale, Kato et al. 2007) found the better performance on cancer knowledge for a game playing group compared with a control

group. Miller and Hegelheimer (Miller and Hegelheimer 2006) uses authentic computer simulation games in language teaching. Akkerman and Admiraal (Akkerman, Admiraal et al. 2009) did a research on mobile and multimedia game designed for History Education, in the game studied, students explore the history of Amsterdam by walking in the city, experiencing characters, buildings, and events, while using UMTS/GPS phones for communication and exchange of information.

Many general purpose learning applications or research are mainly focused on 'during the lecture', Like Kahoot! (Wang and Lieberoth 2016), WordCloud, Post-It, Categories (Wang, Elvemo et al. 2014), Lecture Quiz (Wang, Øfsdahl et al. 2007), Lecture Quiz 2.0 (Wu, Wang et al. 2011), Poll Everywhere (Tremblay 2010).

The Quiz Class is aimed at a general purpose DGBL application, which is not limited by any subject, not limited by user groups, not limited by the time or the place for the end-uses to use. It takes the best part of Buzz!, fun to play; it has the advantage of Kahoot!, general purpose and user-defined contents; it may be used as management tools inspired by ClassDojo. And most importantly, the Quiz Class is designed to boost student or players learning motivation and results in multiple ways, which has not been designed and implemented so far, in addition, the Quiz Class can be used by teachers to track students' homework process situation, analysis the homework results, and get better understanding of their students.

5 Chosen Technologies

Designing and programming large-scale software is a complicated job. As a result, a software system is usually divided logically into subprograms that may be independently designed, programmed and tested by separate programmers' groups (Ampatzoglou and Chatzigeorgiou 2007). Quiz Class is decomposed into two independent modules, namely the front-end and back-end. It plays an important role in the architecture further design and technology choices.

When building a software application, it is important to start with a proper foundation. Every software technology has their strengths and weaknesses. To make the project successful, wise choices are important. This chapter describes what technologies will be used for Quiz Class, analysis how and why certain technologies were chosen. Following core technologies were chosen for Quiz Class:

1. **HTML5**, **JS** and **CSS3**: the essential elements of web page development.
2. **TypeScript**: JavaScript that scales.
3. **Angular**: a development platform for building mobile and desktop web applications.
4. **Bootstrap**: a front-end web framework for designing websites and web applications.
5. **Sass**: a scripting language that is interpreted into Cascading Style Sheets (CSS)
6. **Java**: a general-purpose computer programming language that is concurrent, class-based, object-oriented.
7. **Spring Framework**: an application framework and inversion of control container for the Java platform.
8. **Hibernate ORM**: an object-relational mapping tool for the Java programming language.
9. **MySQL**: relational database management system (RDBMS).
10. **Apache Tomcat**: Java Servlet Container, a web server that interacts with Java servlets.

5.1 Front-end

The ideal situation is to build a web app, an Android app and an iOS app for this project. Because native mobile applications can deliver better usability, can send notifications in a better way. The native app may also reduce the heavy request to Quiz Class server, as static contents can be saved at front-end, requested content can also be safely cached at the front end. But for this master project, the time and resources are limited, so an Android and iOS

app are not planned. Quiz Class will only concentrate on a web app, as web app has great accessibility, all mobiles, computers or any other digital devices which have a web browser can access the web app.

To build a web app, HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript programming language need to be used. For this project TypeScript programming language, Angular 2 framework, Bootstrap Front-end framework and Sass (Syntactically Awesome Stylesheets) style sheet scripting language will be used as well. Many other libraries like ChartJS[7], JQuery[8], JQuery UI[9], Font-Awesome[10], etc. will be used as well but not discussed below, as those libraries can be easily replaced by other alternatives, and they are not the core technologies will be used in Quiz Class front-end, therefore not discussed. Manager tools like NPM (npm n.d.), generating tool like Angular CLI[11], server tool Node.js[12], etc. are not discussed, as all Angular project need NPM and Node.js preinstalled, Angular CLI can be replaced and is not a must have tool. To reduce the size, only important technologies are discussed.

5.1.1 HTML, JavaScript and CSS

The web page showing in a browser is usually a combination of structure, style, and interactivity. It is undertaken by 3 different technologies, namely, HTML, JavaScript, and CSS, which are the three core technologies used for World Wide Web (WWW) content production.

HTML is the standard markup language for creating web pages and web applications, it is the core language of nearly all Web content, which describes the structure and the semantic

---

[7] Chart.js is an open source HTML5 charts library for website (retrieved 15.06.2017, from http://www.chartjs.org/).

[8] jQuery is an open source JavaScript library designed to simplify the client-side scripting of HTML (retrieved 15.06.2017, from https://jquery.com/).

[9] jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library (retrieved 15.06.2017, from https://jqueryui.com/).

[10] Font Awesome is an open source font and scalable vector icon toolkit based on CSS (retrieved 15.06.2017, from http://fontawesome.io/).

[11] Angular CLI is a tool to initialize, develop, scaffold and maintain Angular applications (retrieved 15.06.2017, from https://cli.angular.io/).

[12] Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of server tools and applications (retrieved 15.06.2017, from https://nodejs.org/en/).

content of web pages or web applications (HTML n.d.). The fifth and a current version of the HTML standard HTML5 is used in this project.

CSS is a style sheet language, used for describing the presentation of a document written in HTML, or other markup languages (CSS n.d.). The latest evolution of Cascading Style Sheets language CSS3 is used for this project.

JavaScript (JS) is a high-level, dynamic, and interpreted programming language, which is standardized in ECMAScript language specification. JS is supported by all modern Web browsers without plug-ins (JavaScript n.d.).

For this project, it is possible to build the front-end web app by only use HTML, CSS, and JavaScript. But use frameworks, tools, and technologies can help me achieve better goals in terms of quality, security, and use less time to develop Quiz Class.

5.1.2 TypeScript

TypeScript is a typed superset of JavaScript, that compiles to plain JavaScript (TypeScript n.d.). Why use TypeScript? When working with large applications using JavaScript, there may be naming conflicts, substandard programming tools, complex modularization, hard to re-use common design patterns, difficult to keep a readable and maintainable code base. TypeScript is built for solve those problems (Fenton 2014). I think that this project needs to be continually developed even after my master study, different developers may involve into this project in the further. TypeScript can maximize code readability, maintainability, and reusability.

As described by Fenton, TypeScript is an application-scale programming language that provides early access to proposed new JavaScript features and powerful additional features like static type checking (Fenton 2014). TypeScript programs can be written and run in web browsers or on servers, where TypeScript code can be reused between browser and server applications. The Angular source code is written in TypeScript and can be directly used in TypeScript (Fenton 2014).

5.1.3 Angular

Angular is JavaScript-based, open-source, front-end, web application framework (Angular n.d.). Angular makes building large and complex web applications simpler and easier.

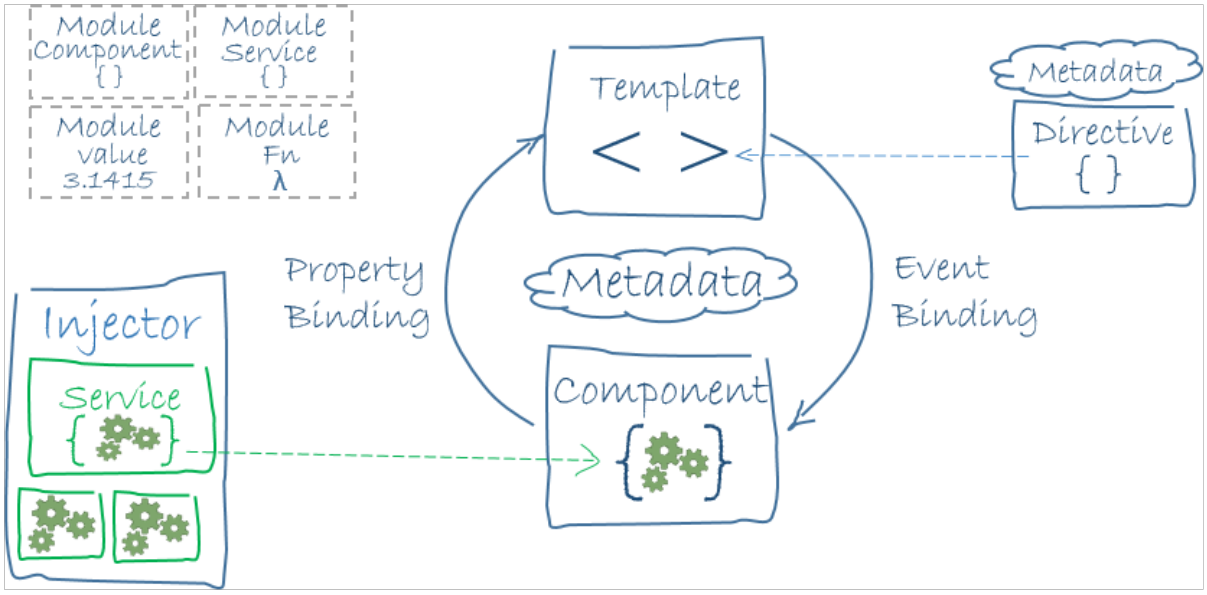


Figure 6 Angular architecture overview(AngularDoc n.d.)

The newest Angular 2.0 (was newest when this project started) will be used for this project (in this report, if not specified, Angular means the newest Angular 2.0), which can be used with TypeScript. An Angular application contains eight main building blocks (AngularDoc n.d.):

1. **Modules**: each module is a cohesive block of code dedicated to an application domain, a workflow, or a closely related set of capabilities.
2. **Components**: are the collection of templates, styles, selector configurations etc, component is one of the basic Angular blocks.
3. **Templates**: is a form of HTML that tells Angular how to render the Component.
4. **Metadata**: tells Angular how to process a class.
5. **Data binding**: a mechanism for coordinating parts of template with parts of a component.
6. **Directives**: changes the appearance or behavior of a DOM element.
7. **Services**: service is a broad category encompassing any value, function or feature that the application needs.
8. **Dependency injection**: is a way to supply a new instance of a class with the fully-formed dependencies it requires.

which is identified in Figure 6: “Angular architecture overview”.

As described by Freeman, Angular taps into some of the best aspects of server-side development and uses them to enhance HTML in the browser, creating a foundation that

makes building rich client-side applications simpler and easier with a specialization on data binding (Freeman 2014).

Angular has a component architecture, when web app has a common UI element and elements with complex functionality, it can be written as a component to get ease of reuse. Angular Dependency Injection model makes it easy to write decoupled business logic from view logic. Angular router module allows creating parent and child routes to components. Angular places an emphasis on creating applications that are (Freeman 2014):

- Extendable: can easily enhance applications to create new and useful features.
- Maintainable: are easy to debug and fix, simplified long-term maintenance.
- Testable: has good support for unit and end-to-end testing.
- Standardized: Angular allows to create standards-compliant web apps that take advantage of the latest features (such as HTML5 APIs) and popular tools and frameworks.

Angular is sponsored and maintained by Google. It has been used in some of the largest and most complex web apps around. Angular will bring a lot of benefits to Quiz Class front-end.

5.1.4 Bootstrap

Responsive web design is important today, as users may use mobile, tablet, laptop, PC or other digital devices with different screen size to interact with a web application through a browser. To achieve responsive web design, a website designer can adapt the layout to the viewing environment by using fluid, proportion-based grids, flexible images etc (RWD n.d.). Figure 7 shows the philosophy and principles of responsive web design. To easily achieve it in this project, Bootstrap is chosen.

> "The world's most popular mobile-first and responsive front-end web framework" (Bootstrap n.d.).

Bootstrap is a free and open-source front-end web framework for design responsive mobile first project for websites and web applications. Which contains HTML, CSS based design templates, and makes front-end web development faster and easier (Bootstrap n.d.).

Figure 7 Principles of Responsive web design (RWD n.d.)

Material Design for Angular[13] was considered as well, but when this project started, Material Design for Angular 2 was still on alpha version, many functionalities were not implemented or tested for use. I am familiar with Bootstrap, have used Bootstrap in some of my other projects. For this project, there are no other better alternatives than choosing Bootstrap.

5.1.5 Sass

> "Sass is the most mature, stable, and powerful professional grade CSS extension language in the world" (Sass n.d.).

Sass (Syntactically Awesome Stylesheets) is open-source style sheet scripting language can be interpreted into Cascading Style Sheets (CSS). Sass is an extension of CSS, adding nested rules, variables, mixins, selector in inheritance and more. Sass will be used in this project to make CSS more maintainable, themeable, and extendable (Sass n.d.).

[13] Material Design is a design language developed in 2014 by Google. Material is a comprehensive guide for visual, motion, and interaction design across platforms and devices (Google. "Material Design." Retrieved 26.03, 2017, from https://material.io/). Angular Material is both a UI Component framework and a reference implementation of Google's Material Design Specification.

There are several CSS preprocessors, which is primarily intended to make writing CSS more dynamic, organized and productive. Two of the most popular CSS preprocessor Less ({less} n.d.) and Sass (Sass n.d.) were taking in consideration.

I have some limited experience with Less but has not been working with Sass. Less is a CSS preprocessor, it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow users to make CSS that is more maintainable, themeable and extensible ({less} n.d.). Less produces a very similar result as Sass, the advantage about Less is that Less is easy to learn and start with if compare with Sass. But Sass offers more advanced functionalities and has bigger user communities than Less. Besides, Bootstrap have switched from using Less to Sass when it comes to Bootstrap 4. Therefore, Sass was chosen over Less for this project.

5.2 Back-end

The Quiz Class end-users should be able to use the application anytime, anywhere they want, therefore cloud computing is needed to run the Quiz Class server. Technologies like, Apache HttpClient[14], Gson[15], JJWT[16], Maven[17], etc. were also chosen for the back-end. This section first introduces the back-end design choices, then presents the core technologies.

5.2.1 Cloud hosting model

Different cloud computing hosting model provides a different set of services, so what technologies to choose may depend on what hosting model fit the project. Two types of cloud

[14] Apache HttpClient provide efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations (retrieved 15.06.2017, from https://hc.apache.org/httpcomponents-client-ga/).

[15] Gson: google-gson, is a Java library that can be used to convert Java Objects into their JSON representation, or convert a JSON string to a Java object (retrieved 15.06.2017, from https://github.com/google/gson).

[16] JJWT: Java JWT: JSON Web Token for Java and Android, is a Java implementation based on the JWT, JWS, JWE, JWK and JWA RFC specifications. JJWT makes transmitting information between two parties in a compact and verifiable form (retrieved 15.06.2017, from https://github.com/jwtk/jjwt).

[17] Apache maven is a software project management and comprehension tool. Based on the concept of a project object model(POM), Maven can manage a project's build, reporting and documentation from a central piece of information (retrieved 15.06.2017, from https://maven.apache.org/).

computing service models are considered for this project, Infrastructure as a Service and Platform as a Service.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) provides visualized computing resources over the internet. A third-party (e.g. Microsoft Azure: Cloud Computing Platform & Services (Azure n.d.) and Amazon Web Services – Cloud Computing Services (AWS n.d.)) provider hosts hardware, software, servers, storage and other infrastructure components on behalf of its users.

IaaS platforms offer highly scalable resources that can be adjusted on-demand, easy to migrate applications. Which makes IaaS well suited for experimental purposes, like this project. Figure 8 shows an architecture solution can be used in IaaS.

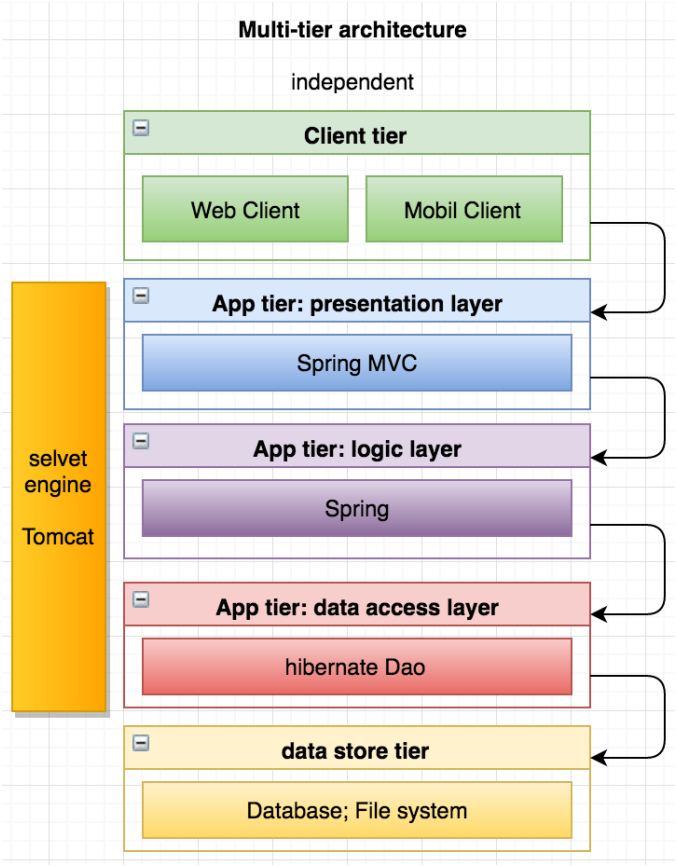


Figure 8 IaaS: multi-tier architecture

The architecture shown in Figure 8 uses Java programming language (JavaLang n.d.), and Spring Framework (Spring n.d.). There are many advantages of using this architecture. First, anything can be built with this architecture, in addition, it provides high customization, scalability at functional level. Second, this can run anywhere, it is easy to migrate to any OS

which has Java virtual machine installed. Java virtual machine is free and easy to install in most of the OS. And this architecture is best for advanced software, may be used for enterprise level.

This architecture also has disadvantages. First, it needs to maintain. IaaS is targeted at the system and network administrators. The application will run on an OS (e.g. Linux), therefore extra time needs to be used for maintaining the server. Second, all the functionalities needs to be implemented and maintained by the developer. It needs a long development cycle. Last, it has relatively lower scalability at run time. When the application starts to run, it is difficult to create or shut-down instances accordingly by the requests worldwide. This is not critical for this project, but worth consideration.

Platform as a Service (PaaS)

A Platform as a Service (PaaS) provider (e.g. Google App Engine (AppEngine n.d.)) hosts the hardware and software on its own infrastructure. Thus, PaaS frees users from having to install in-house hardware and software to develop or run a new application.

An alternative software architecture uses PaaS provided by Google Cloud Platform, as shown in Figure 9.

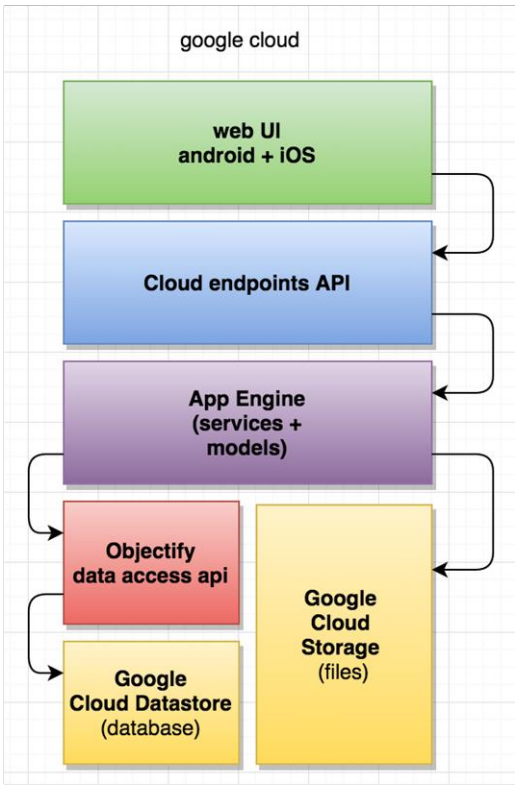


Figure 9 PaaS: architecture - use Google Cloud Platform

The Google App Engine provides the developer with a development environment for Python, Java etc. Google manages to deploy and execute developed code. Google provides database services that can be automatically replicated across data centers (AppEngine n.d.).

There are many advantages of this architecture. First, PaaS is targeted at application developers (suit for this project). Second, it is easy and fast to create and maintain, and best for small applications, like this project. Third, native client APIs (Android, iOS, and JavaScript) can be generated by using Cloud Endpoints (CloudEndpoints n.d.) API. And it provides unlimited scalability at run time. e.g. App Engine can auto scale, Google Datastore (Datastore n.d.) can auto scale. In addition, there are free administration and data analyzing tools to use.

There are many disadvantages as well. Cloud hosting must use Google Cloud Platform. The application must be developed under Google's frameworks. It is difficult (or maybe impossible) to build some customized functions. Users can not access Google services worldwide, e.g. in mainland China, Google services are difficult to access, this may be critical for production use, or test the app in China.

Decision

Google cloud platform was considered for Platform as a Service, which can be implemented by Java programming language. The main advantages of this architecture are summarized below:

- Free daily quotas for App Engine, Cloud Datastore and Cloud Storage (CloudStorage n.d.).
- Cheap to use (pay what is used when exceeds free quotas).
- Auto generates native client APIs for Android, iOS, and JavaScript (by using Cloud Endpoint API), save time to build own APIs for communication between client and server.
- Easy to develop, shorter development cycle. And the free administration, data analyzing tools included.

There are many options if IaaS is used. Two possible ways for application hosting, two possible programming languages are considered. Compare with PaaS, IaaS may need lots of time and resources to build and maintain the server, may need to implement own

administration and data analyzing tools. IaaS structure may be difficult to scale when the server starts to run, SQL database is more challenging to scale at the run time.

I have considered to host Quiz Class on Amazon Web Services (AWS), use Java programming language to build the application, or host on Azure, use the C# programming language to build the application. The AWS offers limited free tier available for 12 months. Microsoft Azure (Azure n.d.) + BizSpark (BizSpark n.d.) + .NET (.NET n.d.) + C# (C# n.d.) solution offers:

- Azure: 1 650kr free credit
- BizSpark, 3 years for free if apply as startups.
- Provide some administration tool and some data analyzing tools.
- Better to use .NET + C# (if work with Microsoft)
    - Disadvantage: I only have limited knowledge and little experiences with .NET Framework and C# language.
    - Better to run on a Windows server, which is not free.

IaaS with Java programming language and Spring framework solution is easy to migrate. This is important as there is uncertainty in this project. Many functionalities need to be implemented self, e.g. authentication, authorization, but with libraries and frameworks, it should be easy.

Amazon Web Service and Microsoft Azure are easy to access by users all over the world, some data analyzing and administration tools are provided. Microsoft Azure give much free stuff for 3 years if applied as a startup through BizSpark, which may not fit for this project.

It is clear that IaaS is the best choice for this project, and Java programming language with Spring framework is the best choice to develop this application. Both Azure and Amazon cloud service provide excellent services. AWS is the dominated cloud host provider today, and it offers 12 month limited free tiers for most of AWS's services, and worldwide distribution center to use. AWS is preferred for this project to use.

5.2.2 Java

"Java is the world's #1 programming language" (Java n.d.).

The Java Programming Language is a general-purpose, class-based, object-oriented, high-level programming language, which was originally developed by Sun Microsystems, and was initiated by James Gosling and released in 1995 (Horstmann 2013).

"Write once, run anywhere", Java can be developed on any device, compiled into a standard bytecode and can run on any device which has Java virtual machine (JVM) installed (Horstmann 2013). Figure 10 shows the Java conceptual diagram.

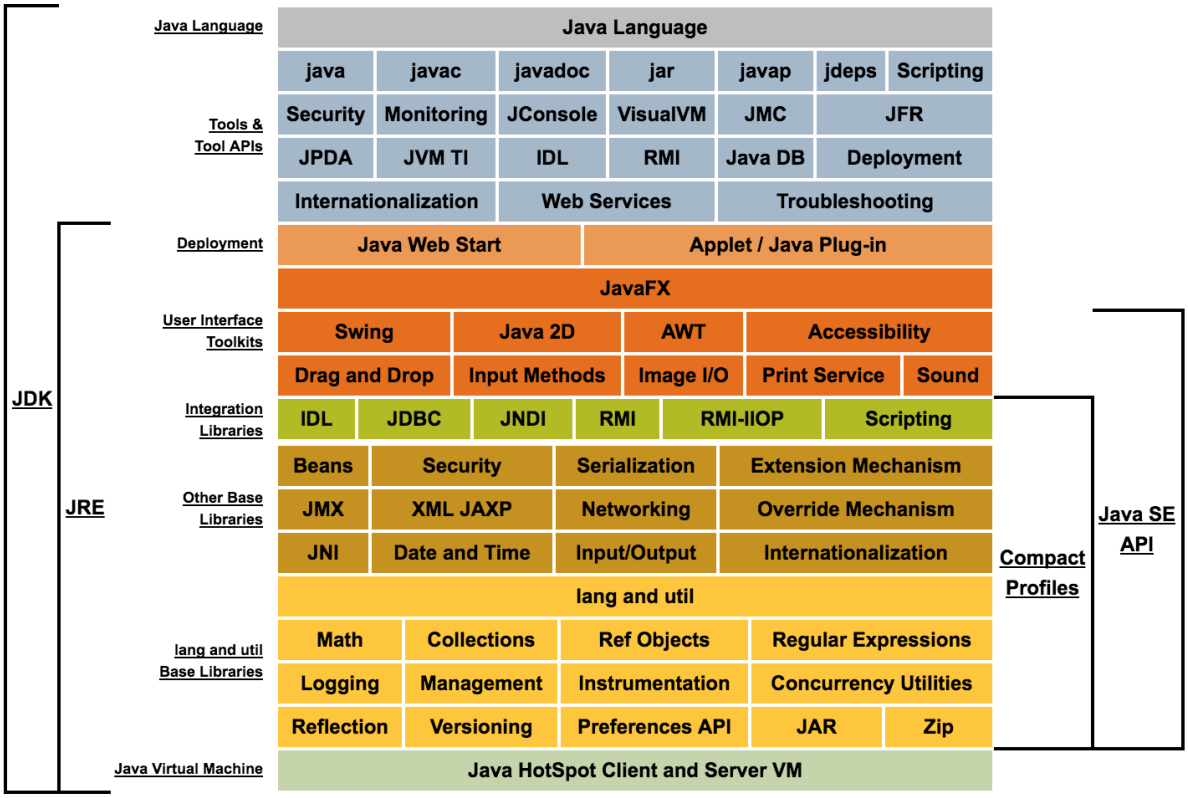


Figure 10 Description of Java Conceptual Diagram (JavaSE n.d.)

In 2006, Sun Microsystems released much of its Java virtual machine as free and open-source software, relicensed most of its Java technologies under the GNU General Public License in 2007. Following Oracle Corporation's acquisition of Sun Microsystems in 2009-2010, Java technology is now owned by Oracle Corporation (JavaLang n.d.).

5.2.3 Spring Framework and Spring Security

> "Spring helps development teams everywhere build simple, portable, faster and flexible JVM-based systems and applications" (Spring n.d.).

Spring is a popular application development framework and inversion of control container for the Java platform. Spring framework can be used to build anything with clean, testable code;

run (portable to) anywhere where JVM is installed; Spring provides an open programming model that is comprehensive, cohesive, widely understood and well supported (Spring n.d.) (Spring n.d.). Spring framework is open source (SpringFramework n.d.).

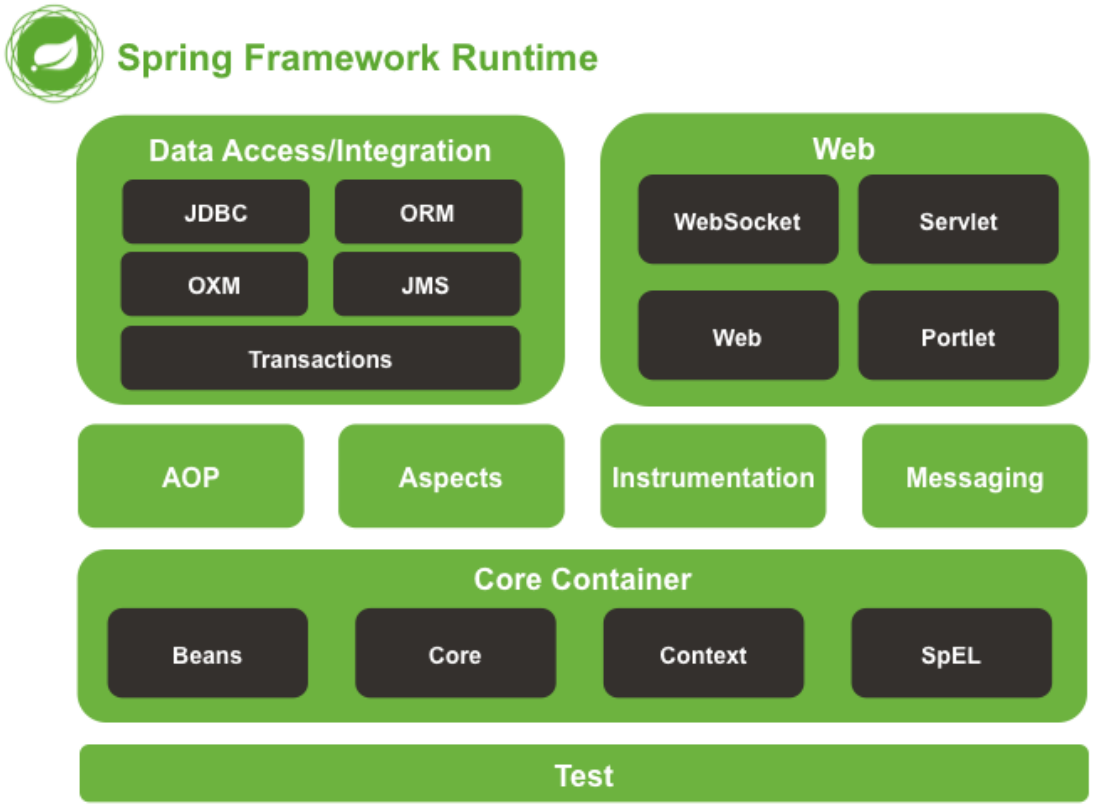


Figure 11 Overview of the Spring Framework (SpringDoc n.d.)

The Spring Framework consists of features organized into different modules. These modules are grouped into Core Container, Data Access/Integration, Web, AOP (Aspect Oriented Programming), Instrumentation, Messaging, and Test, as shown in Figure 11 "Overview of the Spring Framework" (SpringDoc n.d.). Spring framework core features can be used by any Java application.

Spring Security is an open-source Java/Java EE access-control framework. It is the de-facto standard for securing Spring-based applications. Spring Security provides comprehensive and extensible support for both Authentication and Authorization; it provides protection against attacks like session fixation, clickjacking, cross-site request forgery, and other security features (SpringSecurity n.d.).

5.2.4 Hibernate ORM

Hibernate ORM (Hibernate n.d.) is an object-relational mapping framework for Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. In addition to its own "native" API, the Hibernate is also an implementation of the Java Persistence API (JPA) specification. It can be easily used in any environment supporting JPA. Hibernate enables to develop persistent classes following natural Object-oriented idioms including inheritance, polymorphism, association, composition, and the Java collections framework. Hibernate consistently offers superior performance over straight Java Database Connectivity (JDBC) code, both in developer productivity and runtime performance. Hibernate deliver high scalability in any environment, and has excellent stability and quality, it is also highly configurable and extensible. Hibernate is free software distributed under the GNU Lesser General Public License 2.1.

5.2.5 MySQL

"MySQL is the world's most popular open source database" (MySQL n.d.).

MySQL is an open-source relational database management system (RDBMS). MySQL was owned and sponsored by a Swedish company MySQL AB[18], now owned by Oracle Corporation (MySQL n.d.). MySQL is written in C and C++, it works on many system platforms, including Linux, Mac OS X, Microsoft Windows. And it provides Connectors and APIs to enable connect and execute MySQL statements from different languages or environments, including Java, Perl, Python, PHP, Ruby, C, C++, .Net etc. (MySQLdoc n.d.). MySQL server software and the client libraries use dual-licensing distribution. They are offered under GPL version 2 of the GNU General Public License (MySQLServer n.d.). MySQL is only used for development purposes in this project.

5.2.6 Apache Tomcat

Apache Tomcat (Tomcat® n.d.), is an open source Java Servlet web server Container, it is developed by the Apache Software Foundation. Apache Tomcat implements server Java EE

[18] MySQL AB was a software company that was founded in 1995. It was acquired by Sun Microsystems in 2008; Sun was in turn acquired by Oracle Corporation in 2010. MySQL AB is the creator of MySQL, a relational database management system, as well as related products such as MySQL Cluster (MySQL AB, retrieved 15.06.2017, from https://en.wikipedia.org/wiki/MySQL_AB).

specifications including Java Servlet, JavaServer Pages, Java Expression Language, and WebSocket technologies. Tomcat is the component of a web server that interacts with Java servlets. Tomcat is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access-right. Tomcat handles requests to servlets, JavaServer Pages (JSP) files, and other types of files that include server-side code. Tomcat creates servlet instances, loads and unloads servlets, creates and manages request and response objects, and performs other servlet-management tasks. Tomcat implements the web component contract of the Java EE architecture, specifying a runtime environment for web components that includes security, concurrency, lifecycle management, transaction, deployment, and other services (WebContainer n.d.). Tomcat is released under the Apache License version 2.

5.3 HTTPS

"All pages must be served over HTTPS. This includes CSS, scripts, images, AJAX requests, POST data and third party includes. Failure to do so creates a vector for man-in-the-middle attacks" (OWASP 2017). As indicated by OWASP[19], HTTPS connections must be always used, which can protect against eavesdropping, tampering, or forging the contents of communication.

According to Wikipedia (HTTPS n.d.), HTTPS also called HTTP over TLS[20] or SSL[21], and HTTP Secure, which is a protocol for secure communication over a computer network. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by TLS or SSL. HTTPS provides authentications of the website and associated web server with which one is communicating. It provides bidirectional encryption of communication between an applications front-end and back-end, which guarantee that one is communication with precisely the website that one intended to communicate, and ensuring that the contents of communications cannot be read or forget by any third party.

---

[19] The Open Web Application Security Project (OWASP) is an online community which creates freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security (retrieved 15.06.2017, from https://www.owasp.org/index.php/Main_Page).

[20] TLS: Transport Layer Security.

[21] SSL: Secure Sockets Layer. Both TLS and SSL frequently referred to SSL, are cryptographic protocols that provide communications security over a computer network.

5.4 Quiz Cloud

QuizCloud is a framework for creating and managing educational games as well as a storage of game content, and it provides SDKs for JavaScript and GoLang (golang n.d.) development. QuizCloud is developed by my co-supervisor Meng Zhu. The Quiz Class synchronizes all its quizzes to QuizCloud, use it as a backup storage and management tool to store and manage the quiz content. QuizCloud uses Go programming language and MongoDB(MongoDB n.d.) database.

Go programming language (golang n.d.) is a free and open source programming language created by Google in 2007.

> "Go is expressive, concise, clean, and efficient. Its concurrency mechanisms make it easy to write programs that get the most out of multicore and networked machines, while its novel type system enables flexible and modular program construction. Go compiles quickly to machine code yet has the convenience of garbage collection and the power of run-time reflection. It's a fast, statically typed, compiled language that feels like a dynamically typed, interpreted language" (golang n.d.).

The Go program can be compiled and deployed in different platforms include Linux, Mac OS X, and Windows.

MongoDB (MongoDB n.d.) is developed by MongoDB Inc. It is a free and open-source document-oriented NoSQL[22] database. It uses JavaScript Object Notation (JSON), which is an open, human and machine-readable standard that facilitates data interchange. NoSQL database offers better scalability and is increasingly used in big data and real-time web applications (NoSQL n.d.).

[22] A NoSQL (originally referring to "non SQL", "non-relational" or "not only SQL") database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases (retrieved 15.06.2017, from https://en.wikipedia.org/wiki/NoSQL).

5.5 Summary

Choosing the right technology is essential for this project. The RESTful[23] web service makes front-end and back-end independent of each other. The RESTful client-server separation simplifies component implementation, reduces the complexity of the system, improves the modifiability and portability of components, increases the scalability of the server and the client components. However, to choose back-end technologies the cloud hosting model need to be considered first, as the cloud hosting model may limit or prefer some technologies.

The chosen technologies are not only theoretically fit this project, but are also very popular and been used by many software in practice.

[23] Representational state transfer (REST) or RESTful web services are one way of providing interoperability between computer systems on the internet. REST-compliant web services allow requesting systems to access and manipulate textual representations of web resources using a uniform and predefined set of stateless operations (retrieved 15.06.2017, from https://en.wikipedia.org/wiki/Representational_state_transfer).

Part III: Quiz Class

Part III presents Quiz Class's game concept, software requirements, UI Design, software architecture, and the software implementation.

6 Game Concept

The goal of the Quiz Class application is to make learning more social, challenge, competitive, and fun, therefore, boost players' learning motivation. In addition, Quiz Class provides functionalities, which can be used by game creators (usually teachers) to better understand their players (students), therefore, increase teaching efficiency.

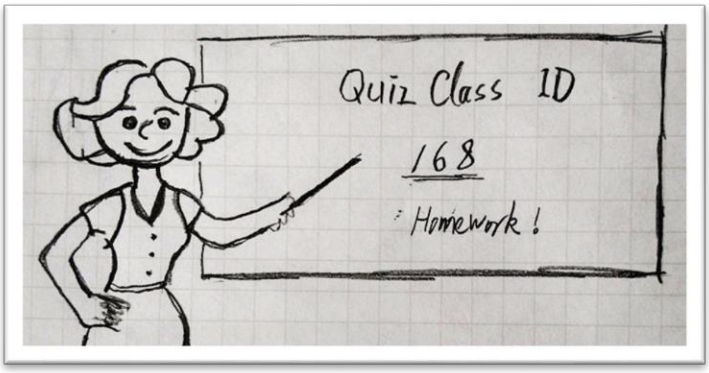


Figure 12 Quiz Class Game Concept: share homework

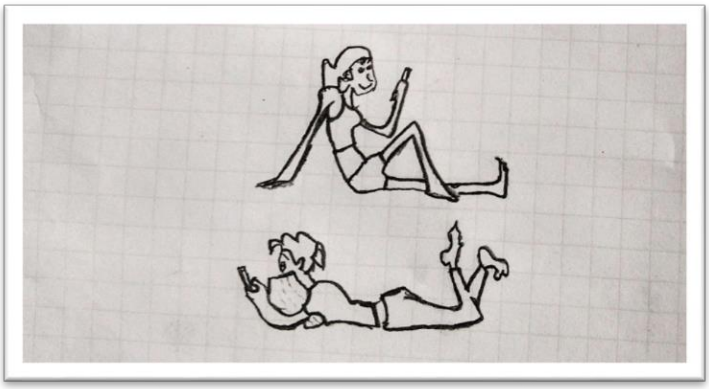


Figure 13 Quiz Class Game Concept: play game

Figure 14 Quiz Class Game concept: evaluation

Malone (Malone 1980) has suggested three essential characteristics that can make a game fun to play: providing an appropriate level of challenge whose attainment is not guaranteed, players should generate fantasy while playing, and the game should trigger players curiosity. Quiz Class is developed with these three characteristics in mind. In addition to Malone's Challenge, Fantasy, and Curiosity, the GameFlow model which consists of eight core elements: concentration, challenge, skills, control, clear goals, feedback, immersion, social interaction (Sweetser and Wyeth 2005), was also used as criteria for achieving enjoyment in Quiz Class. Thus, the Quiz Class concept was developed and illustrated in Figure 12, Figure 13, and Figure 14. The teacher creates a game and shares an ID to the students (Figure 12), the students gets the ID, use his or her digital devices to join the game, and play it anywhere, anytime they like (Figure 13). The teacher may evaluate, and summarize the game with statistical analysis, graphs provided by Quiz Class, and share it on a big screen (Figure 14). Considering Malone's three characteristics, Quiz Class should provide challenges, social interactions among players, fantasy and curiosity should be generated while playing. The level of challenges is depending on the difficulty of questions, which can be controlled by creators. The social interactions are happening both in and out the gameplay, as most players should know each other, and they may discuss the game, which again may increase player curiosity and fantasy. The GameFlow model is used more as criteria for achieving enjoyment both in game concept developing, UI design, and software implementation.

The game has two major roles, player/student, creator/teacher. The players have to compete with each other to get medals. There are three types of medals, Figure 15 shows the earliest medal, Figure 16 shows the Top Score medal, Figure 17 shows the Progress Medal.

Figure 15 Earliest medal

Figure 16 Top Score medal

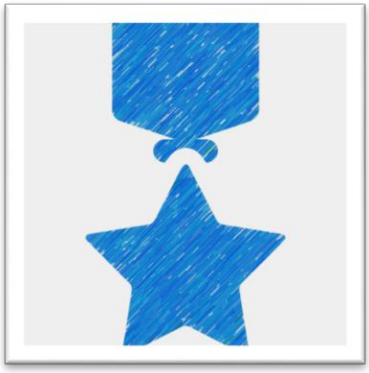


Figure 17 Progress medal

A player must start the game as early as possible, because only the first 10 players, who finished the game earliest can get the Earliest medal. And the earliest list is static and

unchangeable, i.e. if the top 10 is already taken in a game, there is no chance for other players to come in again. Start playing early and finishing the game early is the key to getting this medal.

A player will compete against other players to win a Top Score medal. All top 10 players who have the highest score will get a Top Score medal. Before the game finish, the Top Score medal is unpredictable. A player may lose his or her medal if he or she is not in top 10 list anymore. The top 10 score list is dynamic changing, the only way to be in the top 10 list is to play carefully, and get a higher score than others. Before the game finish, a player has unlimited chance to play the game again, so for some unlucky player, there is still chance to win the game.

A player will get some reward points when he or she plays the game again, and he or she can get more reward points if he or she is making progress (getting a better score than last play). Making progress is the key to win the reward points, and is the key to win a Progress medal. All players in the top 10 progress list will get a Progress medal. This list is also dynamic, so before the game finish, who will finally stay in the top 10 progress list is unpredictable.

A game player is typically a student but can be any individuals who wish to play/learn. The three types of medals are designed to reward students who answer all the questions in a game as early as possible, as correct as possible, and improve themselves continuously. The feature of repetition also considered that anyone who is not satisfied with the result, should have a chance to improve, and most importantly, as said by Zig Ziglar: “Repetition is the mother of learning”. Moreover, the feature of confident level is also innovative. When playing a game, players are encouraged to “gamble”, the idea is to increase the fun of play, and more importantly, give creator/teacher opportunity to know how certain a player/student is, for the answer he/she give.

A creator is typically a teacher but can be any individuals. A creator can create a class (called game collection in Quiz Class), which may also correspond to a course, a topic, or a group. In the created class (game collection), homework (game) can be created, which may be an assignment or summary of a chapter etc.

A quiz creator is the only character who determines the game contents, challenge level, how fun or interesting the game is. A game’s start time and end time is also decided by the creator, and the creator may also decide who can play the game.

A game's start time and end time is conceptually the same as the start time and deadline of homework. The players usually join the game before the game started, but can also join the game before the game end time. With a digital device, the game can be played anywhere, anytime.

7 Requirements

Quiz Class need to be used to investigate how a DGBL application can influence teachers' work, and students learning motivations. Sufficient functionalities need to be provided, so the users can use Quiz Class, and I can collect the data for answering the research questions. This chapter presents Use cases for Quiz Class and the outline of the functional and non-functional requirements for the Quiz Class application.

7.1 Use Cases

Figure 18 shows the Use Case of the Quiz Class, and Figure 19 shows the Use Case of QuizCloud.

Figure 18 Quiz Class Use Case

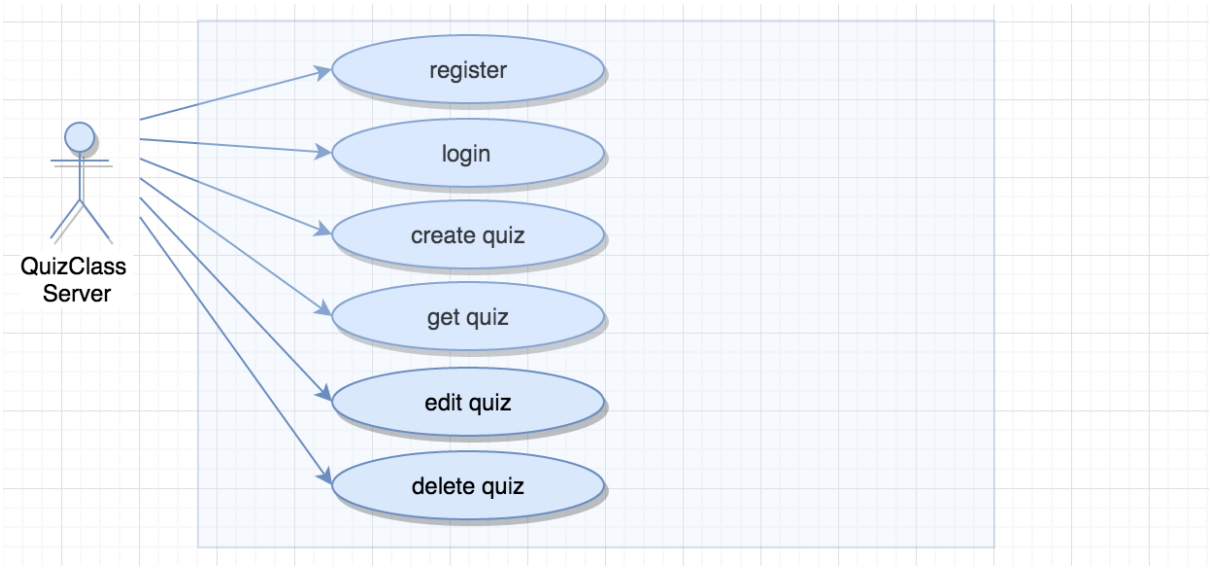


Figure 19 QuizCloud Use Case

The Quiz Class and QuizCloud use cases are used to represent the missions, stakeholder goals, the interactions between roles and systems. Use cases are also used for software requirements, system analysis and communication expression of choice. They are also used as the container for functional requirements capture, analysis, and behavioral specification. For the readers, this also present an overview of the main missions, goals of the system, and interactions of the roles and systems.

7.2 Functional Requirements

This section defines the functionalities that the Quiz Class application is required to have, what the system must do, and how it must behave or react at run time, namely the functional requirements. The functional requirements are analyzed and captured from use cases (from the previous part).

Not all functionalities are at the same critical level, and some might be more important than others. When expectations are high, timelines are short, and resources are limited, it is important to make sure the product contains the essential functionalities. Therefore, explicit priority is important for the requirements. The priority for this project is divided into three levels:

- **High priority:** the application is not acceptable unless these requirements are satisfied.
- **Medium priority:** the requirements are necessary, which should be provided eventually but could be postponed to a later release.

- **Low priority:** a functional or quality enhancement, and they are nice to have someday if resources permit.

User stories are the general-purpose agile substitute for what traditionally has been referred to as software requirements (Leffingwell 2011). User stories are brief statements of intent that describe something the system needs to do for some user. As commonly taught, the user story often takes a standard user-voice form of the following (Leffingwell 2011):

As a <*user role*>, I can <*activity*> so that <*business value*>.

This form focusses on both the user's role and the business benefit that the new functionality provides.

7.2.1 Teacher/Game creator

The functional requirements for teachers or game creators can be found in Table 1.

ID	Name	Description	Priority
FR01	Register	As a teacher, I want to register, so that the system can save my profiles and created homeworks.	High
FR02	Login	As a teacher, I want to login, so that the system know I am back.	High
FR03	Change password	As a teacher, I want to change my password, so that my account security is higher.	Low
FR04	Password reset	As a teacher, I want to reset my password, so that I can access my account even I forget my password.	Medium
FR05	Create public homework	As a teacher, I want to create public homework, so that everyone can access if they want.	High

FR06	Create secret homework	As a teacher, I want to create *secret homework*, so that only student has pin code can apply it.	High
FR07	Approve application	As a teacher, I want to approve the secret homework applications, so that only students approved by me can do the homework.	Medium
FR08	View Analysis	As a teacher, I want to view the results of the homework analysis, so that I know how my student did.	High
FR09	Edit profile	As a teacher, I want to edit my profile, so that I can change information about me.	Low
FR10	View rankings	As a teacher, I want to view the student ranking, so that I know how each students doing.	High
FR11	Start game	As a teacher, I want to set start time, so that I can decide when my student can start doing homework.	Medium
FR12	Stop game	As a teacher, I want to set deadline for the homework, so that I can decide when my student should finish.	Medium
FR13	Edit homework	As a teacher, I want to edit the homework, so that I can edit the homework I created.	High
FR14	Delete homework	As a teacher, I want to delete a homework, so that I can remove a homework which I do not want to have.	Medium

Table 1 Functional requirements for teachers/game creator

7.2.2 Students/players

The functional requirements for students or game players can be found in Table 2.

ID	Name	Description	Priority
FR16	Register	As a player, I want to register, so that the system can save my profiles and I can use other functionalities of the system.	High
FR17	Login	As a player, I want to login, so that the system know I am back.	High
FR18	Change password	As a player, I want to change my password, so that my account security is higher.	Low
FR19	Password reset	As a player, I want to reset my password, so that I can access my account even I forget my password.	Medium
FR20	Edit profile	As a player, I want to edit my profile, so that my profile is updated.	Low
FR21	View ranking	As a player, I want to view the rankings, so that I know the competition results.	High
FR22	Apply secret homework	As a player, I want to be able to apply for a secret homework, so that I can do the secret homework when the approval is granted.	Medium
FR23	Get homework	As a player, I want to get the homework, so that I can do my homework.	High

FR24	Do homework	As a player, I want to do my homework, so that I can compete with others and have fun.	High
FR25	View feedback	As a player, I want to have feedback of each quiz, so that I know how I am doing.	High
FR26	Rate homework	As a player, I want to rate the homework, so that I can give feedback to teachers about the homework they created.	Medium

Table 2 Functional requirements for students/players

7.2.3 QuizCloud

The functional requirements for QuizCloud can be found in Table 3.

ID	Name	Description	Priority
FR26	Register	As a Quiz Class server, I want to register at QuizCloud, so that QuizCloud know I am a registered user.	High
FR27	Login	As a Quiz Class server, I want to login to QuizCloud, so that QuizCloud know I am back.	High
FR28	Create quiz	As a Quiz Class server, I want to create quiz at QuizCloud, so that my quiz can be saved at QuizCloud.	High
FR29	Get quiz	As a Quiz Class server, I want to get quiz from QuizCloud, so that I may use the quiz I created.	High
FR30	Edit quiz	As a Quiz Class server, I want to edit a quiz I created, so that my user updated quiz will be also updated at QuizCloud.	Medium

FR31	Delete quiz	As a Quiz Class server, I want to delete a quiz, so that my user deleted a quiz will also be deleted at QuizCloud.	Low

Table 3 Functional requirements for QuizCloud

7.3 Quality Requirements

The functional requirements have defined what the Quiz Class system should do, while quality (non-functional) requirements cover the quality attribute of the Quiz Class system.

Quality attribute requirement should be unambiguous and testable. To specify quality attribute requirements, Bass, Clements et al. suggest to use six-part scenarios, Figure 20 show the relation of the scenarios, and the six-art scenarios are listed below (Bass, Clements et al. 2012):

1. ***Source of stimulus***. This is some entity (a human, a computer system, or any other actuator) that generated the stimulus.
2. ***Stimulus***. The stimulus is a condition that needs to be interpreted when it arrives at a system.
3. ***Environment***. The stimulus occurs within certain conditions. The system may be in an overload condition or may be running when the stimulus occurs, or some other condition may be true.
4. ***Artifact***. Some artifact is stimulated. This may be the whole system or some pieces of it.
5. ***Response***. The response is the activity undertaken after the arrival of the stimulus.
6. ***Response measure***. When the response occurs, it should be measurable in some fashion so that the requirement can be tested (Bass, Clements et al. 2012).

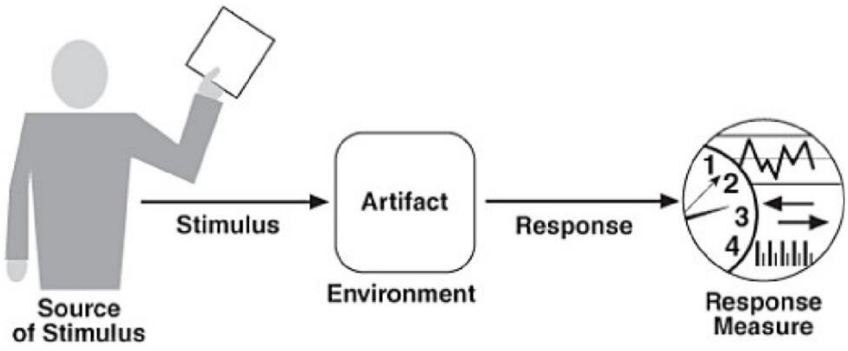


Figure 20 Quality attribute scenario

One of the main goals of this project is to build Quiz Class application, which can be used to experiment and investigate how a Digital Game-Based Learning (DGBL) system influences teachers' work, students' study. The application is not aimed at a finished product, which is ready to release, but a prototype which is good enough to experiment and investigate, so that the I can answer the research questions. This does not mean the quality attribute is not considered but rather prioritized. And availability, modifiability, usability is prioritized for this project.

7.3.1 Availability

Availability defines the proportion of time that the system is functional and working. It can be measured as a percentage of the total system downtime over a predefined period. Availability will be affected by system errors, infrastructure problems, malicious attacks, and system load (Patterns and Team 2009).

Quiz Class system relies on the internet heavily, when we consider quality attribute requirements, the availability is an important factor. The server has to be available all the time so that the game creator can create game/homework, and the players/students can continue their game/homework, otherwise, they will lose interest to continue. Quiz Class system has to be available on a wide range of clients, so that all end-uses can use it anywhere, anytime. The quality attribute scenarios of availability can be found in Table 4 and Table 5.

A1: Reconnect to server

Source of stimulus	Quiz Class front-end
Stimulus	The Quiz Class front-end should reconnect to Quiz Class back-end server if connection is lost.
Environment	Runtime.
Artifact	Network connection.
Response	Reestablish connection to server.
Response measure	The Quiz Class front-end should be reconnected to Quiz Class back-end server within 5 seconds if the internet is available.

Table 4 Availability: reconnect to server

A2: Run on different client types

Source of stimulus	Quiz Class front-end (player client)
Stimulus	Quiz Class front-end should be available on different client types.
Environment	Runtime.
Artifact	Quiz Class front-end
Response	The end-users are able to play the game
Response measure	Quiz Class front-end should support both mobile and desktop OS, and support 95% of today's web browsers.

Table 5 Availability: Run on different client types

7.3.2 Modifiability

Modifiability deals with change and cost in time or money of making a change, including the extent to which this modification affects other functions or quality attributes (Bass, Clements et al. 2003). Modifiability is important for Quiz Class, as it is a prototype of DGBL system, there are many possibilities for changing, e.g. to add new features, to modify existing features, to upgrade security, to improve performance, to fix defects, etc. In addition, changes are also needed to enhance the user experience, to embrace new technology, new platforms, new standards, new business requirements. Modification for Quiz Class may also be needed in many other unexpected situations.

M1: Change UI

Source of stimulus	Developer
Stimulus	Developer wishes to change the UI.
Environment	Design time.
Artifact	Front-end code.
Response	UI is changed.
Response measure	The developer should be able to change the UI within three hours.

Table 6 Modifiability: Change UI

M2: Change database

Source of stimulus	Developer
Stimulus	Developer wishes to change SQL database to NOSQL database.
Environment	Design time.
Artifact	Back-end code.
Response	Make modification, Test modification, Deploy modification
Response measure	The developer should be able to change the SQL database to NOSQL database within three days.

Table 7 Modifiability: Change database

Modifiability is important for developing Quiz Class software system, as it is critical to be able to easily make changes. Quiz Class system needs to be designed in such a way that future changes will be relatively easy to implement, since this reduce the time needed to implement the changes, and decreases maintenance cost. The quality attribute scenarios of modifiability can be found in Table 6 and Table 7.

Portability is a special form of modifiability (Bass, Clements et al. 2012), which is also prioritized in Quiz Class. Portability refers to the ease with which software that was built to on one platform can be changed to run on a different platform (Bass, Clements et al. 2012). As Quiz Class application may be tested and run on different platforms, e.g. on a local Ubuntu server (UbuntuServer n.d.), Amazon Web Services (AWS n.d.), Microsoft Azure

(Azure n.d.) etc. Quiz Class application need to be designed and implement in a way that is as platform independent as possible, moving to a new platform should relatively easy, and demands little modification. A quality attribute scenario of portability can be found in Table 8.

M3: Change platform

Source of stimulus	Developer
Stimulus	Developer wishes to change the platform form local Server to Amazon Web Services (AWS n.d.).
Environment	Design time.
Artifact	Quiz Class back-end and front-end.
Response	Server is up and running again.
Response measure	The developer should be able to change platform within 5 hours.

Table 8 Modifiability(portability): change platform

M3: Change Database server

Source of stimulus	Developer
Stimulus	Developer wishes to change the Database server form local Database Server to Amazon Relational Database Service (Amazon **RDS**) (AWS n.d.).
Environment	Design time.
Artifact	Quiz Class back-end.
Response	Server is up and running again.
Response measure	The developer should be able to change Database server and replicate data to new Database within 1 hour.

Table 9 Modifiability (portability): change Database server

7.3.3 Usability

Usability defines how well the application meets the requirements of the user and consumer by being intuitive, easy to localize and globalize, providing good access for disabled users, and resulting in a good overall user experience (Patterns and Team 2009).

End-users may lose interest on Quiz Class and stop use it, if the Quiz Class front-end UI is cumbersome and difficult to use. It is important to make the UI as user-friendly as possible, ensuring everything is easy to find, read, and use. A quality attribute scenario of usability can be found in Table 10.

U1: Input interface

Source of stimulus	Player/student
Stimulus	The player/student wants to play or do their homework.
Environment	Runtime.
Artifact	Quiz Class front-end.
Response	Easy to use input interface.
Response measure	At least 90 percent of the students using the system should know how to use the user interface.

Table 10 Usability: input interface

8 UI design

This Chapter introduces the details about Quiz Class UI design, and how each function works. The Quiz Class is designed to be mobile friendly as well, screenshots for Mobile View are shown in Appendix: Quiz Class UI. Figure 21 show the Quiz Class Logo, which represent the letter Q, C, and it is cloud based.

Figure 21 Quiz Class Logo

Ease of use affects the users' performance and their satisfaction, while acceptability affects whether the product is used (Holzinger 2005). Andreas Holzinger (Holzinger 2005) has suggested five essential usability characteristics which were considered during Quiz Class UI design. The five essential usability characteristics are listed below:

- Learnability: the user can rapidly begin working with the system;
- Efficiency: enabling a user who has learned the system to attain a high level of productivity;
- Memorability: allowing the casual user to return to the system after a period of non-use with-out having to relearn everything;
- Low error rate: users make fewer and easily rectifiable errors while using the system, and no catastrophic errors occur;
- Satisfaction: making the system pleasant to use.

I want the participants of Quiz Class have smooth user experience with the UI so they can focus on the game, and enjoy the fun of learning while gaming. When Quiz Class Front-end UI is being designed, those goals are kept in mind. Keeping the user interface as simple and clean as possible, so the UI is almost "invisible" to the users. Keeping UI consistent and using common UI elements, so the users feel more comfortable and can get things done quickly. Keeping UI colorful, joyful, and using colors strategically, making them consistent and meaningful, so the users can get information by seeing the colors, therefore increase

productivity of themselves. Make sure that the users feel that they have control of Quiz Class, which also include design dynamic feedback systems, and help systems.

8.1 Login and Register

This section introduces how a new end-user can get start with Quiz Class app by register, login, and what to do when a password is forgotten.

8.1.1 Game Register and login

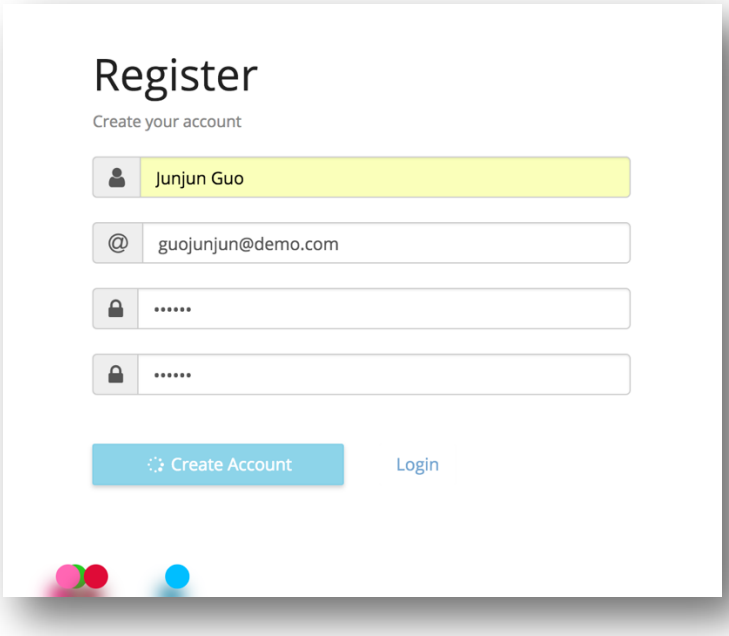


Figure 22 QuizClass UI: Register

To use Quiz Class an end-user need to be registered and logged in. As playing game/doing homework process need to be saved to users account in the cloud. Figure 22 shows the Quiz Class Register User Interface (UI). Figure 23 shows the Quiz Class Login UI.

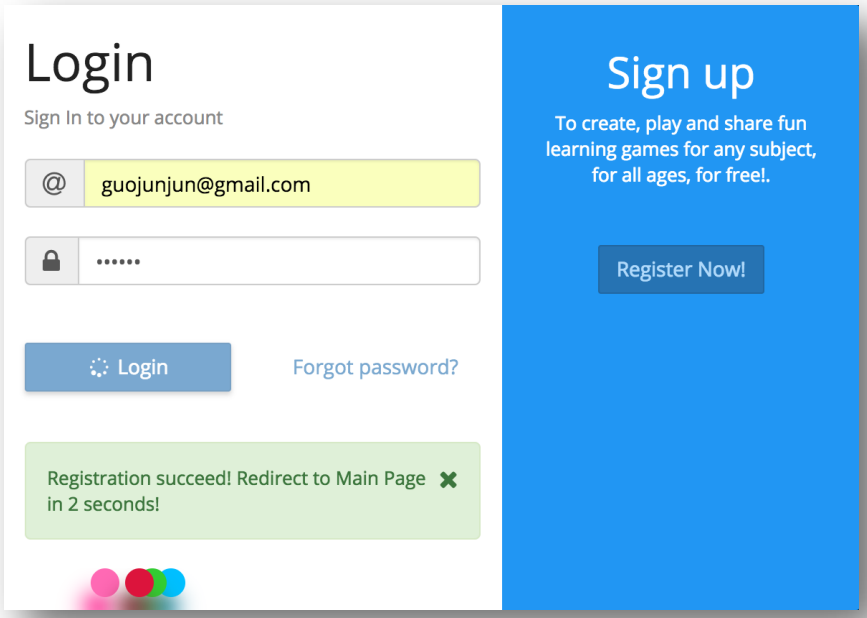


Figure 23 QuizClass UI: Login

8.1.2 Password reset

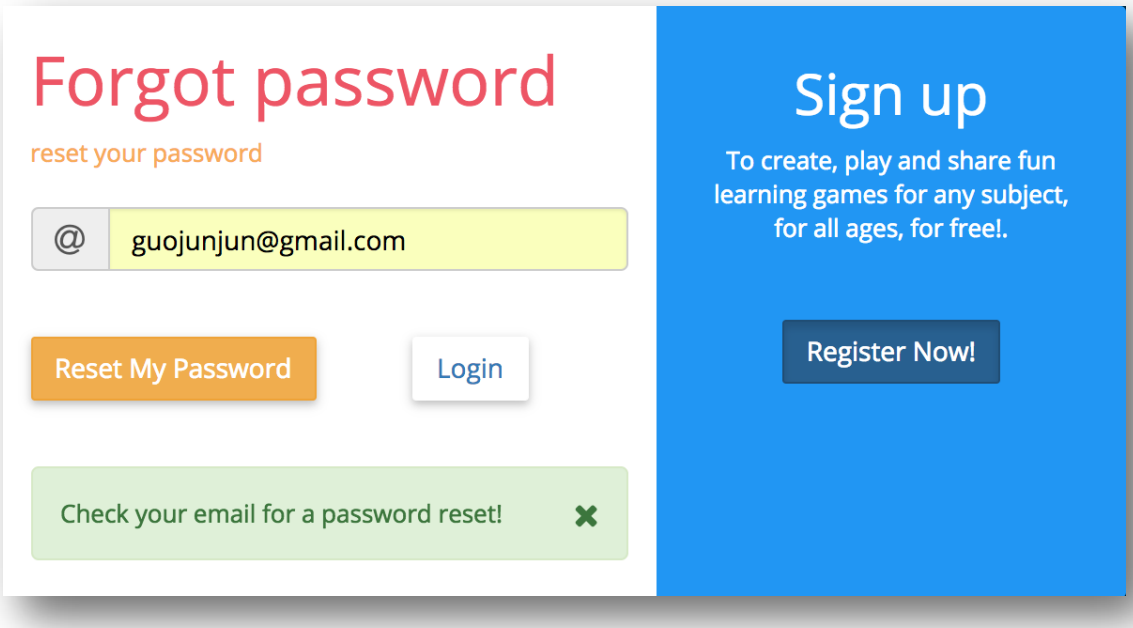


Figure 24 QuizClass UI: request for password reset

It is important for players/students or creators/teachers to access their account, even if they forget their password. End-user can easily reset the password by Forgot Password function.

Figure 24 shows the request for password reset UI. Figure 25 shows password reset link in user's email. The link is valid within 24 hours; a user can request a new one at any time.

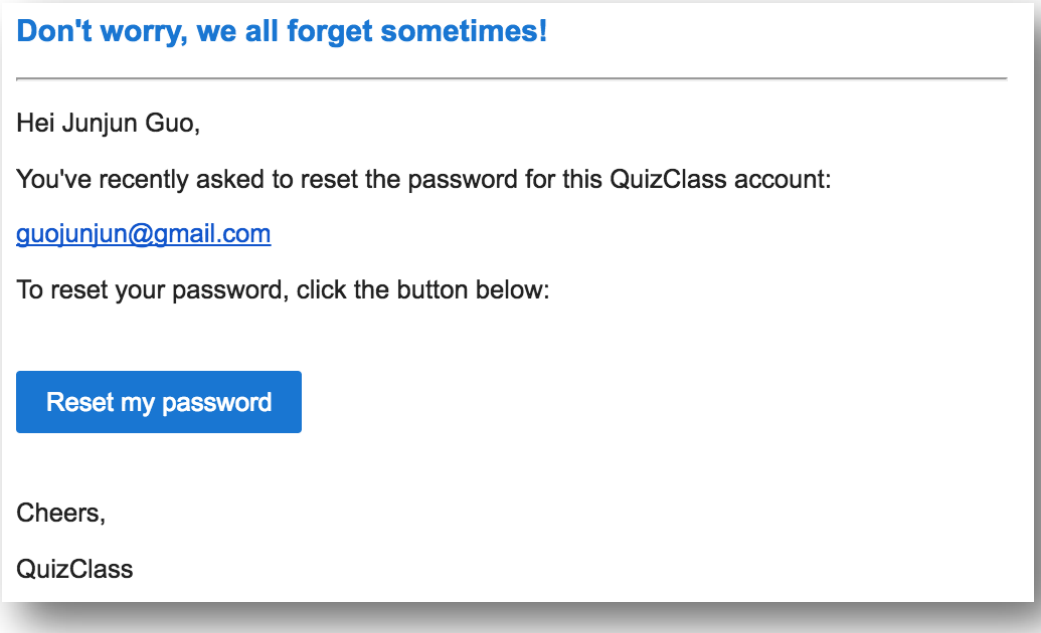


Figure 25 QuizClass UI: password reset link in email

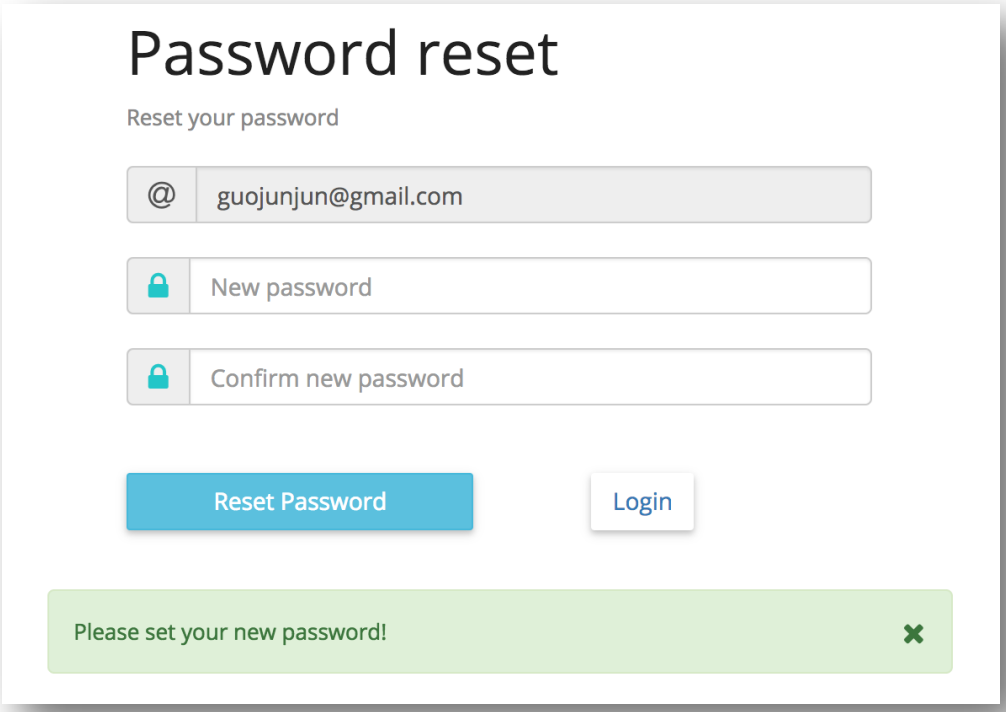


Figure 26 QuizClass UI: reset password

To reset password, end-user can click the password reset link from his or her email (Figure 25), and the password reset screen will be opened (as showing in Figure 26).

8.2 UI for the Users as *Player*

This section presents the Quiz Class UI and features for players, which includes Game Collection, Game, Rating system, Score system, and Ranking system.

8.2.1 Game collection view

Figure 27 shows the player's main screen. In the screen, the red color card represents a game collection that is suggested by the Quiz Class server. The best and public game collections will be recommended to the user/player. The light blue card represents a game collection that the user has already subscribed. The purple color represents a game collection that the user applied, but is waiting for an approval from the creator/teacher to enter the game collection. This is because this game collection is not public, as the small lock icon is in locked state, indicate that the collection is non-public. This is useful for teachers who only want their own students to enter the game collection (the course or classroom). There is also a green color card, which represents a game collection a user searched or recently searched, but not subscribed yet.

Navigation menu

The sidebar menu is showing on the left side, with which users can browse among Profile, Creator, and Analysis view. The sidebar menu has three status, full size (Figure 36), mini size (Figure 27), and invisible. How the sidebar menu is showing is depended on the user's browser screen size, e.g. desktop will show full size or mini size; mobile will show mini size or invisible.

Search bar

The search bar with a dropdown list (Figure 27) shows the matching results from search, and different colors are used to consistently represents the game collection (as described in Game collection view each color represents the game collection status).

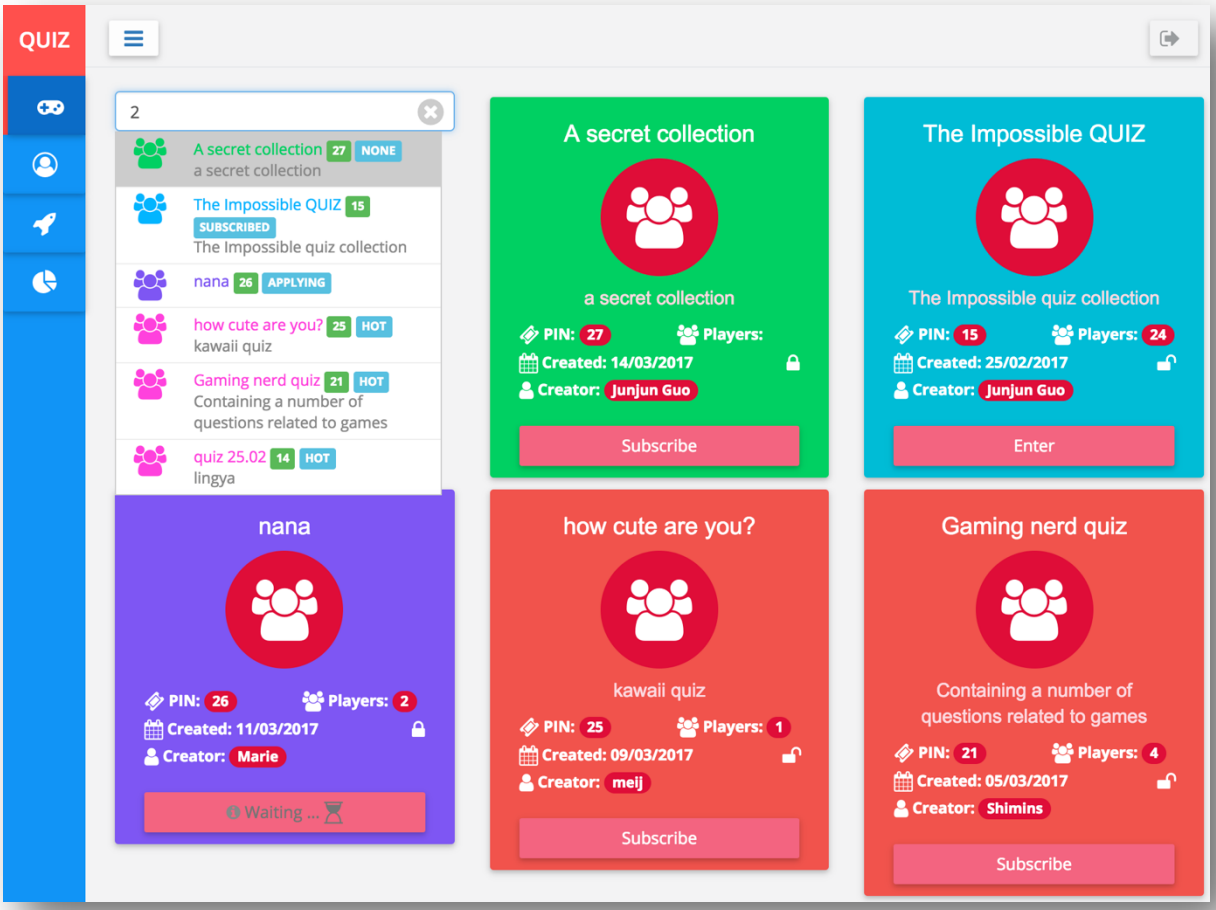


Figure 27 QuizClass UI: game collections for player

8.2.2 Game view

When a game collection is subscribed, a user can click enter button to get into the game collection. Figure 28 shows the game view. A game has 4 statuses, under construction, ready, started, and finished. And they are also indicated by the card background color, the orange color indicates that a game is ready. When the start time comes, the game status will be set to start. The start time and finish time tells when the game start and when the game finish, this can be used as when the student can start doing homework/assignment and when the deadline is. The light green color indicates the game is finished, which is also written on the button with an achieved flag. The players cannot enter the game anymore, but can still see the game ranking list (Figure 35). The gray color card represents a game that is still under construction. And the left side of Figure 28 shows an active game, in the real application the active game background color is continually transformed from code color to warm color (to indicate that the game is active).

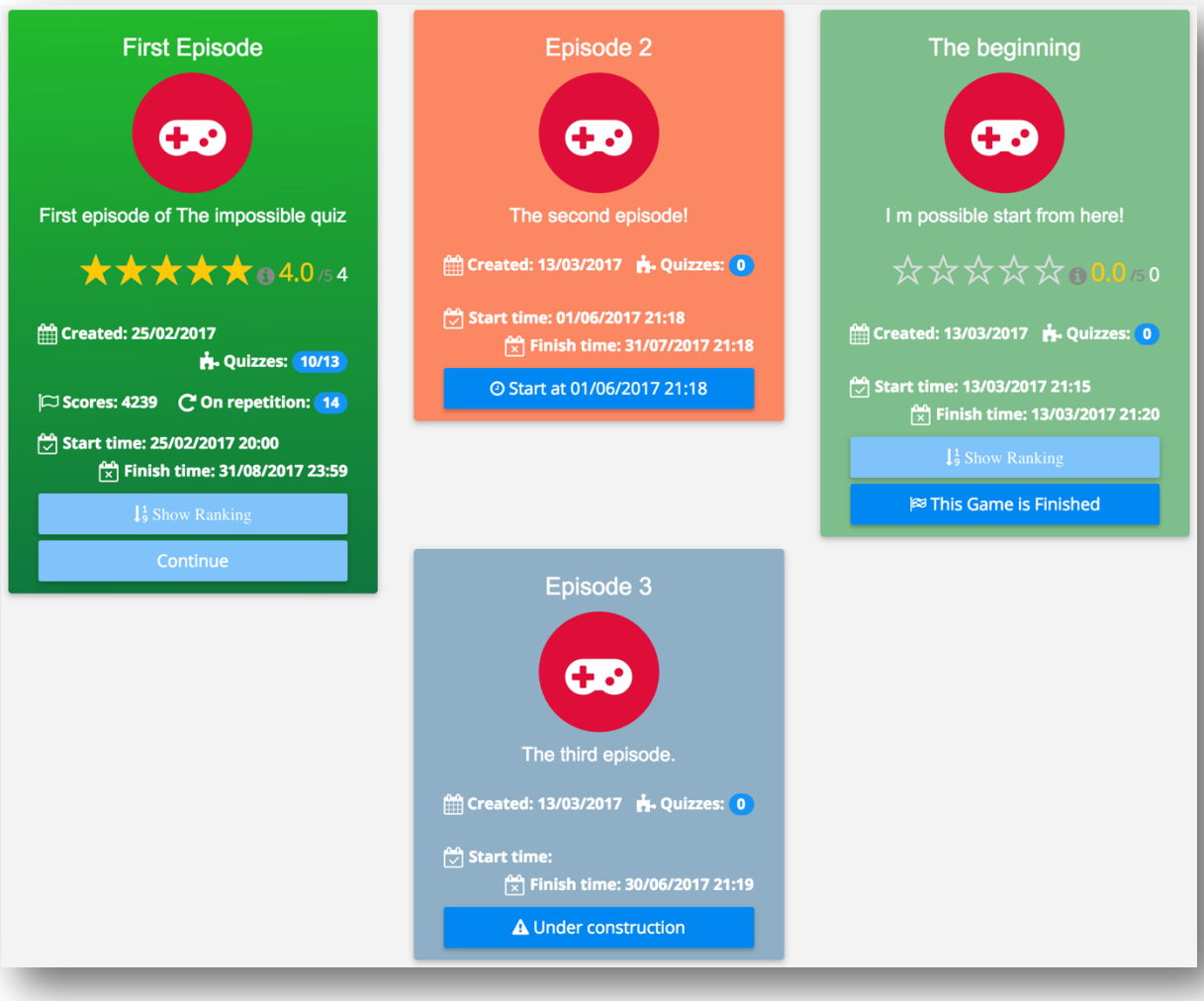


Figure 28 QuizClass UI: games view for player

8.2.3 Game rating

When a player finished a game, the player can rate the game, providing feedback to the creator about how the player feels about the game.

Figure 29 shows the end of a game screen, here a user can choose to rate the game or play it again.

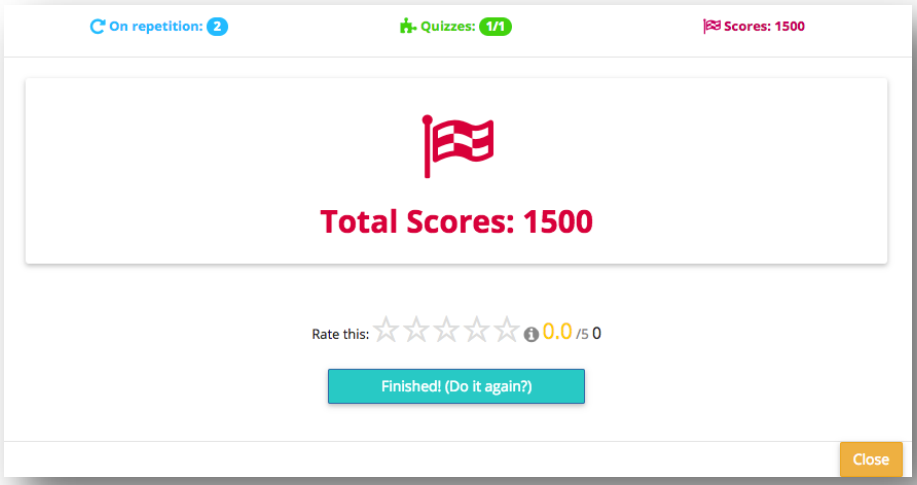


Figure 29 QuizClass UI: game finished

A player can also rate or update their rating in a game view (Figure 28).

8.2.4 Help System

Figure 30 shows that by click the help information icon 'i', help information will be displayed. The help information in Figure 30 shows how the rating system works, and what each number represent.

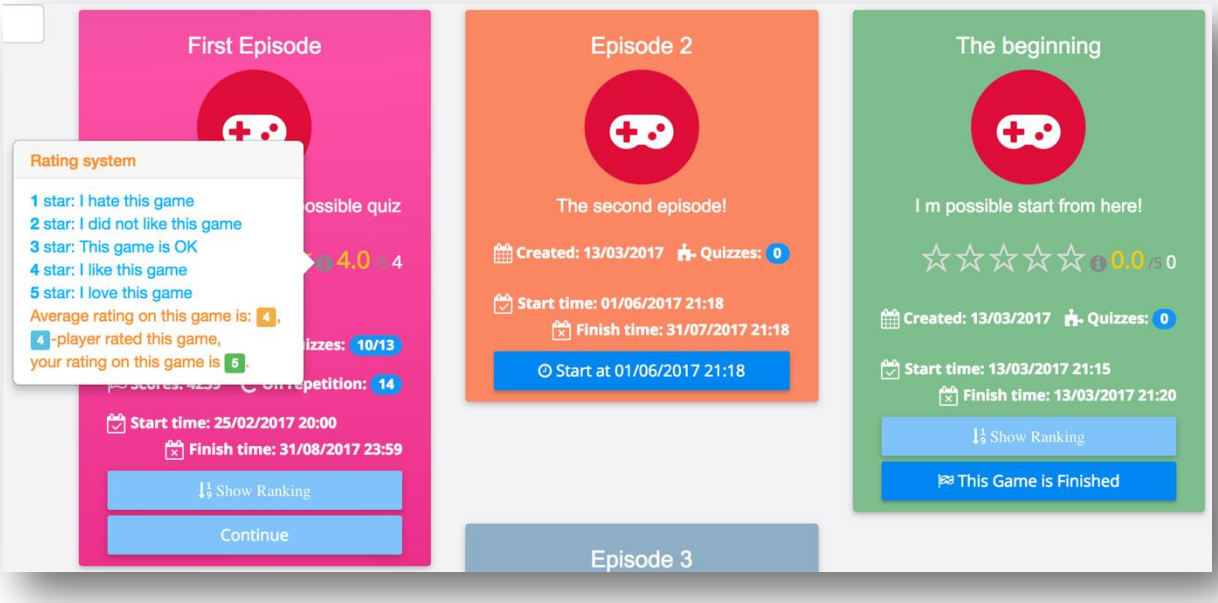


Figure 30 QuizClass UI: help information

8.2.5 Answering question and confidence level

When answer a question, a player can bet up to 500 of their "confidence" points, if only correct answer is selected, the betting points will be added to the final score as extra reward, and on the contrary, if any wrong option is selected, the betting points will be taken from their total score as punishment. The gamble-like feature is called 'confidence level' in Quiz Class, which represents how certain the player is, for the answer they give. This design gives the teacher possibility to know if the knowledge is clearly understood by the student.

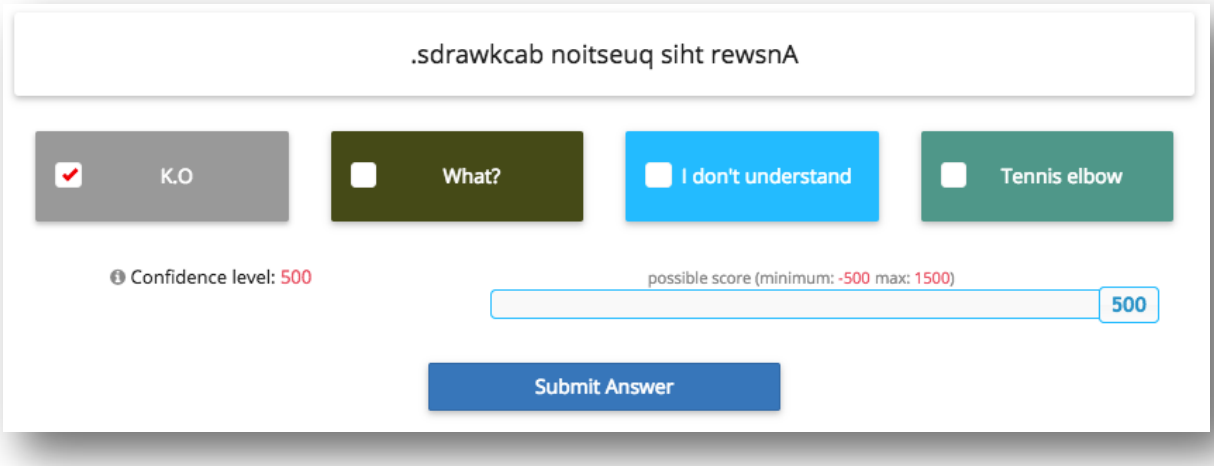


Figure 31 QuizClass UI: answer question

Figure 31 shows the answering question UI. A player can adjust the confidence level bar to set the confidence level. By adjust the confidence level bar, the system will immediately show both the chosen confidence level and the possible scores a player may get. So, the player is more clear about the consequences and feels a sense of control of the system. The answering UI is designed to be simple, clear, and no distracting information.

8.2.6 Score system

In the middle of the play (answering question), a player can click the Help icon 'i' to check the details of Scoring system (Figure 32). Players do not need to quit the game for querying the help information.

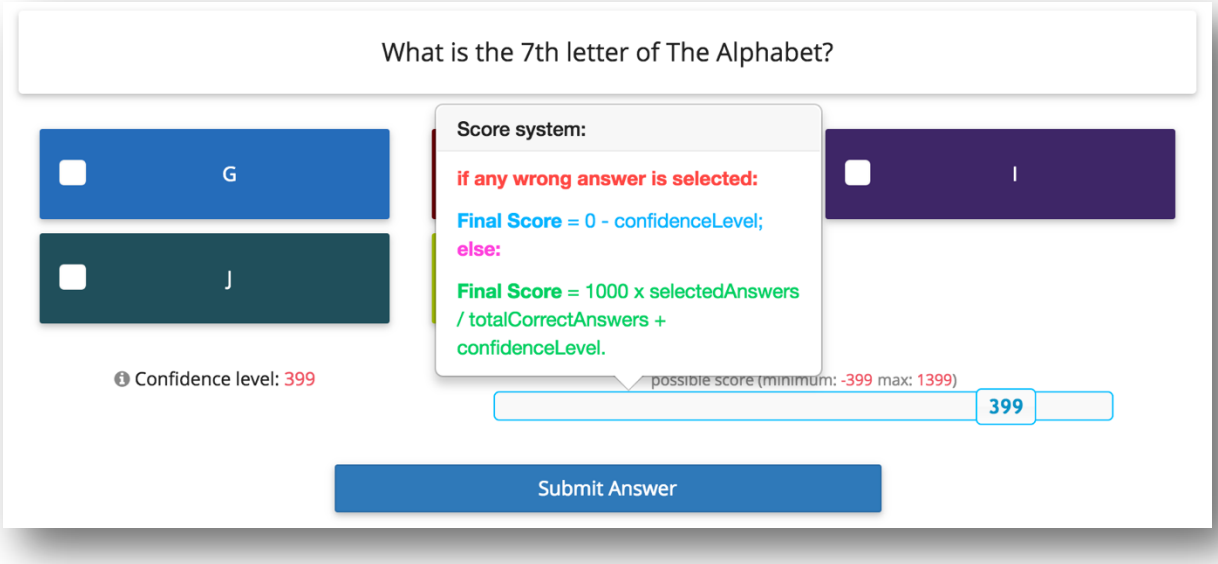


Figure 32 QuizClass UI: help info - score system in game play

8.2.7 Feedbacks for an answer

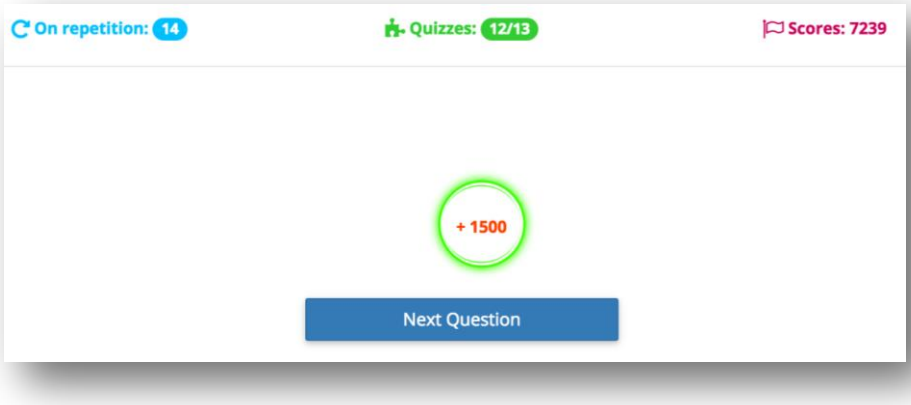


Figure 33 QuizClass UI: correct answer feedback

Figure 33 shows that the player has repeated this game 14 times, there are 13 quizzes in this game and the player has answered 12 of them. The current total score is 7239. The result from the last answer was +1500 points.

Figure 34 shows a red cycle with -500, which tells that the player has chosen a wrong answer, and the score for the answer is -500 points. There is a counting animation for every result to make it more game-like.

Figure 34 QuizClass UI: wrong answer feedback

8.2.8 Ranking

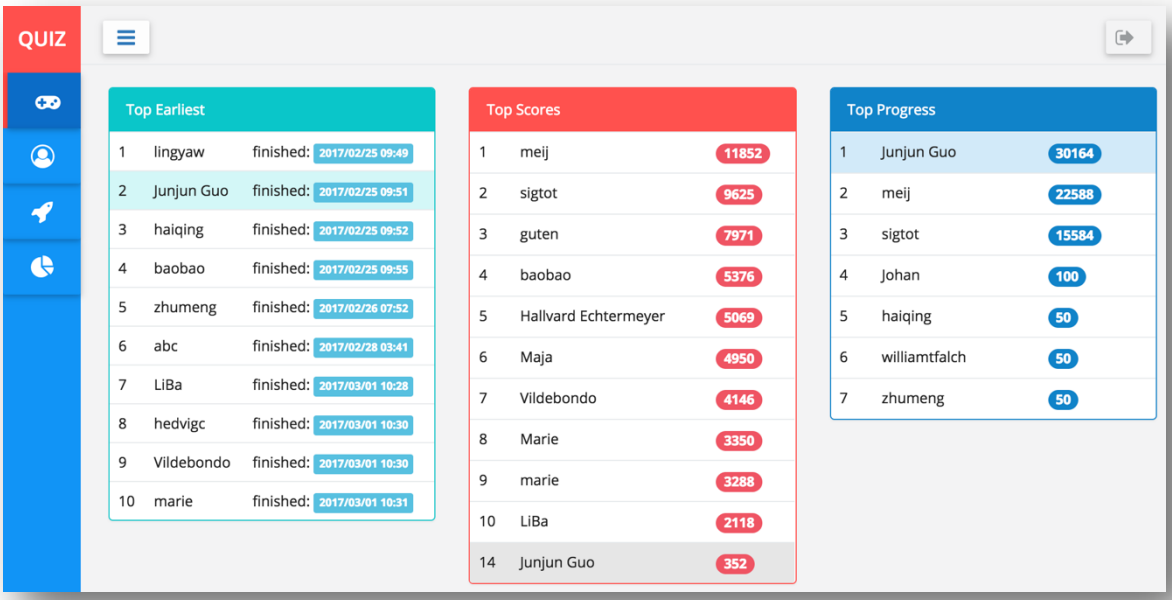


Figure 35 QuizClass UI: ranking lists

The game is designed to use multiple ranking systems, which brings different competition methods, so game players can win in different categories. The players' motivation is boosted up in three different ways, which are designed to achieve higher learning results.

Top scores

The Top Scores is total score a player get from each quiz. This list is dynamically changed, not only because there are upcoming students to do the homework, but also encourage students to do the homework again to get a better score. The idea is that anyone should have chances to improve themselves, and repetition is important for learning. To keep on top of the list, the student might have to play the game again to get a higher score, because if everyone

else is repeating, and getting better scores, the one who temporarily at the top of the list might lose their ranks.

Top earliest

Top earliest is a list of players who finished the homework earlier. This is the only ranking list that does not change, the only way to stay on top is start and finish as early as possible. The main purpose of this list is to encourage student to start doing homework as early as possible (doing homework when the deadline almost up is not encouraged).

Top progress

Top progress is a list for players with high progress scores. The purpose of this list is to encourage students to progress and do better by repetition.

To see the ranking lists (Figure 35) a player can click the 'show ranking' button on a started or finished game. Figure 35 shows the player 'Junjun Guo', ranks 2nd on the top earliest, with light blue background-color. On top scores, the rank of the player is 14th, which is not in top 10, therefore indicated by grey background-color. And on Top Progress list, the player ranking is 1st, and is indicated by light blue background-color.

8.3 User Profile

This section introduces the user profile, where users can view and edit their profile, change their password, and select language.

8.3.1 Language

The system will automatically match user's browser language (only English and Chinese were implemented). If there are any other languages is detected the system will use the English language, but the use can change it in their profile. As shown in Figure 36 users can use their preferred language by click the language button.

8.3.2 Contact Information

An end-user can check and update their profile at any time. Figure 36 shows the user contact information view. The system will always try to show use's username first. If the username is not set, then the system tries to show user's first-name and last-name, and if both are empty again, the system will show user's email address. This system allows players to have a

nickname (username) and a real name. The real name is important when a player is applying for a secret game collection. As the creator/teacher needs to know who the player is.

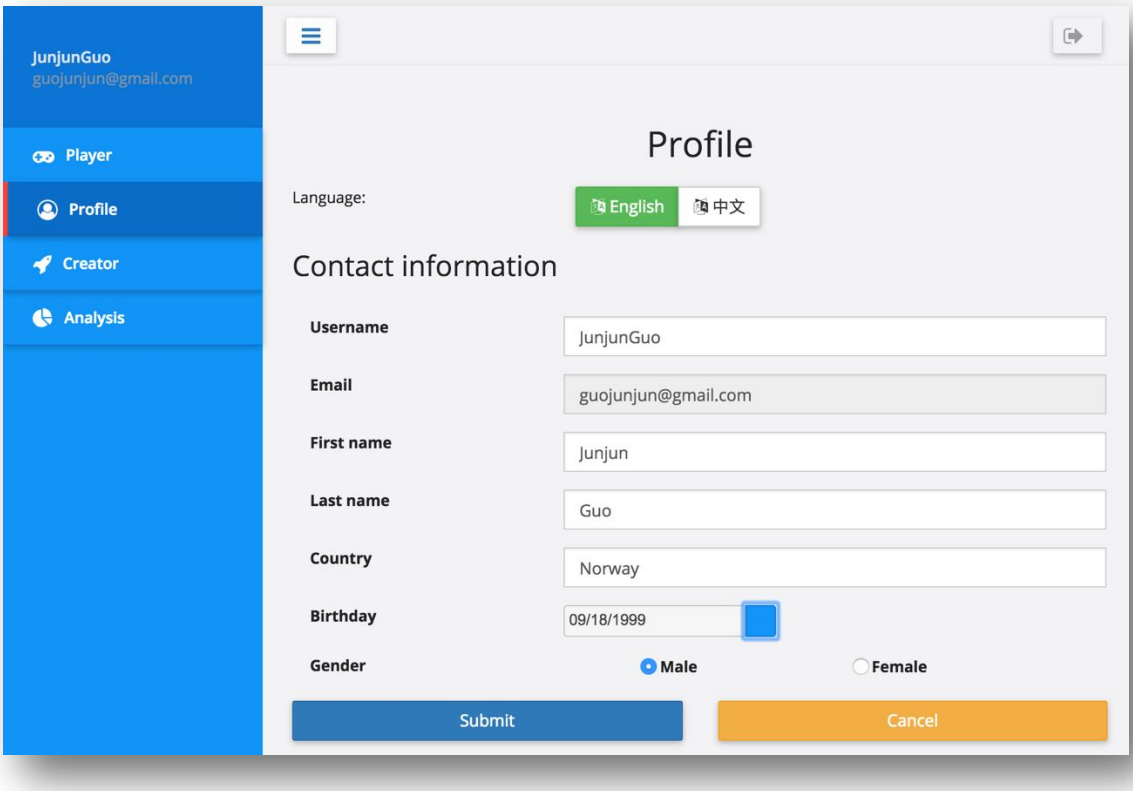


Figure 36 QuizClass UI: contact information

8.3.3 Change password

A user can change his/hser password at any time. Figure 37 shows change password view.

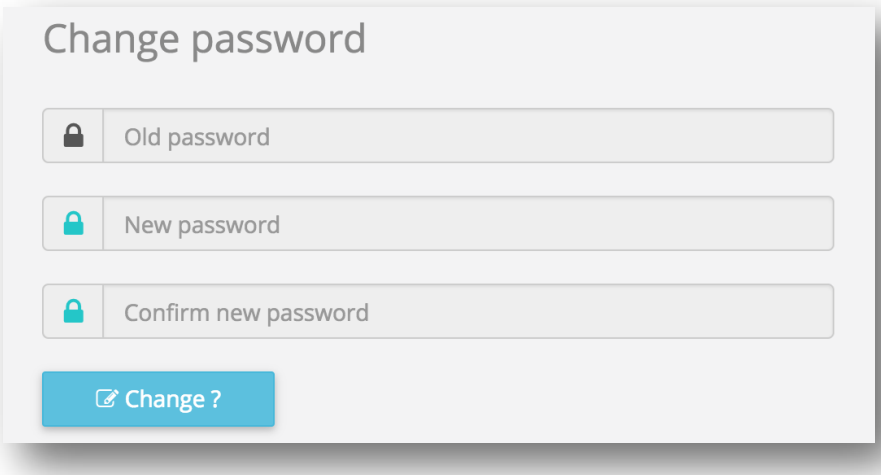


Figure 37 QuizClass UI: change password

8.4 UI for the Users as *Creator*

This section presents the Quiz Class UI and features for creators, which includes how to create a Game collection, a Game, a Question, and how to approve applications.

8.4.1 Creator's Game Collection view

Figure 38 shows all created game collections when creator view is open.

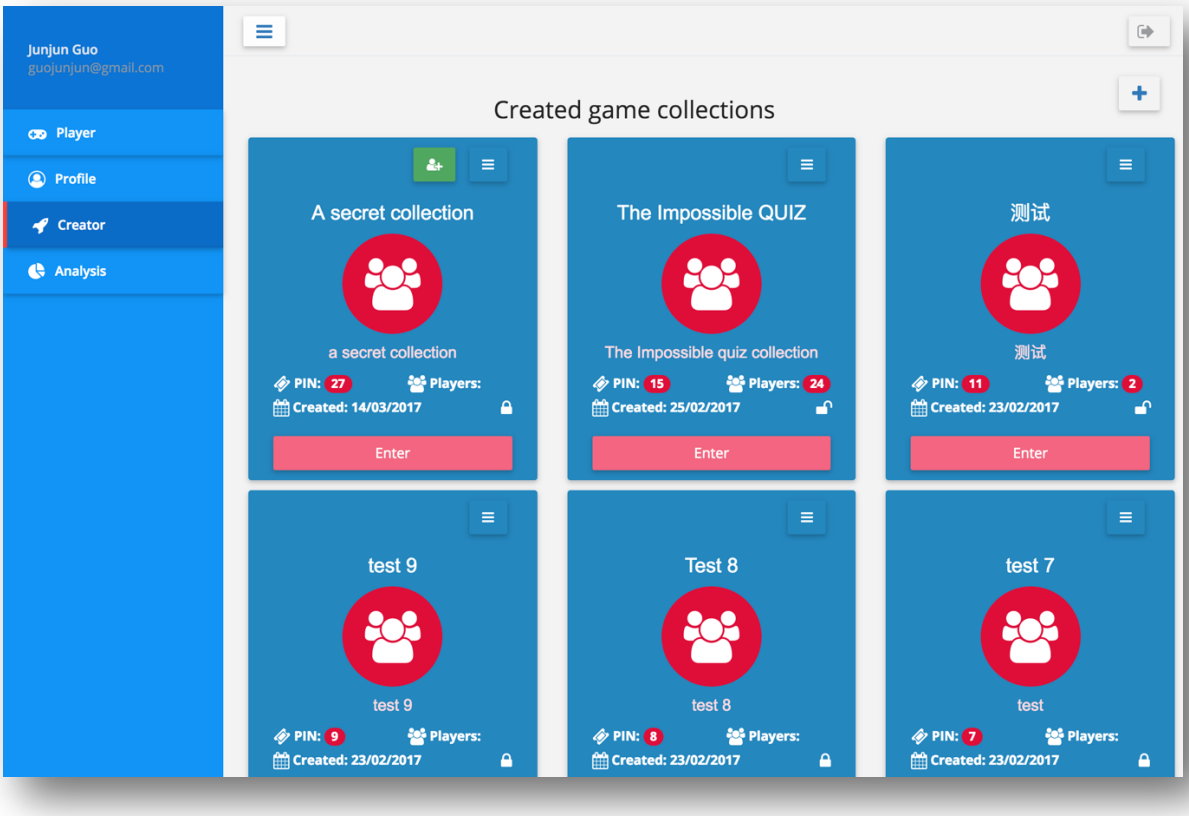


Figure 38 QuizClass UI: creator's game collections

8.4.2 Create new Game Collection

By clicking the '+' on the top right side of the 'Collection view' (Figure 38), a new game collection can be created. Figure 39 shows the Create New Game Collection window.

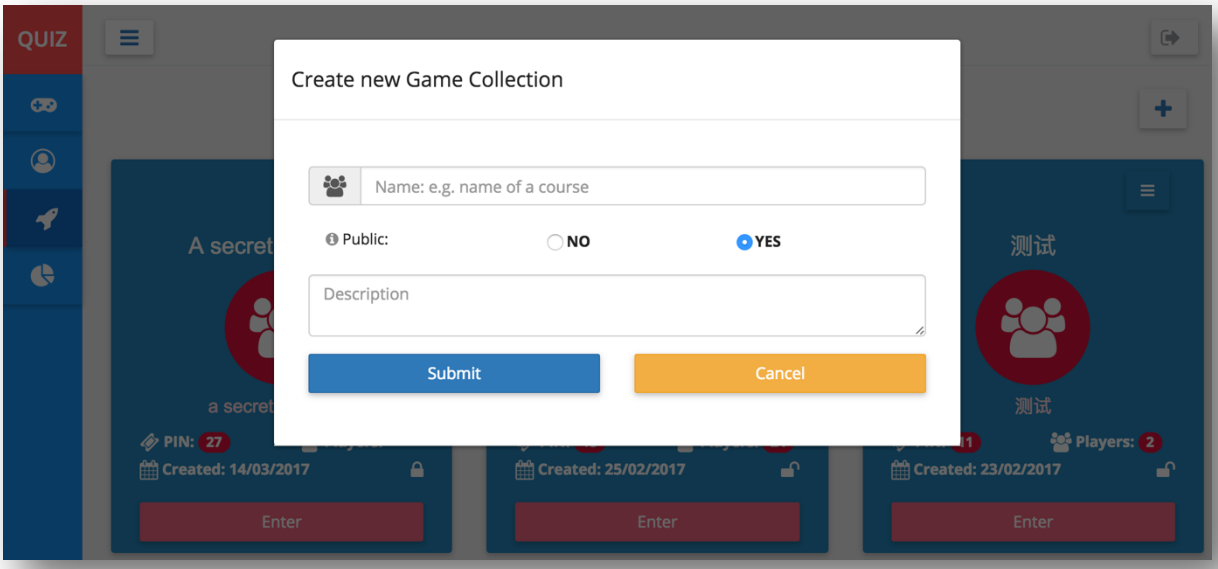


Figure 39 QuizClass UI: Create a new game collection

8.4.3 Edit or Delete a Game Collection

A created game collection can be edited or deleted at any time if there is no game inside. Figure 40 shows the Edit and Delete button on a game collection card.

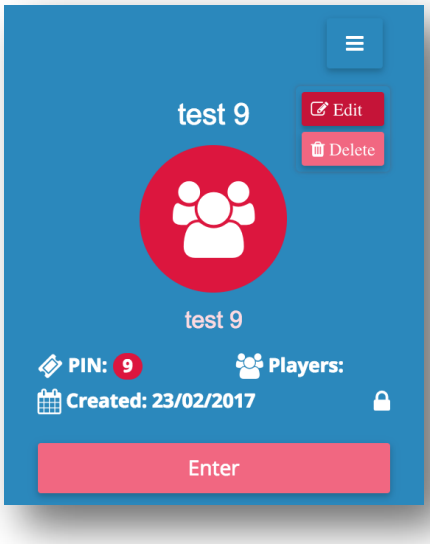


Figure 40 QuizClass UI: edit and delete a game collection options

To delete a game collection, a confirmation window will open with detailed information for the game collection, to make sure a wrong game collection may be deleted by mistake.

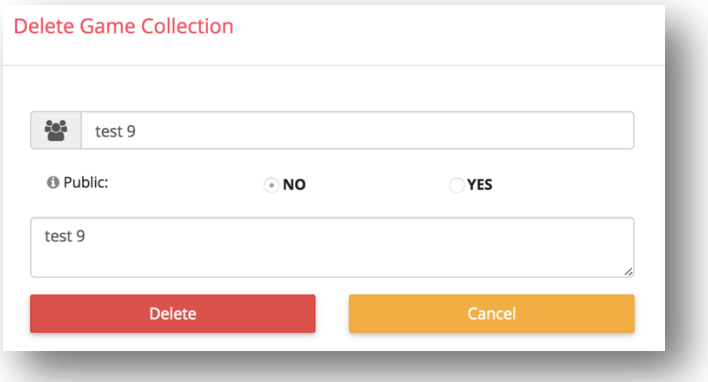


Figure 41 QuizClass UI: delete game collection

8.4.4 Approve Game Collection applications

When there are unhandled applications to a secret game collection, a light blue button with a person and plus icon will be shown on a game collection (Figure 42). The background color is changing to get attention from the creator.

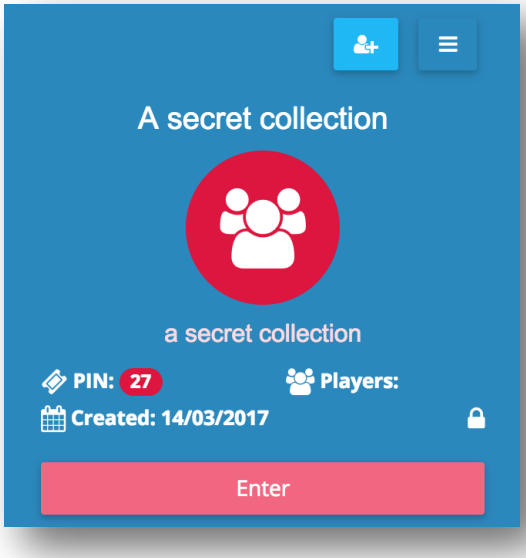


Figure 42 QuizClass UI: creator has application

By clicking the light blue button, a popup window will appear. Figure 43 shows a popup window with two applicants applying for this game collection. The players' email address, username, first name and last name is shown in one line. Names are optional for users, so the system will only show the names which existed. In the screenshot below, the first user has username, first-name, and last-name, the second user only has a username. By click the

applicant, a correct sign with green background color will show up, which tells the creator that if the 'Confirm' button is clicked this applicant will be approved to join the game collection.

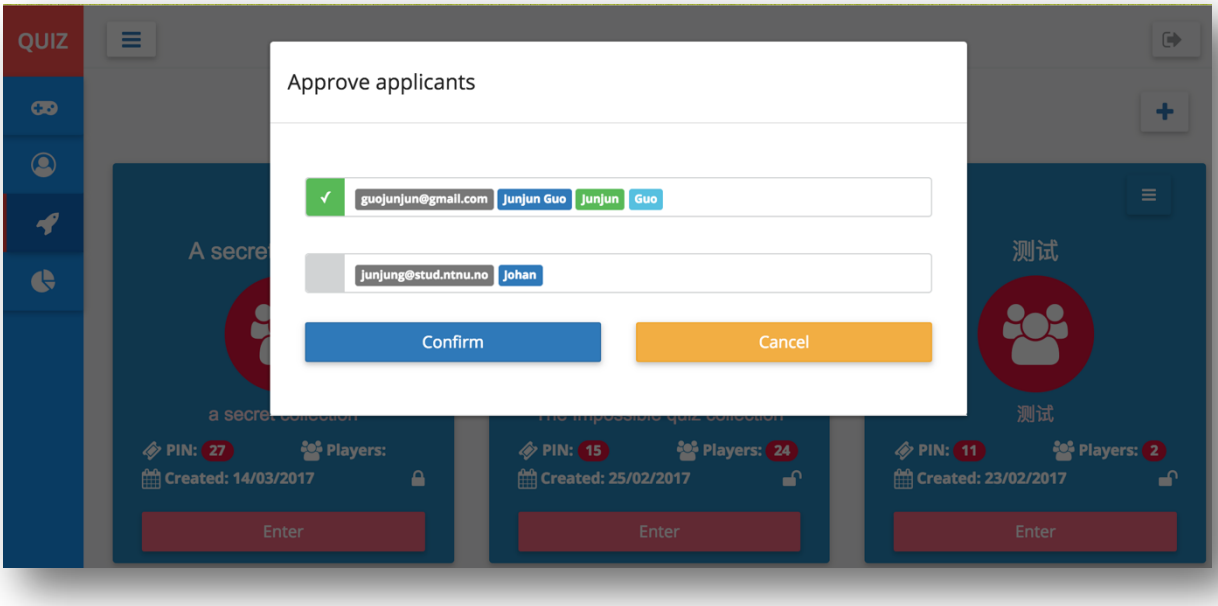


Figure 43 QuizClass UI: creator approve game applicants

8.4.5 Creator's Game view

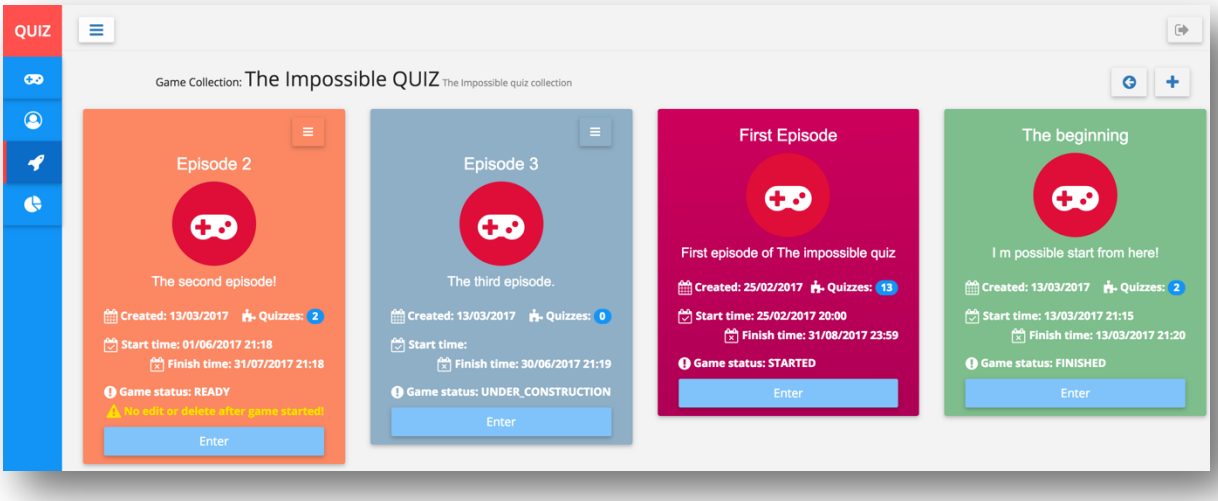


Figure 44 QuizClass UI: creator's games view

A creator's Game view (Figure 44) is like 'Game view' for players, but the creator can add new games, edit, and delete existing games. If a game is on ready status, an attention sign is

showing to alert the creator that no edit or delete can be made after the game started (so they may want to double check everything is correct made).

8.4.6 Create new Game

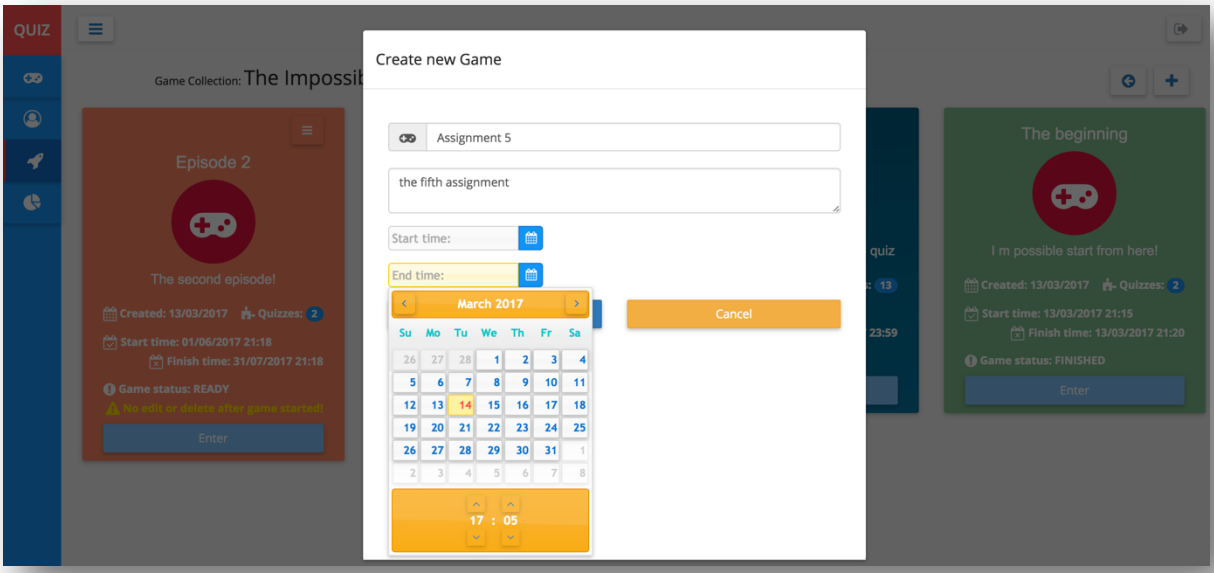


Figure 45 QuizClass UI: create a new game

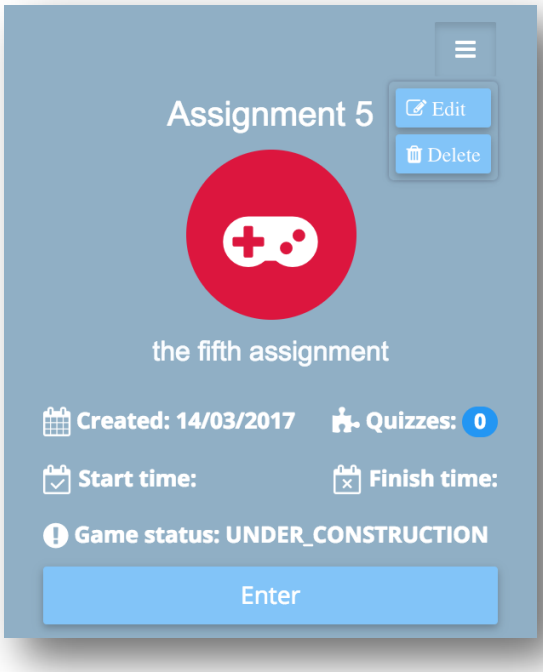


Figure 46 QuizClass UI: edit and delete option UI

To Create a new game, a creator only needs to write the name of the game, and can optionally give a description for the game. The start time and end time can be added or edited any time if the game is not started. And there is a data picker to help creator easily plan the time. Figure 45 shows the create new game UI. Figure 46 shows the buttons for edit or delete a created game.

8.4.7 Questions view

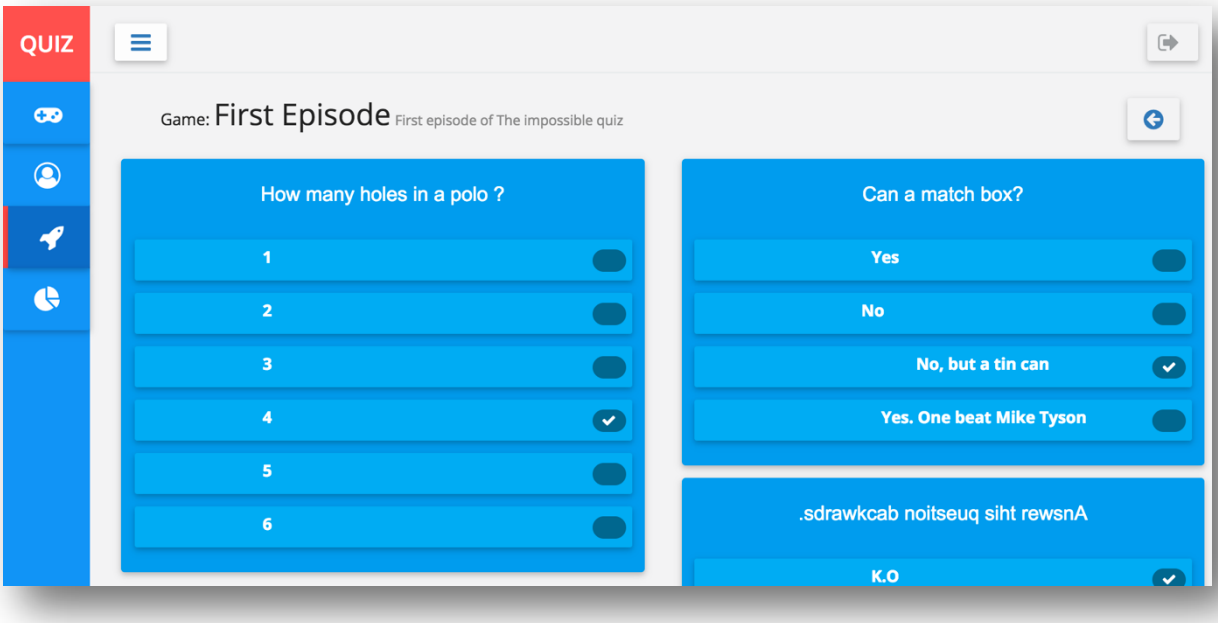


Figure 47 QuizClass UI: questions view

The creator can enter to a game and have an overview of the quizzes. The creator can add, edit, delete any questions if the game is not started. Figure 47 shows a stated game; its quizzes cannot be edited or deleted.

8.4.8 Create new Question

Figure 48 shows the Create New Question in a popup window. A question is required, description is optional. There must be at least one option for the question, and there must be at least one option is correct. The creator can add up to 6 options for a question.

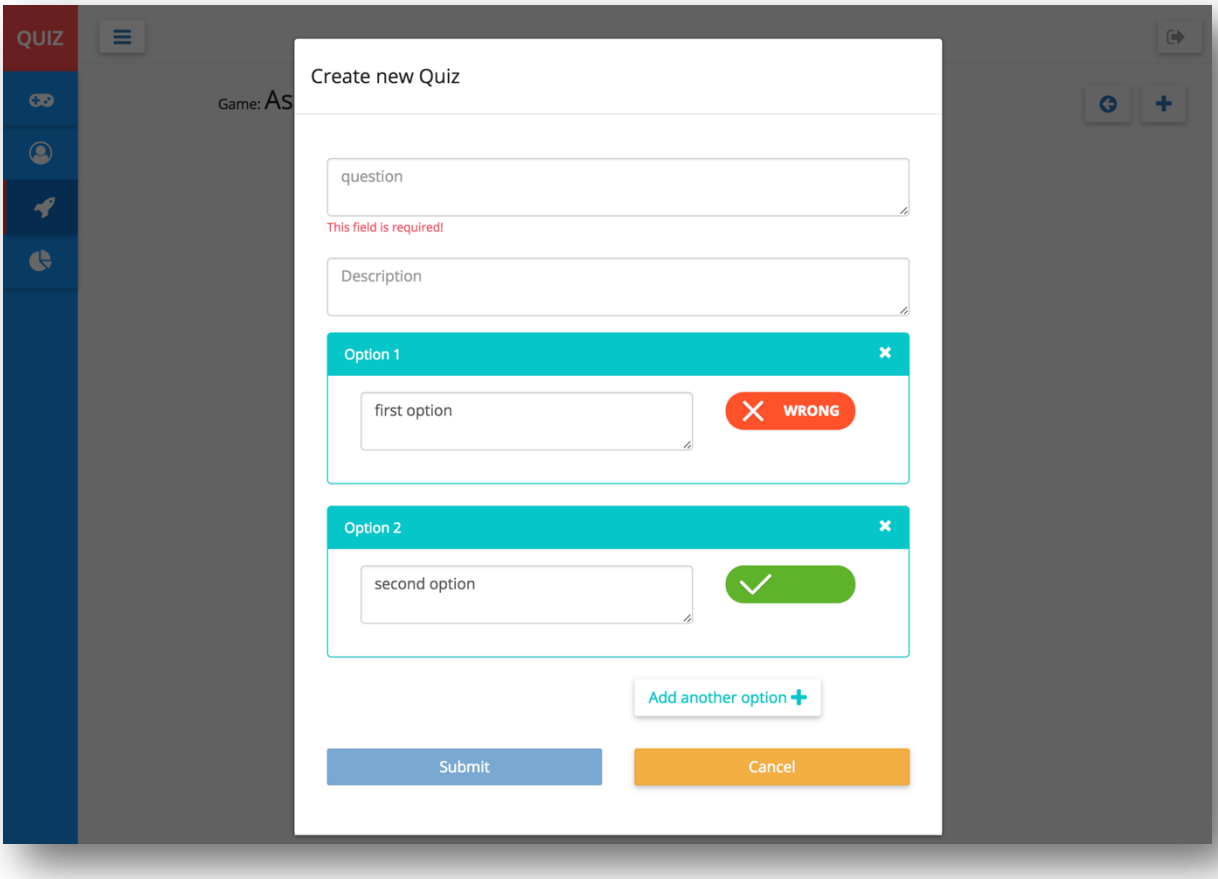


Figure 48 QuizClass UI: create new question

Figure 49 shows a question can be edited or deleted by click the edit or delete button.

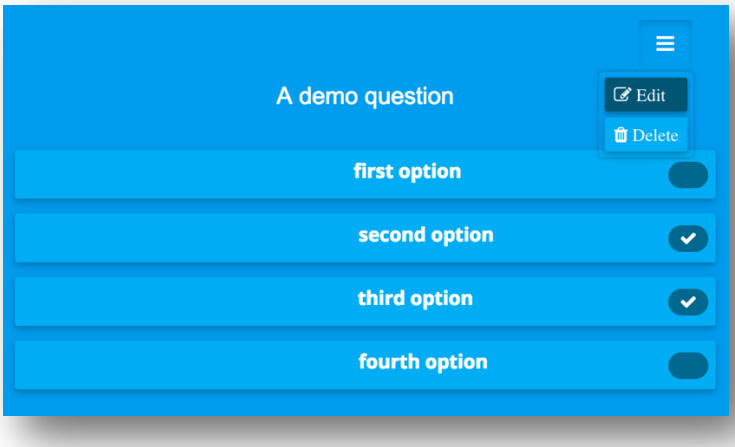


Figure 49 QuizClass UI: edit and delete question option

8.5 Analysis UI

This section introduces the analysis tools, how it looks, what features it have, and how it can be used.

8.5.1 Game collections view

Figure 50 shows a table of the game collections a creator has created. Each row represents a game collection. The table shows each game collection's Id, the number of games in the collection, the number of players subscribed the game collection, the game collections name, and the description. The table can be sorted by any of the columns. Figure 50 shows the default sorting, sort by game collection ID increasingly.

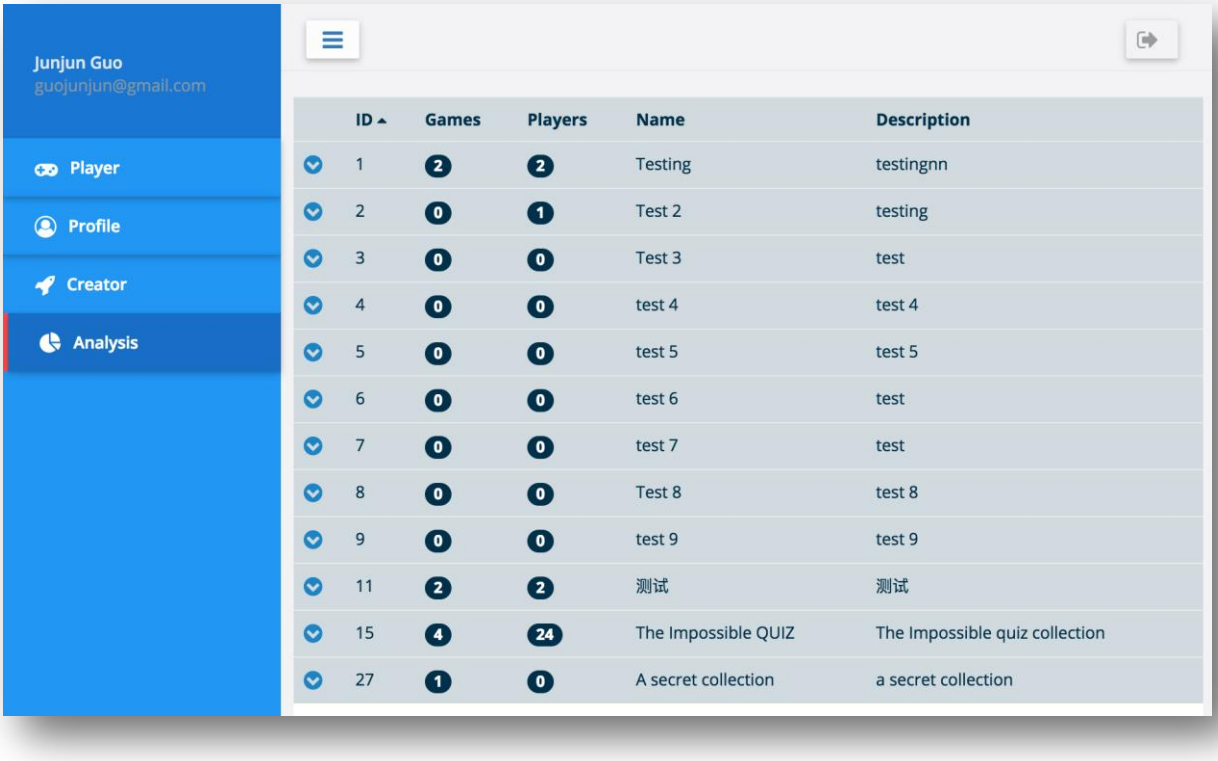


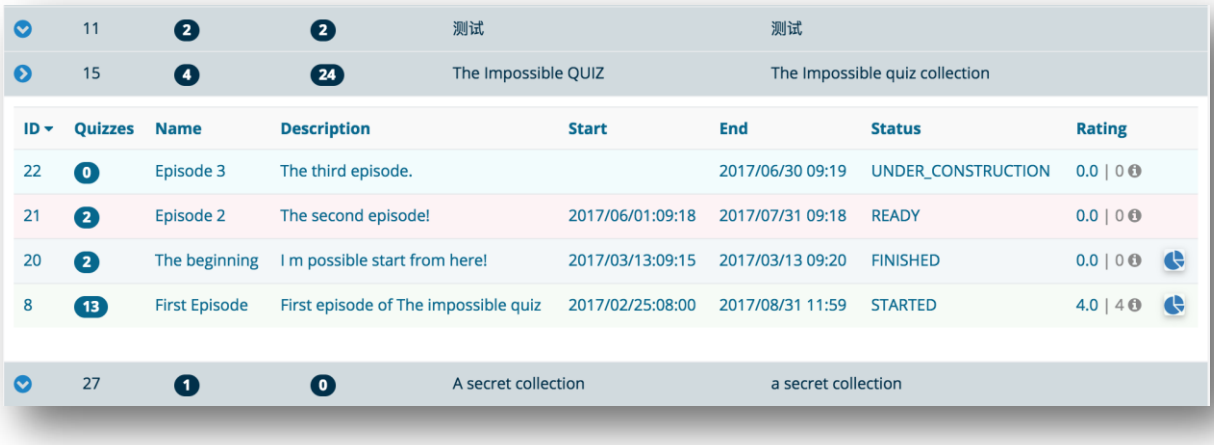
Figure 50 QuizClass UI: analysis game collection table

By click on a row (a game collection) an inner table will open, to close the inner table, the user just needs to click the row again. The inner table will show the games in the game collection (Figure 51).

8.5.2 Game collections and games view

Figure 51 shows 'The Impossible QUIZ' with game collection id 15 is opened. The inner table shows Game Id, the number of questions in the game, game name, description, game start time, game end time, game status, game rating and a small pie icon indicate that there is analysis data ready to view. Each row of the game table represents a game, and can be sorted by any column, Figure 51 shows the default sort is by game id decreasingly.

The game table (game collection's inner table) has also different background colors for each row, which is depend on the game status.



	11	2	2	测试		测试		
	15	4	24	The Impossible QUIZ		The Impossible quiz collection		
ID	Quizzes	Name	Description	Start	End	Status	Rating	
22	0	Episode 3	The third episode.		2017/06/30 09:19	UNDER_CONSTRUCTION	0.0 | 0	
21	2	Episode 2	The second episode!	2017/06/01:09:18	2017/07/31 09:18	READY	0.0 | 0	
20	2	The beginning	I m possible start from here!	2017/03/13:09:15	2017/03/13 09:20	FINISHED	0.0 | 0	
8	13	First Episode	First episode of The impossible quiz	2017/02/25:08:00	2017/08/31 11:59	STARTED	4.0 | 4	
	27	1	0	A secret collection		a secret collection		

Figure 51 QuizClass UI: analysis game table

A user can click anywhere in a row to open the analysis view (if there is a pie icon at the end of the row).

8.5.3 Open game analysis

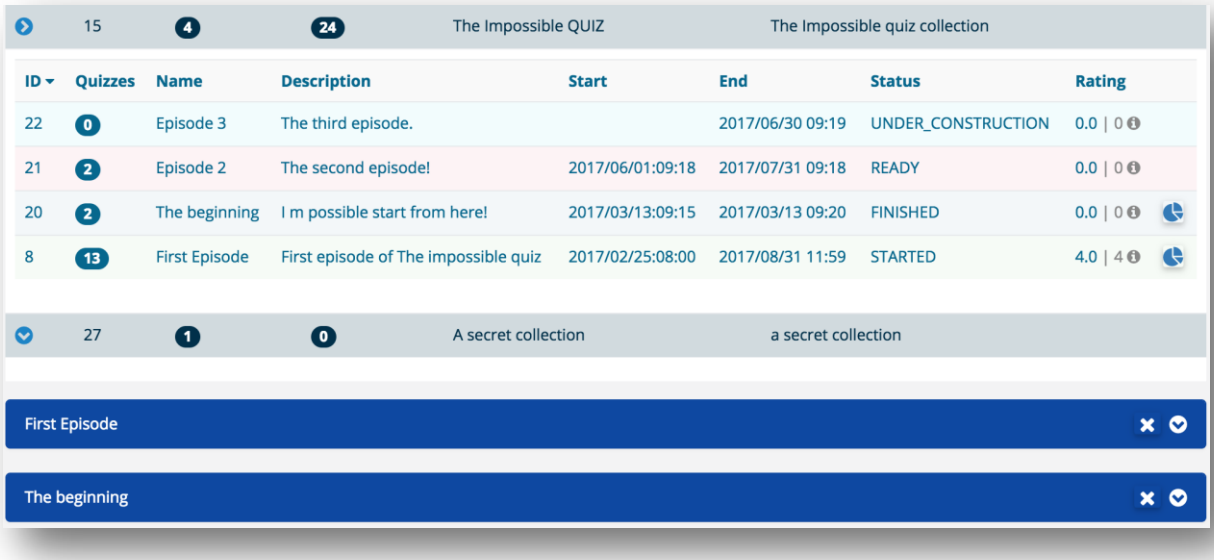


Figure 52 QuizClass UI: open game analysis

Figure 52 shows two games has been clicked for analysis. A user can click anywhere on the deep-blue bar to collapse or open the analysis data for the game. It can also be removed by click on the 'x' close icon.

Figure 53 shows when a game analysis bar is open, it shows four types of data analysis: Top list, Quizzes overview, Quizzes, and Time View.

Figure 53 QuizClass UI: analysis types

8.5.4 Top List

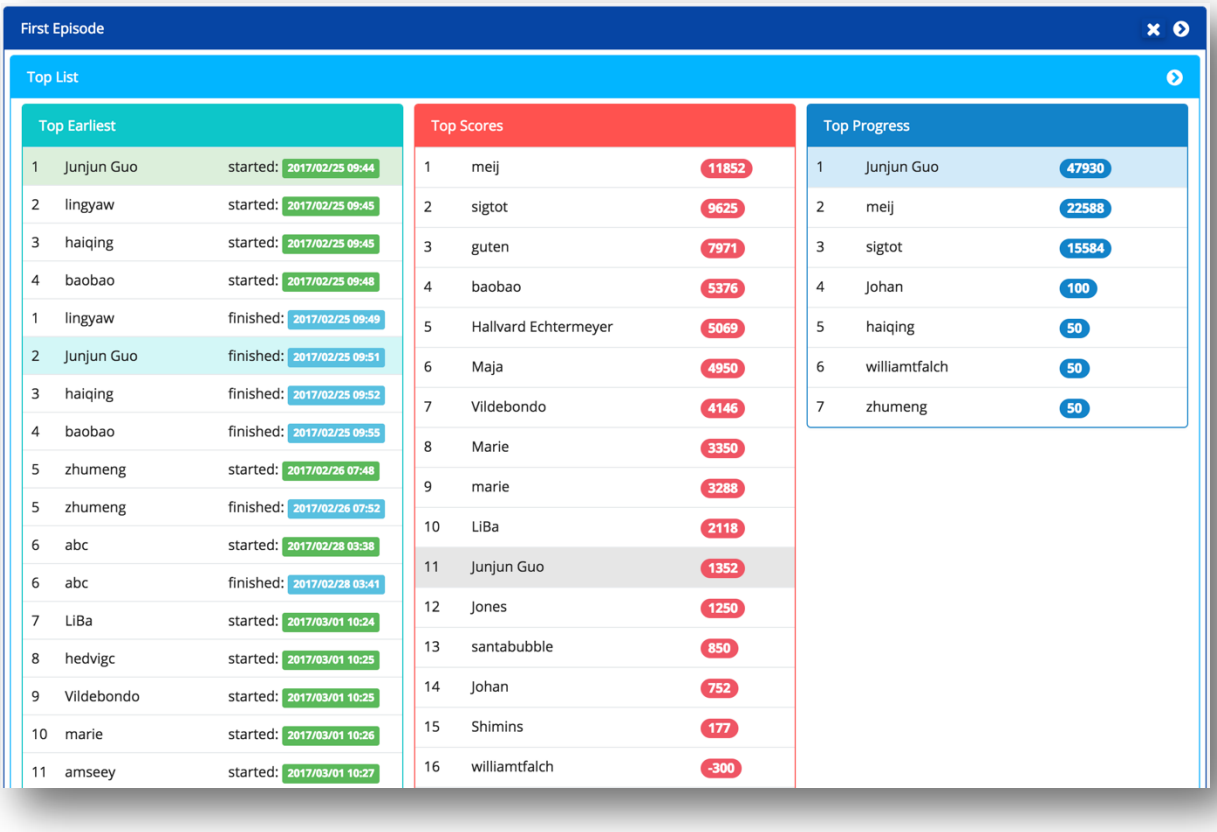


Figure 54 QuizClass UI: analysis top list

The creator/teacher has a full list of all the game participants. A creator self can also be a player for the created game (which may be useful for testing). Figure 54 shows the creator has also played the game he created.

8.5.5 Questions overview

The quizzes overview (Figure 55) represents the average score and average confidence for quizzes in a game. The light blue line shows the average score for each quiz, and the pink line indicates the average confidence for each quiz. The x-axis shows each quiz; the y-axis shows the scores. As indicated in Figure 55, the third quiz has the highest average score, but its average confidence level is not very high; the seventh quiz has the lowest average score, but its average confidence level is even higher than the third quiz. This may tell the creator/teacher that the players/students may have a serious problem with this question, that the teacher may need to put some effort on it.

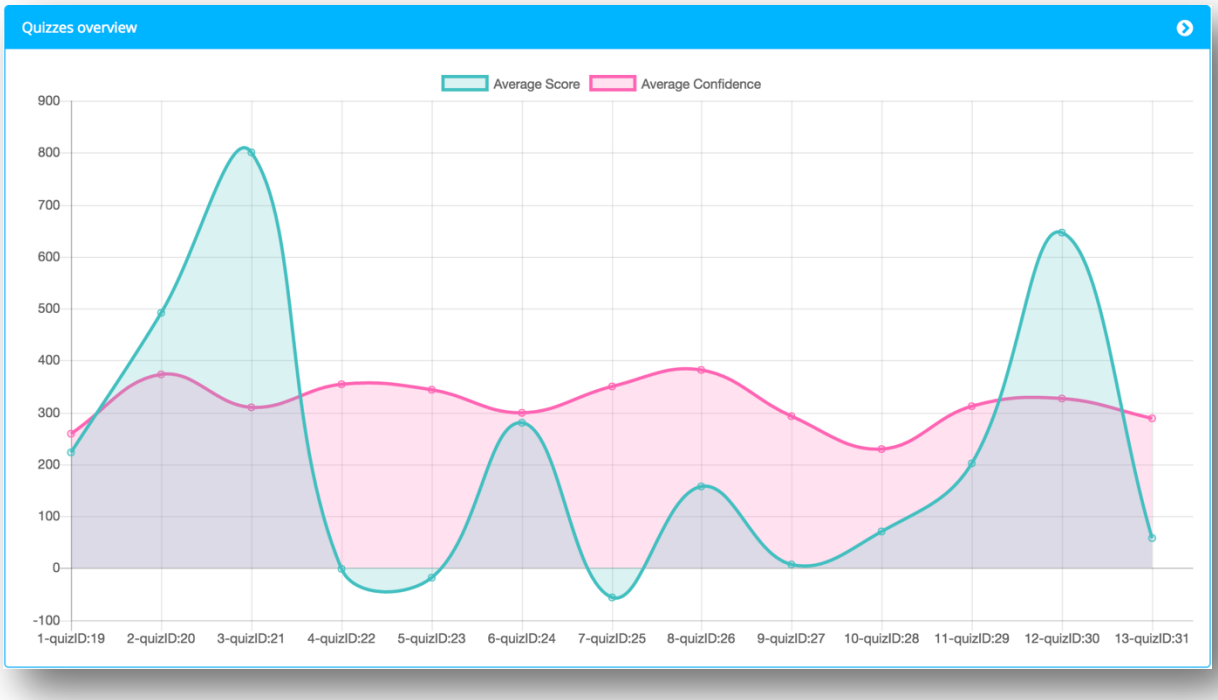


Figure 55 QuizClass UI: questions overview

8.5.6 Question view

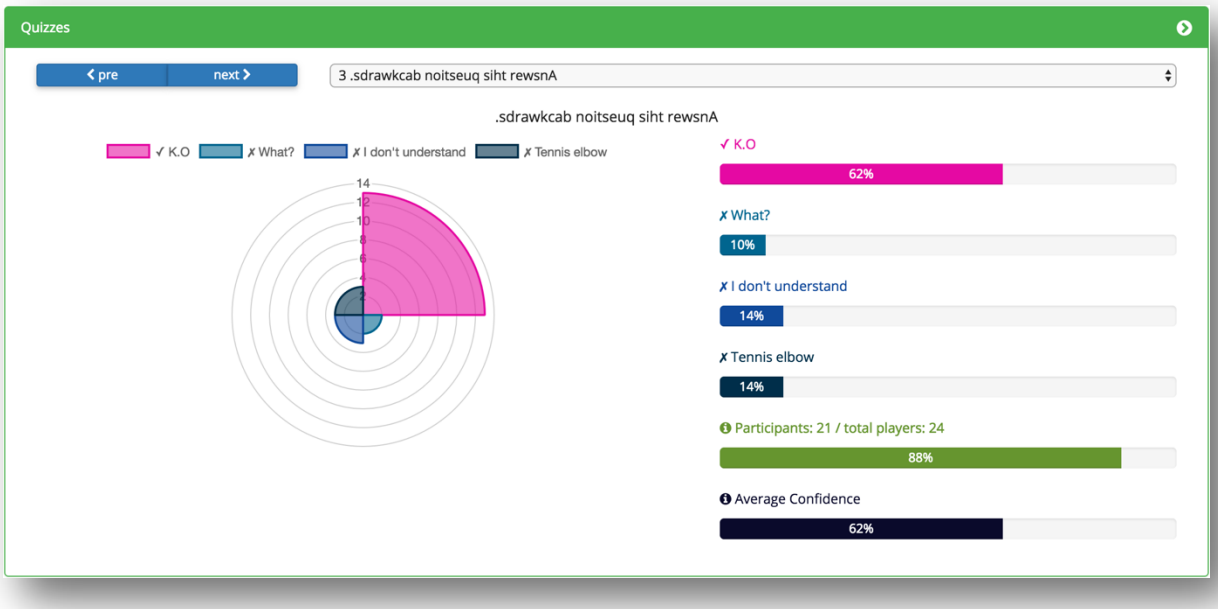


Figure 56 QuizClass UI: question result analysis

Quiz view analysis each individual quiz. On a big screen the quiz analysis will show both pie view and bar view, on a smaller screen like a mobile, only bar view will be shown. Both pie

view and bar view shows the question and each option (options got different colors), each option shares the same color on pie view and the bar view. The pie view is more intuitive and general; the bar view is more detailed. Figure 56 shows the third quiz. On pie view, it shows there are 13 players has chosen the correct answer, and it is the biggest piece, represent that most players got it right. The bar view shows that 62% have chosen this correct answer. The bar view also tells that there are 21 out of 24 players answered this question, and the average confidence on this quiz is 62%.

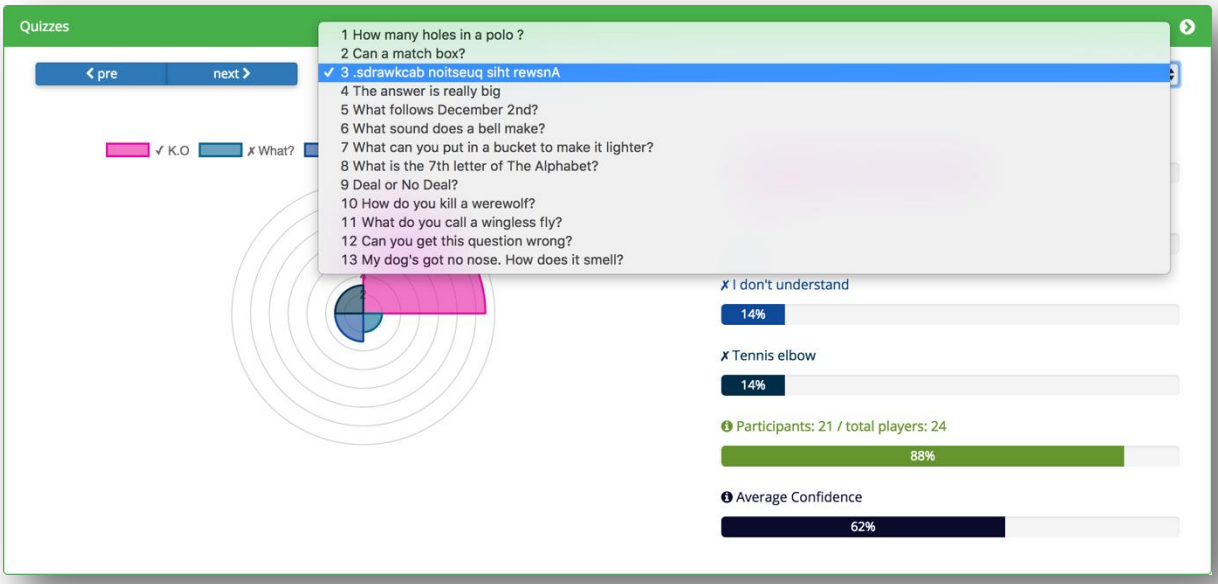


Figure 57 QuizClass UI: choose a question to analyse

The ‘pre’ and ‘next’ button lead to the previous or next question. The dropdown list (Figure 57) can also be clicked to quickly jump to any of the questions.

8.5.7 Time analysis: bar view

Figure 58 shows that the Time analysis is divided into bar view and bubble view.

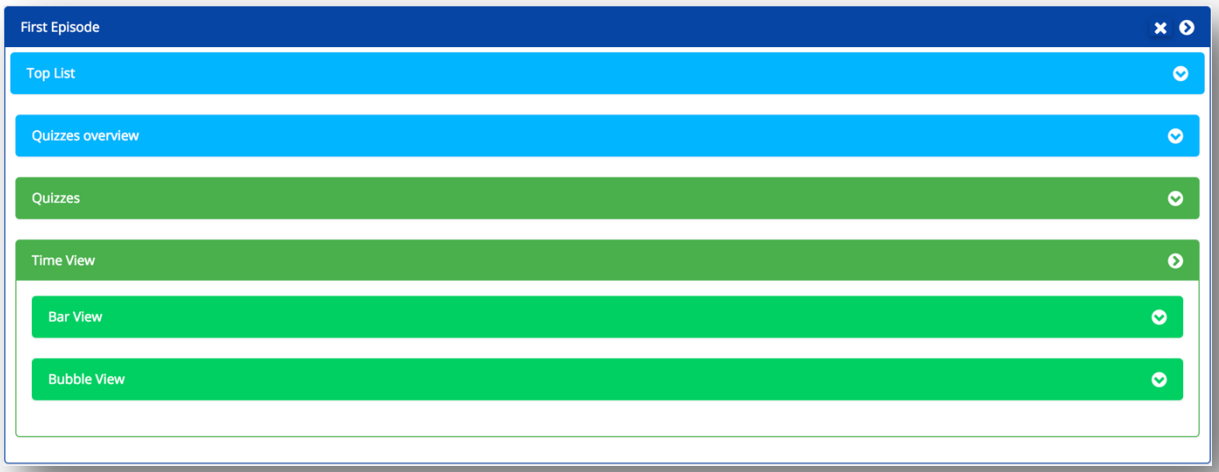


Figure 58 QuizClass UI: type of time analysis

Figure 59 QuizClass UI: bar view for time analysis

The bar view Figure 59 shows how many players (y-axis) started at a day (x-axis) with stronger border color. And how many players finished the game at a day with bar color that has lighter border color. This view simply tells the teacher on a given day how many students start doing homework/assignment and how many are finished.

8.5.8 Time analysis: bubble view

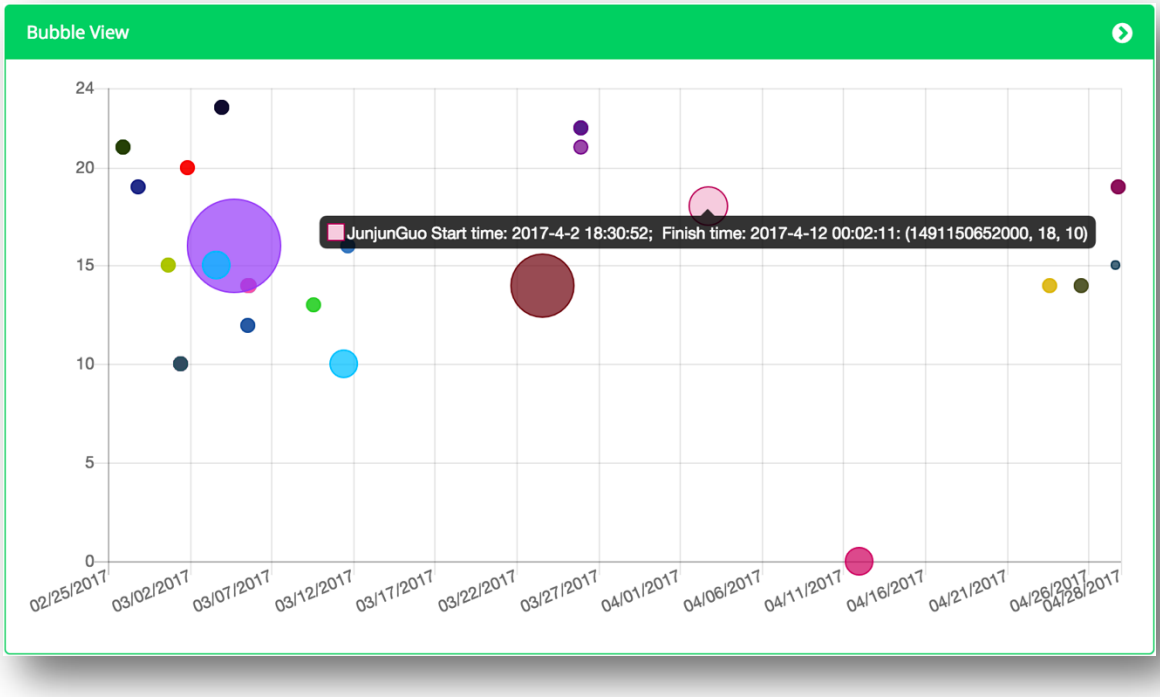


Figure 60 QuizClass UI bubble view for time analysis

The bubble view gives a general view about what time (y-axis) a player started and finished a game, and at which day (x-axis). The teacher may also track each individual player for the start and end time for a game (homework/assignment). The view also tells how long time a player use for a game. The bigger the bubble is the longer time the player used to finish a game. This may be useful for the teacher to know when students started and finished the homework/assignment, and how long time each student used. Figure 60 shows most player use a little time to finish the game (most of the bubbles are small). This view also let the teacher know the time distribution of each homework. Teachers can easily find out what time of a day most of the students started to do their homework (Figure 60 shows most players started and finished around 15 o'clock), and when most students started to do the homework (they started early or near the deadline, in Figure 60 most players started at the very beginning). And if there are too many big bubbles, the teacher might consider that the homework is difficult or needed too much time to do.

9 Architecture design

Software should build on a solid foundation, fail to consider key scenarios, long-term consequences of key decisions, may put application at risk. Good architecture reduces the business risks associated with building a technical solution (Patterns and Team 2009). This chapter presents the software architectural of the Quiz Class. The software architecture will fulfill the functional and non-functional requirements described in Chapter 7.

> The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them (Bass, Clements et al. 2003).

Various stakeholders may read the architecture of this project: the end-users include students and teachers with different backgrounds, software developers, my supervisor, and co-supervisor. In order to address large and challenging architectures, Kruchten introduced "4+1" View Model of Software Architecture (Kruchten 1995), the four views are:

1. ***The logical view***, which is the object model of the design (when an object-oriented design method is used);
2. ***the process view***, which captures the concurrency and synchronization aspects of the design;
3. ***the physical view***, which describes the mapping(s) of the software onto the hardware and reflects its distributed aspect;
4. ***the development view***, which describes the static organization of the software in its development environment (Kruchten 1995).

The description of an architecture—the decisions made—can be organized around these four views, and then illustrated by a few selected use cases, or scenarios which become a fifth view (Kruchten 1995). Figure 61 shows the "4+1" View Model.

The advantage of the "4+1" View Model is that it maps stakeholders (end-user, developers, systems engineers, project managers, etc.) to the type of information that they need, without requiring specific modeling notations to be used. The emphasis is on ensuring that all stakeholders have the information to understand the system. And the functional and non-functional requirements can be handled separately.

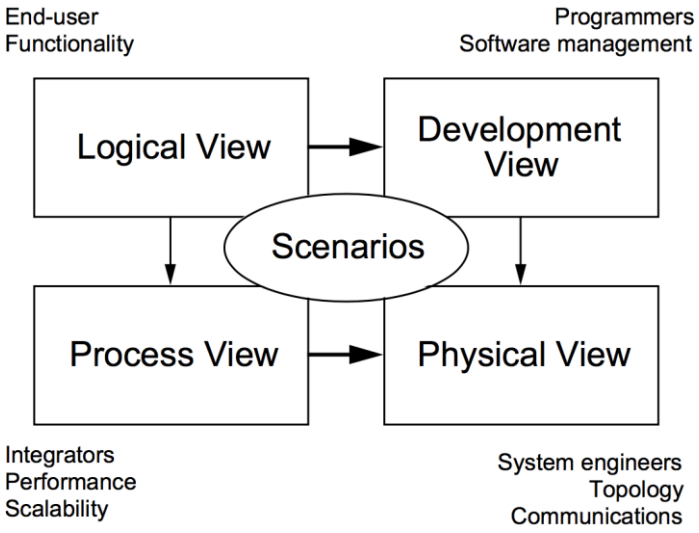


Figure 61 "4+1" View Model (Kruchten 1995)

The Unified Modeling Language (UML) is chosen to describe the architecture views. UML has quickly been adopted as the standard modeling language for modeling software systems (Eriksson and Penker 2000). UML is a general-purpose modeling language used in the field of software engineering, which provides a standard way to visualize the design of a system (UML® n.d.).

9.1 Physical View

The Physical View describes the mapping(s) of the software onto the hardware and reflects its distributed aspect (Kruchten 1995). This part provides an overview of the physical component structure, and how physical components are connected and how physical components communicate. This view is intended for system engineers and administrators.

As shown in Figure 62, the server is deployed on a cloud service, where Quiz Class server, QuizCloud, and Quiz Class Database are running separately. The Quiz Class server need services from the Quiz Class database, and QuizCloud. All end-users need the internet connection to be able to use the services from Quiz Class. All end-users can use all types of digital devices, teachers may also want to show the game result on a large screen, to analysis or summarize the game/homework with students.

Quiz Class server will be deployed and run in a Tomcat (Tomcat® n.d.) HTTP web server environment. A database system needs to be installed for storing information about users, and their homework situations. The web front-end component can be deployed independently

from the Quiz Class server, without communicating with the server the front-end component is just static web app contains HTML, JavaScript, and CSS.

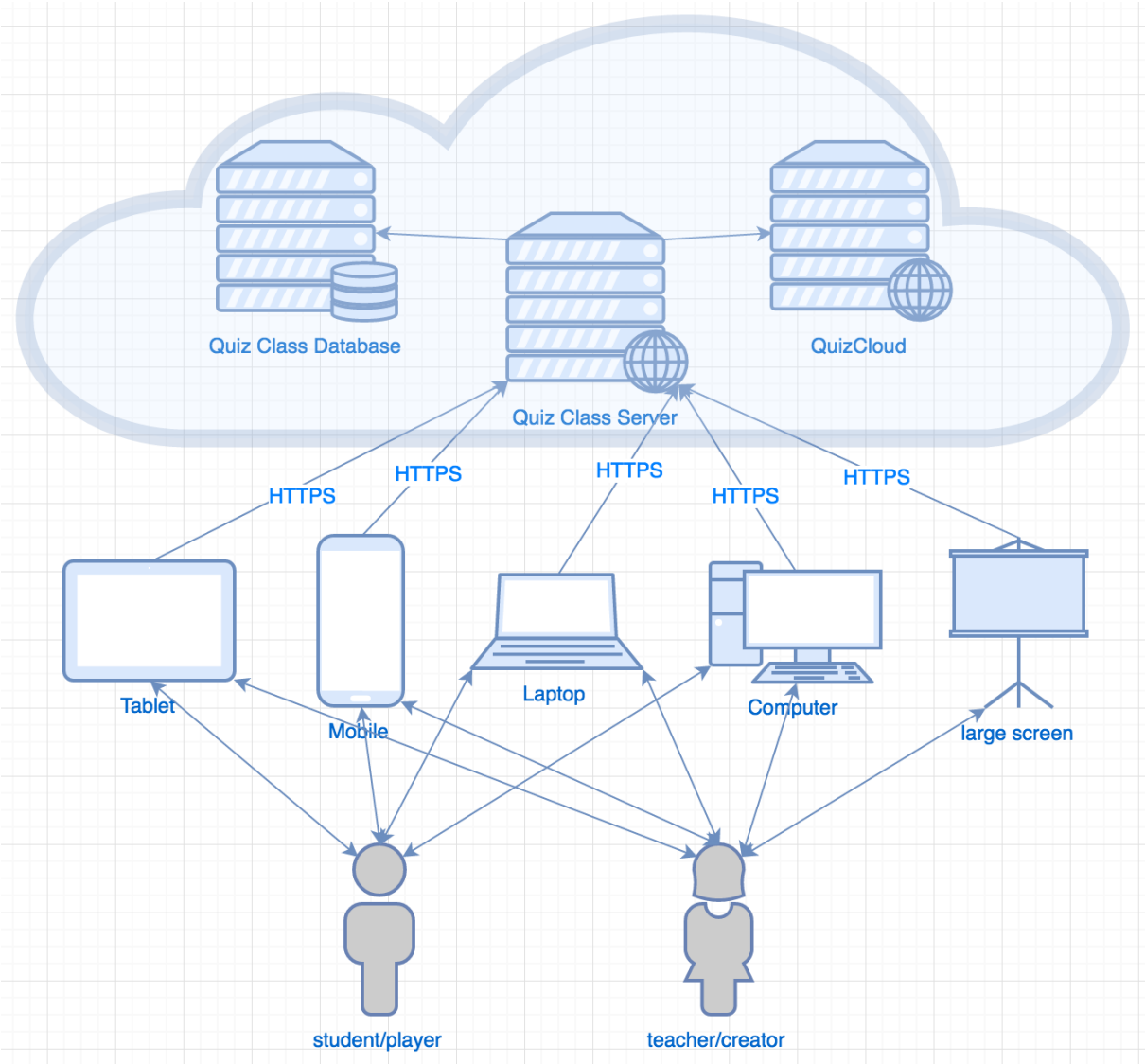


Figure 62 Physical view

Figure 63 illustrates the technologies involved in the front-end and the back-end communication. The back-end application (build with Spring framework) is a provider of RESTful Web Services, serving various front-end applications like a front-end web application (build with the Angular framework), smartphones, and tablets. The Angular browser-based application can also be opened in smartphones and tablets. To communicate

between front-end and back-end, all data is published in the form of JSON[24] or JavaScript files.

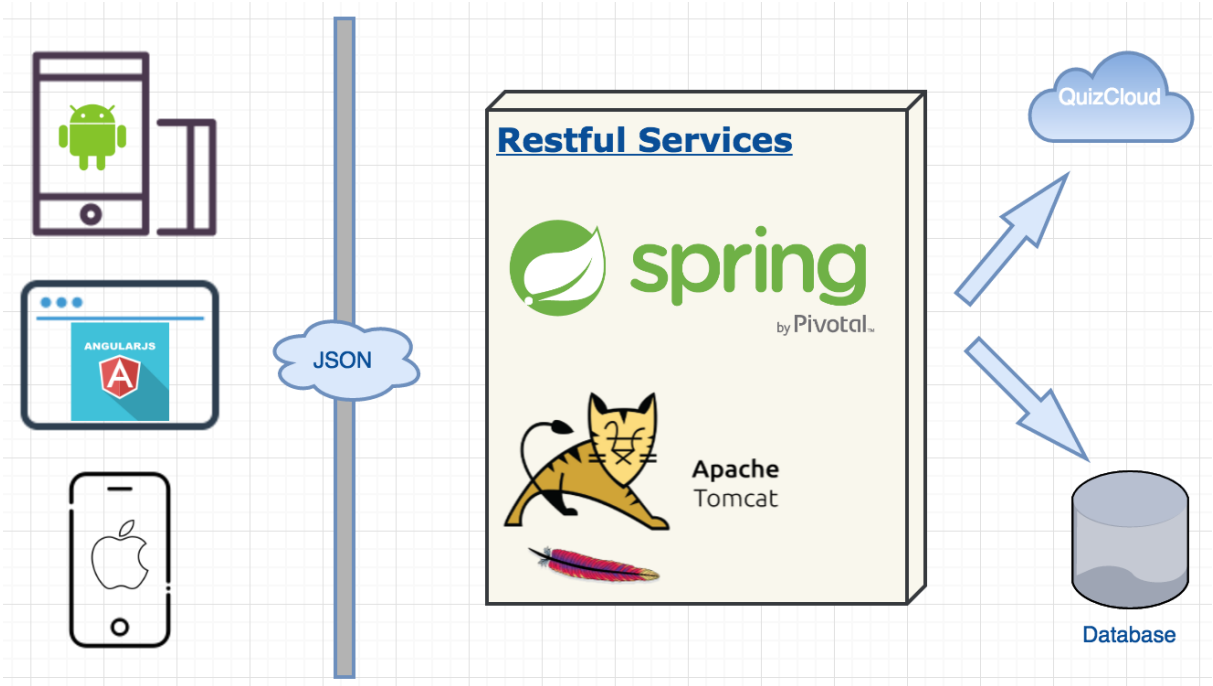


Figure 63 Physical View: technologies

The teachers or game creators are expected to create a game, fill the content beforehand by using any kind of digital devices which have internet connections. Student or game players can only access a game if the game was shared by the create or teacher.

9.2 Development View

The development view is a subsystem decomposition, which describes the static organization of the software in its development environment (Kruchten 1995). This part illustrates the system from a programmer's perspective and is concerned with software management, represent the architecture insight on the actual software module organization on the software development environment. Figure 64 illustrates the development view of both front-end and back-end of the application system.

[24] JSON (sometimes JavaScript Object Notation) is an open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is a language-independent data format, the most common data format used for asynchronous browser/server communication, largely replacing XML which is used by AJAX (retrieved 15.06.2017, from http://www.json.org/).

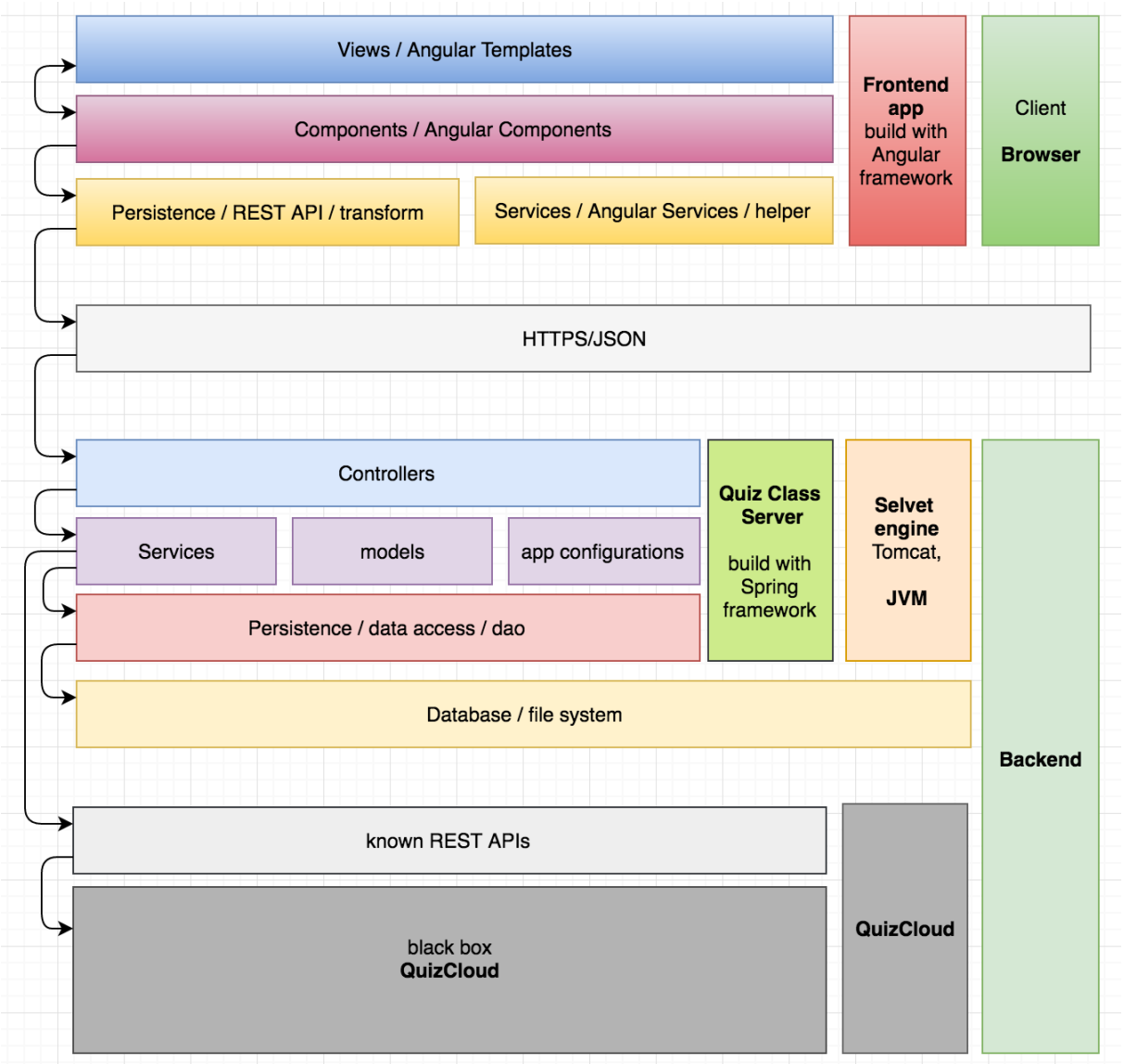


Figure 64 Development view

Front-end uses Typescript programming language (TypeScript n.d.) to generate JavaScript, Angular Framework (Angular n.d.) to manage and organize JavaScript HTML and CSS, Sass stylesheet language is also used to generate dynamic CSS. With Angular's two-way binding, Templates and Components are connected, with services from Services and data from Persistence, the dynamic front-end is formed.

Back-end uses Java Programming Language (JavaLang n.d.), and spring framework. The 'controllers' is the only layer that front-end can talk with, if the services being requested by client can be provided, the controller will request the 'services' to provide service, 'services' perform the logic and request the 'persistence' layer, which can communicate with database or file system, to get, update, save, or delete data, the 'services' together with 'modes' create

or return data to ‘controller’, ‘services’ also need to communicate with QuizCloud to handle quizzes.

9.3 Process View

Process View captures the concurrency and synchronization aspects of the design (Kruchten 1995). This part explains the system activities, processes, workflow rules and how each part communicates, with the focus on the runtime behavior of the system.

The login process

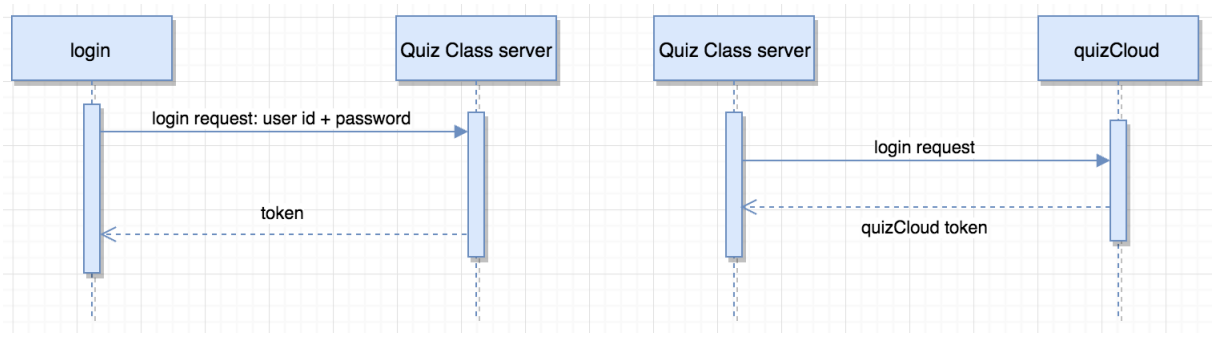


Figure 65 Sequence Diagram: login process

The Quiz Class system has three main parts: front-end (client UI), Quiz Class server, and quizCloud server, those three-part was designed and implemented independently, but they need to communicate with each other to make a functional system.

When an end-user opens Quiz Class front-end UI, the end-user need to login to the system, the process is showing as Sequence Diagram in Figure 65. The end-user uses user id and user password to login, and the Quiz Class server will generate a unique and secret token, which will be saved in user’s browser, for later communication with Quiz Class server. The right side of Figure 65 shows how Quiz Class server get a token from quizCloud, as Quiz Class need services from quizCloud (create, update, delete, and get a quiz), the end-user or the front-end app do not need to know this process.

Figure 66 shows the sequence of creating a quiz arranged in time sequence. When a quiz is made at a client side, it will be formed to a request and being sent to Quiz Class server with (the end-user has to login first) the end user’s token. When Quiz Class server received the request, it will save the relevant data, and get the token for quizCloud, then send formulated quiz to quizCloud with the token, then quizCloud will organize and save the quiz.

Create Quiz process

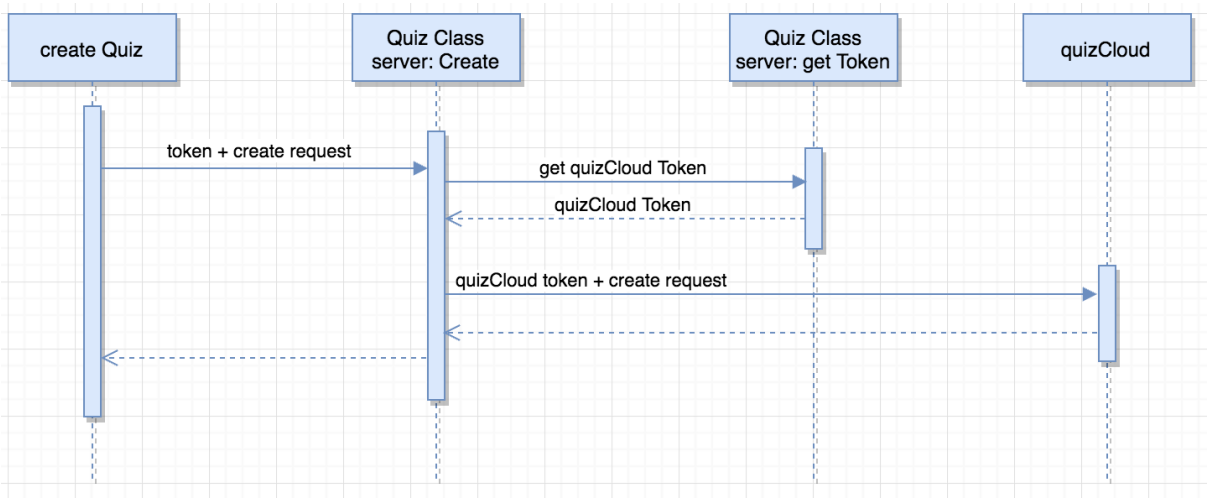


Figure 66 Sequence Diagram: create quiz

Gaming / answer quiz process

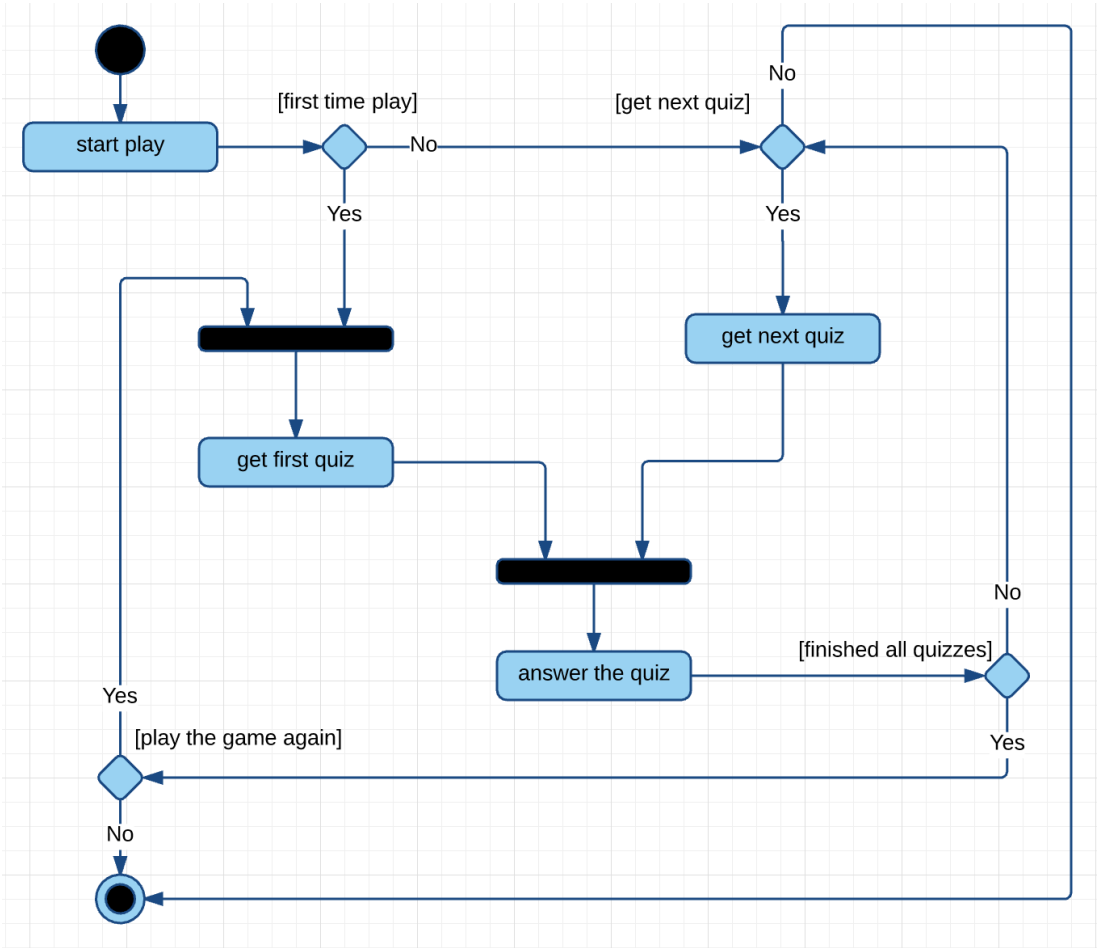


Figure 67 Activity Diagram: doing homework

The activity diagram represents the game workflow of stepwise activities and actions. When a player is logged in and start the game, the system will first checks if the player is first time playing the opened game. For a player who is first time play the game (or finished the game and open the game again repeat), the system will let the player get his/her first quiz. For a player who is returned to a played game, the system will let the player decide to get next quiz (continue to play), or quit the game. If a player decided to get next quiz and answer it, he or she can repeat this process until all questions in the game are answered. When all quizzes in a game are answered, a player can play the game again or quit the game. Figure 67 illustrates how the process is done.

9.4 Logic View

The logical architecture primarily supports the functional requirements: what the system should provide in terms of services to its users (Kruchten 1995). This part decomposed the system into a set of key abstractions, in the form of object classes. The class diagram is used to show the set of classes, packages, and their logical relationships.

Figure 68 “Class Diagram: controllers, services and Dao” show the three packages of controllers, services and Dao, the services, and Dao also have sub-package which is the implementation of the interfaces in the package. The class diagram shows the logical relationships between the classes.

Class Diagram: controllers, services and Dao

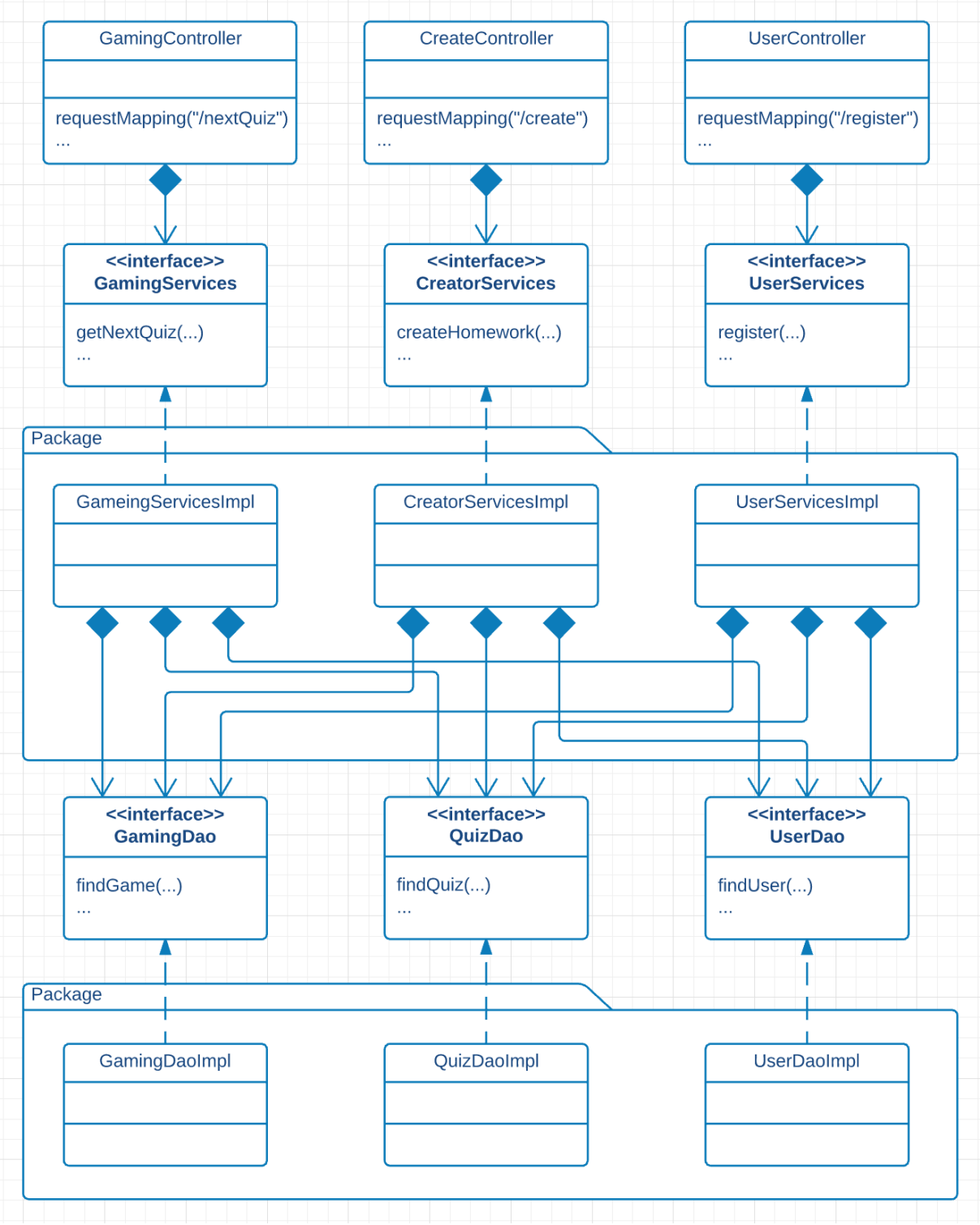


Figure 68 Class Diagram: controllers, services and Dao

10 Implementation

Quiz Class was developed by using the technologies studied in Chapter 5, fulfill the requirements defined by Chapter 7, following the architecture designed from Chapter 9. This Chapter represents the implementation of Quiz Class application, where all functional requirements were successfully implemented.

10.1 Front-end

The front-end source code was written in TypeScript, Sass, and HTML, and it was compiled to pure HTML, CSS, and JavaScript for deployment, which can be supported by all browsers.

The package manager NPM (npm n.d.) was used to manage front-end libraries and dependencies. The front-end source code was organized into 8 main packages:

1. **commons**: the common components like navigation, alert, popup, pipes etc.
2. **components**: the main UI components like gameplay, player UI etc.
3. **models**: all the defined model classes.
4. **services**: all the services like talk to the server, authentication service etc.
5. **layouts**: the public layout and the secured layout.
6. **pages**: where all the public components like register, login, pape_404 are her.
7. **theme**: mainly styles like Sass files.
8. **assets**: this package has all the files that do not need to compile, like icons, languages files.

Angular CLI[25] was used to maintain the front-end application, and to compile and organize source code to pure HTML, CSS, and JavaScript. Angular CLI also performing the Tree Shaking, where any code that was not actually using in Quiz Class will be removed, this reduced the size of the deployed code (many libraries were very big, but not all functions were used in Quiz Class, Tree Shaking them is necessary).

[25] Angular CLI is a tool to initialize, develop, scaffold and maintain Angular applications (retrieved 15.06.2017, from https://cli.angular.io/).

10.2 Back-end

The server side provides restful services. As introduced in the Chosen Technologies part, Java programming language was used, Spring and Hibernate ORM frameworks were used, MySQL database and Apache Tomcat servlet container were used.

The build manager and automation tool Maven (Maven n.d.) was used to managing the server side project's build. If Maven is installed the server side source code can be edited or developed in any editor.

The backend was organized into 7 packages: configuration, security, controller, service, dao, model, and utilities. The configuration package contains all configurations for the application, no XML was used, configurations were done in Java code, which is easier to read and modify than XML configurations. The security package handles the authorization, authentication, and other security issues. The controller, dao, and service packages work as layers, the incoming request only talks with controller layer, and the controller can communicate with service layer interface, service layer may speak with dao layer interface. The interface was used to define the functions needed to be implemented, which makes the modification of one layer do not influence other layers, e.g. if MySQL latter need to be changed to no-SQL like MongoDB, only the implementation 'UserDaoImpl' need to modify.

10.3 Client & server communication

The front-end and back-end communication are important for Quiz Class. This section introduces the Password Reset function as an example of front-end and back-end communication.

Password Reset is one of the many functionalities that need client and server interact to complete a process. In addition, the Password Reset needs a third part user-email to complete the process.

As shown in Figure 69, when end-user clicks the Forgot Password button at the front-end login UI, the user will be redirected to Forgot Password page. When the user writes the email as input, it will be first checked at front-end. The user will get an error message if the input is not a valid email address, otherwise, the email address will be sent to the server at 'forgotPassword' function. The server has a Restful API, which means other clients can talk with the server as well, so the server again checks the email is a valid format, and then check

the email has been registered at Quiz Class. The server will send an error message if there is anything wrong. If all correct, the server will generate a password reset token, a password reset link, an email written in HTML, and send them as an email to the user. This simple process ensures that only the email owner can get the password reset link, and this link only works for the email address. The user can simply ignore this email if he/she remembers the password again. The link is active within 24 hours, and a user can apply a new link at any time.

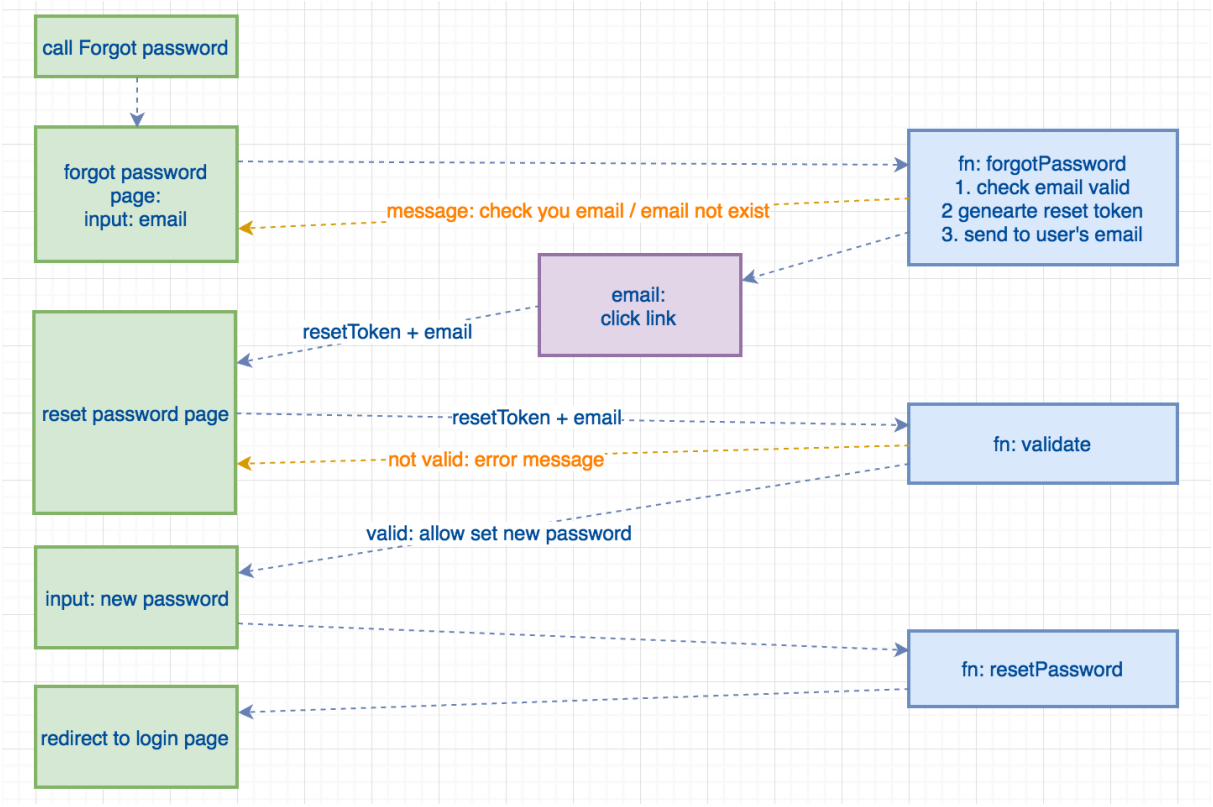


Figure 69 Implementation Password Reset

When the user actives the Password Reset link from his/her email, Quiz Class front-end will be open at Password Reset page. Quiz Class front-end will first check if the link is valid, the user will get a not valid message if the link is not valid. For a valid link, Password Reset token and user email will be extracted from the link and being sent to the server. The token is generated by the server and only the server can decrypt it. The server decrypts the password reset token, check the email is same with the one who is requesting at the client, and check the token is still valid. The server will send an error message if the token is not valid, otherwise, allow client to reset a password if everything is correct. The user can then write his/her new password and request for a password reset. When client gets a succeed password reset

message from the server, the user will be redirected to the login page. The Password Reset process is finished.

10.4 Cloud implementation

Cloud implementation is an important part of Quiz Class. Good cloud implementation practice can maximize Quiz Class accessibility, availability, reliability, capacity, extensibility, scalability, maintainability, modifiability, stability, testability, operability, performance, and security.

The Quiz Class is hosted at Amazon Web Services (AWS), using 3 main cloud services. For Computing, Quiz Class server is deployed at AWS Elastic Beanstalk (AWS n.d.), which can automatically initiate an Amazon EC2[26] and runs in it. Quiz Class Database uses Amazon Relational Database Service (RDS), which is easy to deploy, cost-efficient, and resizable. Quiz Class front-end is hosted at Amazon Simple Storage Service (Amazon S3), which is designed to deliver high durability, and massively scalable objects storage (AWS n.d.). In addition, Amazon CloudFront-Content Delivery Network(CDN) is used to use my own domain name and SSL certificate to deliver content over HTTPS, it also accelerates delivery of Quiz Class.

[26] Amazon Elastic Compute Cloud (EC2) provides secure, scalable computing capacity in the AWS cloud, allowing users to run virtual computers on which to run their own computer applications (retrieved 15.06.2017, from http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html).

Part IV: Experiment

Part IV presents the experiment of the preliminary test, the final test, and the testing results.

11 Preliminary test

Quiz Class needed to be tested in the real world with real users, the preliminary test functions as testing but also an experiment of how Quiz Class works. The major goals of preliminary test include: to test the prototype; find bugs, errors, or defects that was not discovered yet; to test if all functions work for all user groups and environments (there are lots of different devices, browsers, and operating systems out there); to test if the UI works as planned (users may use it differently than the app was designed to); to examine if the users are able to carry out a task successfully; to find the situations where the users get confused, hesitate, or don't know how to proceed; to find the situations where the users tend to make the most errors; to test if it is fun for the users to play; to test how the ranking and confidence level system works; to get feedbacks, suggestions, ideas, and make sure the Quiz Class is ready for the final test.

11.1 Participants and process

In the preliminary test, I knew almost all the participants, and testing was usually taking place at the University. There were 28 valid participants, they participated either as a group or participated alone. This section introduces all the participants.

The first test has three participants, a Ph.D. candidate, and two 5th year MSc. Students. Everyone made a game, I observed one who was creating a game, the other two who had created games before we met. I made a game for the test too. We four played each other's game one by one, so the creator could watch and receive feedbacks about the game he or she created.

The second group testing has 3 participants: two third year informatics students and one second year cybernetics student. I did not participate this test. Participants firstly played a game I created, then they created their own games, and they played each other's game afterward.

The third group testing is done by two participants, a first-year cybernetics student, and a first-year medicine student. One of the players repeated the game two times to achieve a better score.

The fourth group testing was done by 5 participants: one fifth-year Biotechnology student, one fourth-year Applied Physics and Mathematics student, one fourth-year MSc in Communication Technology, one fourth-year Industrial Economics and Technology Management student, and one third-year Clinical Programme in Psychology student. This group only played the game, while they did not build their own game.

The fifth group testing was done by 4 participants, one chemistry Ph.D. candidate, one Energy and Process Engineering post-Ph.D. candidate, and two others who tested already in the first test group. They all played two games, one created by me, and the other one created by a player from the individual test group, and one of them repeated a game six times to achieve better scores.

There was another group testing taking place in China, which I cannot directly observe. However, I "observed" through an online call, where I heard they were excited while playing. Both participants from this group are retired.

Individual participants include: 2 software developers, one of them build a game, the other one only played a game (they played the game at different time and different location); one fifth-year MSc in Communication Technology student, who only played a game; one third-year Civil and Environmental Engineering student; one retired from statistics; one high school teacher; one fifth-year Computer Science student, who played all the opening games in Quiz Class, she repeated some of the games for top scores, and built a game as well.

11.2 Observations and informal interviews

The participant observation in preliminary test helped a lot for improving Quiz Class user experience (UX), user interface (UI), and many bugs were found during this process. It also confirmed that Quiz Class is easy to use and fun to play.

In one of the tests, I found that the animation was not working as expected, but the user had no idea about how it should look like. It is the advantage of observations, that there were some issues which were difficult for users to know or tell. I told the user about the animation issue, but he was so engaged with answering questions while did not even notices the animation. This was partly confirmed that the player was engaged while playing.

The password reset feature was planned to be a low priority because I thought it would not be critical. But in the first group test, one of the users forgot his password, and after many

attempts, he still could not log in. With the password reset feature, he finally reset his password and logged in. As the prototype was targeted on test, experiment, and investigation, the password validation rule was very weak (valid if input password is more than 5 characters long). I thought this would make users set an easy to remember the password, but it turned out the password reset feature is more important than I thought, as it was revealed in this forget-password case.

In the first group test, there were two users who used an iPhone 5, and an iPhone 7 respectively. When a question has too many options, the user need to scroll down the screen to see all the options. We found that it was difficult for iPhone to scroll down to the bottom of the screen. They could not choose the highest or lowest confidence level, and the slide function for confidence level wasn't working either. On iPhone 5 it was also difficult to change screen orientation, while PC, Mac, and Android browsers, did not have those problems. Apart from those issues that were fixed afterward, the participants thought that the UI was easy to use, easy to understand, work with, and they like the idea.

In the second group test where everyone only used mobile devices, a lot of positive suggestions, and potentials were provided by participants. The most critical issue found from this test was that calendar popup did not work well when it was triggered from parent popup (E.g. create Game, set game start or end time). Participants thought that the scoring system was not clear enough, and I told them that there was help information they could read, but they suggested that it would be better if users could see the possible minimum or max points they may get. Those issues were fixed right after the test.

In the third group test, I was suggested to make an auto-scroll down to bottom feature when answering questions. She thought it would be nice if she clicks an option, the app could automatically scroll down, so that she can click submit answer button without manually scroll down to find the button. Like all the other tests, this group also provided lots of good suggestions, ideas, and creative thoughts.

The fourth group test was very smooth, where users were able to quickly register, login, find the game, and play the game. No issues, bugs were found. I was surprised that the participants used Quiz Class like they were familiar with it already.

On the fifth group test, I asked how did they felt about the UI. Participant A responded: "What is UI, I didn't feel any UI, I just paly and answered the questions". Participant B

added: “If you didn’t feel the UI, which means the UI was good”. And later, we moved on to another game, Participant A complained: “This game is so boring!!!”. Participant B answered: “yes, the game was so boring, and now you are on your 6th repetition of the same game”. This group had many good suggestions and ideas, most of them I cannot implement within this project, except one: “the player should be able to see the ranking during play”, and it was implemented.

Both group tests and individual tests have received positive results. The tests tell that the prototype works, and it is ready for investigation in the teacher-student situation. Many bugs, errors, defects were found and fixed. The functionalities work on different devices, browsers, and operating systems (for those were tested). The UI works as expected. The users can easily carry out tasks successfully. The places or situations where some users were confused, hesitated, or tend to make most errors are fixed or improved. Most of the users had fun to play, and some of them thought it was very fun to play. Many good suggestions, ideas, and creative thinking were provided, and most of them are implemented or will be implemented in the future.

11.3 Improved UX

One of the improvement during the preliminary test was the Internationalization and Localization. The Internationalization is the process of developing products in such a way that they can be localized for languages and cultures easily(InternationalizationLocalization n.d.). Which was critical when the tests were spread into China. The Quiz Class has improved that the language will auto fit the users’ browser language (currently only support Chinese and English), and the user can still choose their preferred language when they logged into the app. This improvement makes it possible to test in China.

To optimize the user experiences in China, different technologies and cloud hosting were tested. The server was hosted at AWS Singapore and Frankfurt regions, and AWS China was also tried with server hosted at Beijing region. Heroku (Heroku n.d.) a cloud application platform was also used to host Quiz Class server, and ClearDB (ClearDB n.d.), a geo-distributed data services Platform was used to host MySQL database. The AWS Frankfurt region delivers best results, it has higher accessibility, availability, stability, higher performance, and shorter response time, compared with Heroku and ClearDB solutions.

11.4 Preliminary test results

In the end of the preliminary test, there were 47 registered users, all participants were invited, and the participants were asked not to spread the Quiz Class link to others, as Quiz Class was created for experiments. There were some accounts registered by the same person. Based on my observation and informal interviews, there were 28 valid preliminary test participants.

Group No.	Participants	No. of participants repeated a game	No. of participants created a game	No. of participants finished the game
1	3	1	3	3
2	3	0	3	3
3	2	1	0	2
4	5	0	0	5
5	4	3	0	4
6	2	0	0	1

Table 11 Preliminary test: group participants

There were two types of participants in the preliminary test: join as a group, or participate individually. Table 11 illustrated the group test results. There were 6 groups, the group number is ordered by testing time ascendingly. There were 2 participants joined both group 1 and group 5. “No. of participants repeated a game”: A participant who finished all the questions in a game on a repeating (including repeated a game more than once).

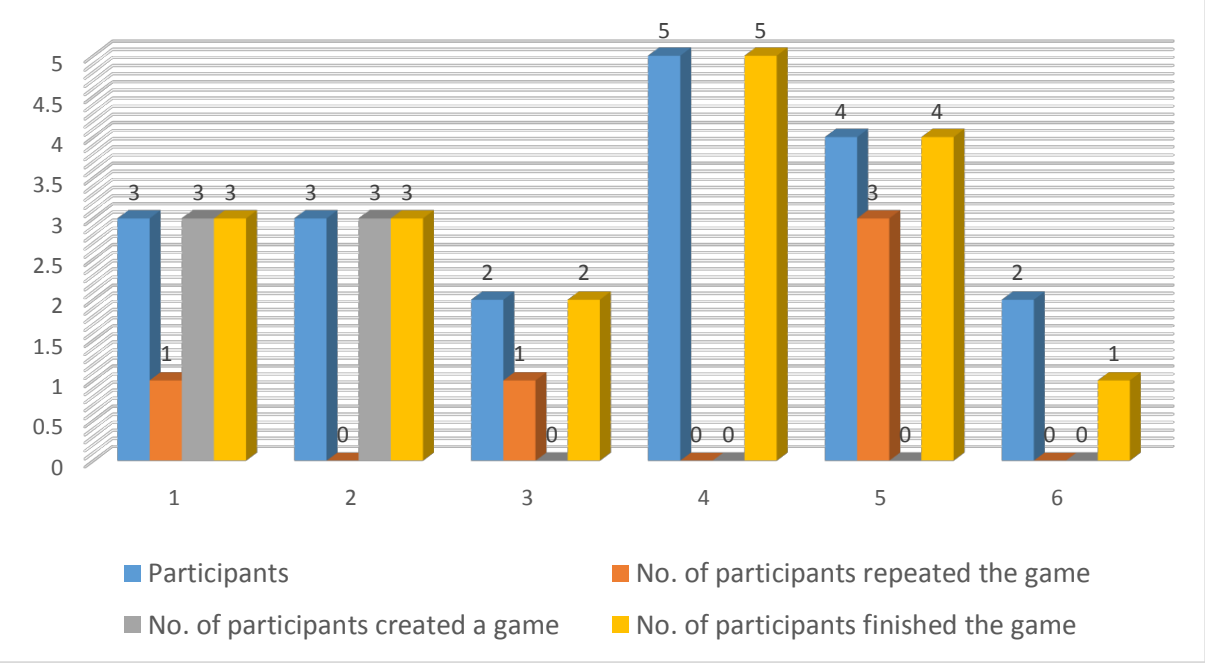


Figure 70 Preliminary test: group participants

Figure 70 converted the data from Table 11 and illustrated it as a chart view, which shows data more intuitively. As the chart shows, there was only one participant from group 6 did not finish the game.

Participant No.	Repeated the game	Created a game	Finished the game
1	0	0	0
2	0	0	0
3	0	0	1
4	0	0	1
5	0	1	1
6	1	0	1
7	0	0	0
8	1	1	1
9	0	1	1
10	1	1	1
11	1	1	1

Table 12 Preliminary test: individual participant

Table 12 shows the 11 valid individual participants, the participant No. is ascendingly ordered by time. In Table 12, '0' represent negative, '1' represent positive. To make it more intuitive, Table 12 is converted to a chart view shown in Figure 71. Participant No. 1, No. 2, and No. 7 did not create any game, did not finished all the questions in the game they played, and they did not repeat any game.

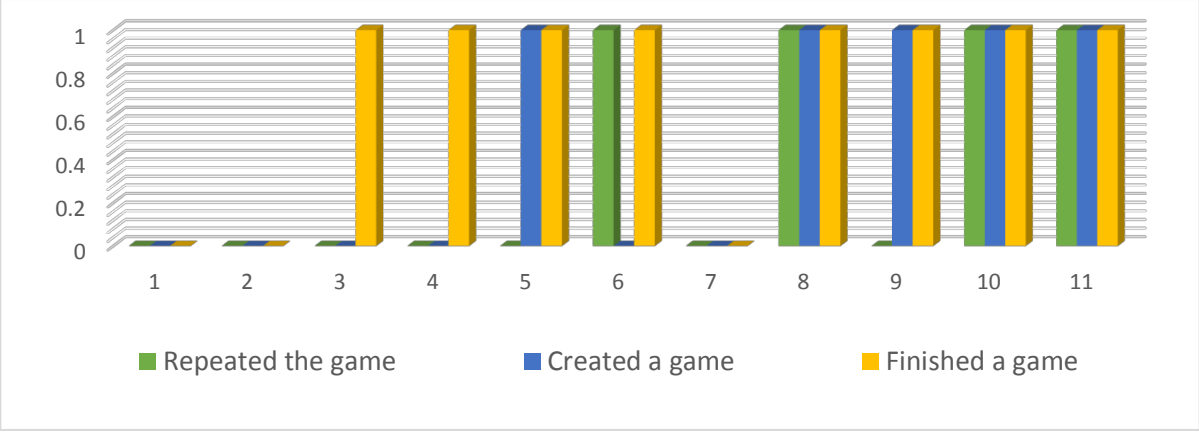


Figure 71 Preliminary test: individual participant

The result from two groups had some differences. To compare the two groups, I made Figure 72. Where the percentage of participant repeated a game, created a game, or finished a game

is showing as two groups in Figure 72. Participant repeated a game or created a game are lower in both sets. But most of the players tend to finish the game whey played.

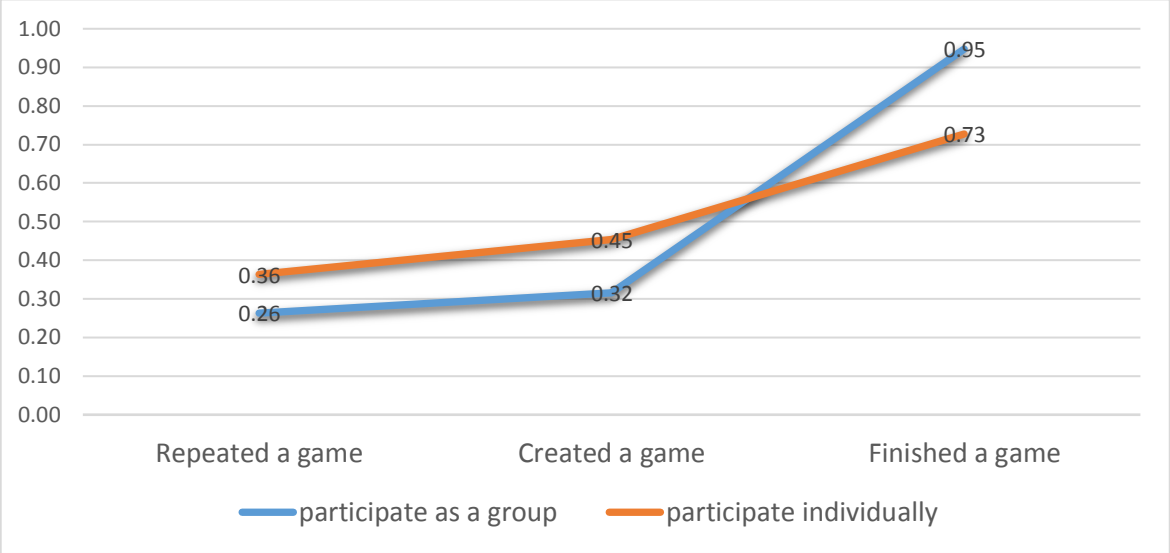


Figure 72 Preliminary test: participant results distribution

Figure 72 illustrated that 95% of group participant finished the game they played, and only 73% of the individual participant finished the game they played. I told all the participants that they may stop playing at any time they want. This may cause an individual player to stop play as there were only two of us, so the participant might feel that I wanted to talk with them instead. There was one participant who stopped when I told her that she can stop the test at any time (I usually told twice, before we start and during the test). But she finished the game after the observation and informal interview.

The observations indicated that the participants joined as a group behaved more exciting, and their emotions changed dramatically. One of the participants repeated a game 6 times to compete against her group. Where the individual participants did not look so exciting, and their emotion changes were not so obvious, compare with group participants.

12 Final test

This chapter presents the final test of Quiz Class, where the participants, and how the test was done will be presented. The final test is the experiment which investigates and discovers how Quiz Class influence students' study, teachers' teaching. The questionnaire was made from the final test to collect data.

12.1 Participants

The final test took place at NTNU, where all player participants were students taking TDT4240 Software Architecture course. My supervisor participated as a teacher, where he created one game collection (the Software Architecture course) called "TDT4240", and three games (three chapters). There were 116 registered users who joined "TDT4240" game collection. Figure 73 is a photo of participants playing Quiz Class during the lecture, where everyone was engaged.

Figure 73 Quiz Class participants: playing game

12.2 Testing environments

The testing was done in two testing environments: during the lecture, and outside the lecture. The teacher first created a game collection, then created three games in the game collection.

The first game is called “Chapter 1: What is Software Architecture?”, with start time 20.04.2017 19:58, and finish time 05.05.2017 19:58. This game contains 13 multiple choice questions, was played by 48 players, and this game got 4.4 rating point, rated by 7 players.

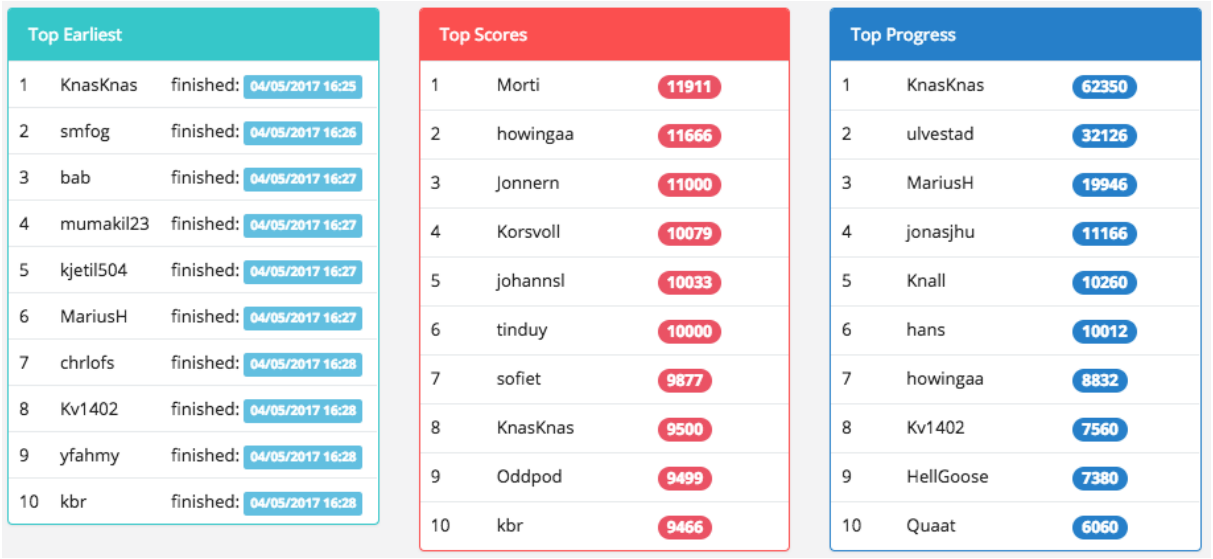
The second game is called “Chapter 2 - Why is Software Architecture Important?”, with start time 20.04.2017 19:57, and finish time 05.05.2017 19:55. The second game contains 3 multiple choice questions, was played by 34 players and this game got 4.3 rating point, rated by 10 players.

The game was shared through It’s Learning, it was announced as a way to test students’ knowledge which was relevant to their upcoming exam, and it was optional for students to do.

Figure 74 Quiz Class participants: game analysis

The third game was created before the last lecture of the semester, and after the two first created game has been played nearly 2 weeks. The third game is called “Chapter 14 - Quality Attribute Modeling and Analysis”, with start time: 04.05.2017 15:45, and Finish time: 18.05.2017 13:01. The third game contains 8 multiple choice questions, played by 96 players, and this game got 4.6 rating point, rated by 7 players.

The third game start time was set half an hour before the lecture started. It was used during the lecture, where students got about 10 minutes[27] to play the game, and the winner will get a prize[28]. Figure 73 shows that players were playing the third game during the lecture. Figure 74 shows the teacher was summarizing the gaming results. Figure 75 shows the competition result, where 5 winners were chosen from the Top Scores. Figure 75 also indicated that many players have made good progress, the top earliest list gives extra information about players' finish time.



Top Earliest

1	KnasKnas	finished: 04/05/2017 16:25
2	smfog	finished: 04/05/2017 16:26
3	bab	finished: 04/05/2017 16:27
4	mumakil23	finished: 04/05/2017 16:27
5	kjetil504	finished: 04/05/2017 16:27
6	MariusH	finished: 04/05/2017 16:27
7	chrlofs	finished: 04/05/2017 16:28
8	Kv1402	finished: 04/05/2017 16:28
9	yfahmy	finished: 04/05/2017 16:28
10	kbr	finished: 04/05/2017 16:28

Top Scores

1	Morti	11911
2	howingaa	11666
3	Jonnern	11000
4	Korsvoll	10079
5	johannsl	10033
6	tinduy	10000
7	sofiet	9877
8	KnasKnas	9500
9	Oddpod	9499
10	kbr	9466

Top Progress

1	KnasKnas	62350
2	ulvestad	32126
3	MariusH	19946
4	jonasjhu	11166
5	Knall	10260
6	hans	10012
7	howingaa	8832
8	Kv1402	7560
9	HellGoose	7380
10	Quaat	6060

Figure 75 Quiz Class Participants competition result

[27] Quiz Class is designed to play after class, but not limited to it. In this event, students got 10 minutes to intensively play the game during the class, the teacher announced the winners at the agreed time.

[28] Because this game can be played before the class started, the early started players will gain advantages of this competition. This is a perfect way to let students learn to start doing homework as soon as possible. The game can still be played until the deadline. This strategy can be used for teachers to adjust the homework, encourage student's motivation, or just to surprise their students.

13 Results

This chapter first presents the accessibility of Quiz Class, then the results of the questionnaire, the assignment repetition and the teacher's perspective was also presents.

The Questionnaire contains 2 mandatory part, one is about the usefulness of Quiz Class, the other one is SUS, and the third part contains only two optional questions, where participants can express their opinions. The results of the questionnaire were divided into two parts: Quiz Class usefulness, and the SUS questionnaire. The questionnaire was made with Google Forms. The contents of questionnaire were attached in Appendix Quiz Class Questionnaire.

13.1 Accessibility

One of the goals of this project is to make Quiz Class easy to access by all users, this section will evaluate the accessibility of Quiz Class. To analyse the accessibility, Google Analytics[29] was implemented in Quiz Class. The data analysed below was based on the 173 registered users at Quiz Class. Figure 76 shows the operating systems (OS) have been used to access Quiz Class, which were tracked by Google Analytics. The major OS includes Macintosh, Windows, and Linux, could access Quiz Class; major mobile OS, Android, and iOS could access Quiz class.

[29] Google Analytics is a freemium web analytics service offered by Google that tracks and reports website traffic (https://analytics.google.com).

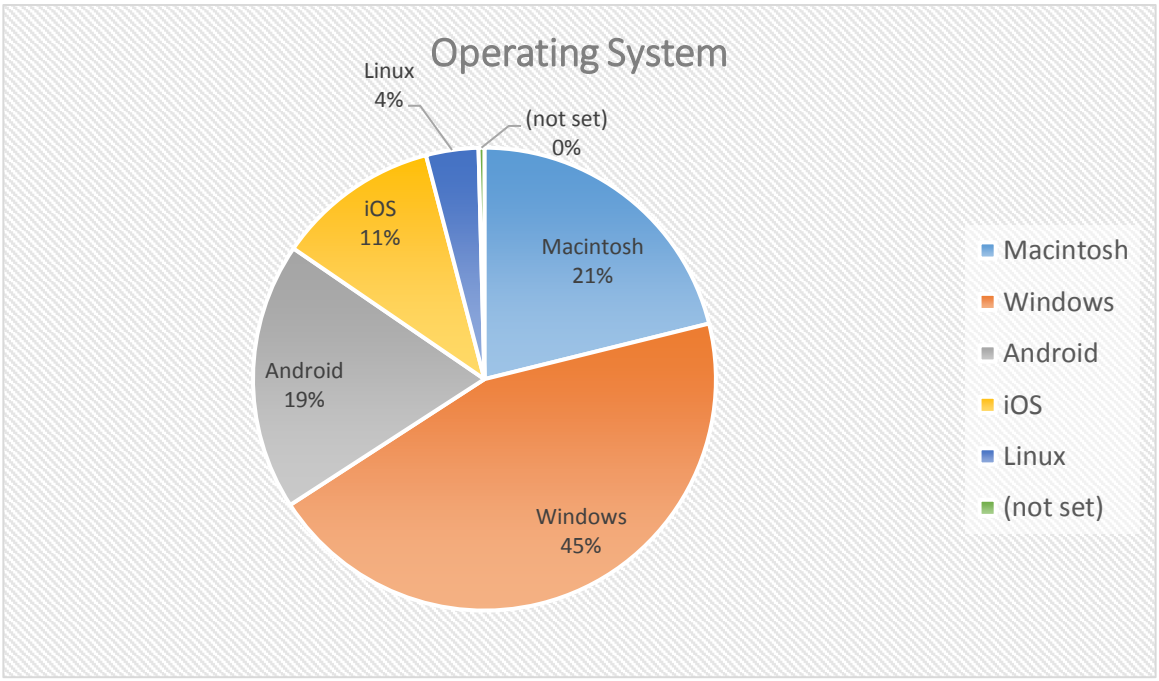


Figure 76 Accessibility: Operating Systems' distribution

Figure 77 shows different browsers have been used to access Quiz Class. Chrome was the most used browser to access Quiz Class, other major browsers include Safari, Firefox, Edge, Internet Explorer, and Opera were also used to access Quiz Class.

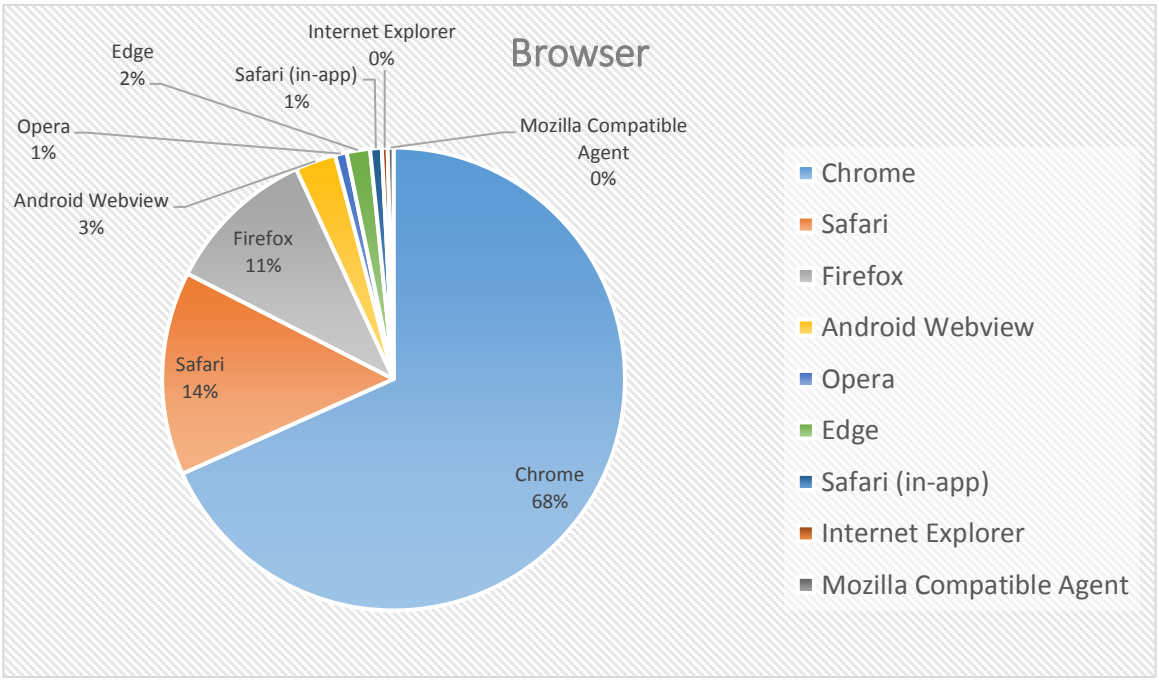


Figure 77 Accessibility: Web Browsers' distribution

Figure 78 shows the mobile brand have been used to access Quiz Class. Which include Xiaomi, Apple, Huawei, Samsung, Sony, Google, Motorola etc. brands.

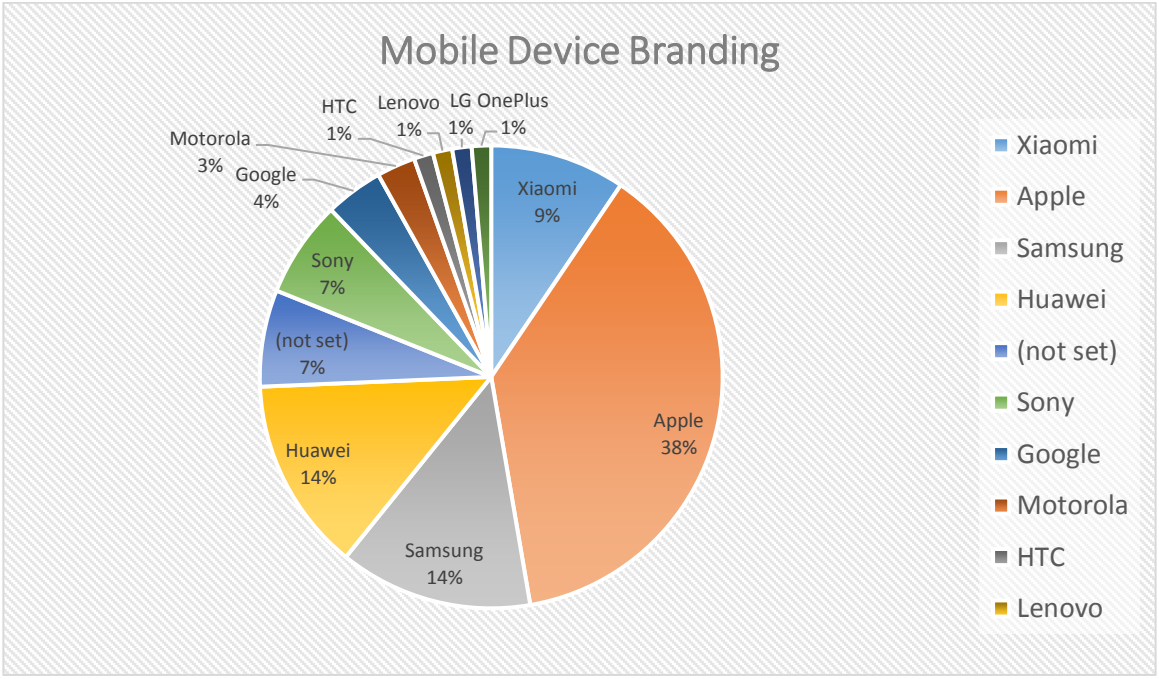


Figure 78 Accessibility: Mobile Devices distribution

Figure 79 shows the device category, where the desktop was mostly used to access Quiz Class, few users have been using the tablet to access Quiz Class.

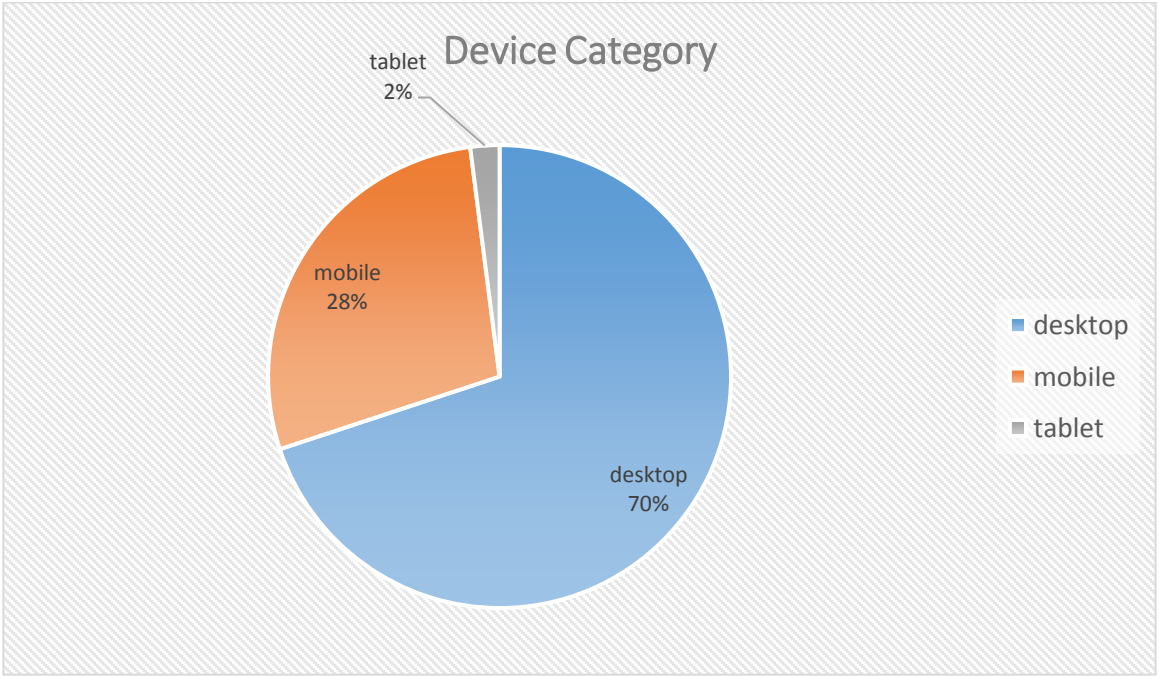


Figure 79 Accessibility: Device Category distribution

The screen resolution may influence user experience when using Quiz Class, and to improve UI, UX, the screen resolutions need to be considered. The Quiz Class UI was designed to fit all kinds of screens, Figure 80 shows the screen resolution distribution of Quiz Class user.

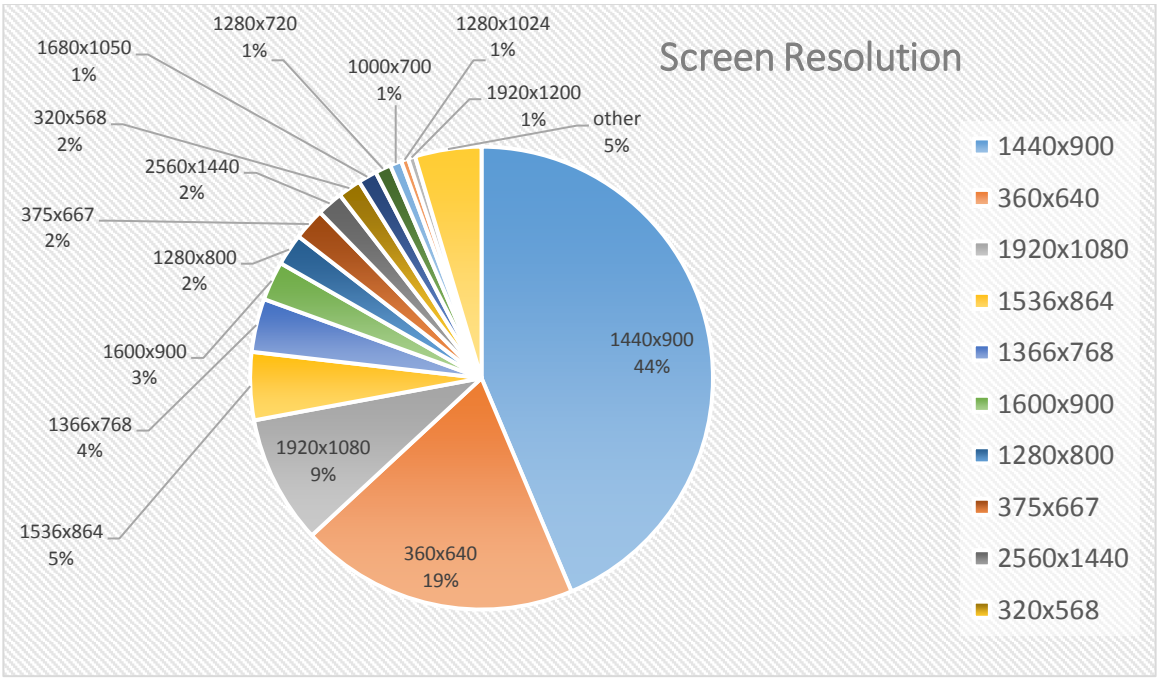


Figure 80 Accessibility: Screen resolution distribution

Quiz Class has been successfully accessed by users with different OS, browsers, device categories, device brands, and its UI has been running in various screen resolutions. According to the observation, and user feedbacks, no user is unable to access Quiz Class.

13.2 Demographics

There were 81 total responses, some responses were not answered seriously, like wrote their age to 0, or wrote comments on the optional fields with meaningless characters like “123”, “aa bb” etc. In the end, there were 78 valid responses, where everyone answered all mandatory questions. There were 44 answered the first optional question, and 35 answered the second optional question.

The first question of the questionnaire was the participants’ age, this is important to know, as it might influence the test result, and it is also important to know the age distribution of the respondents. Figure 81 illustrated that the two largest age group are 22, and 23. Most of the participants are between 21 to 24, and 97% of the participants are between 21 to 30.

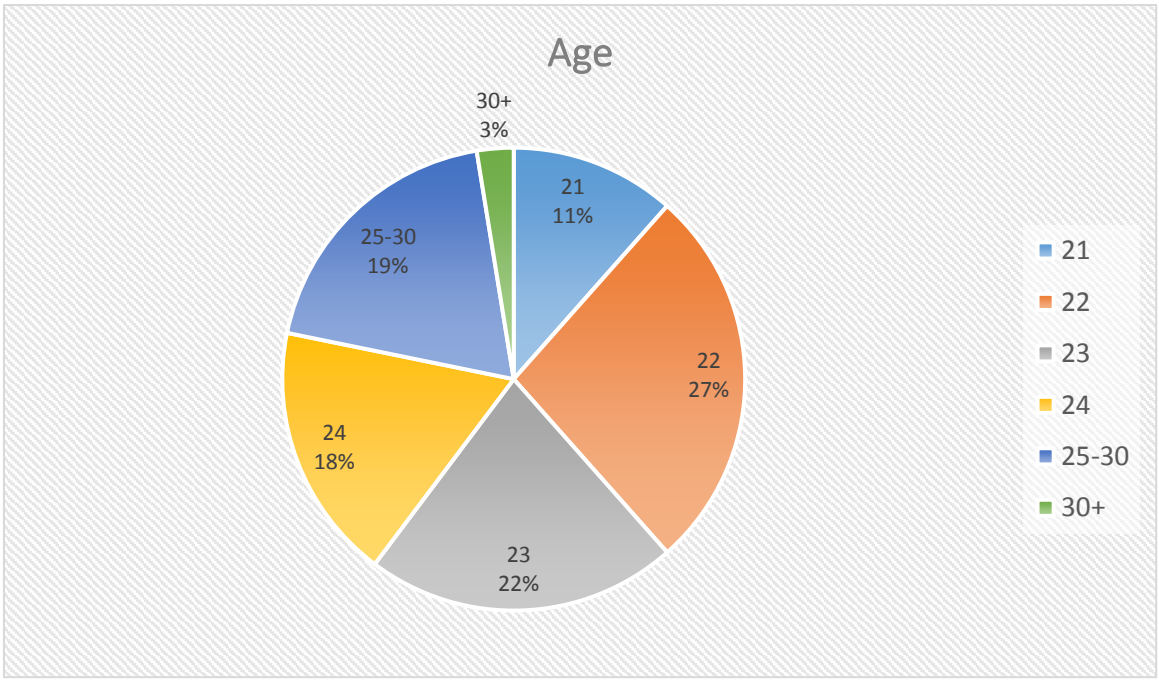


Figure 81 Age distribution of the questionnaire

The second question of the questionnaire was participants' gender. Figure 82 shows the gender distribution of the questionnaire respondents. Where the males are the dominated group.

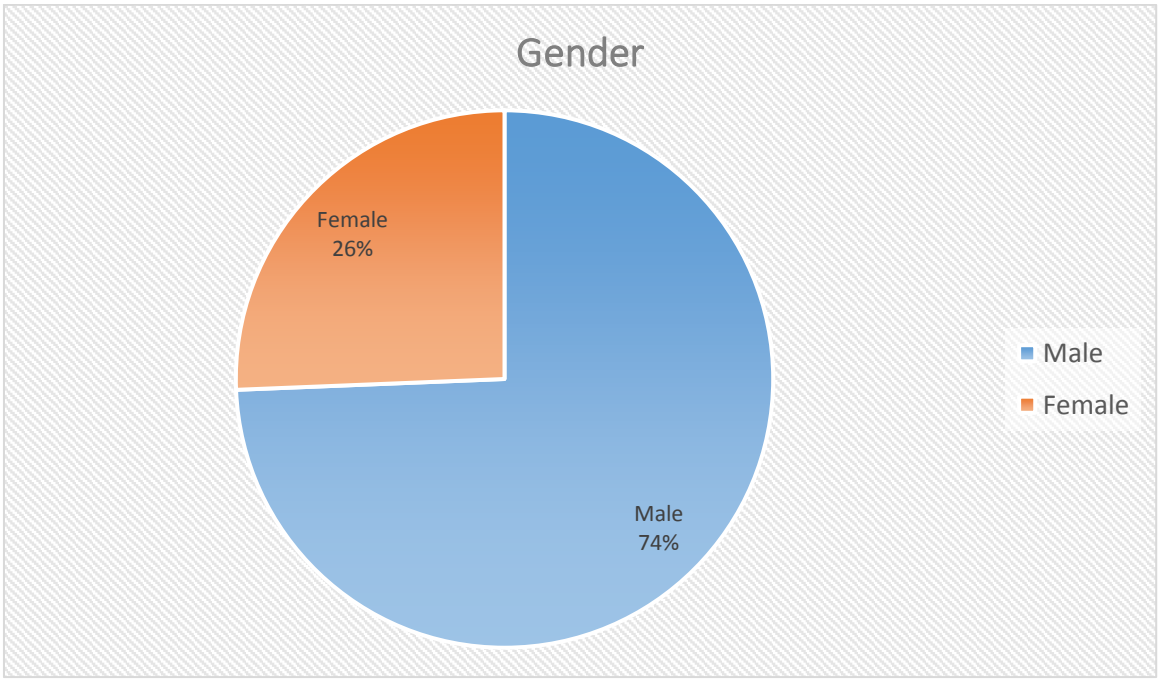


Figure 82 Gender distribution of the questionnaire

13.3 SUS score

The second part of the questionnaire contains the System Usability Scale (SUS). The SUS is a simple, ten-item attitude Likert scale giving a global view of subjective assessments of usability. It was developed by John Brooke (Brooke 1996). SUS yields a single score on a scale from 0 to 100 points. Each question has a scale position from 1 to 5. For items 1, 3, 5, 7 and 9, the score contribution is given by subtracting 1 from the scale position. For item 2, 4, 6, 8 and 10, the contribution is 5 minus the scale position. This implies that each question has a SUS contribution of 0 - 4 points. Finally, the sum of the scores is multiplied by 2, 5 and divided by the number of replies to obtain the SUS score (Brooke 1996).

#	**Question**	**Avg.**	**Score**
1	I think that I would like to use this system frequently	3.24	5.61
2	I found the system unnecessarily complex	2.28	6.83
3	I thought the system was easy to use	4.00	7.50
4	I think that I would need support of a technical person to be able to use this system	1.33	9.20
5	I found the various functions in this system were well integrated	3.49	6.22
6	I thought there was too much inconsistency in this system	1.92	7.72
7	I would imagine that most people would learn to use this system very quickly	4.00	7.50
8	I found the system very cumbersome to use	1.97	7.60
9	I felt very confident using the system	3.81	7.02
10	I needed to learn a lot of things before I could get going with this system	1.50	8.78
	Total SUS Score		**73.97**

Table 13 Questionnaire result: Quiz Class SUS score

The results from 78 valid response of the SUS was calculated and is shown in Table 13. The "Avg." is the average points from 1 to 5, where strongly disagree is 1 point, strongly agree is 5 points. The average point shows the reader what is the average answer of each question. The "Score" shows each SUS score. Quiz Class scored 73.97 out of 100 points, which indicates that the system is average easy to use.

In the optional questions, many participants wrote that they like the UI, it was “Easy to use”, has “Neat design”, “simplicity and intuitive”, “pretty” and “fun” colors, “It’s simple design, because that made it really easy to use”.

13.4 The usefulness of Quiz Class

The usefulness of Quiz Class was measured by 12 questions, which was the first part of the questionnaire. The whole questionnaire is attached in Appendix: Quiz Class Questionnaire. All the questions had the same style as the System Usability Scale (SUS). Scores are ranked from 1 to 5, where 1 represents “strongly disagree”, 5 represent “strongly agree”, and the respondents will find a number from 1 to 5 that can best describe themselves.

To make the results more intuitive and easy to analyze, I have combined “1: strongly disagree” and “2: disagree” to “disagree”, “3: neutral” as “neutral”, combined “5: strongly agree” and “4: agree” to “agree”, as shown in Table 14. In addition, male, female, and total were presented as well, where male indicate the percentage of male respondents who were “disagree”, “neutral”, or “agree”, the female represents the female respondents and the total sums male and female respondents. For detailed data, Table 16 (in Appendix: Tables) provides the percentage of the respondents who selected each number (between 1: “strongly disagree” to 5: “strongly agree”) and the average position (between 1 to 5) on each question.

Table 14 shows that 66.7% of participants found the app make doing assignment more fun. Only question 2, 4, 5, 8 have less than 50% participants agree, rest of the questions have more than half of participants agree on. Question 2, 3, 4 did not reach half, but still have more participants agree than disagree. Question 8: “The top early list encourages me to start doing assignment early” has more participants disagree than agree, which indicated that top early list did not work well in the final test.

#	Question		Disagree	Neutral	Agree
1	It was fun to compete against others	Male	17.2%	24.1%	58.6%
		Female	15.0%	15.0%	70.0%
		Total	16.7%	21.8%	61.5%
2	I pay more attention on doing the assignment because of the app	Male	10.3%	48.3%	41.4%
		Female	30.0%	55.0%	15.0%
		Total	15.4%	50.0%	34.6%
3	I was engaged while playing	Male	13.8%	20.7%	65.5%
		Female	20.0%	30.0%	50.0%
		Total	15.4%	23.1%	61.5%
4	I was more positive towards topic after playing the game	Male	13.8%	37.9%	48.3%
		Female	35.0%	30.0%	35.0%
		Total	19.2%	35.9%	44.9%
5	I found the app made me learn more	Male	20.7%	34.5%	44.8%
		Female	40.0%	25.0%	35.0%
		Total	25.6%	32.1%	42.3%
6	I found the app made doing assignment more fun	Male	10.3%	20.7%	69.0%
		Female	15.0%	25.0%	60.0%
		Total	11.5%	21.8%	66.7%
7	I found the confidence level made the game more exciting	Male	15.5%	31.0%	53.4%
		Female	35.0%	10.0%	55.0%
		Total	20.5%	25.6%	53.8%
8	The top early list encourages me to start doing assignment early	Male	29.3%	41.4%	29.3%
		Female	45.0%	50.0%	5.0%
		Total	33.3%	43.6%	23.1%
9	The top score list encourages me to do well in assignment	Male	12.1%	27.6%	60.3%
		Female	15.0%	15.0%	70.0%
		Total	12.8%	24.4%	62.8%
10	The progress list encourages me to repeat the assignment	Male	19.0%	32.8%	48.3%
		Female	30.0%	15.0%	55.0%
		Total	21.8%	28.2%	50.0%
11	The app increases my motivation for learning/doing the assignment	Male	6.9%	34.5%	58.6%
		Female	35.0%	20.0%	45.0%
		Total	14.1%	30.8%	55.1%
12	I wish this app could be used for more of my assignments	Male	8.6%	31.0%	60.3%
		Female	30.0%	20.0%	50.0%
		Total	14.1%	28.2%	57.7%

Table 14 Questionnaire result: usefulness of Quiz Class

To compare the difference between males and females, Figure 83 has summarized the data from Table 14 to a chart view. As shown in Figure 83, each number in x-axis represents the number of the question, the female has red color, male has blue color. The male group is more positive than the female group in most of the questions. It is interesting that the female group extremely agree with question 1 and question 9, but extremely disagree with question 8.

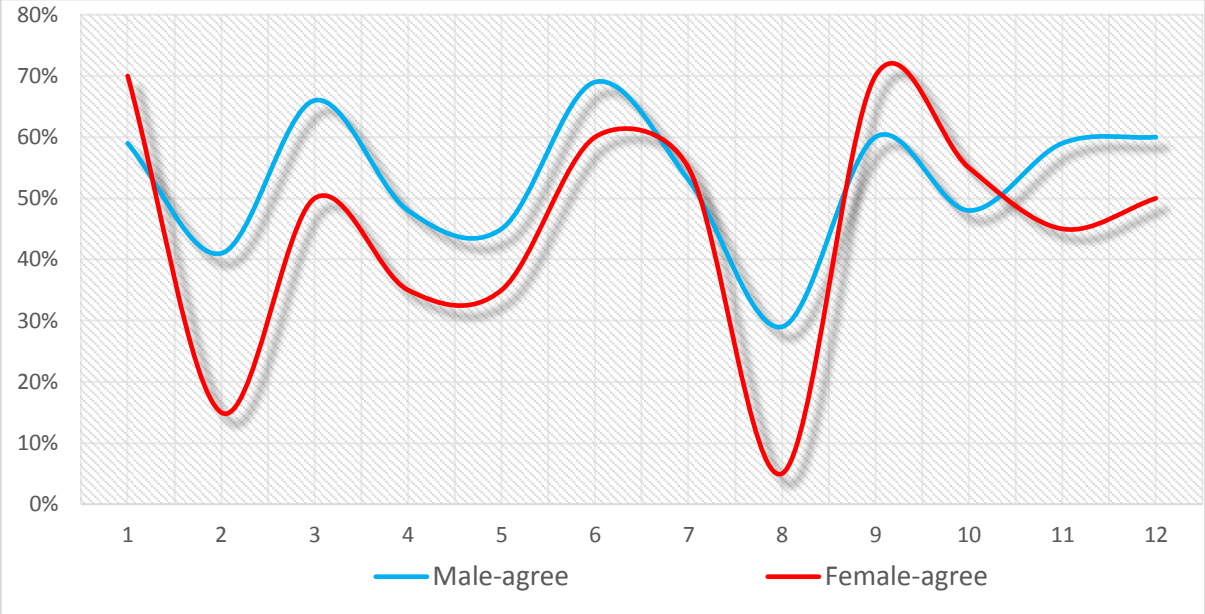


Figure 83 Questionnaire result: usefulness of Quiz Class

13.5 The repetition

It is believed that the repetition is important for learning, and it is also important that everyone should have a chance to improve themselves. This part presents the result of repetition, where many students selves motivated to repeat their assignment, few students even repeated their assignment 9 times.

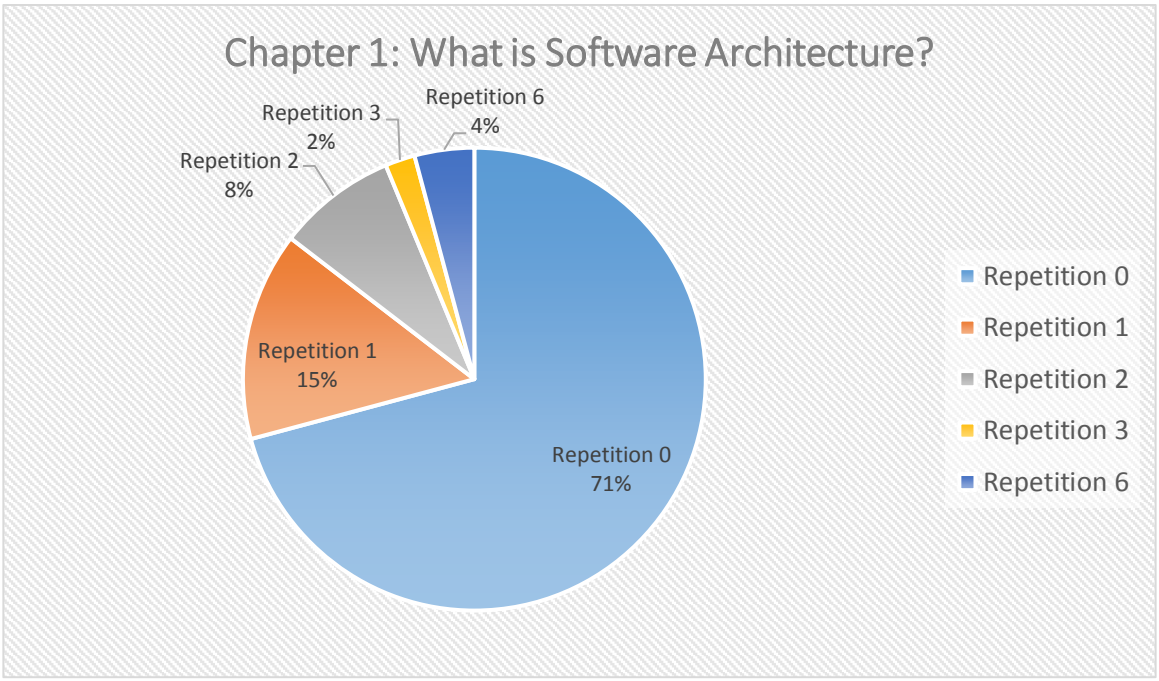


Figure 84 Game 1 repetition distribution

Figure 84 shows the repetition distribution from the first game "Chapter 1: What is Software Architecture?", where 29% of the students repeated the game. And some students repeated this game 6 times.

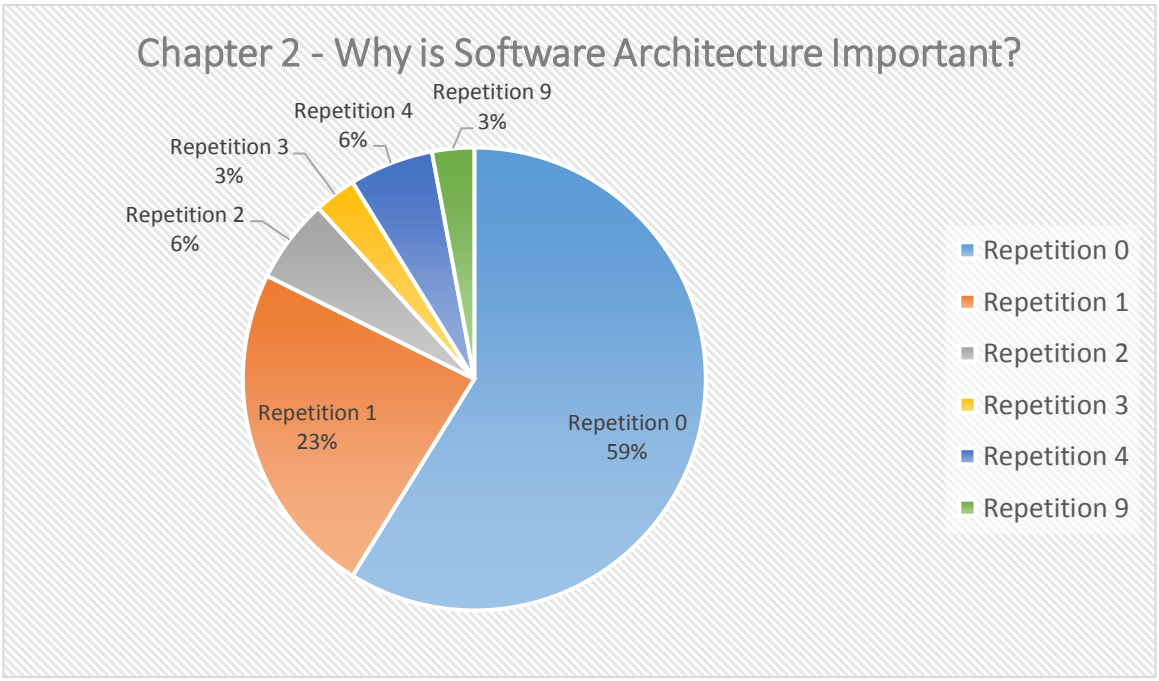


Figure 85 Game 2 repetition distribution

Figure 85 shows the repetition distribution from the second game “Chapter 2 – Why is Software Architecture Important?”. There were 41% students repeated this game, and the highest repetition is 9 times.

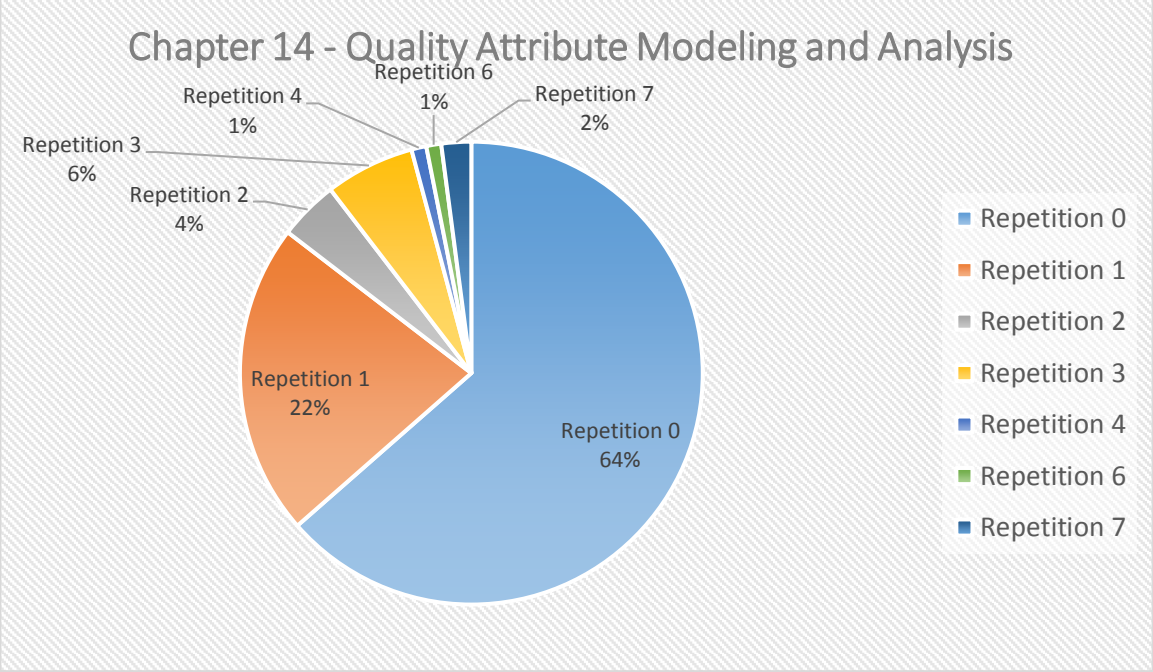


Figure 86 Game 3 repetition distribution

Figure 86 shows the repetition distribution of the last game, where 36% of students have repeated this game, and some students even repeated this game 7 times.

Homework/assignment is an important part of learning, but it is difficult to let student doing their homework, even it is mandatory, and not to mention the quality (is the student doing it only because they must, did they learned anything from it). Where Quiz Class managed to let student doing homework/assignment on their free will (optional assignment), and 41% of students did the assignment (game 2: Figure 85) again just to improve themselves, some students even repeated their assignment 9 times. Is this revolutionary for doing homework?

13.6 Teacher’s perspective

In the final test, there was only one teacher participated, it was not possible to present reliable data to analyze the usefulness of Quiz Class for teachers. This section will illustrate the relevant data, so the readers’ selves can get an idea, and evaluate how those data may help teachers.

I got some screenshots from the teacher participant. The first game was presented as the main example, some data from the second and the third game were used to compare.

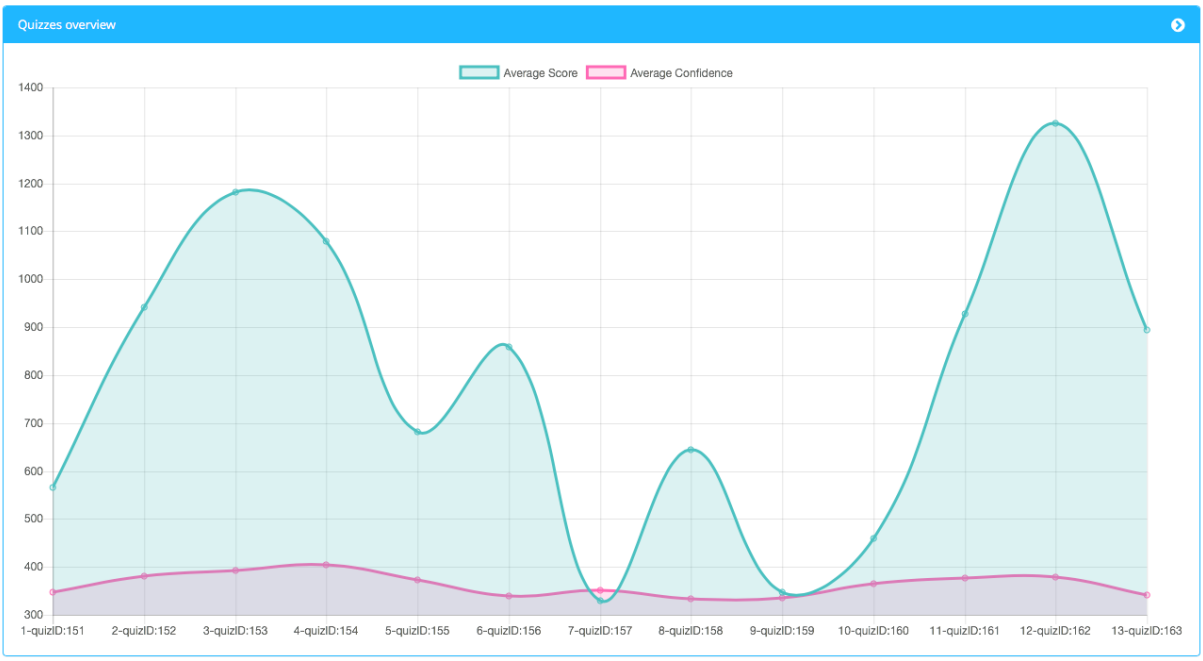


Figure 87 Game 1 – overview for teacher

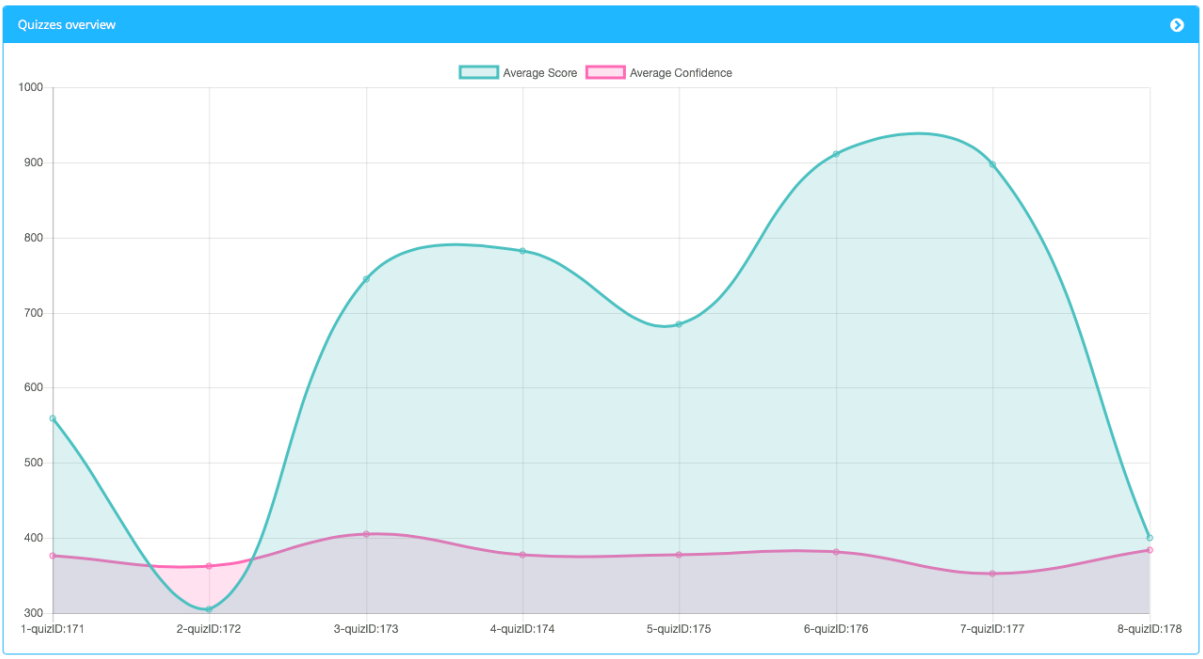


Figure 88 Game 3 - overview for teacher

Figure 87 illustrated the overview of Average Score and Average Confidence from the first game. The figure shows that the students had problems with question 7, question 9, and

question 10. Among those 3 questions, question 7 was worst because the average confidence level is relative high, but the average score is lowest.

Figure 88 shows the quiz overview for the teacher for the third game. Figure 88 shows that question 2 has the average lower confidence level, which indicates that students were not sure about this question; the average score was the lowest, which indicate that most of the students were wrong to this question. This means that most of the students knew that they did not understand this question, and the result shows that they did not.

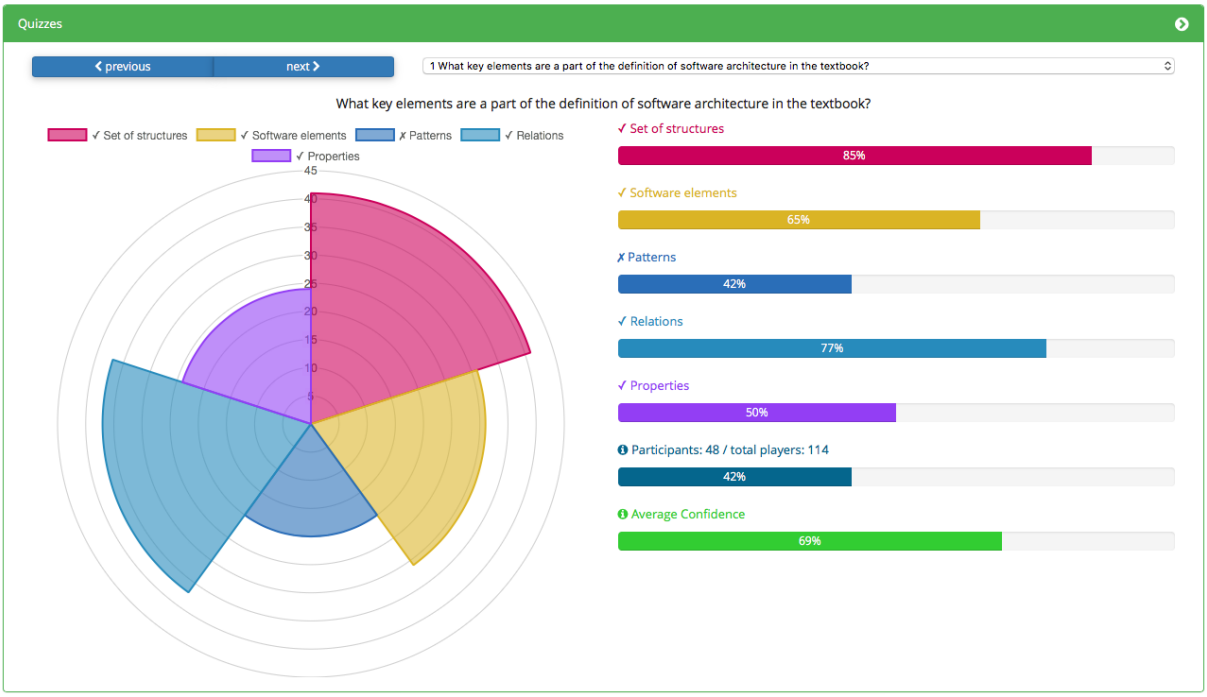


Figure 89 Game 1 - question view for teacher

Figure 89 shows the question view for the teacher, where 85% of participants of this game selected the first option, 77% selected the fourth option, 65% selected the second option, only 50% selected the fifth option, and all those options are correct. There was least percentage of participants selected the wrong option. The average confidence level was 69% on this question.

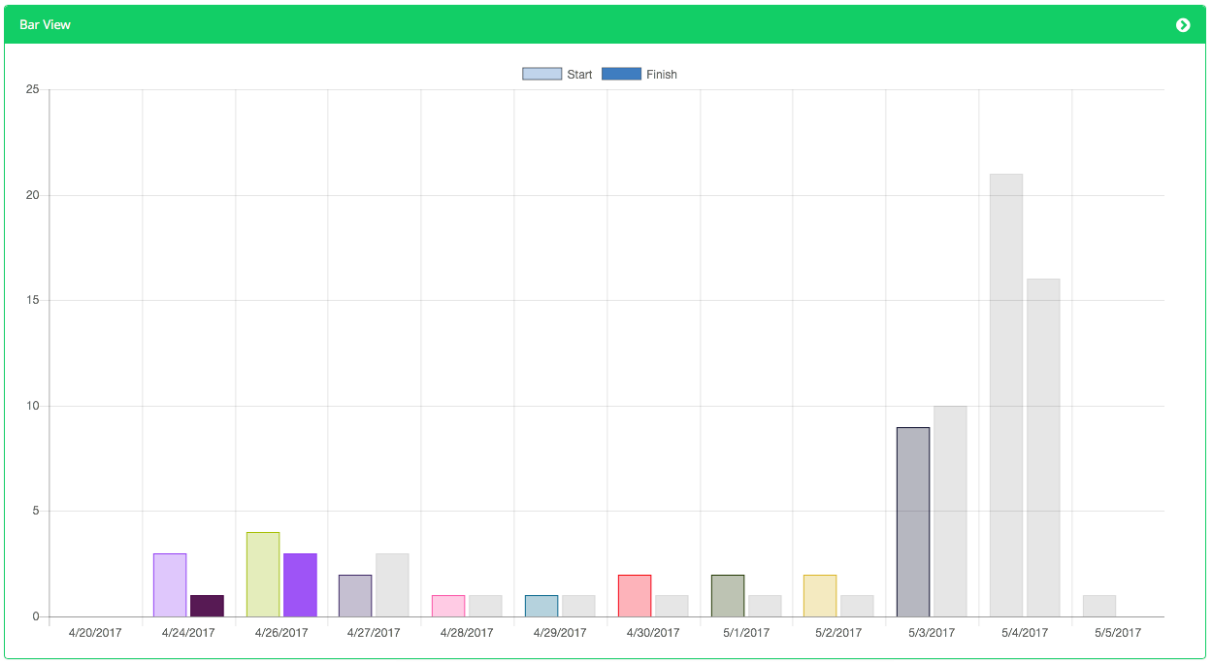


Figure 90 Game 1 - bar view for teacher

Figure 90 shows the bar view, which indicates that many students waited almost to the very end to start and finish their assignment. Most of the students started or finished their assignment at the very end of the game (deadline for the assignment). This time distribution of students doing their assignment represents what students usually do. However, one of the goals of Quiz Class was to motivate students start doing the assignment as early as possible.

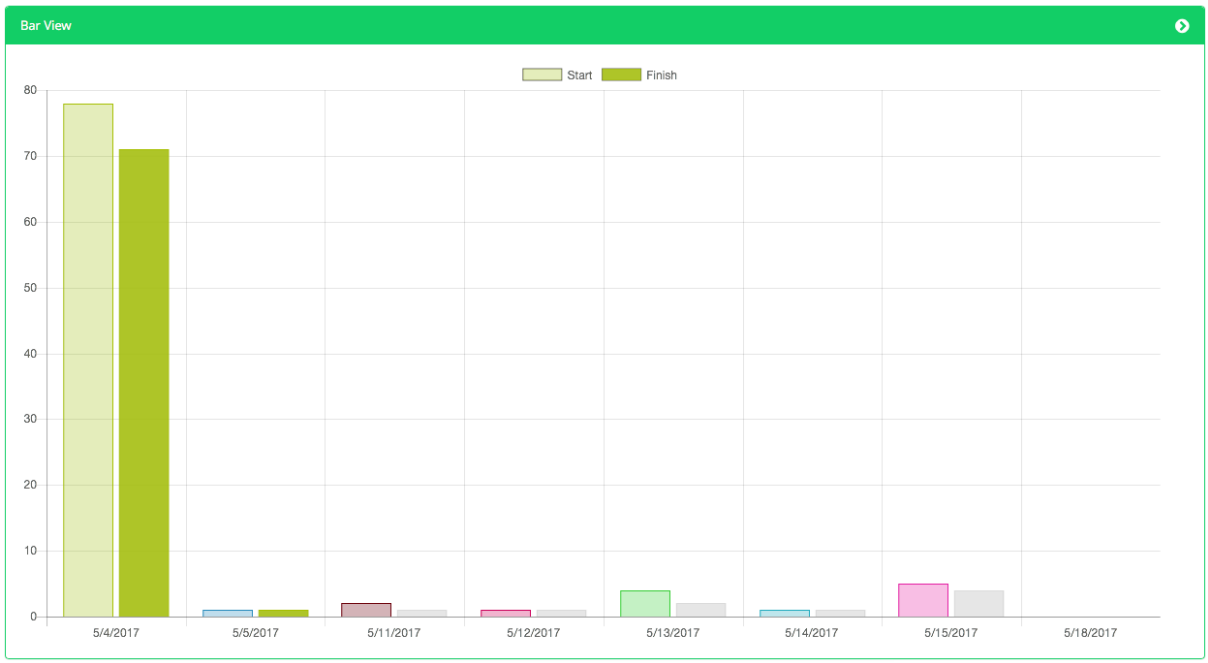


Figure 91 Game 3 - bar view for teacher

Figure 91 shows the bar view of Game 3. As there was a competition at the beginning of the game, most of the students started early. Such kind of competition may motivate students start doing homework early. Figure 90 shows that how most homework is done, but Figure 91 represents the most ideal state.

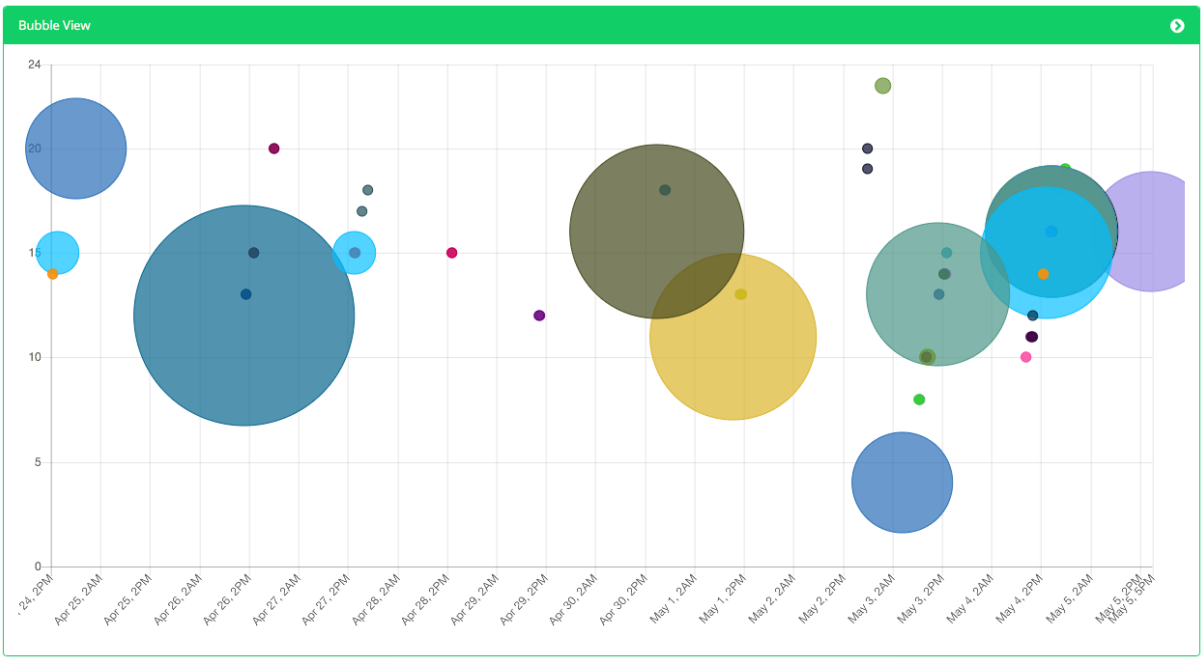


Figure 92 Game 1 - bubble view for teacher

Figure 92 illustrated the bubble view from Game 1 for the teacher. The figure shows that most of the students finished the game within a short time (small bubbles), but there were students who used longer time (bigger bubbles). Most of the student did their homework between 10:00 to 20:00. There was one student (the blue bubble) who started at 20 o'clock in the evening, and some days later he/she finished the homework about 4 o'clock in the morning.

The teacher testing Quiz Class found analysis tools useful, especially the quizzes overview which made it possible to both look at the confidence level of the students as well as looking at the correctness. This view is very useful in improving both teaching and homework assignments, as well as getting a clearer picture of the students' performance.

The analysis of every question was also found to be useful. The most interesting finding here was to see which of the alternatives the students picked and at the same time looking at the confidence level.

The bar/bubble views did not give any big surprises in terms of what was expected by the teacher. However, it would be very interesting to see how this view would change, if the students got motivated to deliver their homework earlier due to playing Quiz Class.

Part V: Evaluation and Conclusion

Part V This part evaluates the project, presents the conclusion of the study, limitations, challenges, and the further work.

14 Project Evaluation

This chapter presents the evaluation of the research, game design, chosen technologies, implementation, and the security of Quiz Class.

14.1 Research and game design

Before this project started, I was planning to create a quiz based application, where people can challenge their friends or be challenged for fun. The main idea was that people spend a huge amount of time on their digital devices with different apps, but most of the apps are not educational. The goal of the planned app was to let users have fun, and at the same time, gain useful information and knowledge, learn skills, and improve abilities based on the quiz contents.

In this project, my original idea evolved. The app is now concentrated on curriculum learning but is not limited to it. Most importantly, the game concept was combined with learning activities. I studied game theories, and thought about how to make game fun through the game design. Various related research about using game concepts in learning have been reviewed, and relevant games and applications have been investigated and compared.

By the time of this project started, many related studies and products have been produced, but none of them have the same game concept as this project. At the end of this project, more related game-based learning research, and products were discovered, but to my best knowledge, ideas such as the three-list ranking system, and the gambling feature (confidence level) were not found in any related work. Moreover, the analysis tools implemented in Quiz Class provides statistics and graphs, can deliver more useful information for teachers than most of the game-based learning applications.

GameFlow was used as a design guideline and evaluation tool during Quiz Class design and implementation. I did not run a GameFlow Criteria evaluation, but a checklist was made as shown in Table 15, where “+” indicate that the Criteria was implemented or can apply to Quiz Class in some degree, where “-” indicate that the Criteria have not been implemented in Quiz Class, and “/” indicate that the Criteria is not applicable to Quiz Class.

Element	Criteria	
Concentration	- games should provide a lot of stimuli from different sources. - games must provide stimuli that are worth attending to. - games should quickly grab the players' attention and maintain their focus throughout the game. - players shouldn't be burdened with tasks that don't feel important. - games should have a high workload, while still being appropriate for the players' perceptual, cognitive, and memory limits. - players should not be distracted from tasks that they want or need to concentrate on.	+ + + + / +
Challenge	- challenges in games must match the players' skill levels. - games should provide different levels of challenge for different players. - the level of challenge should increase as the player progresses through the game and increases their skill level. - games should provide new challenges at an appropriate pace.	+ + + +
Player Skills	- players should be able to start playing the game without reading the manual. - learning the game should not be boring, but be part of the fun. - games should include online help so players don't need to exit the game. - players should be taught to play the game through tutorials or initial levels that feel like playing the game. - games should increase the players' skills at an appropriate pace as they progress through the game. - players should be rewarded appropriately for their effort and skill development. - game interfaces and mechanics should be easy to learn and use.	+ + + - + + +
Control	- players should feel a sense of control over their characters or units and their movements and interactions in the game world. - players should feel a sense of control over the game interface and input devices. - players should feel a sense of control over the game shell (starting, stopping, saving, etc.). - players should not be able to make errors that are detrimental to the game and should be supported in recovering from errors. - players should feel a sense of control and impact onto the game world (like their actions matter and they are shaping the game world). - players should feel a sense of control over the actions that they take and the strategies that they use and that they are free to play the game the way that they want (not simply discovering actions and strategies planned by the game developers).	/ + / / / /
Clear Goals	- overriding goals should be clear and presented early. - intermediate goals should be clear and presented at appropriate times.	- +

Feedback	- players should receive feedback on progress toward their goals. - players should receive immediate feedback on their actions. - players should always know their status or score.	+ + +
Immersion	- players should become less aware of their surroundings. - players should become less self-aware and less worried about everyday life or self. - players should experience an altered sense of time. - players should feel emotionally involved in the game. - players should feel viscerally involved in the game.	+ + + + +
Social Interaction	- games should support competition and cooperation between players. - games should support social interaction between players (chat, etc.). - games should support social communities inside and outside the game.	+ - -

Table 15 GameFlow check

There are six items does not apply to Quiz Class. There are four items have not implemented in Quiz Class. The item "players should be taught to play the game through tutorials or initial levels that feel like playing the game" was not implemented, quiz Class for the player part is easy to use, all the tested players can play it without learning. The item "overriding goals should be clear and presented early" was implemented, but when a user repeat a game they should get a warning about the previous score will be lost, and the score only applies to the newest repetition (there was one player complained about this). The social interactions like chat functions were not implemented in Quiz Class, because of time limitation. In the implemented items, some of them were only partly applied, i.e. the challenge element, which is essentially depending on the creator, it can be applied if and only if the creator uses the games in a right way.

14.2 Chosen technologies

The chosen back-end technologies made developing the back-end server much easier and effective. There are sufficient documentations for Java programming language, Spring, and Hibernate framework, so most of the technical issues can be solved by reading the documentation. The Spring framework reduced the complexity of the back-end development, its annotations made the code more clean, readable, and easy to implement. The Spring framework increased the maintainability and reduced the development time for Quiz Class. The Hibernate framework made queries to the MySQL database more clear, easy to read and maintain, as it offers objects mapping to or from a database table. Hibernate also provides

auto create, drop, or update a database table based on an entity class, so there is no need to create, modify, or update a MySQL database table manually. Those increase the modifiability of the back-end server. The dependency management and build automation tool, Maven dynamically downloads Java libraries and describes the dependencies, which made add, remove, upgrade, or downgrade libraries effortless, and increased the project portability.

For the front-end, using TypeScript made coding much more productive than JavaScript. TypeScript delivered high code readability, made the implementation of Quiz Class much easier and more maintainable. The Angular framework made front-end development more organized, and the component architecture made the code more reusable, easy to organize and maintain. With Angular's data binding, a rich client-side application was built effectively. The Bootstrap framework made responsive web implementation faster, more effective, and easier. Bootstrap is well documented, and have a lot of examples code online, which makes it easy to learn and use. The Sass language made CSS more dynamic, and the ability to create variables in Sass made Quiz Class theme more consistent, and easy to use, which is currently not possible by pure CSS.

Quiz Class is deployed in both Heroku (Heroku n.d.) cloud application platform, and Amazon Web Services. Heroku is best for demonstration as it is always free to use, and easy to implement, but has lots of limitations. When the code was uploaded to Heroku all the settings were automatically done, including auto-connecting to a third-party database service. The back-end server was automatically running in HTTPS secure connection, which could save a lot of time for the developer.

Compared to Heroku, AWS was much more complex to setup and deploy, the learning curve was also shaper. I had to learn the VPN settings, security group settings, port settings, to secure Quiz Class communication on the cloud. To optimize Quiz Class, technologies such as AWS EC2, Elastic Beanstalk, S3, RDS, CloudFront, Load Balancers were used, which also required time to learn. I am satisfied with chosen AWS as my main cloud services, not only because it is more stable, easy to customize, powerful, functionally sufficient, but also gives me the opportunity to use some of the knowledges learned from my Computer Science education in practice. Putting all those technologies together and making them work effectively needs solid understanding of computer network, security, software architecture, etc.

14.3 Implementation

In my specialization project (TDT4501, autumn 2016), I implemented a simple front-end and back-end, running on a local server, which was a prototype to test if the chosen technologies were appropriate. In the follow-up project (TDT4900: Master's Thesis), the front-end and back-end were fully implemented.

The implementation process was mainly organized in 4 sprints. After the 4 sprints, some new features were implemented, improvements were made during the preliminary test. Each sprint took 3 weeks long, as shown in Figure 93. The first sprint focused on front-end and back-end structure implementation. The goal of the second sprint was to implement an alpha version with main functions. A beta version was implemented on the third sprint, where the server was moved from the local to the cloud. The fourth sprint brought Quiz Class on stage, where the focus was to make Quiz Class ready for real users to test and play with.

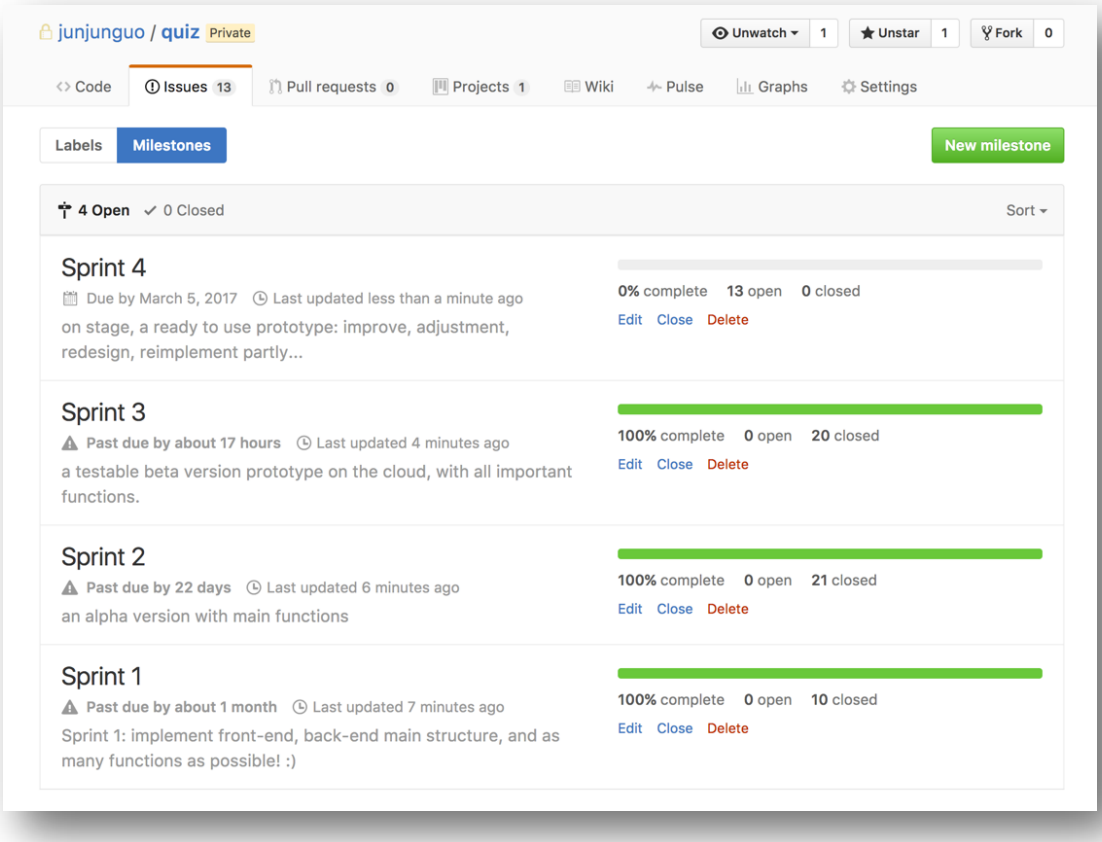


Figure 93 Sprints

Figure 93 is a screenshot took from the beginning of sprint 4 (the last sprint). The

Scrum iterative and incremental agile software development strategy was partly used in this project, as I was developing the Quiz Class alone. To better organize the implementation process, some of the useful ideas from Scrum framework was used. The backlog was initially made with most important functionalities. The first sprint had more general items in the backlog, and along with the implementation, more items were added with more concrete items. In each sprint, I tried to implement backlog items as many as possible, and always tried to implement higher priority items first. As shown in Figure 94, "L" represents low priority, "M" represent medium priority, and "H" represent high priority. Figure 94 is a screenshot took from the end of the last sprint, where all the high priority items from backlog were implemented, only some low priority items were left on the TODO list, the testing is already taken place. After the last sprint, the development continued, which mainly focused on bug fixing, improving UI, UX, and functionalities under the preliminary test.

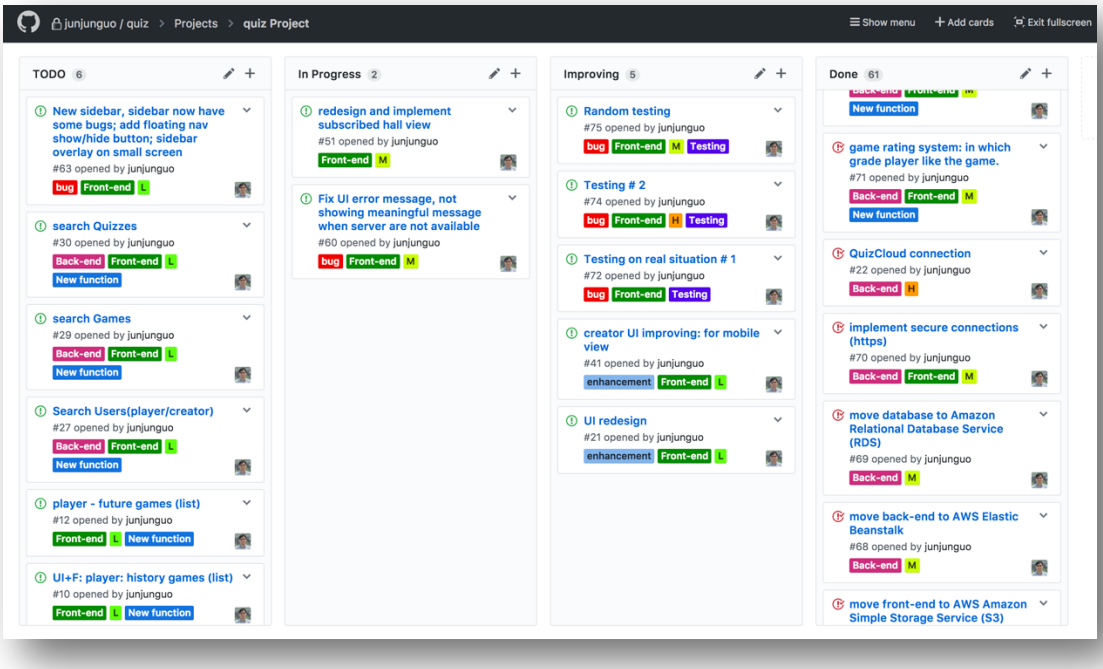


Figure 94 Last Sprint

The implementation took more time than planned. The original plan had 3 sprints, but it was not enough, thus Sprint 4 was added. All functional requirements from Chapter 7 were implemented. Much effort was used to improve UI, UX, fix errors, bugs, and add new functionalities after Sprint 4. Many new functions were implemented which did not listed on the functional requirements. There are still a lot of work need to be done, e.g. add social

functions: chat, comments, discussion etc. but for this project, which has very limited time and resources, I am satisfied with the implementation.

14.4 Software quality

This section mainly evaluates the quality requirements (section 7.3). In addition, the software security is also evaluated.

Availability

The Quiz Class uses the client-server model, where the client is a static web app (when not communicate with the server), therefore, the front-end can be cached by any node, e.g. user web browser caching, proxy caching etc. Different caching technologies were used to allow Quiz Class front-end being cached at the nearest node to end-user. The benefit is that even the original front-end hosting server went down, the client can still be available via caching, this also increases the client download speed, decrease the internet cost. The AWS cloud service ensures the back-end server uptime. In addition, the Elastic Load Balancing was also used, which enables the app to achieve fault tolerance, offers high availability, and automatic scaling (AWS n.d.). Most importantly, no server down was detected during testing.

Modifiability and Portability

The Angular framework was used at the front-end, which made Quiz Class component based. All main functions or often used functions were organized as components, which increases Quiz Class maintainability, code readability, and most importantly modifiability. The front-end can be switched between local and cloud within a second, it was hosted both at Github pages and AWS S3, which indicated that the front-end has high portability.

The back-end modifiability and portability is high because of the chosen technologies (evaluated in section 14.2) and the implementation (evaluated in section 14.3). The authentication, authorization, and security functions were quickly added to Quiz class, which were not planned. This indicated the high modifiability. Quiz Class back-end server was often switched between running locally, running on Heroku cloud, and AWS cloud. I never used Heroku before this project, but Quiz Class server was deployed on Heroku within an hour, which indicated that the back-end portability is high.

Usability

Quiz Class achieved 74 total scores on SUS, which indicated that the usability and learnability are good.

Security

Security had low priority in this project, but it is very important for every software, especially a web-based application. Quiz Class takes the advantage of web-based application, that end-users do not need to install the app, and they can use the app anywhere, anytime. All Quiz Class end user's data are saved in the cloud, so the end-user never need to worry about missing data. But those made security significant and essential for Quiz Class. As a computer science student with a specialization in software engineering, I cannot ignore security even the focus of the project is not security.

The Spring Security framework was implemented at the back-end, which provides both authentication and authorization, and it was extended to meet the Quiz Class requirements. The Spring Security provides protection against attracts like session fixation, clickjacking, cross-site request forgery, etc.

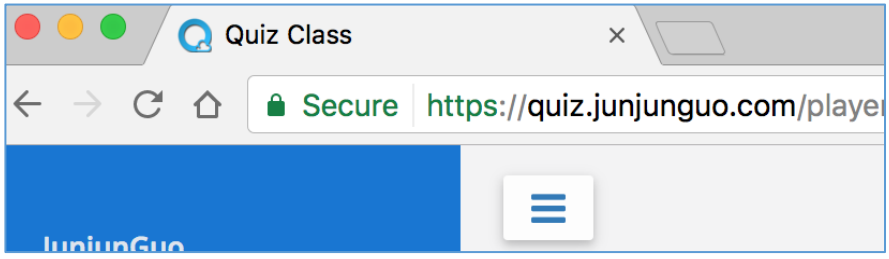


Figure 95 Quiz Class https secure connection

As shown in Figure 95, HTTPS was implemented in Quiz Class, the front-end and back-end communication are therefore secured. This implementation can not only protect Quiz Class users' privacy and integrity of the data exchange, but also protect against man-in-the-middle attacks, eavesdropping, tampering, forging, and guarantee the Quiz Class users are communicating with the Quiz Class server (not malicious servers).

Figure 96 shows the Quiz Class SSL Server Test from QUALYS SSL LABS[30], which is a free online service, performs a deep analysis of the configuration of any SSL web server on the public internet. Quiz Class got overall rating A from the SSL Server Test, where the grade is given from A to F, A is the best grade.

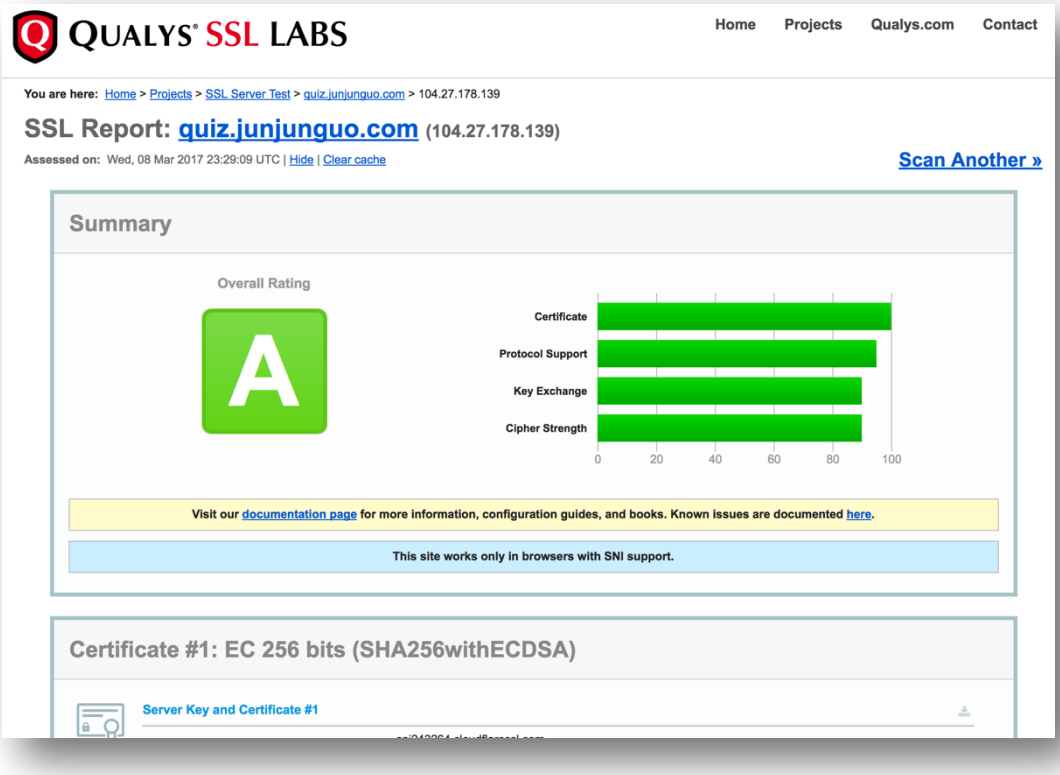


Figure 96 SSL Server Test

[30] SSL Labs is a collection of documents, tools and thoughts related to SSL. It's an attempt to better understand how SSL is deployed, and an attempt to make it better. SSL Server Test is one of the most popular SSL testing tools to check all latest vulnerability and misconfiguration (Ristić, I. (n.d.). "Qualys SSL Labs." Bringing you the best SSL/TLS and PKI testing tools and documentation. Retrieved 22.04, 2017, from https://www.ssllabs.com/index.html).

15 Conclusion

This chapter summarizes the thesis, discusses and answers the research questions, points out limitations and challenges of the study.

In this master project, I developed a game concept for learning, designed the software architecture, implemented a digital game-based learning application (Quiz Class), and tested Quiz Class on different user groups. The goal was to experiment and discover how Quiz Class could influence learning and teaching.

The project was developed as an iterative process, as shown in Figure 1. I started with a simple idea, then developed the objectives of the study, research questions, literature study, technology study were made. The idea was developed and formulated to a game-based learning concept, software requirements and software architecture were developed, software prototype was implemented, and tested.

173 users registered on Quiz Class. In the preliminary test, there were 28 valid participants. The final test had 116 participants and 78 valid questionnaire responses. The data was collected mainly through questionnaires, but user observations and informal interviews were also made to support the data. As a conclusion, my research questions are answered below:

RQ1. How can we make learning fun?

> Studies and research have indicated that game concept can be used in learning and make learning fun. Various products have also proved that digital game-based learning can make learning fun.
>
> This study has designed a game concept, which contains challenges, fantasy, and curiosity from Malone (Malone 1980). In addition, the GameFlow (Sweetser and Wyeth 2005) framework was also used as guidance to combine game concept into learning activities, therefore making learning fun. Quiz Class application was implemented to make learning fun. The testing results have indicated that about 67% of participants think that Quiz class has made learning fun (see more on RQ4).

RQ2. How can we design and implement a game-based learning application?

> There are various types of game-based learning applications, they could be designed and implemented in different ways. This project started with study the relevant

theories, research, products, and technologies. Based on the studies, a game concept has been designed, which combined game concept into learning activities. And then, the software requirements, software architecture were made. Finally, a game-based learning application (Quiz Class) has implemented and deployed in the cloud. In addition, Quiz Class has been tested as a game-based learning application and received mainly positive results.

RQ3. How can we make Quiz Class easy to access?

The observations from this study have indicated that every participant has at least one digital devices. Most digital devices have a web browser. A web-based front-end (client) application can be easily accessed by any end-users (both players and game creators). As described by W3C: "The Web is fundamentally designed to work for all people, whatever their hardware, software, language, culture, location, or physical or mental ability" (W3C n.d.). In addition, Quiz Class is like most of the other web-based applications, which do not need any installations or complex settings.

The tests in this study have shown that Quiz Class was successfully accessed and used by its participants. Different OS (Windows 45%, Macintosh 21%, Android 19%, iOS 11%), diverse mobile devices (Apple 38%, Samsung 14%, Huawei 14% …), various device categories (desktop 70%, mobile 28%, tablet 2%), various web browsers (Chrome 68%, Safari 14%, Firefox 11% …), and diverse screen resolutions (1440*900 44%, 360*640 19%, 1920*1080 9% …) were used to access Quiz Class.

RQ4. How do students being affected by using Quiz Class?

RQ4.1. Students reactions on a game-based application being used for their study.

From the preliminary test, most of the participants were very positive with Quiz Class and thought it would be nice to use it in their study. The questionnaire indicated positive quantitative results, where 62% of participants think it was fun to compete against others, 62% of participants thinks that they were engaged while playing, 67% of participants agree that the app made doing assignment more fun, 63% of participants think that the app encourages them to do well in assignment, 58% of participants wish the app could be used for more assignments. Many positive feedbacks were given by the participants, one of a participant wrote:

"It is a wonderful tool for practicing a course. It aims at making teaching more modern and interactive for students. The thing I like the most is the ranking: The fact that you get points and overall ranking for doing well is a great indication of yourself improving, which is very vague in traditional practicing of a course."

RQ4.2. Students interest and motivation in doing the assignment.

The observations have shown that the players/students were engaged while playing. The questionnaire indicated that 67% of participants found the app made doing assignment fun, 55% of participants agree that the app had increased their motivation on doing an assignment, and 63% of participants think the app encourages them to do well in the assignment. They were even motivated to do the assignment many times (see RQ4.3.), which was based on self-interest and self-motivation. The students doing the assignment again can be considered as strong support to students' interest and motivation in doing the assignment.

RQ4.3. Students doing the assignment again.

In the preliminary test, there were many participants repeated the game over and over to improve the scores. The questionnaire results had also pointed out that most of the participants agree that they were encouraged to repeat the assignment.

The three games from the final test have indicated that 29% participants repeated the first game, 41% participants repeated the second game, and 36% of participants had repeated the last game. In addition, some participants even repeated the game 9 times. As a support to this, the preliminary test has shown that 26% of group participants repeated the game, and 36% of individual participants repeated the game.

In the traditional way of doing an assignment, if not forced, it is hard to believe that someone would do an assignment 9 times to improve it, but by using a digital game-based learning app some students did, and it was motivated by students' themselves.

15.1 Limitations and Challenges

There are, of course, limitations and challenges in this study, which cannot be ignored. It is important to recognize the limits, address the challenges, and set realistic expectations. In this

section, some limitations and challenges encountered in this study are listed out, so that the readers could know the extent of the investigation, and the circumstances that this study has conducted.

15.1.1 Limitations and challenges of the Quiz Class App

The Quiz Class is developed as a prototype, which is aimed for investigation and experimentation. Some important functions could influence user experiences, i.e. social login[31] was not implemented. Some users complained about they must register an account to use Quiz Class.

15.1.2 Challenges on finding participants

To gain better testing results, two high schools in Trondheim were visited; two middle schools were contacted in China and a Norwegian high school teacher was contacted.

The first high school I visited two times. The first time, one teacher promised to contact other teachers and find someone who would like to help with testing. In this school, to talk with any teacher, I must ask the reception to get a teacher out of their office area. After a week, I did not get any feedback, so I visited again, waited for half an hour to meet the one who promised to help. But she said she was too busy to talk, and I should make an appointment. And there was no reply until this section was written.

At the other high school, I met the principal, she was willing to help at the beginning, but when I clarified that the testing needs the teacher to be involved, then she said she was busy. I then asked if I could talk with other teachers, she refused and said all teachers in her school were busy.

I thought maybe I could test it in a Chinese school, most of the schools in China have bigger classes. A Chinese middle school usually have about 50 students per class, and each teacher may teach 2 to 4 classes. And more importantly, there are usually a lot of homework for each course. Those conditions fit Quiz Class. If a teach teaches two classes, there will be about 100

[31] Social login, also known as social sign-in, is a form of single sign-on using existing information from a social networking service such as Facebook, Twitter or Google+, to sign into a third-party website instead of creating a new login account specifically for that website. It is designed to simplify logins for end users as well as provide more and more reliable demographic information to web developers (Retrieved 15.06, 2017, from https://en.wikipedia.org/wiki/Social_login).

students, which means, the teacher must correct each student's homework, that is 100. Quiz Class could easily reduce teacher's work, and plus the analysis tool which will give them a lot of advantages.

I could not travel to China, as the time was very limited, so I called my brother for help. I moved the server from AWS to Heroku, trying to speed up the connection to Quiz Class, which did not help much. And then I opened a new region at AWS Singapore to serve Chinese users, it did not increase the connection speed. And finally, I decided to open a server in China. AWS in China is operated by a Chinese company, which is separate from AWS worldwide. I must apply a new account to use AWS China. But to apply an AWS China account, I must apply as a registered company, not as an individual person. I called help from my father, after a week, my AWS China account was approved. In the meantime, I implemented the Chinese language to Quiz Class. Have a server in China does not mean that my server can be accessed by anyone. I must apply an ICP licence[32] to permit Quiz Class can be operated in China. I did not know the ICP licence before I debugged Quiz Class code, and AWS server settings for few days, because no one told me this, and the server always responded that I am not authenticated to communicate to Quiz Class server, so I thought the AWS cloud settings in China were different.

My brother visited two middle schools, several times. At the first middle school, one of a school teacher promised to test in her class, but when the head teacher heard about the testing, she stopped the planned testing. The reason was that students are not allowed to use mobile at school. They were afraid that this may break the rule, and they may lose the control of their students, which was too risky. And at the other middle school, my brother met the headmaster. There were lots of good feedbacks about use technologies and innovative ways to improve the existing teaching methods. But they could not help to test Quiz Class in their school.

The last attempted was a Norwegian high school teacher. The teacher was surprised when she tried Quiz Class and saw the analysis tools. She reflected that the analysis tools not only gave her opportunities to better understand her students but also reflects how was she doing on teaching. She was so excited, asked if she must pay to use the software, if she could show

---

[32] **ICP licence** (abbreviation for *Internet Content Provider) is a permit issued by the Chinese Ministry of Industry and Information Technology to permit China-based websites to operate in China (Retrieved 25.05, 2017, from https://en.wikipedia.org/wiki/ICP_license).*

Quiz Class to her headmaster and other colleagues. But Quiz Class is built for my master project, and the app might not be available after my master project. When this was told, she answered that she was busy, but she will ask the headmaster if she could help with testing, and she will consider it and give me feedback. I wrote two emails but have not got a reply until this part was written.

15.1.3 Limitations of Sampling

The methods of sampling conducted in this study may have limitations. Because of the limited access to schools and teachers, convenience sampling was used for this study. As criticized by Bryman, the problem with a convenience sampling strategy is that it is impossible to generalize the findings, because we do not know of what population this sample is representative (Bryman 2015). In addition, only one class was tested, and the test was made optional to students to test their knowledge at the end of the semester. It was not a real assignment/homework situation that the Quiz Class was originally built for.

16 Further work

The study presented in the thesis have raised more questions than that it has answered. There are many research questions arising from this work which should be pursed.

The original research goal of this study was to investigate: how Quiz Class could influence teachers work, students study and their interactions. As the time was limited, Quiz Class was only tested in one class. It was not possible to investigate how Quiz Class influence teachers work with only one teacher, and their interactions cannot be investigated neither. If more courses and teachers can participate Quiz Class testing, following research questions can be answered:

RQ1. How does Quiz Class influence teachers work?

1. To what extent does the application improve / increase effective teaching?
2. To what extent does the app reduce teachers work (on correcting student homework)/or make teacher works more effectively?
3. To what extent does the app provide good feedback to the teachers and they can use that feedback to improve teaching?
4. What are the disadvantages for the teachers to use this application?

RQ2. How Quiz Class influences student-teacher interaction?

1. To what extent does the application increase student-teacher interaction?
2. To what extent does the application let the teacher better known by their students?
3. To what extent does the application increase the teacher's knowledge about their student/student's ability/how much student understand/what student do not understand?
4. To what extent does the application help students to be better known toward each other?

RQ3. To what extent does Quiz Class influence students to do the assignment again?

RQ3 can be answered if more testing can be done in real homework situations. Quiz Class records all players answered questions, it is easy for Quiz Class to extract and analyse those data, to answer questions like:

RQ4. To what extent does student improve their knowledge/learning by repetition?

More interesting questions such as:

RQ5. How many times does repetition give best learning results?

Can be answered by bigger sampling size. I can build a function to show those results to the players, so they know the results of general population. AI maybe implemented by using machine learning, to study and give each individual person suggestions for better learning results.

There are many other potentials to further work with this study, both academically and in practice.

Bibliography

.NET (n.d.). ".NET - Powerful Open Source Cross Platform Development - Microsoft." Retrieved 24.10, 2016, from https://www.microsoft.com/net.

{less} (n.d.). "Less extends CSS with dynamic behavior such as variables, mixins, operations and functions.". Retrieved 27.03, 2017, from http://lesscss.org/.

Akkerman, S., et al. (2009). "Storification in History education: A mobile game in and about medieval Amsterdam." Computers & Education **52**(2): 449-459.

Aldrich, C. (2003). Simulations and the Future of Learning: An Innovative (and Perhaps Revolutionary) Approach to e-Learning, Pfeiffer; 1 edition.

Allan, K., et al. (2008). "theguardian: Games." Retrieved 06.06, 2017, from https://www.theguardian.com/technology/2008/jan/24/games.sony.

Ampatzoglou, A. and A. Chatzigeorgiou (2007). "Evaluation of object-oriented design patterns in game development." Information and Software Technology **49**(5): 445-454.

Anderson, J. L. and M. Barnett (2013). "Learning Physics with Digital Game Simulations in Middle School Science." Journal of Science Education and Technology **December 2013, Volume 22, Issue 6, pp 914–926**.

Angelo, T. A. and K. P. Cross (1993). "Classroom assessment techniques: From Classrom Assessment Techniques, A handbook for college teachers." Jossey-Bass; 2 edition.

Angular (n.d.). "Angular: One framework." Retrieved 01.19, 2017, from https://angular.io/.

AngularDoc (n.d.). "Angular Docmuentation: Architecture overview." Retrieved 25.05, 2017, from https://angular.io/docs/ts/latest/guide/architecture.html.

AppEngine (n.d.). "App Engine - Platform as a Service | Google Cloud Platform." Retrieved 24.01, 2017, from https://cloud.google.com/appengine.

AWS (n.d.). "Amazon Web Services (AWS) - Cloud Computing Services." Retrieved 16.01, 2017, from https://aws.amazon.com/.

Azure (n.d.). "Microsoft Azure: Cloud Computing Platform & Services." Retrieved 16.01, 2017, from https://azure.microsoft.com/.

Bass, L., et al. (2003). Software Architecture in Practice, Addison-Wesley Professional; 2 edition (April 19, 2003).

Bass, L., et al. (2012). Software Architecture in Practice, Addison-Wesley Professional; 3 edition (October 5, 2012).

Beale, I. L., et al. (2007). "Improvement in cancer-related knowledge following use of a psychoeducational video game for adolescents and young adults with cancer." J Adolesc Health **41**(3): 263-270.

BizSpark (n.d.). "Microsoft BizSpark: BizSpark gives startups 3 years of free stuff – software, services, tech support, and Azure cloud." Retrieved 24.10, 2016, from https://bizspark.microsoft.com/.

Bootstrap (n.d.). "Bootstrap · The world's most popular mobile-first and responsive front ...". Retrieved 05.10, 2016, from http://getbootstrap.com/.

Boyle, E., et al. (2011). "The role of psychology in understanding the impact of computer games." Entertainment Computing **2**(2): 69-74.

Brooke, J. (1996). "SUS-A quick and dirty usability scale." Usability evaluation in industry **189**(194): 4-7.

Bryman, A. (2015). Social Research Methods, OUP Oxford 5 edition.

Buzz! (n.d.). "Buzz!: The Schools Quiz." Buzz! Retrieved 16.01, 2017, from https://en.wikipedia.org/wiki/Buzz!:_The_Schools_Quiz.

C# (n.d.). "C Sharp Programming language." Retrieved 24.10, 2016, from https://en.wikipedia.org/wiki/C_Sharp_(programming_language).

ClassDojo (n.d.). "Build wonderful classroom communities with parents and students.". Retrieved 16.01, 2017, from https://www.classdojo.com/.

ClearDB (n.d.). "The Ultra Reliable, Geo Distributed Data Services Platform." ClearDB on Heroku enables you to build your apps using native MySQL databases so that you can focus on your applications and get your ideas to market ... Retrieved 20.04, 2017, from http://w2.cleardb.net/.

CloudEndpoints (n.d.). "Cloud Endpoints: Develop, deploy and manage APIs on any Google Cloud backend ". Retrieved 24.10, 2016, from https://cloud.google.com/endpoints/.

CloudStorage (n.d.). "Cloud Storage - Online Data Storage | Google Cloud Platform." Retrieved 24.10, 2016, from https://cloud.google.com/storage/.

Connolly, T. M., et al. (2012). "A systematic literature review of empirical evidence on computer games and serious games." Computers & Education **59**(2): 661-686.

Csikszentmihalyi, M. (2002). Flow: The Classic Work on How to Achieve Happiness, with a new Introduction by the author, Rider & Co (August 1, 2002).

CSS (n.d.). "CSS developer guide." Retrieved 05.01, 2017, from https://developer.mozilla.org/en-US/docs/Web/Guide/CSS.

Datastore (n.d.). "Datastore - NoSQL Schemaless Database | Google Cloud Platform." Retrieved 24.10, 2016, from https://cloud.google.com/datastore/.

DeWalt, K. M. and B. R. DeWalt (2002). Participant Observation: A Guide for Fieldworkers, Rowman Altamira.

Duolingo (n.d.). "Duolingo: Learn Spanish, French and other languages for free." Retrieved 13.10, 2016, from https://www.duolingo.com/.

Ebner, M. and A. Holzinger (2007). "Successful implementation of user-centered game based learning in higher education: An example from civil engineering." Computers & Education **49**(3): 873-890.

Eck, R. V. (2006). "Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless...." EDUCAUSE Review **vol. 41, no. 2 (March/April 2006)**: 1-16.

Eden, D. (2012). "ECAR Study of Undergraduate Students and. Information Technology (Research Report)." EDUCAUSE Center for Applied Research.

Eriksson, H.-E. and M. Penker (2000). "Business Modeling with UML." Business patterns at work(Citesser).

Fenton, S. (2014). Pro TypeScript Application-Scale JavaScript Development, Apress.

Fetterman, D. M. (2009). Ethnography: Step-by-Step, SAGE Publications.

Freeman, A. (2014). Pro AngularJS, Apress.

Garris, R., et al. (2002). "Games, Motivation, and Learning: A Research and Practice Model." Simulation & Gaming **33**(4): 441-467.

Gibson, D., et al. (2007). Games and simulations in online learning: research and development frameworks, Information Science Pub., 2007.

golang (n.d.). "The Go Programming Language." Retrieved 23.11, 2016, from https://golang.org/.

GooglePlay (n.d.). "Google Play: Enjoy millions of the latest Android apps, games, music, movies, TV, books, magazines & more. Anytime, anywhere, across your devices.". Retrieved 12.10, 2016, from https://play.google.com/store?hl=en.

Hammonds, L., et al. (2013). "Gateway tools: Five tools to allow teachers to overcome barriers to technology integration." Delta Kappa Gamma Bulletin **80**(Delta Kappa Gamma Society International): 36-40.

Hatch, J. A. (2002). Doing Qualitative Research in Education Settings, State University of New York Press.

Hattie, J. and H. Timperley (2007). "The Power of Feedback." Review of Educational Research **77**(1): 81-112.

Heroku (n.d.). "Heroku: Cloud Application Platform." Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. Retrieved 20.04, 2017, from https://www.heroku.com/.

Hibernate (n.d.). "Hibernate ORM: Idiomatic persistence for Java and relational databases.". Retrieved 16.11, 2016, from http://hibernate.org/orm/.

Holzinger, A. (2005). "Usability engineering methods for software developers." Commun. ACM **48**: 71-74.

Horstmann, C. S. (2013). Big Java: Early Objects, 5th Edition, Wiley; 5 edition (January 1, 2013).

HTML (n.d.). "HTML developer guide." Retrieved 05.10, 2016, from https://developer.mozilla.org/en-US/docs/Web/Guide/HTML.

HTTPS (n.d.). "Wikipedia: HTTPS." HTTP. Retrieved 27.03, 2017, from https://en.wikipedia.org/wiki/HTTPS.

InternationalizationLocalization (n.d.). "Wikipedia: Internationalization and localization." Retrieved 19.04, 2017, from https://en.wikipedia.org/wiki/Internationalization_and_localization.

Java (n.d.). "Java Software | Oracle." Retrieved 14.10, 2016, from https://www.oracle.com/java/.

JavaLang (n.d.). "Java (programming language)." Retrieved 14.10, 2016, from https://en.wikipedia.org/wiki/Java_(programming_language).

JavaScript (n.d.). "JavaScript." Retrieved 05.10, 2016, from https://en.wikipedia.org/wiki/JavaScript.

JavaSE (n.d.). "Java Platform Standard Edition 8 Documentation:Description of Java Conceptual Diagram." Retrieved 14.10, 2016, from http://docs.oracle.com/javase/8/docs/index.html.

Johnson, S. (2006). Everything Bad is Good for You: How Today's Popular Culture is Actually Making Us Smarter, Riverhead Books; 1 Reprint edition (May 2, 2006).

Kahoot! (n.d.). "Kahoot!: Making Learning Awesome!". Retrieved 11.10, 2016, from https://getkahoot.com/.

Kiili, K. (2005). "Digital game-based learning: Towards an experiential gaming model." The Internet and Higher Education **8**(1): 13-24.

Kruchten, P. (1995). "Architectural Blueprints—The "4+1" View Model of Software Architecture." IEEE Software 12 (6) November 1995, pp. 42-50.

Kvale, S. and S. Brinkmann (2009). InterViews: Learning the Craft of Qualitative Research Interviewing, SAGE Publications, Inc; 2nd edition (July 16, 2008).

LearningGamesForKids (n.d.). "Learning Games For Kids." Retrieved 13.10, 2016, from http://www.learninggamesforkids.com/.

Leffingwell, D. (2011). agile software requirements: Lean RequiRements PRactices foR teams, PRogRams, and the enteRPRise, Pearson Education, Inc.

Magerkurth, C., et al. (2005). "Pervasive Games: Bringing Computer Entertainment Back to the Real World." ACM Computers in Entertainment **Vol. 3, No. 3, July 2005. Article 4A.**: 1-19.

Malone, T. W. (1980). What makes things fun to learn? heuristics for designing instructional computer games. Palo Alto, California, USA, Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems. **10.1145/800088.802839:** 162-169.

Maven (n.d.). "Apache Maven." Retrieved 17.11, 2016, from https://maven.apache.org/.

McGonigal, J. (2010). "Gaming can make a better world." Retrieved 13.10, 2016, from https://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world/transcript?language=en.

Méndez, D. and J. Slisko (2013). "Software Socrative and smartphones as tools for implementation of basic processes of active physics learning in classroom: An initial feasibility study with prospective teachers." European Journal of Physics Education **4**(European Journal of Physics Education).

Miller, M. and V. Hegelheimer (2006). "The SIMs meet ESL Incorporating authentic computer simulation games into the language classroom." Interactive Technology & Smart Education (2006) 4:311– 328(© 2006 Troubador Publishing Ltd.).

Moeller, B. and T. Reitzes (2011). "Integrating Technology with Student-Centered Learning." Education Development Center, Inc. (EDC)(Nellie Mae Education Foundation).

MongoDB (n.d.). "MongoDB for GIANT Ideas - Build innovative modern applications that create a competitive advantage.". Retrieved 23.11, 2016, from https://www.mongodb.com/.

MySQL (n.d.). "MySQL: The world's most popular open source database." Retrieved 16.11, 2016, from http://www.mysql.com/.

MySQLdoc (n.d.). "Chapter 25 Connectors and APIs." Retrieved 16.01, 2017, from http://dev.mysql.com/doc/refman/5.7/en/connectors-apis.html.

MySQLServer (n.d.). "MySQL Server Source code." Retrieved 16.11, 2016, from https://github.com/mysql/mysql-server.

NoSQL (n.d.). "NoSQL database." Retrieved 23.11, 2016, from https://en.wikipedia.org/wiki/NoSQL.

npm (n.d.). "npm is the package manager for JavaScript and the world's largest software registry. Discover packages of reusable code — and assemble them in powerful new ways.". Retrieved 29.05, 2017, from https://www.npmjs.com/.

Overmars, M. (2012). "A Brief History of Computer Games."

OWASP (2017). "Transport Layer Protection Cheat Sheet." Open Web Application Security Project. Retrieved 27.03, 2017, from https://www.owasp.org/.

Papastergiou, M. (2009). "Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation." Computers & Education **52**(1): 1-12.

Patterns, M. and P. Team (2009). Microsoft® Application Architecture Guide (Patterns & Practices), Microsoft Press; Second Edition edition (November 22, 2009).

Prensky, M. (2001). "Digital Natives, Digital Immigrants." MCB University Press **Vol. 9 No. 5, October 2001**: 1-6.

Prensky, M. (2003). "Digital Game-Based Learning." ACM Computers in Entertainment **Vol. 1, No. 1, October 2003, Book 02.**: 1-4.

Prensky, M. (2004). Digital Game-based Learning, McGraw-Hill.

Prensky, M. (2006). Don't Bother Me Mom--I'm Learning!, Paragon House; 1St Edition edition

ProProfs (n.d.). "ProProfs Build & Test Knowledge." Retrieved 30.09, 2016, from http://www.proprofs.com/.

Quizlet (n.d.). "Quizlet: Learning tools & flashcards, for free." Quizlet makes simple learning tools that let you study anything. Retrieved 31.03, 2017, from https://quizlet.com/.

Rosas, R., et al. (2003). "Beyond Nintendo: design and assessment of educational video games for first and second grade students." Comput. Educ. **40**(Elsevier Science Ltd.): 71-79.

RWD (n.d.). "Wikipedia: Responsive web design." Retrieved 14.10, 2016, from https://en.wikipedia.org/wiki/Responsive_web_design.

Sass (n.d.). "Sass: Syntactically Awesome Style Sheets." Retrieved 05.10, 2016, from http://sass-lang.com/.

Socrative (n.d.). "Socrative is your classroom app for fun, effective classroom engagement." Retrieved 08.06, 2017, from https://www.socrative.com/.

Spradley, J. P. (2016). The Ethnographic Interview, Waveland Press, Feb 17, 2016.

Spring (n.d.). "Spring by Pivotal: Let's build a better Enterprise.". Retrieved 14.01, 2017, from https://spring.io/.

SpringDoc (n.d.). "Spring Documentation: Introduction to the Spring Framework." Retrieved 14.03, 2017, from http://docs.spring.io/spring/docs/current/spring-framework-reference/html/overview.html.

SpringFramework (n.d.). "GitHub: Spring Framework source code." Retrieved 14.03, 2017, from https://github.com/spring-projects/spring-framework.

SpringSecurity (n.d.). "Pivotal Software: Spring Security." Spring 4.2.2. Retrieved 27.03, 2017, from http://projects.spring.io/spring-security/.

Sweetser, P. and P. Wyeth (2005). "GameFlow: A Model for Evaluating Player Enjoyment in Games." ACM Computers in Entertainment **Vol. 3, No. 3, July 2005. Article 3A**: 24.

Tomcat® (n.d.). "Apache Tomcat®." Retrieved 18.10, 2016, from http://tomcat.apache.org/.

Tremblay, E. A. (2010). "Educating the Mobile Generation – using personal cell phones as audience response systems in post- secondary science teaching." Jl. of Computers in Mathematics and Science Teaching **20(2)**: 217-227.

TypeScript (n.d.). "TypeScript - JavaScript that scales." Retrieved 05.10, 2016, from https://www.typescriptlang.org/.

UbuntuServer (n.d.). "Scale out with Ubuntu Server." Retrieved 16.10, 2016, from https://www.ubuntu.com/server.

UML® (n.d.). "OMG'S UNIFIED MODELING LANGUAGE® (UML®)." Retrieved 17.10, 2016, from http://www.uml.org/what-is-uml.htm.

Vargas, J. M. (2011). Modern learning: Quizlet in the social studies classroom. Dept.of Curriculum and Instruction., Wichita State University, College of Education. **Master:** 48.

W3C (n.d.). "Accessibiliby." The World Wide Web Consortium. Retrieved 30.04, 2017, from https://www.w3.org/standards/webdesign/accessibility.

Walford, G., et al. (2010). The SAGE Handbook of Measurement, SAGE.

Wang, A. I. (2015). "The wear out effect of a game-based student response system." Computers & Education **82**: 217-227.

Wang, A. I., et al. (2014). "Three Social Classroom Applications to Improve Student Attitudes." Education Research International **2014**: 1-14.

Wang, A. I. and A. Lieberoth (2016). "The effect of points and audio on concentratio, engagement, enjoyment, learning, motivation, and classroom dynamics using Kahoot!".

Wang, A. I., et al. (2007). "An Evaluation of a Mobile Game Concept for Lectures."

WebContainer (n.d.). "Wikipedia: Web container." Retrieved 16.11, 2016, from https://en.wikipedia.org/wiki/Web_container.

Wu, B., et al. (2011). "IMPROVEMENT OF A LECTURE GAME CONCEPT
- Implementing Lecture Quiz 2.0."

Zyda, M. (2005). "From Visual Simulation to Virtual Reality to Games." Published by the IEEE Computer Society **38**(9): 25-32.

Part VII: Appendix

Abbreviation

app	Application software
AWS	Amazon Web Services
CS	Computer Science
CSS	Cascading Style Sheets
DGBL	Digital Game-Based Learning
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
HTML	Hypertext Markup Language
IaaS	Infrastructure as a Service
IDE	Integrated Development Environments
Java EE	Java Platform, Enterprise Edition
Java SE	Java Platform, Standard Edition
JDBC	Java Database Connectivity
JVM	Java virtual machine
JS	JavaScript
JSON	JavaScript Object Notation
JPA	Java Persistence API
NTNU	Norwegian University of Science and Technology
ORM	Object/Relational Mapping
OS	Operating System
PaaS	Platform as a service
RDBMS	Relational database management system
RWD	Responsive web design
RQ	Research Question
Sass	Syntactically Awesome Stylesheets
SSL	Secure Sockets Layer
SUS	System Usability Scale

SQL	Structured Query Language
SRS	Student Response System
TLS	Transport Layer Security
UI	User Interface
UX	User Experience
UML	Unified Modeling Language
WWW	World Wide Web

GameFlow

GameFlow model by Sweetser and Wyeth (Sweetser and Wyeth 2005):

Concentration: Games should require concentration and the player should be able to concentrate on the game.

- games should provide a lot of stimuli from different sources.
- games must provide stimuli that are worth attending to.
- games should quickly grab the players' attention and maintain their focus throughout the game.
- players shouldn't be burdened with tasks that don't feel important.
- games should have a high workload, while still being appropriate for the players' perceptual, cognitive, and memory limits.
- players should not be distracted from tasks that they want or need to concentrate on.

Challenge: Games should be sufficiently challenging and match the player's skill level.

- challenges in games must match the players' skill levels.
- games should provide different levels of challenge for different players.
- the level of challenge should increase as the player progresses through the game and increases their skill level.
- games should provide new challenges at an appropriate pace.

Player Skills: Games must support player skill development and mastery.

- players should be able to start playing the game without reading the manual.
- learning the game should not be boring, but be part of the fun.
- games should include online help so players don't need to exit the game.
- players should be taught to play the game through tutorials or initial levels that feel like playing the game.
- games should increase the players' skills at an appropriate pace as they progress through the game.
- players should be rewarded appropriately for their effort and skill development.
- game interfaces and mechanics should be easy to learn and use.

Control: Players should feel a sense of control over their actions in the game.

- players should feel a sense of control over their characters or units and their movements and interactions in the game world.
- players should feel a sense of control over the game interface and input devices.
- players should feel a sense of control over the game shell (starting, stopping, saving, etc.).
- players should not be able to make errors that are detrimental to the game and should be supported in recovering from errors.
- players should feel a sense of control and impact onto the game world (like their actions matter and they are shaping the game world).
- players should feel a sense of control over the actions that they take and the strategies that they use and that they are free to play the game the way that they want (not simply discovering actions and strategies planned by the game developers).

Clear Goals: Games should provide the player with clear goals at appropriate times.

- overriding goals should be clear and presented early.
- intermediate goals should be clear and presented at appropriate times.

Feedback: Players must receive appropriate feedback at appropriate times.

- players should receive feedback on progress toward their goals.
- players should receive immediate feedback on their actions.
- players should always know their status or score.

Immersion: Players should experience deep but effortless involvement in the game.

- players should become less aware of their surroundings.
- players should become less self-aware and less worried about everyday life or self.
- players should experience an altered sense of time.
- players should feel emotionally involved in the game.
- players should feel viscerally involved in the game.

Social Interaction: Games should support and create opportunities for social interaction.

- games should support competition and cooperation between players.
- games should support social interaction between players (chat, etc.).
- games should support social communities inside and outside the game.

For evaluation purpose, each of these criteria can be given a score between 0 - 5 where

0 - N/A

1 - Not at all

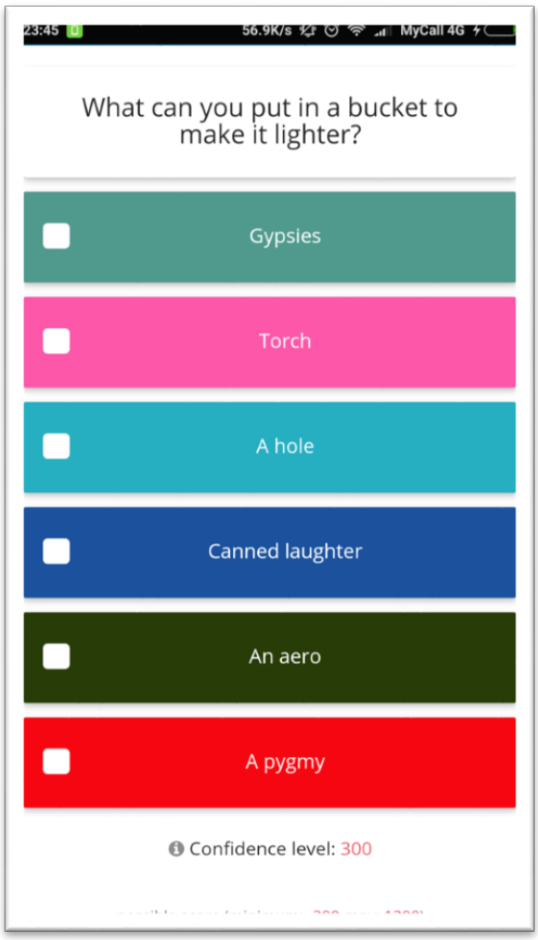
2 - Below average

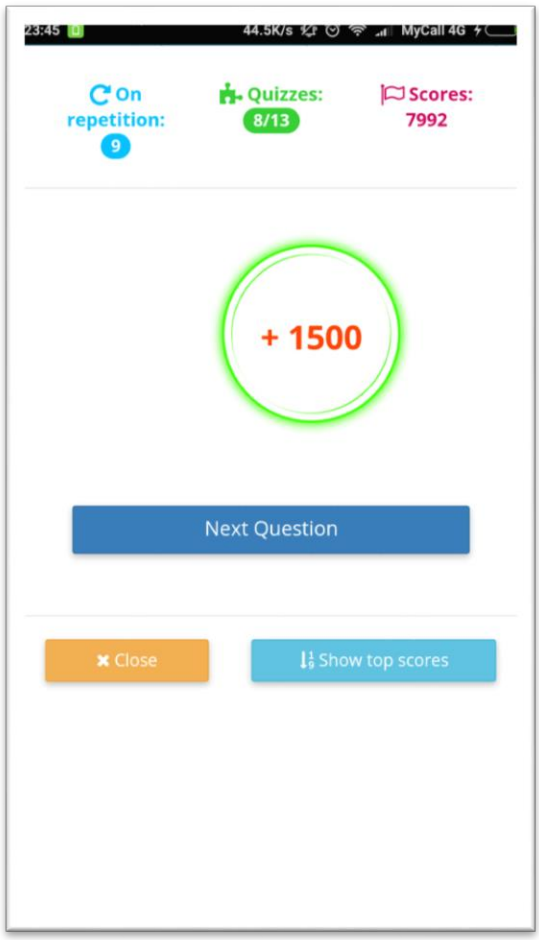
3 - Average

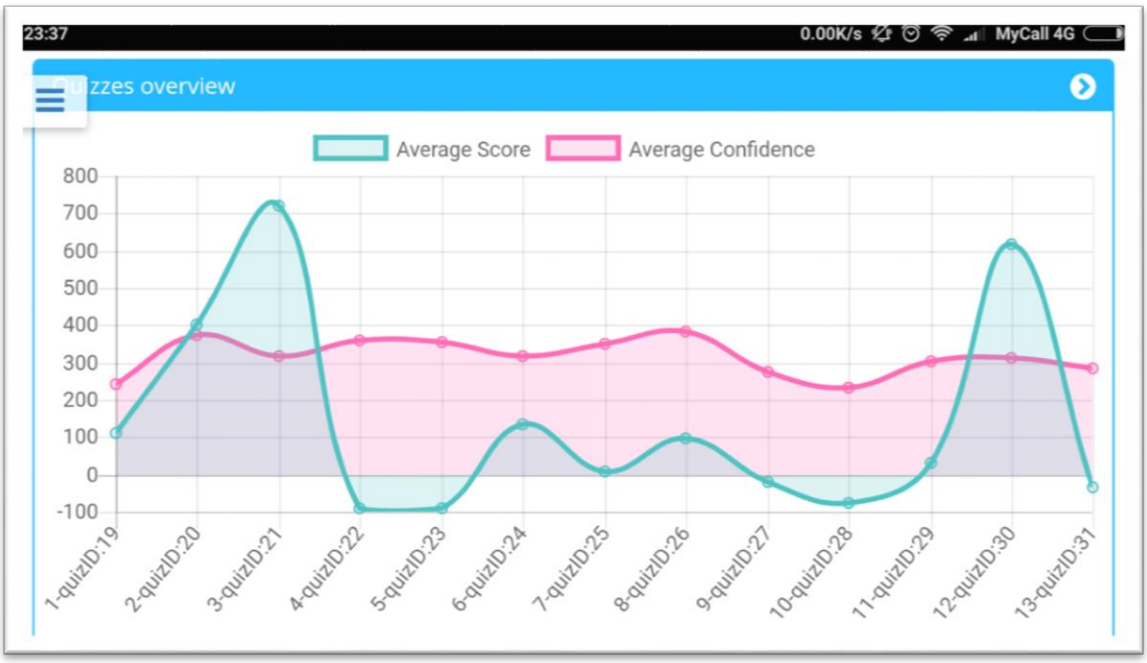
4 - Above average

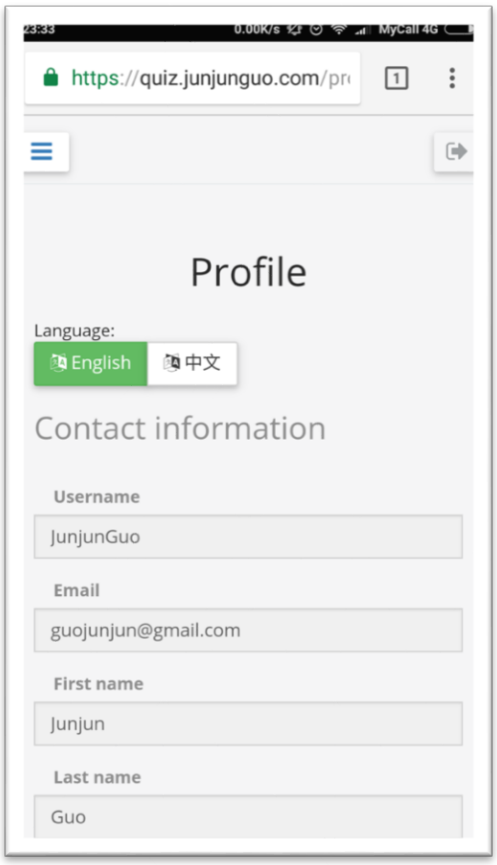
5 - Well done

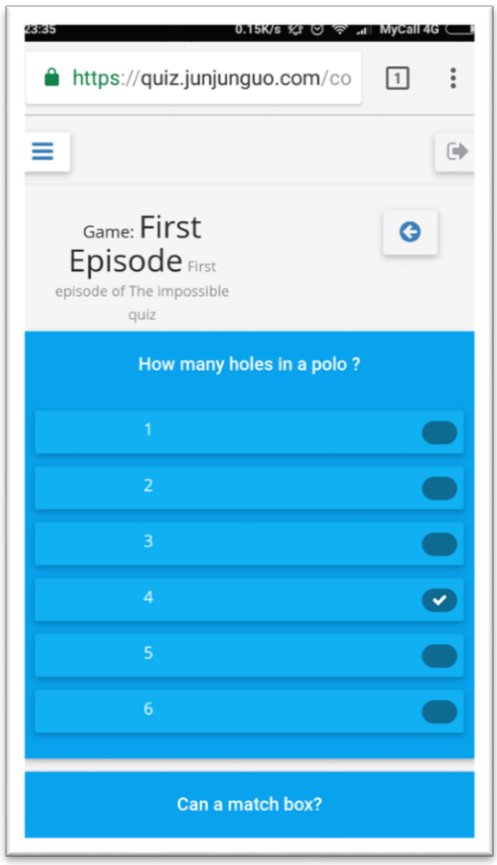
Quiz Class UI: Mobile View

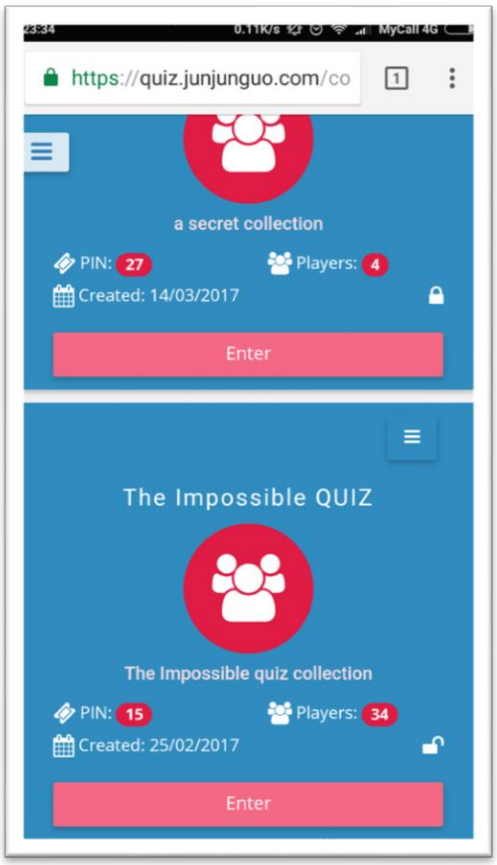




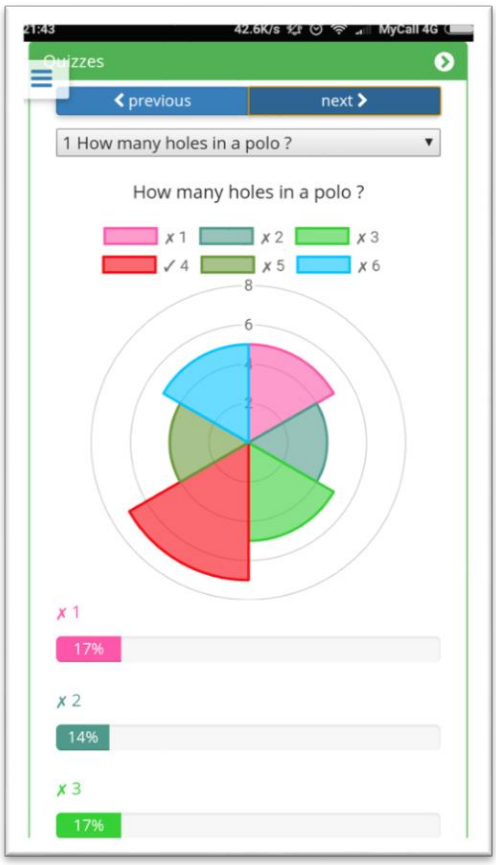


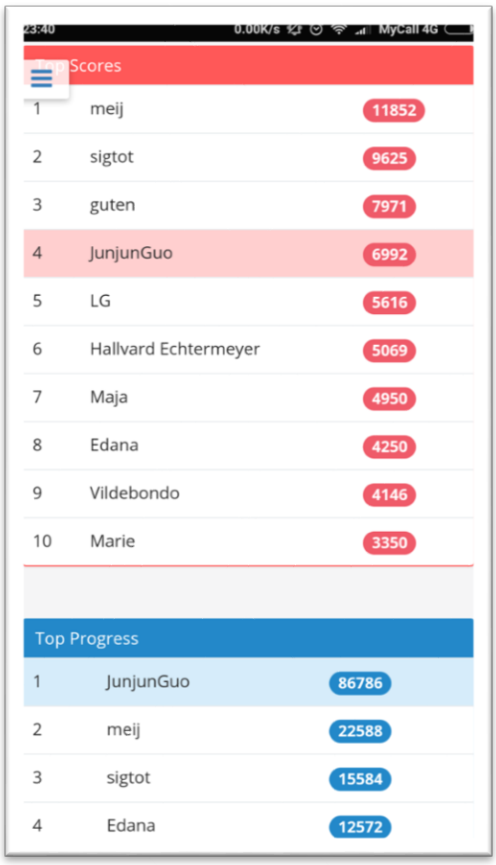

https://quiz.junjunguo.com/pr
Profile
Language:
English
中文
Contact information
Username
JunjunGuo
Email
guojunjun@gmail.com
First name
Junjun
Last name
Guo


https://quiz.junjunguo.com/co
Game: First Episode
First episode of The impossible quiz
How many holes in a polo ?
1
2
3
4
5
6
Can a match box?

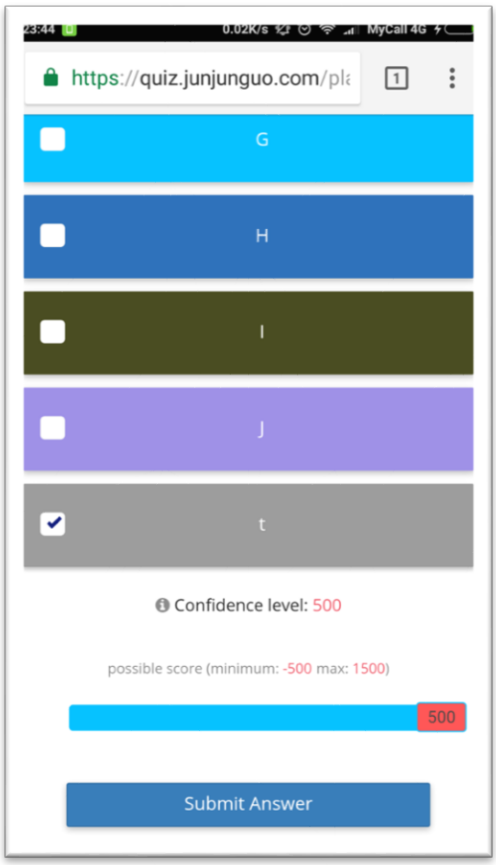

https://quiz.junjunguo.com/co
a secret collection
PIN: 27
Players: 4
Created: 14/03/2017
Enter
The Impossible QUIZ
The Impossible quiz collection
PIN: 15
Players: 34
Created: 25/02/2017
Enter

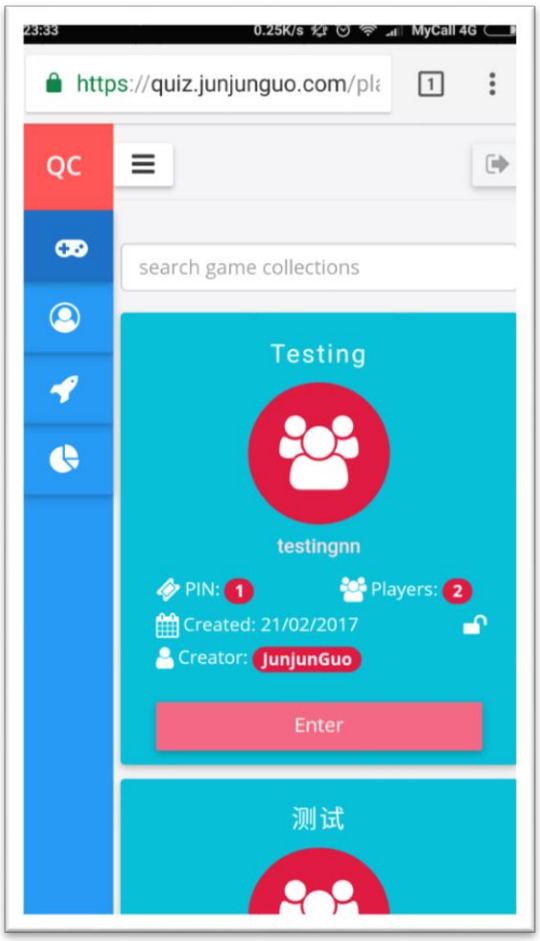

https://quiz.junjunguo.com/an
First Episode
Top List
Quizzes overview
Average Score
Average Confidence
800
600
400
200
0
-200
1-quizID:19
3-quizID:21
5-quizID:23
7-quizID:25
9-quizID:27
11-quizID:29
13-quizID:31
Quizzes
Time View
Bar View
Bubble View

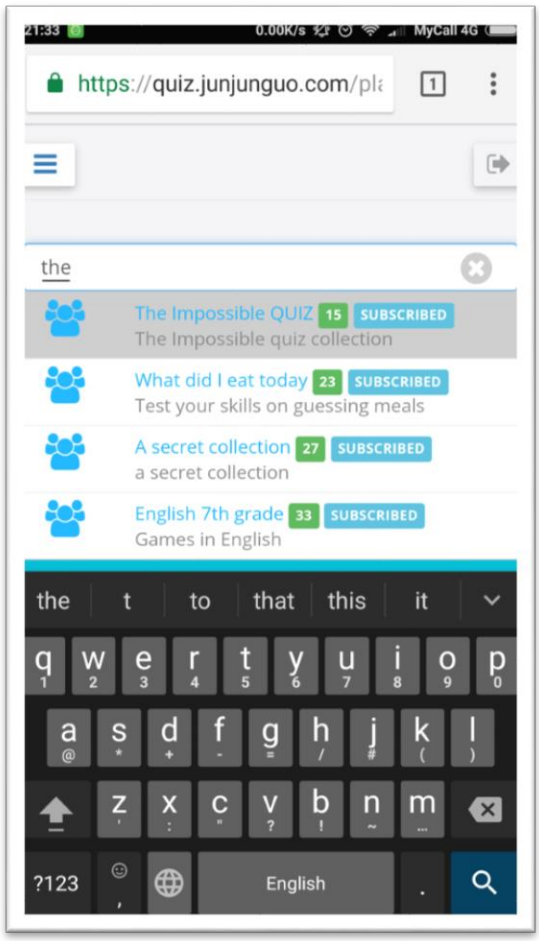

Quizzes
previous
next
1 How many holes in a polo ?
How many holes in a polo ?
x 1
x 2
x 3
✓ 4
x 5
x 6
x 1
17%
x 2
14%
x 3
17%

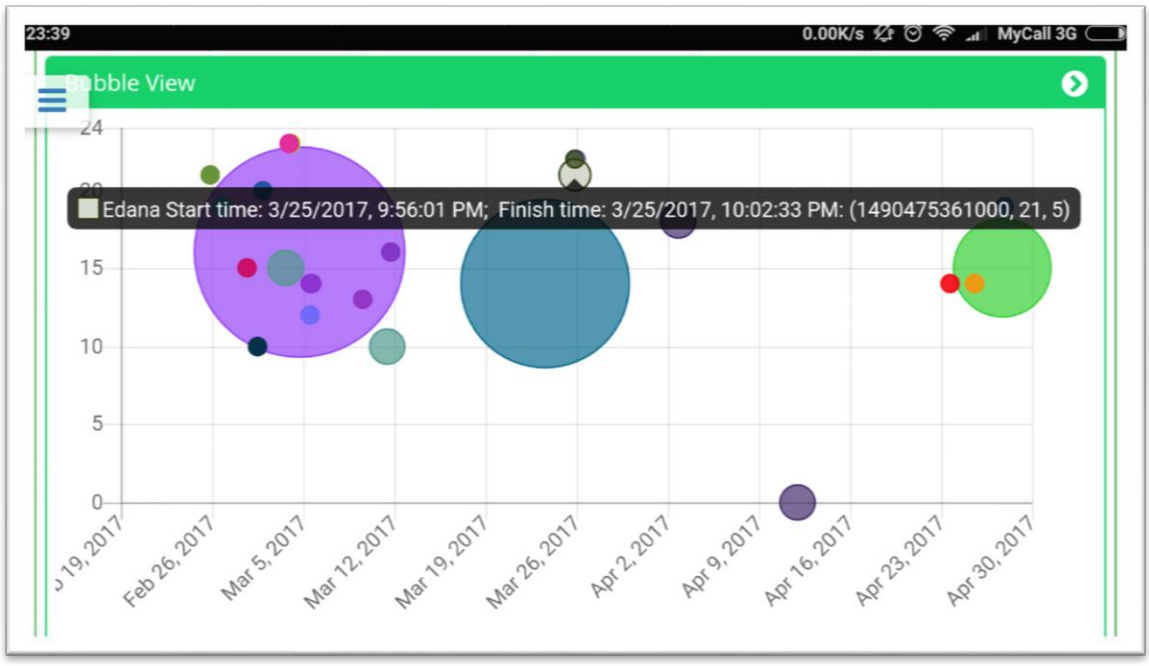

Top Scores
1 meij 11852
2 sigtot 9625
3 guten 7971
4 JunjunGuo 6992
5 LG 5616
6 Hallvard Echtermeyer 5069
7 Maja 4950
8 Edana 4250
9 Vildebondo 4146
10 Marie 3350
Top Progress
1 JunjunGuo 86786
2 meij 22588
3 sigtot 15584
4 Edana 12572

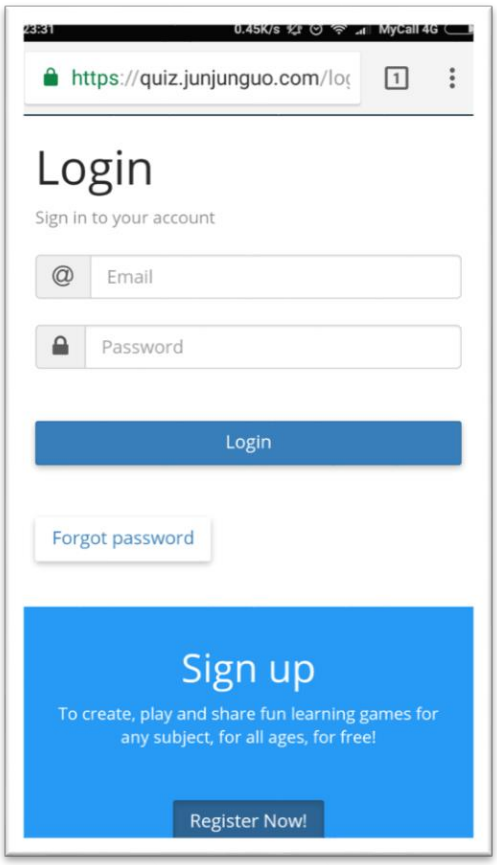

search game collections
First Episode
First episode of The impossible quiz
4.0 /5 4
Created: 25/02/2017
Quizzes: 6/13
Scores: 6992
On repetition: 9
Start time: 25/02/2017 20:00
Finish time: 20/07/2017 11:00
Show Ranking
Continue

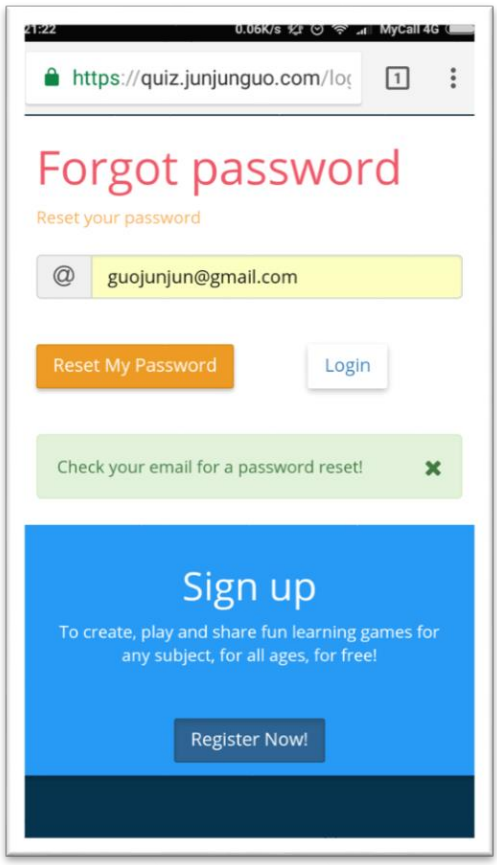

https://quiz.junjunguo.com/pl
G
H
I
J
t
Confidence level: 500
possible score (minimum: -500 max: 1500)
500
Submit Answer

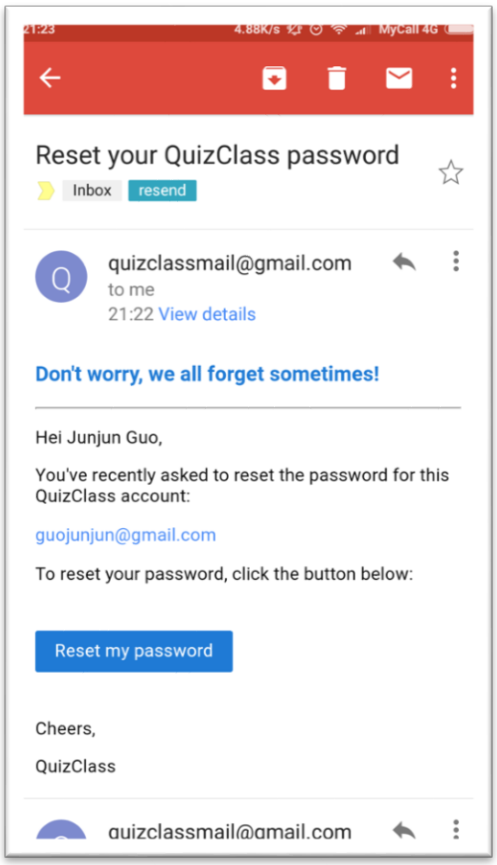

https://quiz.junjunguo.com/pl
QC
search game collections
Testing
testingnn
PIN: 1
Players: 2
Created: 21/02/2017
Creator: JunjunGuo
Enter
测试

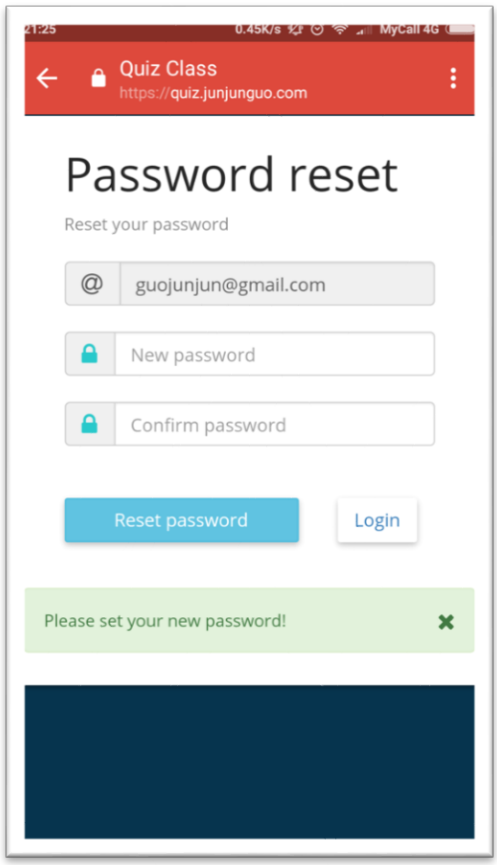

https://quiz.junjunguo.com/pl
the
The Impossible QUIZ 15 SUBSCRIBED
The Impossible quiz collection
What did I eat today 23 SUBSCRIBED
Test your skills on guessing meals
A secret collection 27 SUBSCRIBED
a secret collection
English 7th grade 33 SUBSCRIBED
Games in English
English

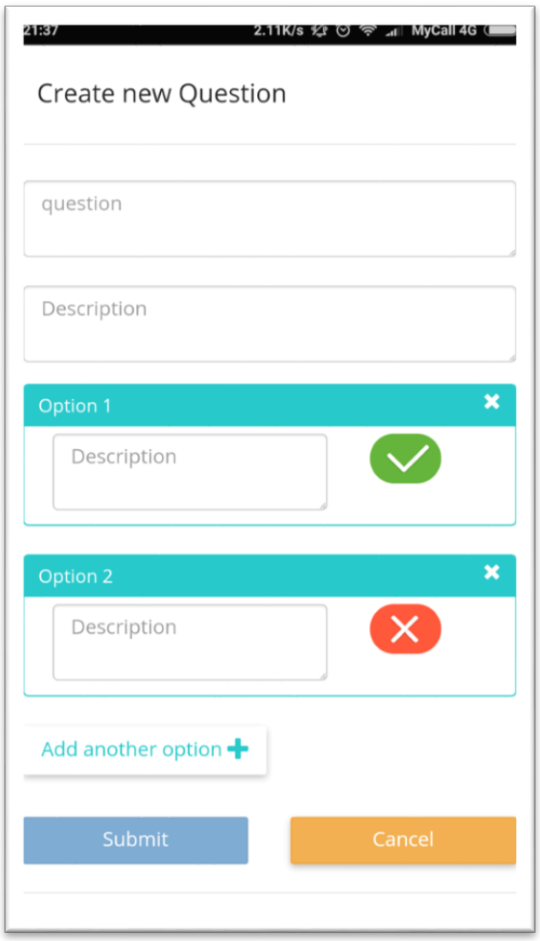

Bubble View
Edana Start time: 3/25/2017, 9:56:01 PM; Finish time: 3/25/2017, 10:02:33 PM: (1490475361000, 21, 5)
24
15
10
5
0
Feb 26, 2017
Mar 5, 2017
Mar 12, 2017
Mar 19, 2017
Mar 26, 2017
Apr 2, 2017
Apr 9, 2017
Apr 16, 2017
Apr 23, 2017
Apr 30, 2017

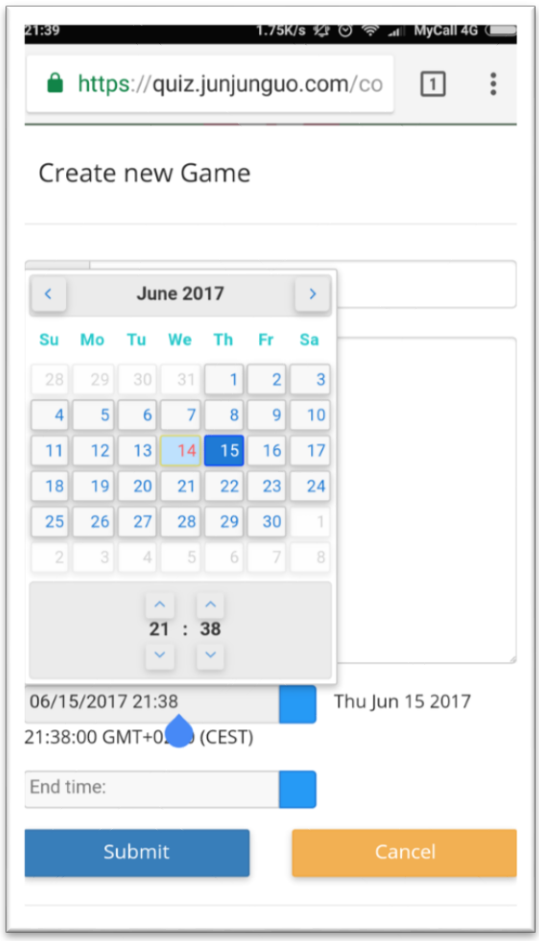

https://quiz.junjunguo.com/log
Login
Sign in to your account
Email
Password
Login
Forgot password
Sign up
To create, play and share fun learning games for any subject, for all ages, for free!
Register Now!

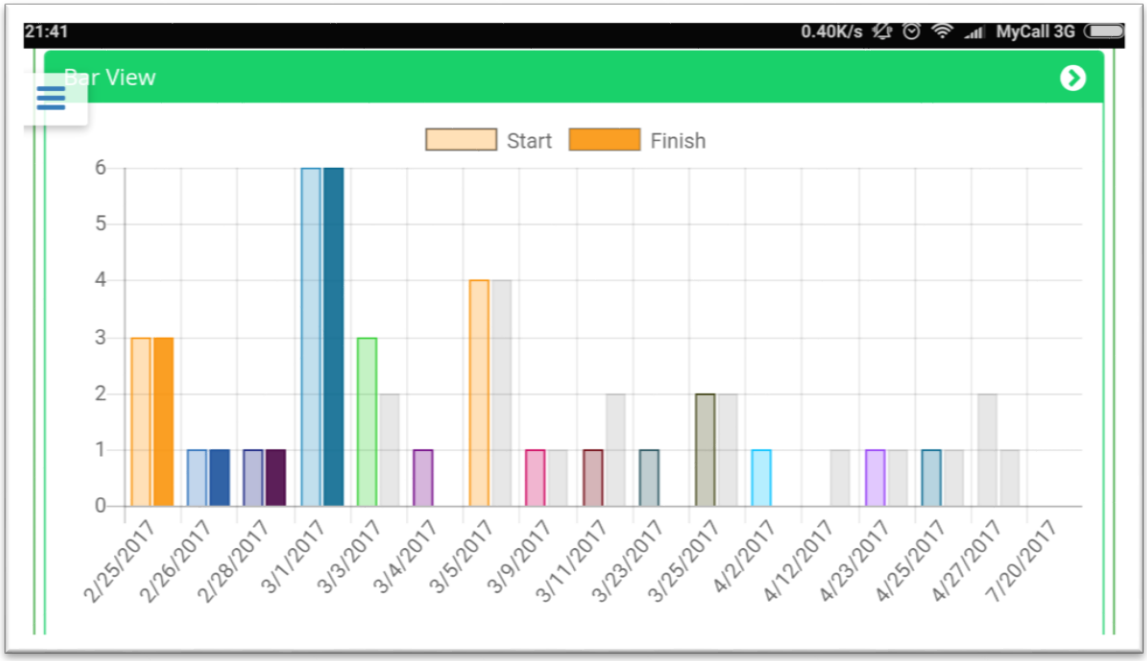

https://quiz.junjunguo.com/log
Forgot password
Reset your password
guojunjun@gmail.com
Reset My Password
Login
Check your email for a password reset!
Sign up
To create, play and share fun learning games for any subject, for all ages, for free!
Register Now!


Reset your QuizClass password
Inbox
resend
quizclassmail@gmail.com
to me
21:22 View details
Don't worry, we all forget sometimes!
Hei Junjun Guo,
You've recently asked to reset the password for this QuizClass account:
guojunjun@gmail.com
To reset your password, click the button below:
Reset my password
Cheers,
QuizClass


Quiz Class
https://quiz.junjunguo.com
Password reset
Reset your password
guojunjun@gmail.com
New password
Confirm password
Reset password
Login
Please set your new password!


Create new Question
question
Description
Option 1
Description
Option 2
Description
Add another option
Submit
Cancel


https://quiz.junjunguo.com/co
Create new Game
June 2017
Su Mo Tu We Th Fr Sa
21 : 38
06/15/2017 21:38
Thu Jun 15 2017
End time:
Submit
Cancel


Bar View
Start
Finish
6
5
4
3
2
1
0
2/25/2017
2/26/2017
2/28/2017
3/1/2017
3/3/2017
3/4/2017
3/5/2017
3/9/2017
3/11/2017
3/23/2017
3/25/2017
4/2/2017
4/12/2017
4/23/2017
4/25/2017
4/27/2017
7/20/2017

Quiz Class Questionnaire

Age: _____ **Gender**: ☐ male ☐ female

There are no right or wrong answers, just answer as accurately as possible. If you are strongly agreed with the statement, check on 5; if you are strongly disagreed with the statement, check on 1, otherwise, find the number between 1 and 5 that best describes you.

		1	2	3	4	5	
1. It was fun to compete against others	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
2. I pay more attention on doing the assignment because of the app	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
3. I was engaged while playing	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
4. I was more positive towards topic after playing the game	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
5. I found the app made me learn more	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
6. I found the app made doing assignment more fun	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
7. I found the confidence level made the game more exciting	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
8. The top early list encourages me to start doing assignment early	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
9. The top score list encourages me to do well in assignment	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
10. The progress list encourages me to repeat the assignment	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
11. The app increases my motivation for learning/doing the assignment	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
12. I wish this app could be used for more of my assignments	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

System Usability Scale

		1	2	3	4	5	
1. I think that I would like to use this system frequently	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
2. I found the system unnecessarily complex	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
3. I thought the system was easy to use	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
4. I think that I would need the support of a technical person to be able to use this system	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
5. I found the various functions in this system were well integrated	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
6. I thought there was too much inconsistency in this system	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
7. I would imagine that most people would learn to use this system very quickly	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
8. I found the system very cumbersome to use	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
9. I felt very confident using the system	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree
10. I needed to learn a lot of things before I could get going with this system	Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

Optional questions

1. If you could change one thing about the app what would it be and why?

2. What do you like the best of the app and why?

Table: Questionnaire result

#	Question	Strongly disagree 1	disagree 2	neutral 3	agree 4	Strongly agree 5	Average
1	It was fun to compete against others	2.6%	14.1%	21.8%	42.3%	19.2%	3.62
2	I pay more attention on doing the assignment because of the app	5.1%	10.3%	50%	24.4%	10.3%	3.24
3	I was engaged while playing	5.1%	10.3%	23.1%	42.3%	19.2%	3.60
4	I was more positive towards topic after playing the game	3.8%	15.4%	35.9%	26.9%	17.9%	3.40
5	I found the app made me learn more	11.5%	14.1%	32.1%	26.9%	15.4%	3.21
6	I found the app made doing assignment more fun	2.6%	9%	21.8%	52.6%	14.1%	3.67
7	I found the confidence level made the game more exciting	7.7%	12.8%	25.6%	25.6%	28.2%	3.54
8	The top early list encourages me to start doing assignment early	14.1%	19.2%	43.6%	14.1%	9%	2.85
9	The top score list encourages me to do well in assignment	7.7%	5.1%	24.4%	34.6%	28.2%	3.71
10	The progress list encourages me to repeat the assignment	5.1%	16.7%	28.2%	30.8%	19.2%	3.42
11	The app increases my motivation for learning/doing the assignment	6.4%	7.7%	30.8%	39.7%	15.4%	3.50
12	I wish this app could be used for more of my assignments	6.4%	7.7%	28.2%	37.2%	20.5%	3.58

Table 16 Questionnaire result: Usefulness of Quiz Class - detailed