

**ORANGECODE: WEBSITE TO LEARN
CODING VIA GAMING**

NASRUL ARIF BIN ZAKRIA

**BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING) WITH HONOURS**

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : NASRUL ARIF BIN ZAKRIA

Date of Birth : 06 AUGUST 1996

Title : ORANGECODE : WEBSITE TO LEARN
CODING VIA GAMING

Academic Session : _____

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)

960806-06-5603
New IC/Passport Number
Date:



(Supervisor's Signature)


AbdulRahman
Name of Supervisor
Date: 07/01/2019



Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in



(Supervisor's Signature)

Full Name : DR. ABDULRAHMAN AHMED MOHAMMED AL SEWARI
Position : SENIOR LECTURER
FACULTY OF COMPUTING & COMMUNICATIONS
SOFTWARE ENGINEERING
Date : UNIVERSITI MALAYSIA PAHANG
LEBUHRAJA UTARA 26100 GAMBANG,
KUANTAN, PAHANG.
TEL: 09-549 2244 FAX: 09-549 2144



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in blue ink, appearing to read 'Arif', written over a horizontal line.

(Student's Signature)

Full Name : NASRUL ARIF BIN ZAKRIA

ID Number : CB15060

Date : 07/01/2019

GAMIFICATION AND VISUALIZATION OF C PROGRAMMING VIA E-
LEARNING

NASRUL ARIF BIN ZAKRIA

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor Of Computer Science (Software Engineering) With Honours

Faculty of Computer Systems and Software Engineering
UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

ACKNOWLEDGEMENTS

First thing first, I would like to express my thankfulness to Allah S.W.T for every single gift and strength that He had given to me throughout the entire development of this thesis. Plus, I would like to put my appreciation to all the people who either directly or indirectly assisted me in the completion of this report. Massive thanks to my supervisor, Dr. Abdulrahmen Ahmad Al-Sewari who has been putting a lot of efforts in supervising and guiding all my doing regarding this report. It would be impossible to complete this report in a good quality without his supervision. Next, I would like to express my credit to my whole family especially my parent, Zakria Bin Mat Arif and Normala Binti Abu Talib who's has been rising me up and motivating me from the beginning of the report until its completion. Then, big thanks to all my friends who has been giving me recommendations and ideas towards the completion of this project. Last but not least, I am also giving my appreciations to the evaluators that helped and advised me regarding my presentation of this project.

ABSTRAK

Belajar untuk mengaturlcaraan adalah sangat sukar tanpa seseorang yang boleh menerangkan atau menggambarkannya. Terutamanya terhadap seseorang yang sememangnya tidak pernah tahu sebarang berkenaan pengaturlcaraan. Hasil ujikaji mendapati terdapat terdapat sebahagian orang yang mengalami kesukaran mempelajari sesuatu tanpa mendapat gambaran visual berkenaannya. Jadi, projek ini akan menerangkan tentang pembinaan OrangeCode dan aplikasinya. OrangeCode merupakan sebuah platform atas talian berdasarkan laman sesawang untuk kegunaan bagi memahami asas-asas pengaturlcaraan bermula daripada mempelajari pembolehubah-pembolehubah, pernyataan bersyarat, gelungan dan banyak lagi. OrangeCode boleh digunakan sesiapa sahaja yang ingin mempelajari tentang pengaturlcaraan daripada awal termasuklah kanak-kanak. Projek ini melibatkan kefahaman konsep pengaturlcaraan tanpa menspeksifikasikan kepada mana-mana bahasa pengaturlcaraan. Projek ini dibina dibawah dasar model proses perisian *Rapid Application Development (RAD)*. OrangeCode menggunakan pendekatan gamifikasi serta visualisasi untuk menanam kefahaman konsep-konsep berkenaan kod kepada pelajar baharu. Pendekatan ini sangat bersesuaian dengan trend dunia sekarang dimana berlakunya reformasi permainan video dikalangan belia. Dengan kewujudan teknologi terbaru seperti komputer dan peranti mudah alih yang berkuasa tinggi, permainan video telah menjadi sebahagian daripada rutin kehidupan seharian seseorang terutamanya golongan belia. Golongan belia menghabiskan waktu hariannya dengan permainan video apabila mereka mempunyai waktu lapang. Oleh itu, strateginya adalah untuk menukarkan konsep pengetahuan kepada permainan video agar golongan belia dapat belajar sambil bermain permainan video tersebut. Atas hasil kepada projek ini, para pelajar kini lebih berminat untuk belajar dan memahami konsep pengaturlcaraan.

ABSTRACT

Learning to code is definitely difficult without anyone explaining or visualizing it. Especially for a person who have never know any basic of coding. Studies show that there are certain person who can hardly learn something without the seeing the visual concept of it. Thus, this project will explain about the development of OrangeCode and their application. OrangeCode is a web-based online platform for the use of to understand the fundamental of coding starting from learning about variables, conditional statements, looping and more. OrangeCode can be used by any people who wanted to learn coding from the very beginning including kids. This project involves the knowledge of programming concept without referring to any specific programming language. This project was developed under the Rapid Application Development (RAD) software process model. It applied the approach of gamification and visualization to embed the concepts of coding among new learners. This approach is suitable with current world trends of video game reformation. With latest technology such as high performance computer and mobile devices, video games have become a part of lifestyle for a person especially the youths. Youths spend their day with video games whenever they have a free time. Therefore, the strategy is to convert knowledge concepts into video games so that the youths can learn while playing them. As a result from this project, students become even more interested to learn and understand programming concepts.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	Error! Bookmark not defined.
LIST OF FIGURES	Error! Bookmark not defined.
LIST OF SYMBOLS	xi
LIST OF ABBREVIATIONS	xii
CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND OF STUDY	1
1.2 PROBLEM STATEMENTS	2
1.3 AIM AND OBJECTIVES	2
1.4 SCOPES	3
1.5 SIGNIFICANCE	3
1.6 THESIS ORGANIZATION	3
CHAPTER 2 LITERATURE REVIEW	5
2.1 INTRODUCTION	5
2.2 GAMIFICATION	5
2.3 VISUALIZATION	5

2.4	EXISTING SYSTEMS	6
2.4.1	Manual Teaching Approach	6
2.4.2	Tutorials through Online Documentation & Videos	7
2.4.3	CodeCombat	8
2.5	COMPARISON OF EXISTING SYSTEMS / APPROACHES	8
2.6	TOOLS AND FRAMEWORKS	10
2.6.1	Unity	10
2.6.2	HTML5	11
2.6.3	Godot	12
2.6.4	Laravel Framework	13
2.7	SUMMARY	14
CHAPTER 3 METHODOLOGY		15
3.1	INTRODUCTION	15
3.2	METHODOLOGY	16
3.2.1	Requirement Planning Phase	17
3.2.2	User Design Phase	18
3.2.3	Construction Phase	19
3.2.4	Cutover Phase	19
3.3	HARDWARE & SOFTWARE REQUIREMENTS	20
3.4	GANTT CHART	21
3.5	TESTING PLAN	21
3.6	GENERAL ARCHITECTURE	21
3.6.1	Model	22
3.6.2	View	22
3.6.3	Controller	23

3.7	PACKAGE MODULE	23
3.7.1	Basic Programming Syllabus	23
3.7.2	Unity Engine	23
3.7.3	Laravel	23
3.8	Implementation	23
CHAPTER 4 RESULTS AND DISCUSSION		25
4.1	INTRODUCTION	25
4.2	SYSTEM DESIGN	26
4.3	SYSTEM IMPLEMENTATION	29
4.3.1	Development Environment	29
4.3.2	Laravel Framework - Setup and Installation	29
4.3.3	Laravel Framework - Creating Views	32
4.3.4	Laravel Framework - Creating Controllers	32
4.3.5	Laravel Framework - Creating Models	33
4.3.6	Laravel Framework - Creating Migration and Migrate	34
4.3.7	Laravel Framework - Routing	35
4.3.8	Laravel Framework - Retrieving Data with Eloquent	35
4.4	SYSTEM TESTING AND RESULTS	37
4.4.1	Unit Testing	38
A.	Authentication	38
B.	Profile	43
C.	Forum	46
D.	FAQ	50
E.	Classroom	53
F.	Administrator Task	55

4.4.2	System Testing	57
4.4.3	User Acceptance Testing	59
4.5	INTEGRATION OF UNITY3D, LARAVEL AND MYSQL	59
4.6	USER MANUAL	60
CHAPTER 5 CONCLUSION		61
5.1	Introduction	61
5.2	User Acceptance Test Discussion	61
5.3	System Aftermath	61
5.4	System Attributes	62
5.5	Commercialization	62
APPENDIX A TEST PLAN		63
APPENDIX B SYSTEM REQUIREMENT SPECIFICATIONS		66
APPENDIX C SOFTWARE DESIGN DOCUMENT		71
APPENDIX D USER ACCEPTANCE TESTING		72
APPENDIX E USER MANUAL		73
REFERENCES		74

LIST OF TABLES

Table 1.1 - Problem Statements	2
Table 2.1 - Comparison Table	10
Table 2.2 - Comparison between game development frameworks	13
Table 2.3 - Comparison between web development frameworks	15
Table 4.1 - Common Commands used in Laravel Framework	36
Table 4.2 - Authentication Decision Table	46
Table 4.3 - Authentication Test Specification	46
Table 4.4 - Profile Test Specification	49
Table 4.5 - Forum Test Specification	54
Table 4.6 - FAQ Test Specification	57
Table 4.7 - Classroom Test Specification	60
Table 4.8 - Administrator Tasks Test Specification	61
Table A.3.1 - Features to Be Tested	69

LIST OF FIGURES

Figure 2.1 - TutorialsPoint website tutorials on C Programming	8
Figure 2.2 - CodeCombat gameplay	9
Figure 2.3 - Unity Framework Logo	11
Figure 2.4 - HTML5 Framework Logo	12
Figure 2.5 - Godot Framework Logo	13
Figure 2.6 - Laravel Framework Logo	14
Figure 2.7 - CodeIgniter Framework Logo	15
Figure 3.1 - Rapid Application Development Model	18
Figure 3.2 - OrangeCode Context Diagram	19
Figure 3.3 - OrangeCode Use Case Diagram	20
Figure 3.4 - Gantt Chart Part 1	23
Figure 3.5 - Gantt Chart Part 2	24
Figure 3.6 - Gantt Chart Part 3	25
Figure 3.7 - Gantt Chart Part 4	26
Figure 3.8 - MVC Architecture Design	27
Figure 3.9 - OrangeCode Package Module	29
Figure 4.1 - OrangeCode MVC Design Part 1	32
Figure 4.2 - OrangeCode MVC Design Part 2	33
Figure 4.3 - OrangeCode MVC Design Part 3	33
Figure 4.4 - OrangeCode MVC Design Part 4	34
Figure 4.5 - Laravel Project Structure	37
Figure 4.6 - web.php File used to Route Controllers and Views together	38
Figure 4.7 - Forum Model	40
Figure 4.8 - Routing Examples	41
Figure 4.9 - Eloquent Implementation in Forum Controller	42
Figure 4.10 - The Login Authentication Interface	44
Figure 4.11 - The Registration Interface	45
Figure 4.12 - Updating the Display Image	49
Figure 4.13 - Forum User Interface	52
Figure 4.14 - Replying to a Post	53
Figure 4.15 - FAQ Interface	56
Figure 4.16 - Classroom Main Interface	59
Figure 4.17 - Illustration of Integrating Unity3D, MySQL and Laravel	65

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

UMP	University Malaysia Pahang
AR	Augmented Reality
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
RAD	Rapid Application Development
SDLC	Software Development Life Cycle
SRS	System Requirement Specification
MVC	Model View Controller

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

Most educational institutes that offer courses related to computer or machine use C programming language as their fundamental syllabus for their enrolled students. The reasons are that most known programming languages such as Java, C#, C++ or even JavaScript and PHP have the similar syntax notation with C language in most general statements (Aki, Güllü, & Kaplanoğlu, 2015). Even in University Malaysia Pahang (UMP) itself, we have been using the language C in the subject Programming Technique for a very long time to consult students registered in the course regarding to the basic of programming. This involved the learning method of lecture session to define the concept of programming and lab session as the practical application of the understanding. However, because of the immense level of abstraction of concepts in programming, students and teachers found that programming is somehow complicated to understand (Sáez-López, Román-González, & Vázquez-Cano, 2016).

To these concerns, the project named ‘OrangeCode: Website to Learn Coding via Gaming’ will be implemented specifically for beginners who are still trying to tame themselves into programming in order to improve their understandings and eventually produce positive impacts upon their feedback. The usage of games for learning and Gamification has a major different in which Gamification intended to create a game like experience through integrating elements of games with real life contexts or environments while the use of games for learning is like designing the game which is totally intended for educational purposes (Souza, Veado, Moreira, Figueiredo, & Costa, 2018). Though, it is also exposed in latest research that visualizing both problem-solving and knowledge-construction processes is indispensable to understand complicated problem-solving contexts (Wang, Wu, Kinshuk, Chen, & Spector, 2013).

1.2 PROBLEM STATEMENTS

There are several groups of students who took *Programming Technique* subject in Universiti Malaysia Pahang (UMP) claim that they have to repeat the subject due to failures. This can affect their studies for example they might need to extend their studies especially when the subject becomes a prerequisite to another advance subjects.

Next is the irrelevant answers in programming tests and application shows that students fail to understand the programming concepts itself. Some academicians report that large amount of students were unable to answers properly. As we know, programming are mostly involves logical thinking and problem solving. This will affect the university's performance in term of the average scores and future employability of those students.

Finally, many students were unable to catch up with the programming concepts taught by their lecturers in class. Taking the possibilities of students who come from different background and very inexperienced in programming, this can contribute to the cause of why student become uninterested to learn programming, affecting their study progress and grade.

1.3 AIM AND OBJECTIVES

The aim of this project is to design and implement a new learning approach for *Programming Technique* subject. To achieve this aim, the following objectives should be done:

- i. To investigate the implementation of the new approach in learning programming.
- ii. To design and implement a new learning perspective that involved games and visualization.
- iii. To evaluate the performance of the new approach in *Programming Technique* in several testing techniques.

1.4 SCOPES

i. Programming Syllabus

This project will cover general programming which is identifying variables and data types.

ii. Development Materials and Platforms

The development tools that will be used in this project are Unity3D with C# and will be integrated into a web site that can be accessed from any web browser from different platforms. The web site will be published in a web server while being developed using PHP Laravel framework and MySQL database as the storage.

1.5 SIGNIFICANCE

- i.** Can increase the number of students that passed the subject *Programming Technique* in UMP.
- ii.** As an aid to academicians in teaching programming subject to students.
- iii.** Provide a new platform for students around the world that interested in programming but having hardship to understand the concept.

1.6 THESIS ORGANIZATION

The thesis consists of an abstract and three distinct chapters. The first chapter will discuss upon the background as well as the basic understanding of the project which includes the problem statements, objectives, scopes, significant and the thesis organization.

Chapter two will focused on the literature review for the project. It will extract any existing system or experiment related to this project in detail.

Chapter three describe regarding on the methodology to be used that will ensure that this study will reach its milestones and its main objective throughout specific model.

Chapter four is all about the aftermath of the system implementation process where the testing and discussion will take place. The chapter explains throughly on the testing of the system from unit testing, integration testing, system testing and user acceptance test. Plus, the status whether the system really achieve its objectives and goals also been discussed here.

Conclusion is what the chapter five of this thesis is consists of. What is the resolution from the system that has been build and what potential improvements can be made from the system in the future.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter will provide and discuss regarding the literature review of current technology of Augmented Reality along with existing systems or approaches that already have been used before to teach programming. Plus, the technologies and tools to be used which are closely related to this study will also be dis.

2.2 GAMIFICATION

Gamification is referring to the use of real-world contexts of non-gaming purpose into application of game design elements that can improve human motivation and performance in that real-world context(Sailer, Hense, Mayr, & Mandl, 2016). Though, the terminology itself has been in discussion for many years among researchers. A simple analogy that can be use to define gamification is to convert theoretical concepts into game concepts. For instance, theoretical concepts for accounting can be converted into game and the players are able to understand it better as with the consistency of the game mechanic and storyline. Seaborn and Fels (2014) concluded that gamification approach can boost users motivation, enjoyment and engagement in non-gaming concepts that is far beyond the computer-mediated environments.

2.3 VISUALIZATION

Boredom among students in lectures in various courses such as Mathematics, Engineering even Science Computer would seldomly be created, especially when they are delivered in a long period or in certain time of the day(Salleh, 2011). Felder and Sivlerman (2015) suggested that there are many ways of learning among students which are by seeing and hearing; reflecting and acting; reasoning logically and intuitively;

memorizing and visualizing and drawing analogies and building mathematical models; steadily and in fits and starts. Many teaching lesson conducted has been proven to be not effective or bored. Thus, some approach has been discovered such as active teaching approach and visualization approach. Visualization is a teaching approach that involved the process of visualizing concepts into graphical ideas which are easily understand by the look of the eyes. This can greatly improve the understanding of the students as well as decreasing the rate of boredom in the session. In an experiment, it is concluded that the implementation of visualization in variables or programming concepts can assists in the learning session (Shi, Min, & Zhang, 2017).

2.4 EXISTING SYSTEMS

There are about three existing systems or approaches that have been identified to be interrelated to this study. First, the manual approach of teaching C Programming through lectures and lab exercises. Next is the teaching approach using website tutorial that provides step by step instructions and concepts explanation by documentation. Finally, the educational game that teaches how to code and programming, *CodeCombat*.

2.4.1 Manual Teaching Approach

The manual teaching approaches are the normal classes or lectures that were conducted for students that registered in programming subject. These approaches include the verbal explanations with demonstrations of how to code a program using desired programming language. Furthermore, this approach may also include lab sessions that give students the freedom to apply what they have learnt during the lectures through given case study or scenario. The most basic elements to conduct these approaches are a tutor and a student. Normally, these approaches were done with a a person that capable to give tutor and a group of students with the lecture environmental room.

2.4.2 Tutorials through Online Documentation & Videos

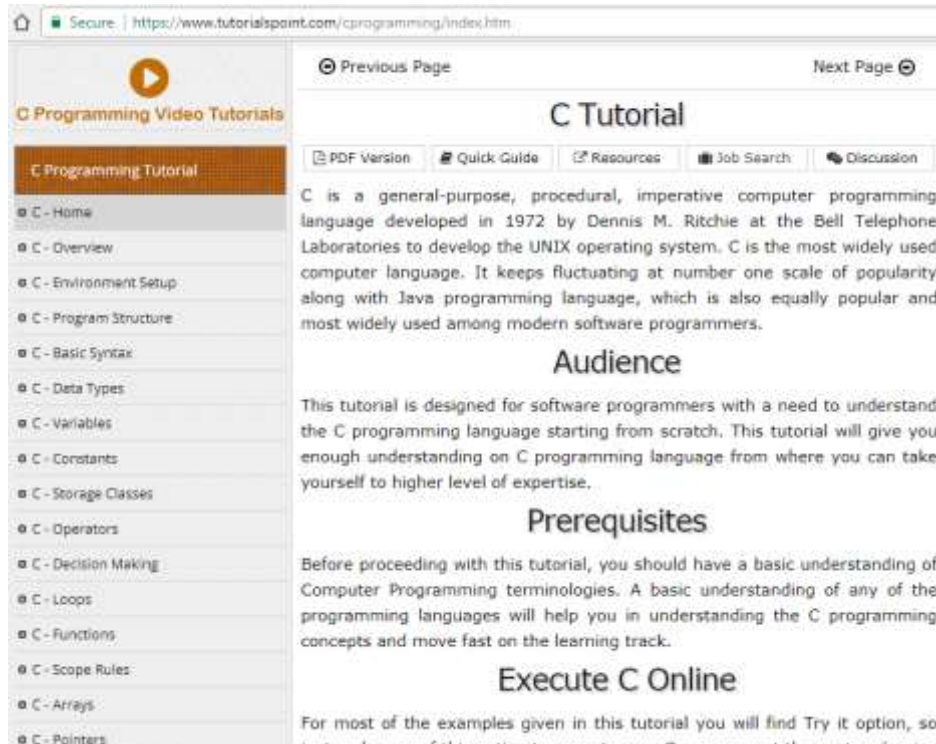


Figure 2.1 TutorialsPoint website tutorials on C Programming

This approach includes the step by step instructions from online documentations or videos created by expert community from all over the world as shown in figure 2.1. The types of instructions were given in detail along with the code examples and their outputs. *TutorialsPoint*, *W3Schools* and *Learn C* are the examples of websites that provide tutorials regarding programming and coding for the beginners via documentations. While website such as *Udemy.com* and *Lynda.com* offers online courses regarding computer related skills through video explanations and visualizations. A student can work on alone along the whole tutorial by following sequence of topics from the most fundamental topics and to the higher level topics.

2.4.3 CodeCombat



Figure 2.2 CodeCombat gameplay

CodeCombat is a website that provides alternative ways of learning programming by playing game. Learners may choose one of several languages that he or she wants to undergo throughout the game session. The game allows learners to write specific coding statements that will then be converted into commands for the hero to pass certain obstacles and to complete the course of the map. The game is free to play except for additional in game purchase. Learners can either play the game anonymously without saving feature or by registering and sign in to be able to play the game with the saving feature. The game starts from basic coding statements into more critical coding statements with high concepts understanding such as looping, arrays and functions. Although, some reviews indicate that CodeCombat is a bit of lacking in term applying what they have learnt. This may be due to the lack of explanations on every code statements during the game session as well as the fundamental concepts such as variables, assignments, data types and etc are not explained even in the earliest map.

2.5 COMPARISON OF EXISTING SYSTEMS / APPROACHES

Table 2.1 shows the comparison between three existing approaches in several elements. In the table is also provided the proposed system to be compared with.

Table 2.1 Comparison table for existing systems

	Manual Approach	Online Courses	CodeCombat	Proposed System
Platform	N/A	Any web browser	Any web browser	Any web browser, Android OS
Availability	Planned session only	All time	All time	All time
Cost	Based on package	Documentation are free; Some videos require payment	Free	Free
Live Tutor	Yes	Yes	No	No
Learning Pre-requisite	Nothing	Fundamental of Programming	Fundamental of Programming	Nothing
Network Access Required	No	Yes	Yes	Yes
Flexibility & Freedom	No	Yes	No	Yes
Understandability	Difficult to understand without visualization	Easy to understand by following step by step but no tutor to explain the detail.	Difficult since it assumes users have the basic understanding.	Easy with visualization, taught the most basic concept.0
Usability	N/A	Good	Good	Good
Expandability	No	Yes	Yes	Yes

When those existing approaches were compared, we can obviously see that *Online Courses* may have a big advantage except that it may require some payment to take the courses with great learning experience.

2.6 TOOLS AND FRAMEWORKS

There are about three tools or frameworks that have been identified as candidates to be used for the development of this project. The first one would be using Unity game engine development platform that uses C# language for the coding part. Next is by using web application framework which is HTML5 with the help of *Phaser.js* for the game along with the use of Phonegap to build the web application so that it will be able to run in mobile platform. Finally is Godot framework which uses similar language to Python and can be use to export in different platforms.

2.6.1 Unity



Figure 2.3 Unity Framework Logo

Unity is a game engine and a game development framework which has become the most popular to most game developer around the world. It can develop either 2D or 3D type of game. The framework provide free and paid version of pre-build assets that can be used to furnish the game environment. Unity provides game developers to build their game using their own preferred language as provided. For instance, Unity games can be developed using either C# or Javascript based on what the developer comfortable with. The interface may be a bit complicated for new user but it is easy to learn because the detailed guide and documentation provided along with helpful forum community. This framework can be used across many platforms for which it can also make game to be embedded into web browser.

After experiencing Unity3D course training, it is found that Unity3D could be the best candidate for the development of this project. When looked into the usability elements, Unity3D has been rated by the world as the easiest game development engine for beginners and advanced users. The existence of big Unity3D community along with support teams, documentations and trainings has contributed to the splendid usability status of this game engine. Unity3D also has AR feature for mobile which can be an ease to this project development.

2.6.2 HTML5



Figure 2.4 HTML5 Framework Logo

HTML5 is the latest version of markup language that mainly used to develop most web application. HTML5 uses the basic web scripts which are the Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS). Those three scripts must be integrated to build a complex application. Plus, HTML5 scripts especially CSS and JS have so many frameworks that can be used to help to build application with specific design or function such as using Bootstrap framework to design professional looking web application or even to implement gaming functionality by using Phaser Javascript framework. This framework can be used across platforms.

HTML5 frameworks are incredible in term of having wide range of plugins that can be installed and used to enhance the game being developed. It is also possible for advanced users to create their own plugins if they want. Although, most HTML5 frameworks require users to create their own game from total scratch, which could be a little bit harder for beginner to understand and implements further functionalities to their game. However, some experts have created frameworks with special GUI that can be easily used by beginners to create their first game.

2.6.3 Godot



Figure 2.5 Godot Framework Logo

An open source game engine that available in many platforms is Godot. Godot can be used to develop 2D or 3D games. It also came with built in animation tools to perform animation on many different assets. Games can be developed using Godot either in C#, C++ or GDScript which is almost similar to Python.

Godot is also a good game development engine to which most developers recommended other than Unity3D. But looking at the age of this engine, it was released in 2014 while Unity3D was released in 2006 and finally able to make stable release in December 2017.

Table 2.2 Comparion between game development frameworks

	Unity3D	HTML5	Godot
Language	C# or Javascript	HTML, JavaScript, CSS	C#, C++ or GDScript (Python)
Asset Store	Yes	No	Yes
Friendly GUI	Yes	Yes, using third-party plugin	Yes
Performance	Very Good	Good	Very Good
2D Support	Lacking but improving	Lacking	Good Support in Configuration
3D Support	Very good	Lacking	Not ready but possible
Tutorials & Community	Active	Active for several such as Phaser	Active

Among the three stated game development framework, Unity3D has been chosen. This is due to the improving features in the framework that can ease the developer. Godot seems to be very competitive with Unity3D with many beautiful attributes. Though, both are good and acceptable for game development.

2.6.4 Laravel Framework



Figure 2.6 Laravel Framework Logo

A simplified PHP Model-View-Controller (MVC) framework normally used to develop a website. Instead of scratching our head doing the backend process, Laravel provide the simplified PHP which can be applied directly in the views at ease.

2.6.5 Laravel Framework



Figure 2.7 CodeIgniter Framework Logo

A competitor in PHP framework is CodeIgniter. This framework is smaller than Laravel but that make it works faster than other framework. Similar to Laravel, CodeIgniter uses MVC pattern for development. Detail comparison can be seen in Table 2.3 below.

Table 2.3 Comparison between web development frameworks

	PHP Laravel	PHP CodeIgniter	Without Framework
Development	Model-View-	Model-View-	Freedom to create

Pattern	Controller	Controller	own pattern but is complicated to do so
Language	PHP	PHP	PHP
Code Portability	No	Yes	No
Performance	Good	Fast	Good
Flexibility	Not very strict	Some features require developers to follow guideline to prevent problems.	Very flexible but can lead to many problems.
Testing Complexity	Easy	Normal	Difficult
Third-party libraries	Yes	Yes	Yes but not much

Based on the comparison above, different framework has its own pros and cons that we cannot get rid of. Though, based on industry needs and simplicity, PHP Laravel framework has been chosen as the development tool for the website.

2.7 SUMMARY

This chapter discussed about how we can categories a system as Augmented Reality System. Also the comparison of Gamification and visualization approach with existing approaches can be a highly recommended since it has capability for most beginners. Gamification and visualization can be interconnected with AR since AR has the ability to provide different perspective to the users and assist in the understanding of the subject. To do so, Unity3D has the capability to develop a game from scratch along with the AR feature which can be implemented to furnish this project using Vuforia.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

Developing a software system is not as simple as ABC. It requires a lot of crucial plans and settings from different perspective in order to develop software within the constraints of time and budget. Thus, a suitable software development methodology or framework must be chosen and utilised from the beginning of the project development until its released version. This chapter mainly discussed about the methodology used in this project together with the representation of its requirements.

Different methodology follows different structures of development procedures. As for this project, **Rapid Application Development (RAD)** will be chosen as the process model. The method of RAD to be used is based on *James Martin*, in which it consists of **four** phases which are the Requirement Planning Phase, User Design Phase, Construction Phase and Cutover Phase. The details of those phases will be discussed more in the next segment.

3.2 METHODOLOGY

Rapid Application Development (RAD) is a process model that is suitable for rapid development of a project in a short time. As mentioned earlier, RAD that is applied in this project will be based on *James Martin* method that involves the **four** main phases as shown in Figure 3.1. This section will define in detail of each phase involved along with how it is applicable with the project.

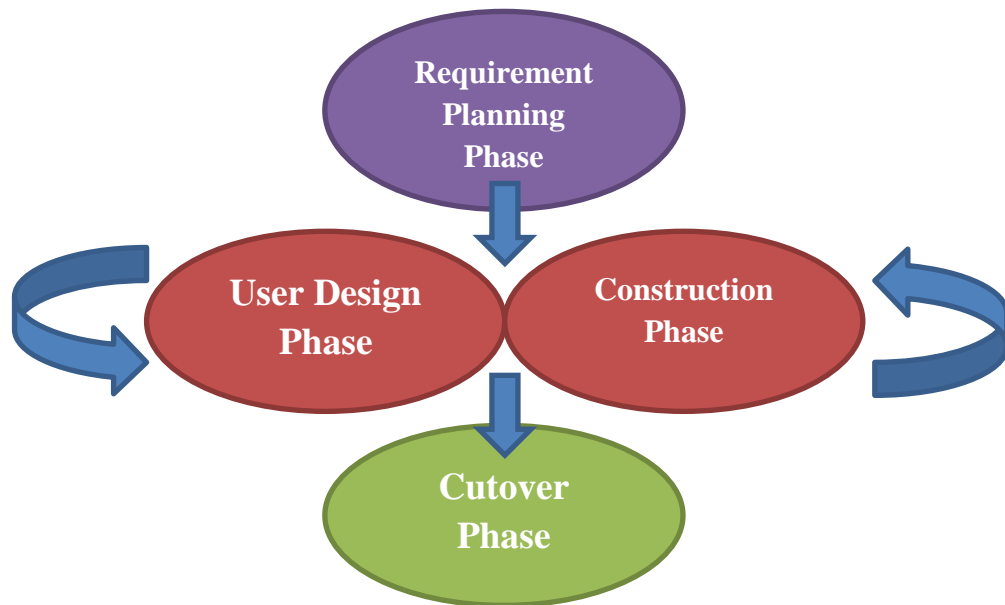


Figure 3.1 Rapid Application Development Model

3.2.1 Requirement Planning Phase

Requirement planning phase basically follows the first two elements in Systems Development Life Cycle (SDLC) which are the planning and analysis. In general, this phase involve the team to make an agreement on the project needs such as the scopes of the project, the requirements of the projects and the constraints of the project. Until the whole team agrees and given the authorization to proceed the project.

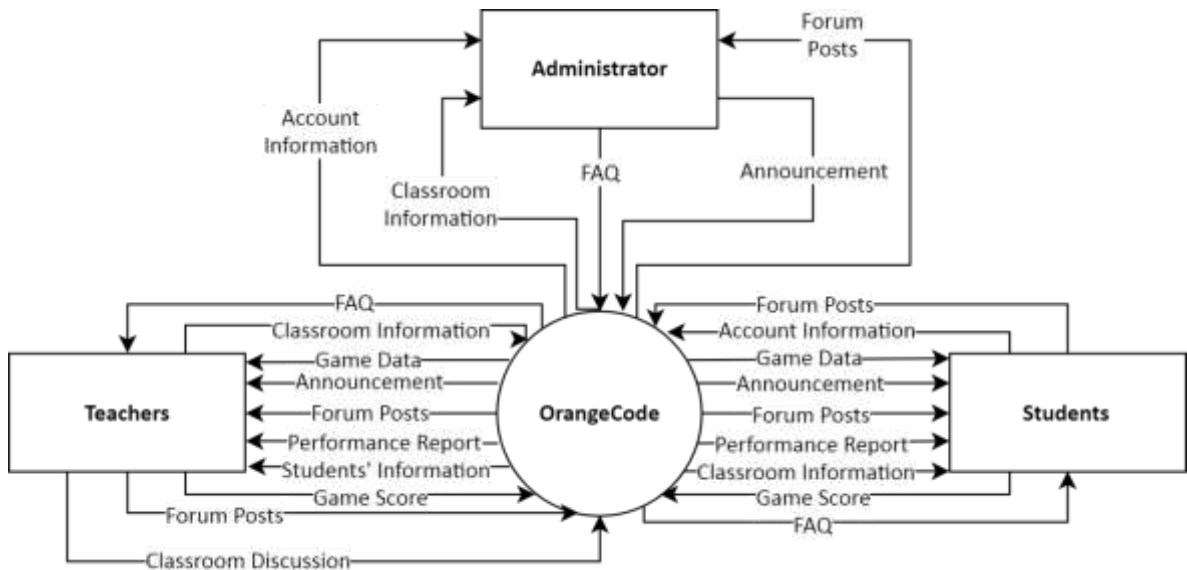


Figure 3.2 OrangeCode Context Diagram

To apply in this project, the first thing to ensure is to define the problem statements of the project, several questions may be asked such as what the problem is, why it needs to be solved, who will be involved, what is the scope of the problem and what is the constraints and risks of the project development. Figure 3.2 shows the context diagram of the system that exposes the overview of the system. Once this phase is considered cleared and completed, the project should proceed to the next phase.

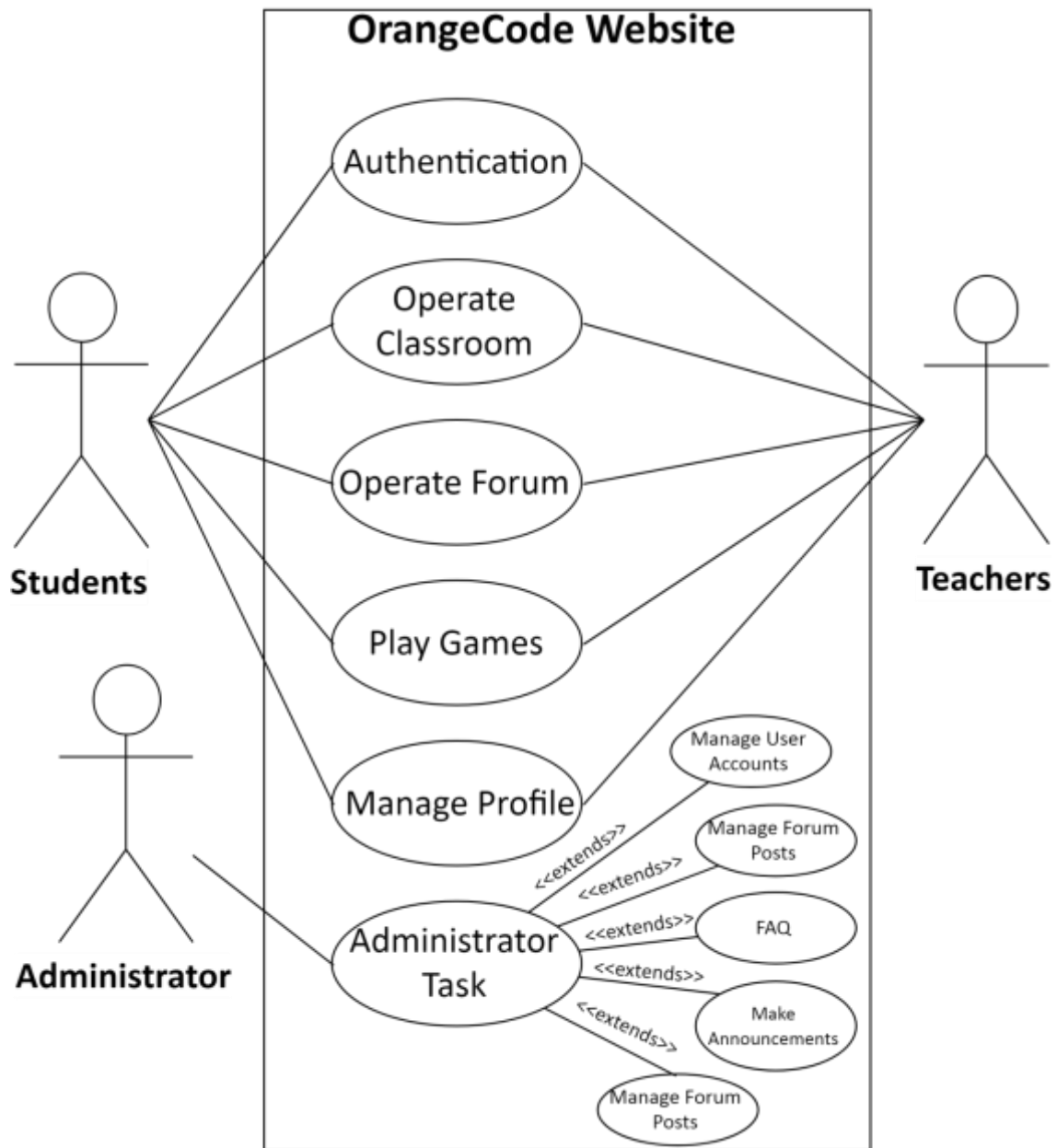


Figure 3.3 OrangeCode Use Case Diagram

Figure 3.3 shows the use case diagram for OrangeCode. The system consists of six main functionalities which are the user able to create account, create class, play games, generate performance report, post forum and manage general admin tasks. There are in total of four actors in the system. Those actors are the students, supervisors, individual learners and the administrator.

3.2.2 User Design Phase

User design phase is where the system will be designed based on the feedback of users. Though, this phase will be the paired with construction phase and can be repeatable

along with construction phase anytime if there is changes of the system design during in the middle construction phase as to meet the requirements.

Applying to this system, first step will be to get users (students) feedback on how they would like to be taught in programming as well as which chapters from *Programming Technique* syllabus that they were having the most problem with. This process is also called the user acceptance test. After certain period of construction phase and that part of the system is ready to be tested by users, users feedback will be recorded and the design of the system will be changed or maintained depends on those feedbacks. This process will be repeated until users are satisfied with the system. Document such as System Requirement Specification (SRS) and System Design Document (SDD) will be the artifacts created and modified from this phase.

3.2.3 Construction Phase

Construction phase also known as the development phase is where the coding and implementation process happen. As mentioned earlier construction phase usually paired with user design phase. This is to conform to the users' feedback made in user design phase so that users will be comfortable and satisfied with the system. If the system failed to get users' satisfaction, then the phase shall be looped back to user design phase. This phase can be repeated for as many times.

Construction phase of this system is where the core of the system which is the game, will be developed to satisfy users. The development will starts from user interface to the game mechanic of the game. Once the core of the system managed to be developed, the construction phase will then focused onto the web design and mobile application design in which will be main platform for users to use the system. Simultaneously, the system will be tested minimally through various testing technique.

3.2.4 Cutover Phase

Cutover Phase also called the deployment phase is when the system is finally achieved users' satisfaction and ready to be tested and published into real environment. This is the phase where the system will be tested carefully and precisely using several testing technique. Once the system is literally free from bugs and errors, then the system shall be deployed and published into the market.

As for the cutover phase of this proposed system, once the system have reached the users satisfaction, the system shall then be brought into the testing process in which the system will be tested throughly and meticulously. The planned testing technique will be discussed in detail at *Chapter 3.5 - Testing Plan*.

3.3 HARDWARE & SOFTWARE REQUIREMENTS

- Hardware Requirements
 - Complete Set of Desktop or Laptop Computer
 - Smartphone with Android Operating System
- Software Requirements
 - Laravel Framework Dependencies
 - Unity3D Game Development Engine
 - Web Server (XAMPP for localhost testing)
 - MySQL Server (XAMPP for localhost testing)
 - Visual Studio Code (VSCode)
 - Web Browsers (Chrome, Firefox, Internet Explorer, Edge)
 - Google Play Console

3.4 GANTT CHART

In this project, RAD has been used as the software process model. The gantt chart was design to have a requirement planning, three cycle of design and construction phase and finally a cutover phase.

Refer appendix B - Gantt Chart.

3.5 TESTING PLAN

The system to be designed shall require testing in order to maintain the quality and improve users' experiences. Thus, testing plan will define the basic planning and requirements of the testing to be conducted throughout the development of the system. The details of the system testing plan are explained in **Appendix A - Test Plan**.

3.6 GENERAL ARCHITECTURE

This subchapter explains the general architecture that will be used by the system. Technically, the system will be using Model-View-Controller (MVC) architecture.

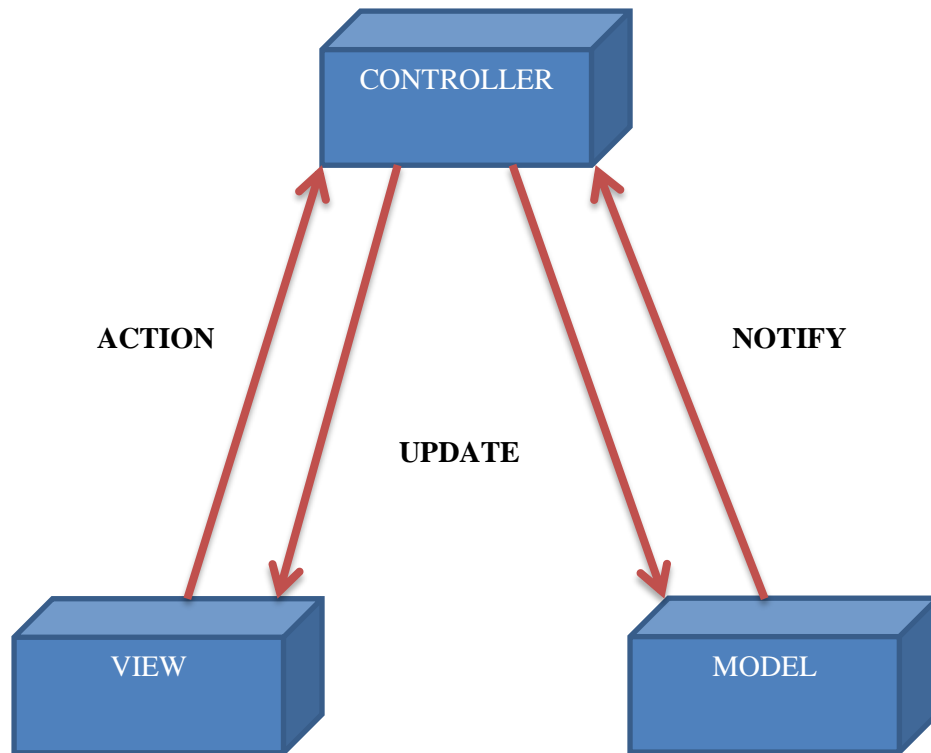


Figure 3.8 MVC Architecture Design

3.6.1 Model

Model is a layer in MVC architecture that plays as a reference to the database. In order to perform CRUD (Create, Read, Update, and Delete) functionalities, model of the table in the database is needed in the controller. When requests are sent to the model by the controller, the model will connect to the hosted database and perform the fetching process and return the values.

3.6.2 View

View act as presentation layer in MVC architecture where view is where the web interface and design is displayed to the users and able to be interacted. When users interacted with view and made requests, controller involved will be triggered depending on the function that the request is included in. For view to be connected to controller, route must be set up first between them. This will be discussed later in Chapter 4.

3.6.3 Controller

Controller is a layer where most functions and back-end processes are being focused. Controller consists of functions that will return a view along with data fetched from model if necessary. In MVC, controller acts as a middle medium for interacting view and model.

3.7 PACKAGE MODULE

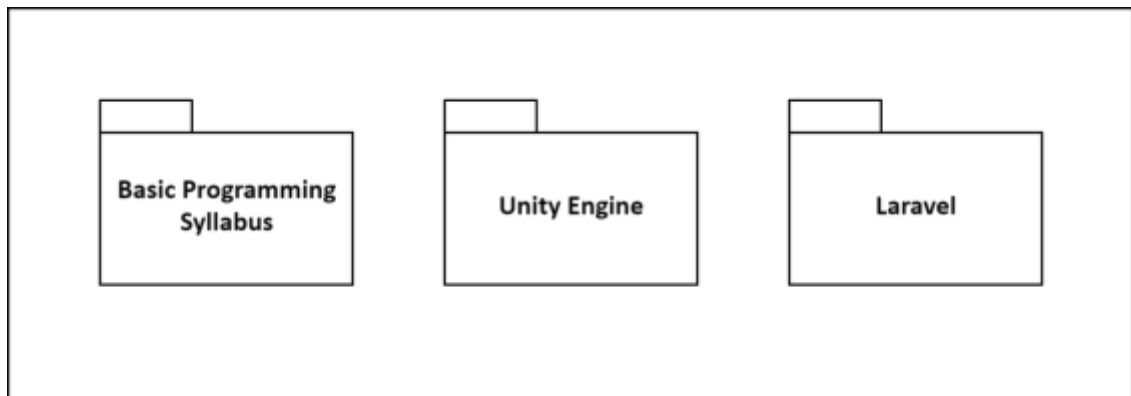


Figure 3.9 OrangeCode Package Module

3.7.1 Basic Programming Syllabus

This package consists of standard programming learning flow based on various sources such as tutorial sites, documentations and etc.

3.7.2 Unity Engine

This package contains the library and modules of Unity Engine. For example, physics, materials, light, audio, renderer and etc.

3.7.3 Laravel

This package consists of Laravel modules to be used in creating website from simplified PHP version.

3.8 Implementation

Using RAD, the implementation of this system development will be undergoing four main phases. The first phase which is the requirement planning shall start from the beginning of this project. By gathering requirements and preparing the context of the

system, the requirement planning phase shall achieve its completion state as the whole system overview is clarified. Requirements can be gathered by benchmarking most existing system available. Plus, the requirement

Next is the user design phase. The design of the system should be ready as it will be used in later phase. The design of the system will be made from most of users' feedback towards either existing system or what they expected to be in this system. This phase can be concurrent with construction phase as the system will be develop and tested regularly with the users to get their feedback along with the SRS and SDD as main reference to the development.

Construction phase is where the system shall be developed. The system will be developed using multiple softwares and languages such as Unity3D software will be focused on using JavaScript language as to script functions in the game class. While the webpage shall be developed using web application language which are the Hypertext Markup Language, JavaScript, Cascading Coding Style and Hypertext Preprocessor languages. As for the assets of the system such as the sound effects, image sprites, banner, logo and etc, they can actually be obtained from Unity3D free assets.

Finally, once the system has been developed to almost its completion and it satisfies the users, the system shall then be deployed and tested in the real environment which is in UMP itself. As mentioned in Chapter 1, one of the target users will be the students who currently are taking the subject Programming Technique or for the students who failed the subject and is repeting it. If the system received any bug reports then it shall then be documented and fixed depending on the priority and the risk level.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this chapter, we will discuss on the system design, implementation, testing and their results of OrangeCode. In order to develop the system, system design is important to ensure that the overall looks of the system are clear so that there would not be any ambiguity during the implementation phase. Once the implementation phase is somehow achieved its exit criteria, then the testing phase will take it place. This is to ensure that the system work as it should be. While the unit testing and integration testing will be run simultaneously during the implementation phase, system testing and user acceptance testing will done once the system development is completed. Should any errors occur, then the debugging and fixing process will be executed and regression testing will be executed for the changes done.

4.2 SYSTEM DESIGN

The architecture pattern used to make the system is Model-View-Controller or MVC architecture. Figure 4.1 until Figure 4.4 below show the MVC Design of OrangeCode. The detail of each subject of this design can be viewed in **Appendix D - Software Design Document**.

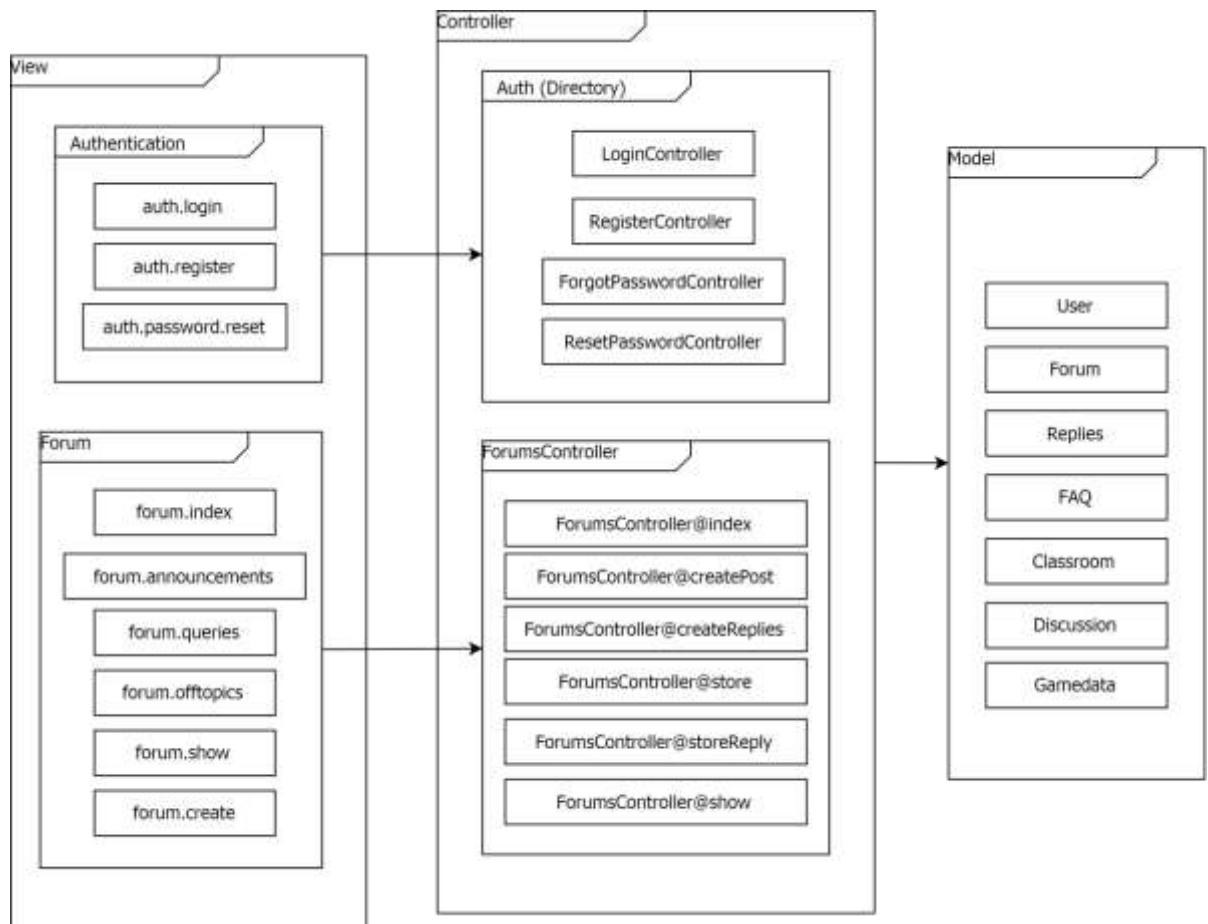


Figure 4.1 OrangeCode MVC Design Part 1

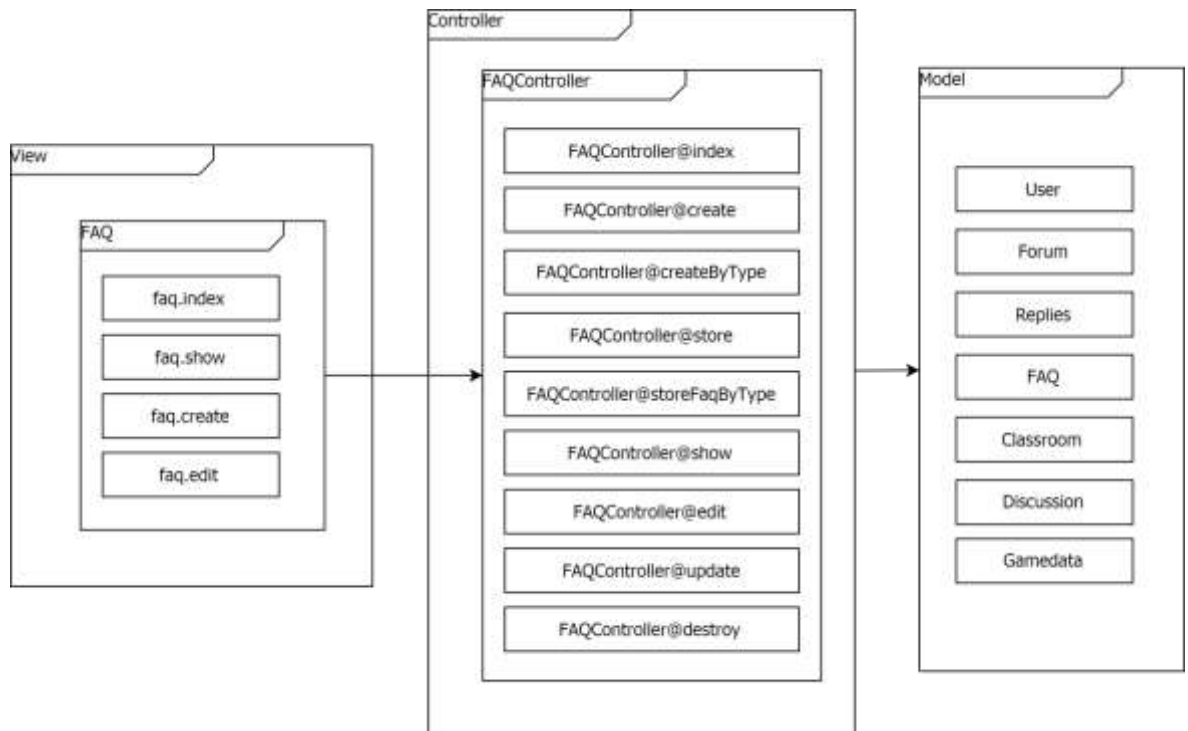


Figure 4.2 OrangeCode MVC Design Part 2

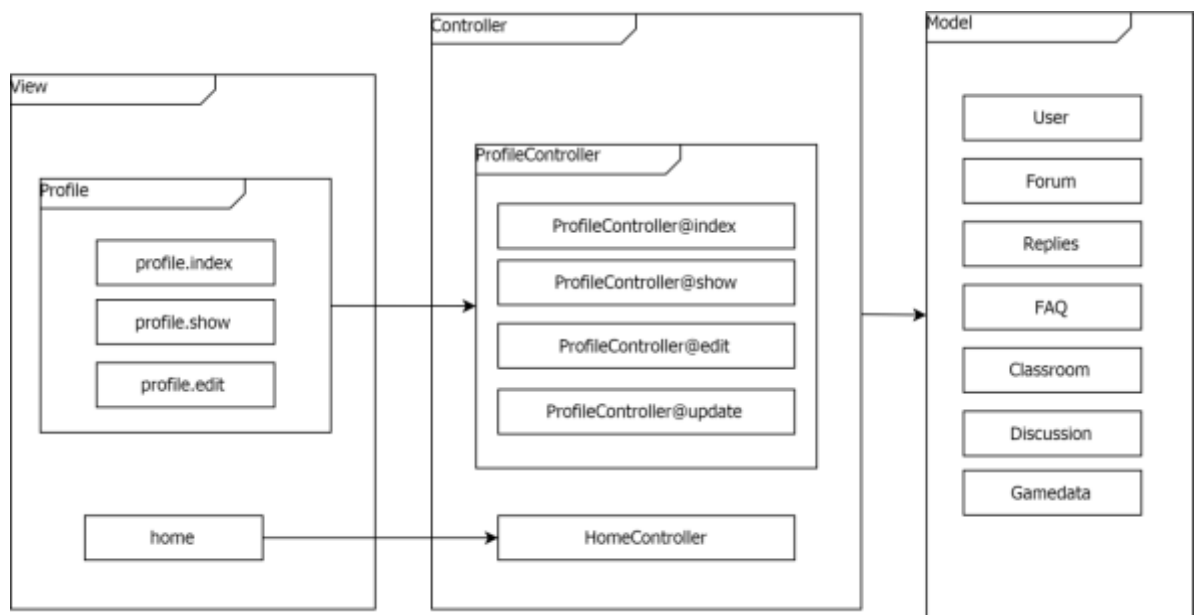


Figure 4.3 OrangeCode MVC Design Part 3

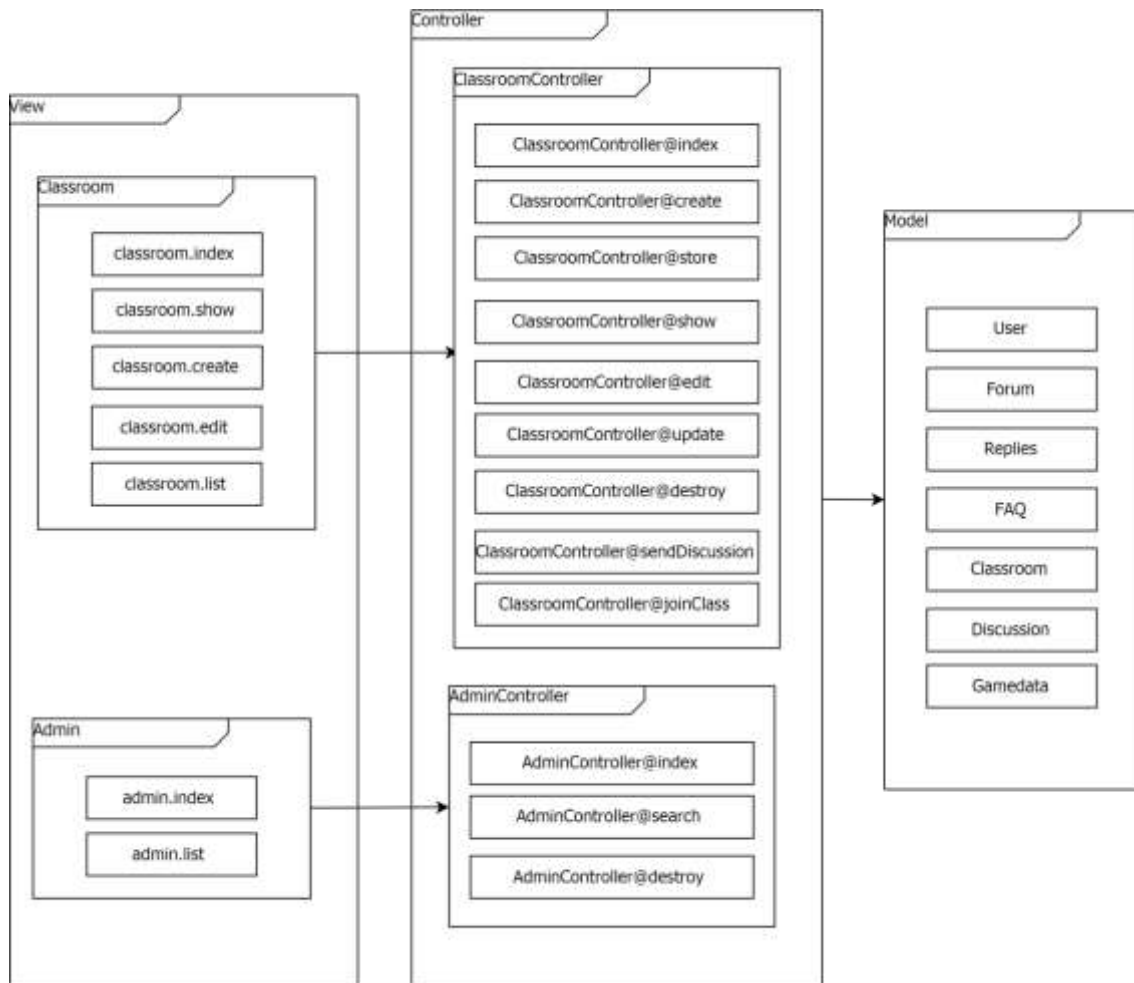


Figure 4.4 OrangeCode MVC Design Part 4

OrangeCode consist of several packages of views and their own controllers which are:

- Authentication Views => Auth Controller Package
- Profile Views => Profile Controller
- Forum Views => Forum Controller
- FAQ Views => FAQ Controller
- Classroom Views => Classroom Controller
- Home View => Home Controller
- Admin View => Admin Controller

As displayed in the figure, those controllers utilised the models. Each controller may use more than one model depending on the functionality desirability. For more detail, please refer Appendix D - Software Design Document.

4.3 SYSTEM IMPLEMENTATION

4.3.1 Development Environment

In order to develop a web-based system, it is a must to have a web development environment which is the server. Hosting a server can be free or at a cost depending on what is going to be delivered. As for testing, then it is the best to host server locally where it is only accessible through the same network of the device. As for a finished system or for environment testing, then it must be hosted with unique domain and published to the internet. Plus, hosting a server also brings the meaning of hosting database for the system to function more dynamically.

In the development of OrangeCode, local hosting software called XAMPP is used as the development environment. It is for the fact that OrangeCode is built using Laravel framework which is a PHP framework. Thus, XAMPP will be the best paired with Laravel framework since it is a PHP development environment.

Apart from that, the database management system used for the development period is the PHPMyAdmin database management system while the Integrated Development Environment (IDE) used is Visual Studio Code (VSCode). VSCode provides many beneficial features such as being able to use terminal internally rather than opening it externally to execute particular commands.

4.3.2 Laravel Framework - Setup and Installation

Before using Laravel, a PHP dependency tool must be installed which is called the *Composer*. The tool is free to be downloaded from the internet and it must be installed in order to execute specific Laravel related command such as to create new model or

controller. Once composer installed, then Laravel command can be used to develop the system. Table 4.1 below shows several commands that are commonly used in Laravel.

Table 4.1 Common Commands used in Laravel Framework

Command	Description
<code>composer create-project laravel/laravel <project name></code>	To create a new project.
<code>php artisan make:model <model name></code>	To create a new model.
<code>php artisan make:controller <controller name></code>	To create a new controller.
<code>php artisan make:controller <controller name> --resources</code>	To create a new controller with auto declaration of basic function: index, create, store, show, edit, update
<code>php artisan make:migration <migration name></code>	To create a migration (table related command to database).
<code>php artisan migrate</code>	To execute the migration and take effect to related tables.

When a Laravel project is created using *Composer*, all the dependencies required by Laravel environment will be automatically appear. Figure 4.3 below shows the project structure of a Laravel environment when initialized.

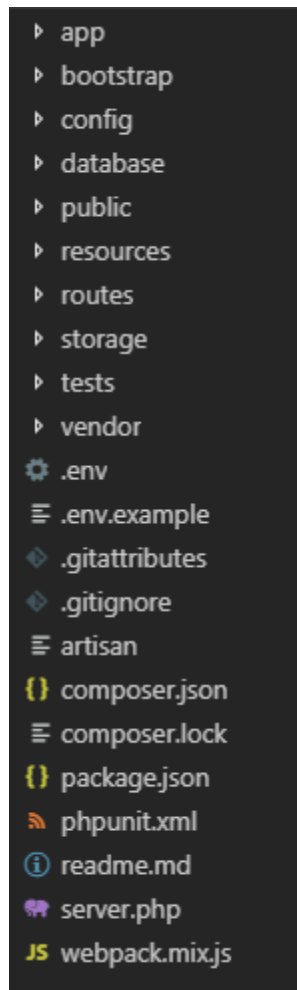


Figure 4.5 Laravel Project Structure

In every directory, for example in *app* directory, there are more sub directories and files which are used to make Laravel functioned properly. In general, *app* directory holds the **models** and **controllers** files. While *database* directory holds the **migrations** classes in which when a specific command is executed, those classes will cause update to the database. As for **views**, they should be located under the *resources/view* directory.

In order for all models, controllers and views to interact with each other, routes must be configured to link them together. Figure 4.4 below shows the **web.php** file under *routes* directory that is used to connect between controllers and views.

```

EXPLORER
├── OPEN EDITORS
├── ORANGECODE
│   ├── app
│   ├── bootstrap
│   ├── config
│   ├── database
│   ├── public
│   ├── resources
│   └── routes
│       ├── api.php
│       ├── channels.php
│       ├── console.php
│       └── web.php
│           ├── storage
│           ├── tests
│           └── vendor
│               ├── .env
│               ├── .env.example
│               ├── .gitattributes
│               └── .gitignore
└── web.php
25
26 Route::get('/forum/announcements', 'PagesController@announcements');
27 Route::get('/forum/queries', 'PagesController@queries');
28 Route::get('/forum/offtopics', 'PagesController@offtopics');
29
30 Route::resource('posts', 'PostsController');
31
32 Route::resource('forum', 'ForumsController');
33 Route::get('/forum/create/{forumtype}', 'ForumsController@createPost');
34 Route::post('/forum/show', 'ForumsController@storeReply');
35
36 Route::resource('faq', 'FAQController');
37 Route::get('/faq/create/{faqtype}', 'FAQController@createByType');
38 Route::post('/faq/show', 'FAQController@storeFaqByType');
39
40 Route::resource('profile', 'ProfileController');
41
42 Auth::routes();
43
44 Route::get('/home', 'HomeController@index');
45

```

Figure 4.6 web.php File used to Route Controllers and Views together

4.3.3 Laravel Framework - Creating Views

As mentioned in 4.3.2, views must be allocated under the *resource/view* directory. By navigating to the directory, it is much easier to just manually create a view by right-clicking the directory and press “New File”. Then, name it by any name except that the extension must always be “.blade.php”.

Laravel uses *.blade.php* extension to its view rather than *.php* in normal PHP framework. For example, the file name for view called “index” will be “index.blade.php”. This is because Laravel uses Blade to create the views. Blade is a template that allows Laravel front-end developers to design views using convenience keywords such as @section to fill up a specific section in the HTML or @extends to extend a HTML segment from another .blade.php file.

4.3.4 Laravel Framework - Creating Controllers

Controllers are very important in an MVC based system. Controllers connect views and models together to perform a fully operational function as per requirements. Controller can be easily created using commands as stated in table 4.1.

```
php artisan make:controller <controller name>
```

It is possible to develop a system with only one controller but the best practice is to divide every requirement as one controller. For example, the requirement states that the

system needed a forum management. Thus, a controller specifically for forum should be created distinctly from other requirement. This will allow developers to differentiate and manage which controller belongs to which views or models. Plus, since controllers are in the back-end side, thus mistakes by developers can occur more frequently in this layer.

4.3.5 Laravel Framework - Creating Models

Every dynamic web-based system must have a database that is hosted on a server. In Laravel or any MVC, a model becomes the reference to a table in the database. It represents the data from database that will be utilised by controller either to be displayed in views or other means. Model should only be created when a new table is needed. For example, a system may require “User” table to store users’ information should create a model for it. A model may be used by multiple controllers and a controller may use multiple models in it. Creating a model can be done with this command:

```
php artisan make:model <model name>
```

Plus, a model that has connection with another model must be declared in the model constructor. For example, a “Forum” model that has connection with “User” model should be bridged. Figure 4.6 shows an example of bridge of “Forum” model with other several models.

```

Forum.php x
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Forum extends Model
8  {
9      // Table Name
10     protected $table = 'forumposts';
11
12     // Primary Key
13     protected $primaryKey = 'forumid';
14
15     // Timestamps
16     public $timestamps = true;
17
18     public function user()
19     {
20         return $this->belongsTo('App\User', 'userid');
21     }
22
23     public function replies()
24     {
25         return $this->hasMany('App\Replies', 'replyid');
26     }
27 }

```

Figure 4.7 Forum Model

4.3.6 Laravel Framework - Creating Migration and Migrate

Migration in Laravel is defined as creating a control for updating tables in database. For example, when a migration is created, developers can create command in the migration to perform table related function such as to create or drop table, to drop or create new field in a table or even to perform rollback of database onto the system. Migration can simply be created using this command:

```
php artisan make:migration <migration name>
```

Best practice is to name the migration that can be easily understood. For instance, it will be readable if the migration is named as “Create_Forum_Table” or “Add_Forum_Type_To_Table_Forum”. In order to execute the migration created, developer must migrate all the migration using command:

```
php artisan migrate
```

4.3.7 Laravel Framework - Routing

Routing is used to define routes between a view and a controller. Routing is not specified in MVC architecture only, even in other framework with different architecture may also have routing. When user directed to a web page, the system will route the web page with its own controller and execute any necessary functionalities. Without routing, it is almost impossible for a system to work properly since the view and controller will never be connected. Figure 4.7 shows an example of routing between views of “Forum” and their controllers.

```
//=====FORUM=====
Route::get('/forum/announcements', 'PagesController@announcements');
Route::get('/forum/queries', 'PagesController@queries');
Route::get('/forum/offtopics', 'PagesController@offtopics');
Route::resource('forum', 'ForumsController');
Route::get('/forum/create/{forumtype}', 'ForumsController@createPost');
Route::post('/forum/show', 'ForumsController@storeReply');
```

Figure 4.8 Routing Examples

In addition, routing main goal is to determine what kind of request needed used by the user, for example the ‘post’ or ‘get’ request which is to send request to the model or to fetch data from the database respectively.

4.3.8 Laravel Framework - Retrieving Data with Eloquent

Eloquent is a special feature in Laravel where developer can easily grab data from a model using simple syntax. Figure 4.8 shows an example of the implementation of eloquent in Laravel.

```

public function show($id)
{
    $replies = Replies::where('forumid', $id)->orderBy('created_at', 'asc')->paginate(10);
    $forumMain = Forum::find($id);

    $data = array(
        'forumMain' => $forumMain,
        'replies' => $replies
    );

    return view('forum.show')->with($data);
}

```

Figure 4.9 Eloquent Implementation in Forum Controller

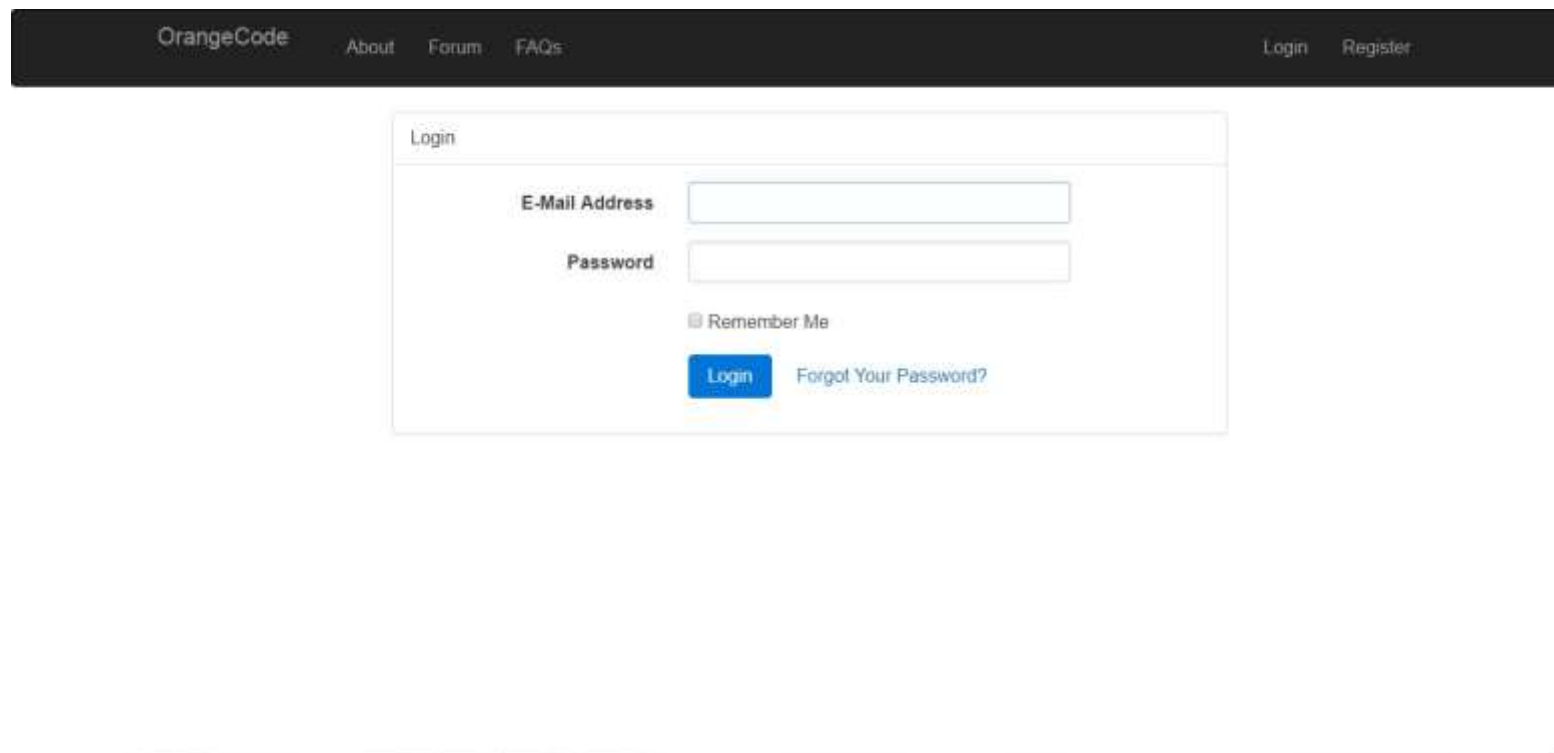
By using ‘Forum::find(\$id)’ for example, will help us to connect to database as well as perform query without the need of writing it manually. For example, normally in MySQL, we have to write our own query such as “SELECT * FROM forum”. But with eloquent feature in Laravel, we just have to write “Forum::all()” which is more convenient and easy to use.

4.4 SYSTEM TESTING AND RESULTS

System developed must always be tested in order to identify if the system work as expected or not. Thus, the next step after developing the system is testing the system itself. This section will discuss on the techniques of testing applied and their results are shown in the subsection below.

4.4.1 Unit Testing

A. Authentication



The image shows a web application's login interface. At the top, a dark navigation bar contains the text "OrangeCode" on the left and "Login Register" on the right. Below this, a white box titled "Login" contains the following elements: an "E-Mail Address" label next to a text input field, a "Password" label next to a password input field, a "Remember Me" checkbox, a blue "Login" button, and a link labeled "Forgot Your Password?".

Figure 4.10 The Login Authentication Interface

Register

Name

E-Mail Address

Password

Confirm Password

Birthdate

Gender

Nationality

You are: A Student A Teacher
Teacher can create classrooms to handle students

Figure 4.11 The Registration Interface

The tests conducted for Authentication module are for the login, register and reset password functionality.

i. Decision Table

Table 4.2 Authentication Decision Table

Test Condition ID	Test Condition
TC_AUTH_C_01	Valid Name
TC_AUTH_C_02	Valid Email
TC_AUTH_C_03	Email Not Taken
TC_AUTH_C_04	Password Matched

Condition																
TC_AUTH_C_01	F	F	F	F	F	F	F	F	F	T	T	T	T	T	T	T
TC_AUTH_C_02	F	F	F	F	T	T	T	T	F	F	F	F	T	T	T	T
TC_AUTH_C_03	F	F	T	T	F	F	T	T	F	F	T	T	F	F	T	T
TC_AUTH_C_04	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T
Output																
Authenticate Into System	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	T
Display Warning Message	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

Table 4.3 Authentication Test Specification

Test Case ID	Test Item	Test Parameters	Test Data	Test Type	Expected Result	Actual Result	Pass / Fail	Type
TC_AUTH_01	Register	Name Email Password Confirm Password	Myname myemail@email.com 123456 123456	Normal	Registered Successfully	Registered Successfully	Pass	Unit Testing

TC_AUTH_02	Register	Name Email Password Confirm Password	Myname2 myemail2@email.com 123456 1234	Erroneous	Password Does not Match message appears.	Password Does not Match message appears.	Pass	Unit Testing
TC_AUTH_03	Register	Name Email Password Confirm Password	Myname2 myemail2@email.com 1234 1234	Erroneous	Password must be at least 6 characters message appears.	Password must be at least 6 characters message appears.	Pass	Unit Testing
TC_AUTH_04	Register	Name Email Password Confirm Password	Myname2 myemail2@com 123456 123456	Erroneous	Email must be valid message appears.	Email must be valid message appears.	Pass	Unit Testing
TC_AUTH_05	Login	Email Password	myemail@email.com 123456	Normal	Login Succesfully, Redirected to Dashboard	Login Succesfully, Redirected to Dashboard	Pass	Unit Testing
TC_AUTH_06	Login	Email Password	myemail@email.com 1234	Erroneous	Failed to login	These credentials do not match our records message appear.	Pass	Unit Testing
TC_AUTH_07	Login	Email Password	myemail 123456	Erroneous	Validation of email format error appear	Validation of email format error appear	Pass	Unit Testing

TC_AUTH_08	Forget Password	Email	myemail@email.com	Normal	Password reset link sent to email	Maximum execution time of 30 seconds exceeded error appears.	Fail	Unit Testing
------------	-----------------	-------	--	--------	-----------------------------------	--	------	--------------

B. Profile



Figure 4.12 Updating the Display Image

The tests conducted for Profile module which include in updating the profile as well as view other people profile.

Table 4.4 Profile Test Specification

Test Case ID	Test	Test Parameters	Test Data	Test Type	Expected	Actual Result	Pass / Fail	Type
--------------	------	-----------------	-----------	-----------	----------	---------------	-------------	------

	Item				Result			
TC_PROF_01	Update Profile	Profile Image	782KB sized image	Normal	Updated successfully	Updated successfully	Pass	Unit Testing
TC_PROF_02	Update Profile	Profile Image	3140KB sized image	Erroneous	Failed to upload image.	Failed to upload image.	Pass	Unit Testing
TC_PROF_03	Update Profile	Profile Image	.PDF type file	Erroneous	Failed to upload file.	The display image must be an image message appears.	Pass	Unit Testing
TC_PROF_04	Update Profile	Name Email Password Confirm Password	Myname2 myemail2@com 123456 123456	Erroneous	Email must be valid message appears.	Email must be valid message appears.	Pass	Unit Testing
TC_PROF_05	Update Profile	Date of Birth	01/01/2000	Normal	Updated successfully	Updated successfully	Pass	Unit Testing
TC_PROF_06	Update Profile	Date of Birth	01/01/2020 (Ahead of current actual year)	Erroneous	Failed to update	Updated successfully	Fail	Unit Testing
TC_PROF_07	Update Profile	Gender Website Company/Institution Nationality Interest	Male "OrangeCode.com" "UMP" "Malaysia" "Fishing"	Normal	Updated successfully	Updated successfully	Pass	Unit Testing

		Biography	“Love to code!”					
TC_PROF_08	Update Profile	Gender Website Company/Institution Nationality Interest Biography	Male *blank *blank *blank *blank	Normal	Updated successfully	Updated successfully	Pass	Unit Testing

C. Forum

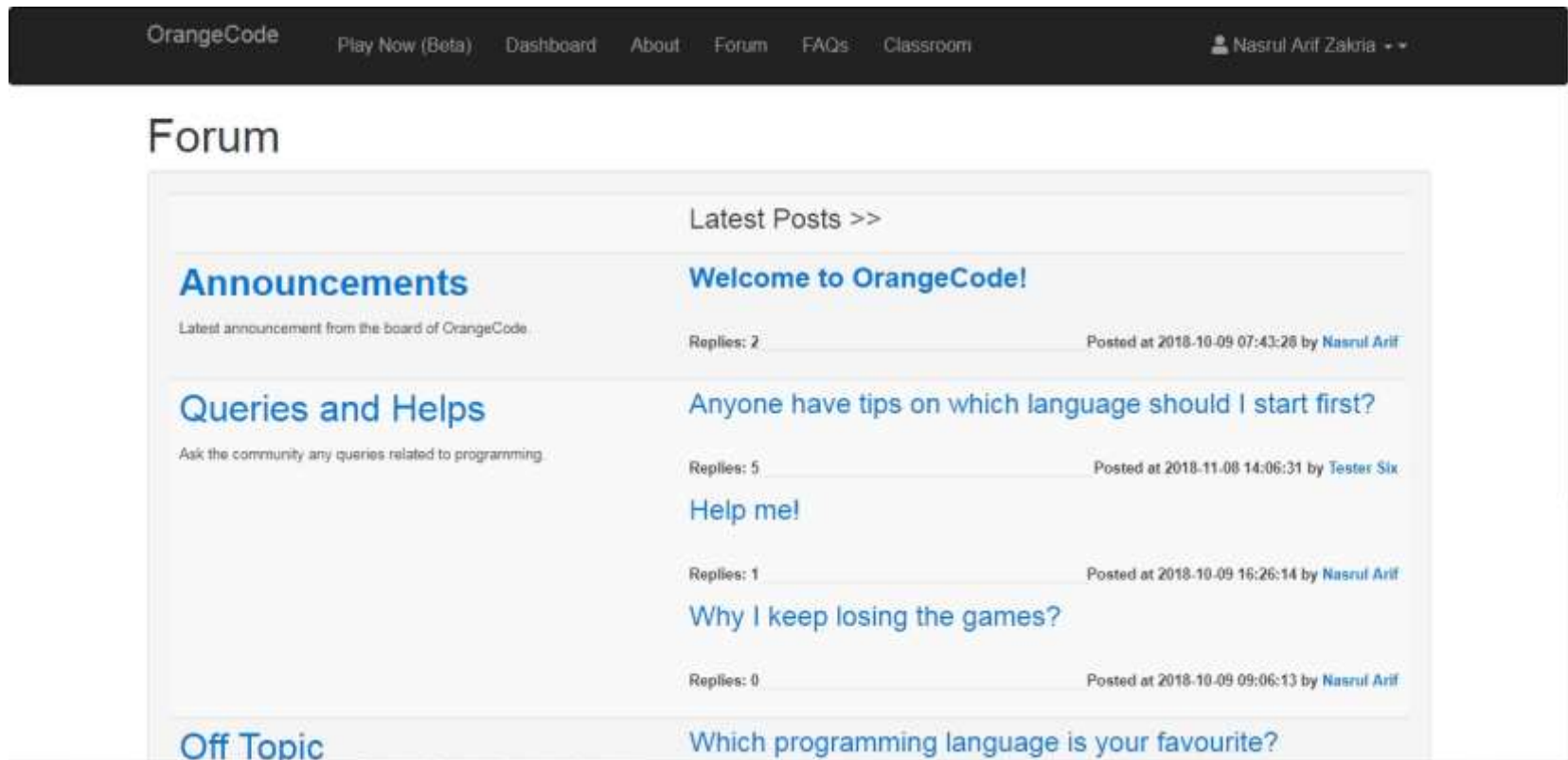


Figure 4.13 Forum User Interface

Meow

You can try the demo from the main page.

Replied at 2018-10-11 18:26:23 | Reply-ID-1

Reply Nasrul Arif Zakria

✂️ 📄 📄 📄 📄 ⬅️ ➡️ ABC 🔗 🚩 🖼️ 📄 📄 Ω 🔄 Source

B I S u ☰ ☷ ☰ ☷ ⌘ ⌘ | Styles - | Format - | ?

Reply

Figure 4.14 Replying to a Post

The tests conducted for Forum module that includes initiating a forum in different types, replying to forum.

Table 4.5 Forum Test Specification

Test Case ID	Test Item	Test Parameters	Test Data	Test Type	Expected Result	Actual Result	Pass / Fail	Type
TC_FORUM_01	Create Forum	Title Body	“Hello” “Hi, I am a user.”	Normal	Created successfully	Created successfully	Pass	Integration Testing
TC_FORUM_02	Create Forum	Title Body	*blank “Hi, I am a user.”	Erroneous	Failed to create a new forum.	The title field is required message appears.	Pass	Integration Testing
TC_FORUM_03	Create Forum	Title Body	“Hello” *blank	Erroneous	Failed to create a new forum.	The body field is required message appears.	Pass	Integration Testing
TC_FORUM_04	View Forum	-	-	Normal	Forum displayed successfully.	Forum displayed successfully.	Pass	Integration Testing
TC_FORUM_05	Reply Forum	Body	“Hello this is a reply”	Normal	Replied successfully.	Replied successfully.	Pass	Integration Testing
TC_FORUM_06	Reply Forum	Body	*blank	Erroneous	Failed to reply to the forum.	The body field is required message appears.	Pass	Integration

								Testing
--	--	--	--	--	--	--	--	---------

D. FAQ

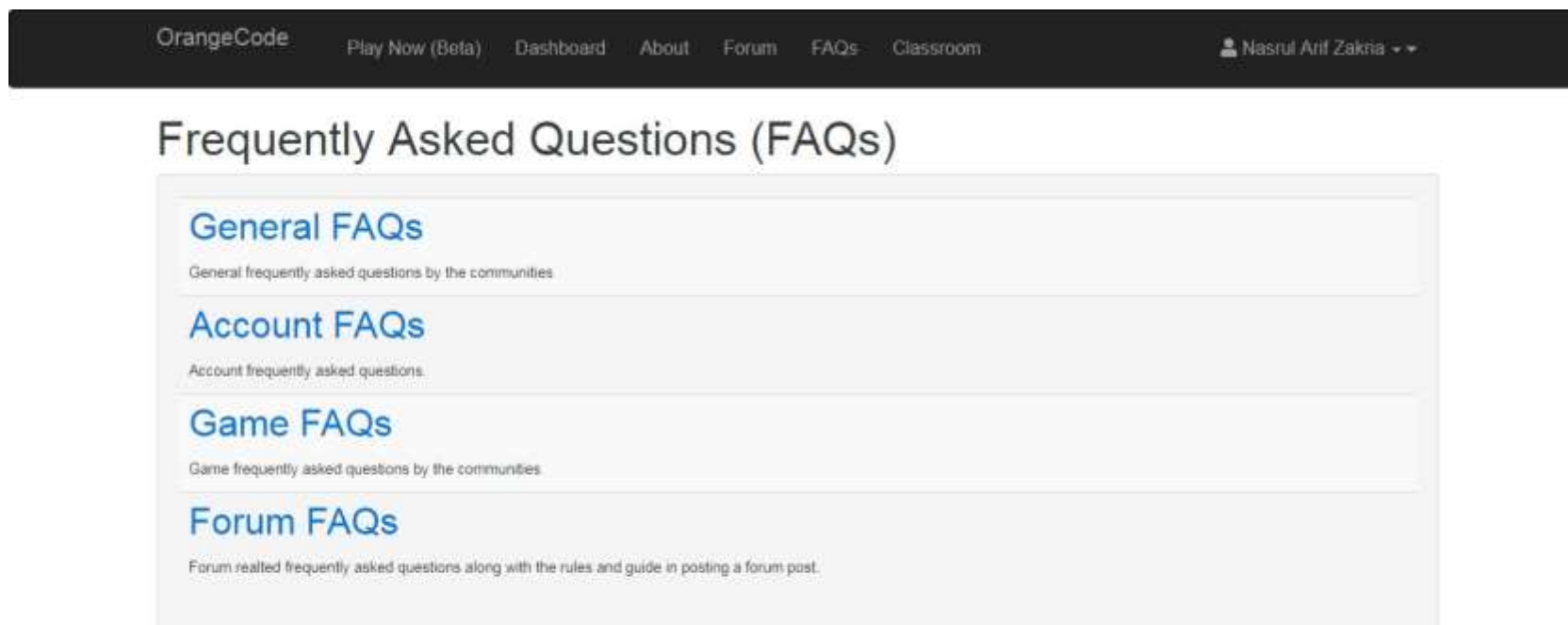


Figure 4.15 FAQ Interface

The tests conducted for FAQ module that includes creating new FAQ, editing, deleting and viewing it.

Table 4.6 FAQ Test Specification

Test Case ID	Test Item	Test Parameters	Test Data	Test Type	Expected Result	Actual Result	Pass / Fail	Type
TC_FAQ_01	Create FAQ	Title Body Type	“This is a FAQ” “Hello” General FAQ	Normal	Created Successfully	Created Successfully	Pass	Integration Testing
TC_FAQ_02	Create FAQ	Title Body Type	*blank “Hello” General FAQ	Erroneous	Failed to create a new FAQ.	The title field is required message appears.	Pass	Integration Testing
TC_FAQ_03	Create FAQ	Title Body Type	“This is a FAQ” *blank General FAQ	Erroneous	Failed to create a new FAQ.	The body field is required message appears.	Pass	Integration Testing
TC_FAQ_04	Create FAQ	Title Body Type	“This is a FAQ” “Hello” Account FAQ	Normal	Created Successfully	Created Successfully	Pass	Integration Testing
TC_FAQ_05	Create FAQ	Title Body Type	“This is a FAQ” “Hello” Game FAQ	Normal	Created Successfully	Created Successfully	Pass	Integration Testing
TC_FAQ_06	Create FAQ	Title Body Type	“This is a FAQ” “Hello” Forum FAQ	Normal	Created Successfully	Created Successfully	Pass	Integration Testing
TC_FAQ_07	Edit/Update FAQ	Title Body	“This is a FAQ” “Hello”	Normal	Updated Successfully	Updated Successfully	Pass	Integration

								Testing
TC_FAQ_08	Edit/Update FAQ	Title Body	*blank "Hello"	Erroneous	Failed to update FAQ.	The title field is required message appears.	Pass	Integr ation Testing
TC_FAQ_09	Edit/Update FAQ	Title Body	"This is a FAQ" *blank	Erroneous	Failed to update FAQ.	The body field is required message appears.	Pass	Integr ation Testing
TC_FAQ_10	View FAQ	-	-	Normal	Able to view FAQ according to type and ID	Able to view FAQ according to type and ID	Pass	Integr ation Testing
TC_FAQ_11	Delete FAQ	*ID	*Dynamically	Normal	Deleted Successfully	Deleted Successfully	Pass	Integr ation Testing

E. Classroom

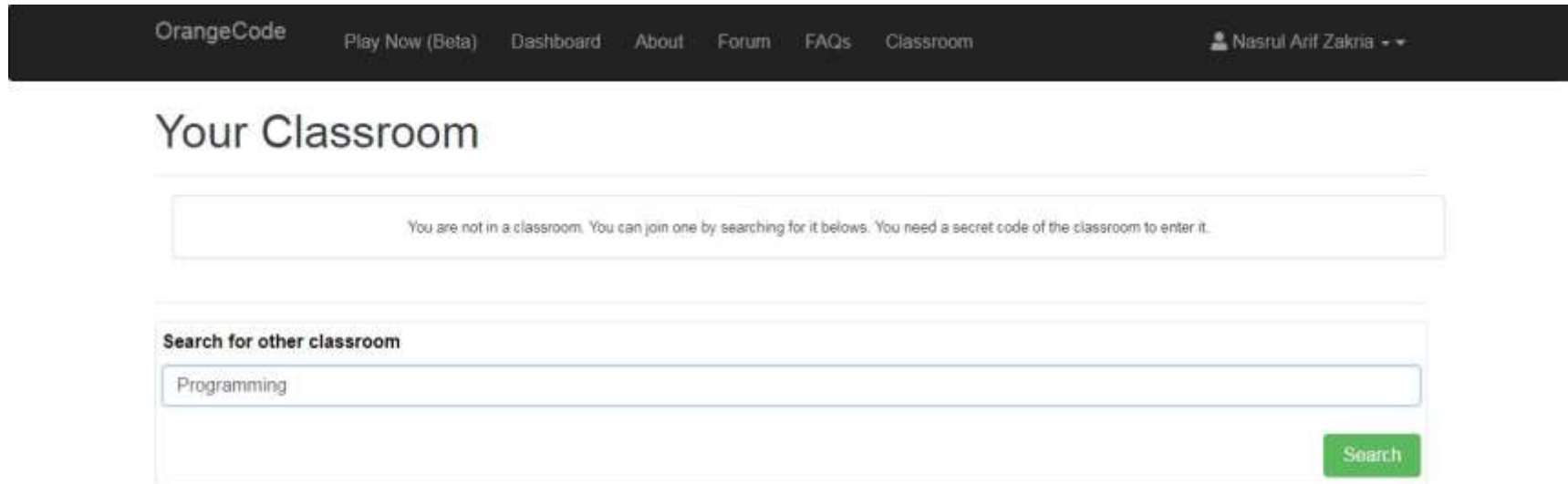


Figure 4.16 Classroom Main Interface

The tests conducted for Classroom module where students are able to view classroom information as well as to join or leave a classroom. Teachers are able to create, edit and update or delete a classroom. Both users can search for existing classroom ever created using keywords. Plus, teacher who owned a classroom as well as students of the classroom can perform discussion at the discussion board of the classroom.

Table 4.7 Classroom Test Specification

Test Case ID	Test Item	Test Parameters	Test Data	Test Type	Expected Result	Actual Result	Pass / Fail	Type
TC_CLASS_01	Create Classroom	Class Title Class Description Class Objective Institution Class Secret Code	“This is a title” “This is description” “This is objective” “UMP” “ABCDEFGH”	Normal	Created successfully	Created successfully	Pass	Integration Testing
TC_CLASS_02	Edit / Update Classroom	Class Title Class Description Class Objective Institution Class Secret Code	“Edited title” “Edited description” “Edited objective” “Edited institution” “HGFEDCBA”	Normal	Updated successfully	Updated successfully	Pass	Integration Testing
TC_CLASS_03	View Classroom	-	-	Normal	View loaded successfully with complete data fetched	View loaded successfully with complete data fetched	Pass	Integration Testing
TC_CLASS_04	Join Classroom	Class Secret Code	*matching secret code	Normal	Student successfully joined the	Student successfully joined the	Pass	Integration

					classroom	classroom		Testing
TC_CLASS_05	Join Classroom	Class Secret Code	*unmatching secret code	Erroneous	Student successfully joined the classroom	Student successfully joined the classroom	Pass	Integration Testing
TC_CLASS_06	Leave Classroom	-	-	Normal	Student successfully leave the classroom	Student successfully leave the classroom	Pass	Integration Testing
TC_CLASS_07	Search Classroom	Keywords	“Classroom”	Normal	A list of classroom with any part of the title represent the keyword will appear.	A list of classroom with any part of the title represent the keyword will appear.	Pass	Integration Testing

F. Administrator Task

Administrator is able to have a list of all users or classroom or forum posts exist in the system database. Plus, an administrator can search for specific user or classroom or forum with the search function by inputing keywords. From the list of users or classrooms or forum posts, an administrator is able to view or delete it as desire. This is useful to handle abusive users that do not follow the rules or using any hack to gain benefits in the game provided.

Table 4.8 Administrator Tasks Test Specification

Test Case ID	Test Item	Test Parameters	Test Data	Test Type	Expected Result	Actual Result	Pass / Fail	Type
TC_ADMIN_01	Search Users	Keyword	“Ali”	Normal	System generate list of users that have “Ali” in its username.	System generate list of users that have “Ali” in its username.	Pass	Integration Testing
TC_ADMIN_02	Search Classroom	Keyword	“Top”	Normal	System generate list of classroom that have “Top” in its classroom title.	System generate list of classroom that have “Top” in its classroom title.	Pass	Integration Testing
TC_ADMIN_03	Search Forum Post	Keyword	“How to”	Normal	System generate list of forum posts that have “How to” in its forum title.	System generate list of forum posts that have “How to” in its forum title.	Pass	Integration Testing
TC_ADMIN_04	Get list of all users	-	-	Normal	System display list of all users.	System display list of all users.	Pass	Integration Testing
TC_ADMIN_05	Get list of all classrooms	-	-	Normal	System display list of all classroom.	System display list of all classroom.	Pass	Integration

								Testing
TC_ADMIN_06	Get list of all forum posts	-	-	Normal	System display list of all forum posts.	System display list of all forum posts.	Pass	Integration Testing
TC_ADMIN_07	Delete Item	-	-	Normal	System delete item successfully	System delete item successfully	Pass	Integration Testing

4.4.2 System Testing

The system has been tested under the computer of specification:

Processor: Intel® Core™ i5-3210M CPU @ 2.50GHz

RAM: 4.00GB

System type: 64-bit

OS: Windows 7

Manufacturer: Hewlett-Packard (HP)

Model: HP Pavilion g4 Notebook PC

There are several issues with the system test, where the Unity3D WebGL is having lag issue and the system load quite **slow**.

4.4.3 User Acceptance Testing

Please refer Appendix E.

4.5 INTEGRATION OF UNITY3D, LARAVEL AND MYSQL

This topic is quite fascinating to discuss as it involve three big components which is two big frameworks and a database. Looking at them standing alone may be good, until that you have dreamed of getting them integrate together to work in harmony.

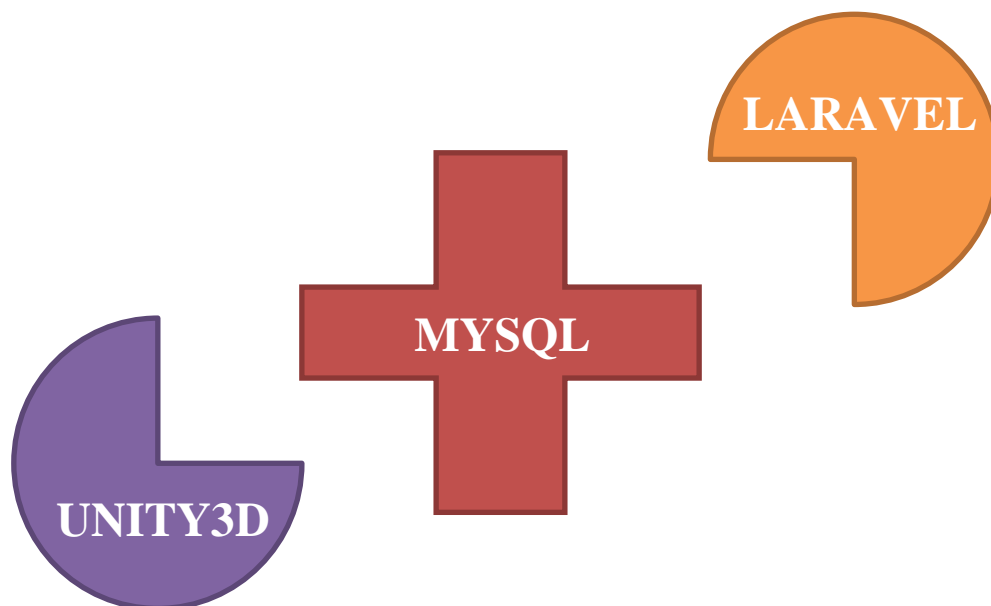


Figure 4.17 Illustration of Integrating Unity3D, MySQL and Laravel

First, let's talk about Unity3D. Few years ago, Unity3D release a way of building game as a WebGL component which is a great advantage for developers who is going to publish their game into the internet especially website. Though, the build product is too general that it requires self skill to integrate to the web development framework that we are using. For example in this project, does it even possible to integrate Unity3D WebGL into Laravel? As we know, Laravel is a MVC framework

that brings a lot of positive benefits to web artisan. In MVC, controller plays a really big role which is as a medium to the views and models. So, logically in order to integrate Unity3D with Laravel, we have to put the WebGL component into one of the views and have a route of controller to return the views to the user. It took several tries and errors trying to integrate Unity3D into one of the Laravel views since the WebGL component require the path of the library to be correct or else the Unity3D renderer would not load in the web page. Simply, putting the folders dependency into one of Laravel public directory would work. Then, the HTML WebGL component must be changed to the extension of “.blade.php” to match the Laravel views.

Now it works, but how about the MySQL integration? There are no possible ways to receive session directly from Laravel in Unity3D since Unity3D does not methods to do so. The only method that could be helpful from Unity3D is to use the “WWW” class to send request to the internet and retrieve the information. Thus, I have created a hidden page to display the user’s raw information regarding game progress and have Unity3D to fetch and convert from it into the game.

4.6 USER MANUAL

Please refer Appendix F.

CHAPTER 5

CONCLUSION

5.1 Introduction

In this chapter, the conclusion from previous chapters will be presented here. This includes the discussion of the User Acceptance Test, to ensure whether the requirements were met as well as to trace if the system functioned as it should be. Plus, we also discuss on the system status after the deployment and the long-term benefits.

5.2 User Acceptance Test Discussion

As the UAT has been done, we can conclude that the user satisfied and approved that the system work as it is and fulfilled the requirement.

5.3 System Aftermath

As the system being deployed, it should just stay as it is. As usual, maintenance, debugging, fixing and testing will always be done throughout the system life. In addition, there will be a plan to improve the system performance in term of launching the Unity WebGL since there are some reports of it being faulty in several types of browsers and computer specifications. Though, it is just a minor performance issue. Plus, more modules will be added such as creating your own level in the game as well as to implement online multiplayer feature. Other features such as the implementation of Augmented Reality (AR) or Virtual Reality (VR) could be a great investment to the system.

Other than that, there might be a mobile version of the game that can be interacted with the current web platform in the future.

5.4 System Attributes

The system can be used for teachers and students to learn programming in more effective and enjoyment rather than a boring class or lecture. Currently, there are several issues in the system where the game would hang while loading the level in several environments. Though, this is considered as minor performance issue since it does not affect any other functionalities or data. Plus, the administrator is lacking on some functionality such as to delete specific replies in a forum or to ban or mute a user. There might be other limitation in the system that remains undercover until it is reaching the limit.

5.5 Commercialization

The system can be a good for advertisement revenue. For example, when playing game, the player can receive special rewards by playing the advertisement video. Furthermore, banner advertisements and other kind of advertisements can be embedded in the web platform for revenue.

APPENDIX A TEST PLAN

A.1 Testing Plan Objective

The objective of this testing plan is:

- To detail the actions needed to organize and to perform system test
- To identify the test tools and environment needed to perform system test

A.2 Test Design Reference

The test design will be based on the following document:

- OrangeCode System Requirement Specifications (OC_SRS_1.1)

A.3 Features to Be Tested

Table A.3.1 Features to be tested

Use Case ID	Use Case Name	Risk
OC_UC01	Authentication	High
OC_UC02	Operate Classroom	Medium
OC_UC03	Play Games	Medium
OC_UC04	Manage Profile	High
OC_UC05	Operate Forum	Medium
OC_UC06	Administrator Task	High

A.4 Testing Approach

The testing technique to be conducted on OrangeCode system will be:

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Test

The test will also be using the black-box testing approach where it may covers the following testing technique.

- Decision Table Testing

A.5 Item Pass / Fail Criteria

In order for the system to pass, it must succeed the following criteria:

- Passed all test cases and without crash
- No deadlock happened throughout the system operation

A.6 Entry Criteria

Before testing may begin, it needed:

- Requirements documents of OrangeCode

A.7 Exit Criteria

Before testing may ended, it needed:

- Test execution has been completed
- Meets the PSM due dates
- Defect at the end of the testing process shall be:
 - Zero fatal defect,
 - Zero major defect,
 - Less than 3 minor defects

APPENDIX B GANTT CHART

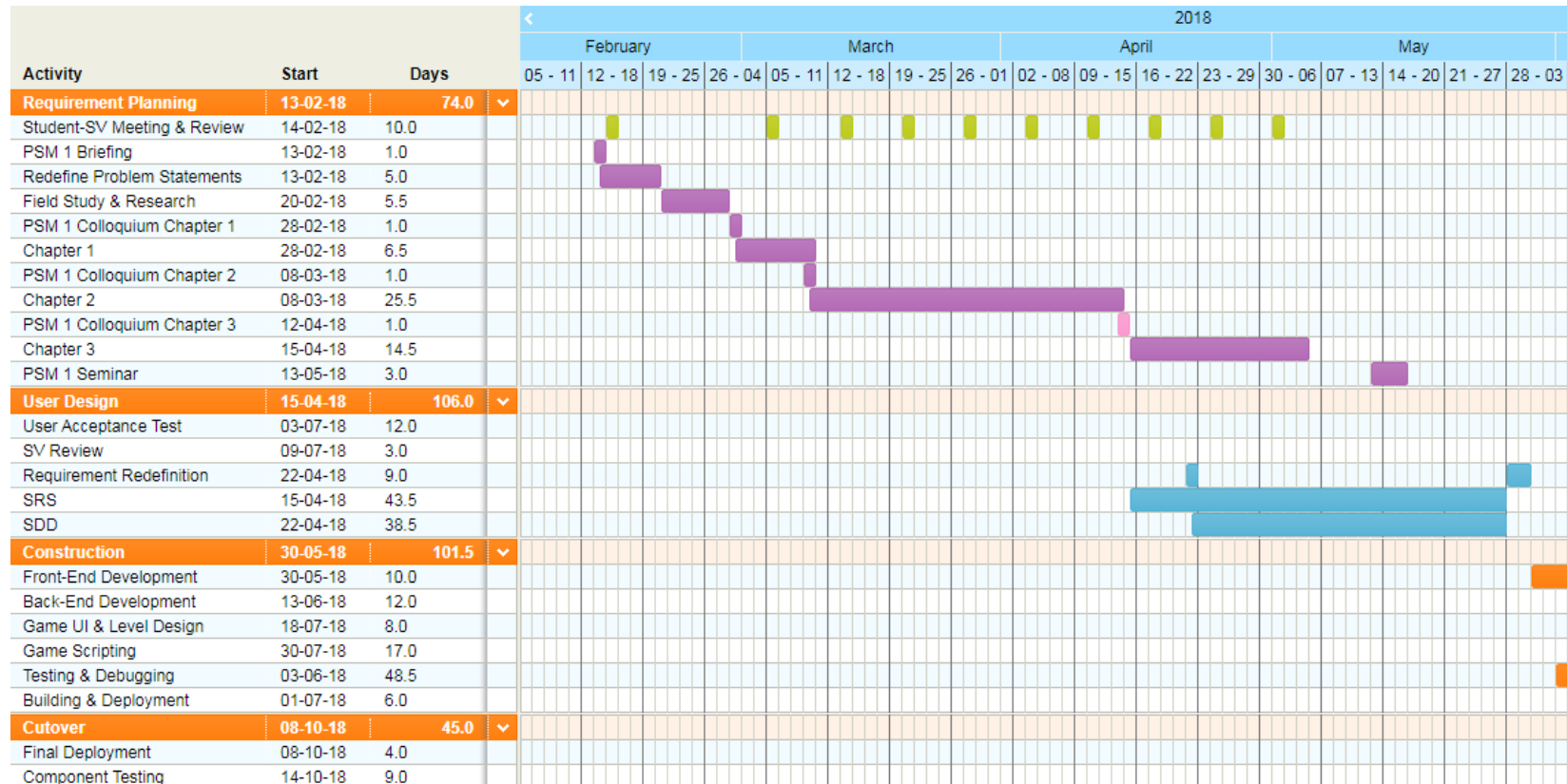


Figure B-1 Gantt Chart Part 1

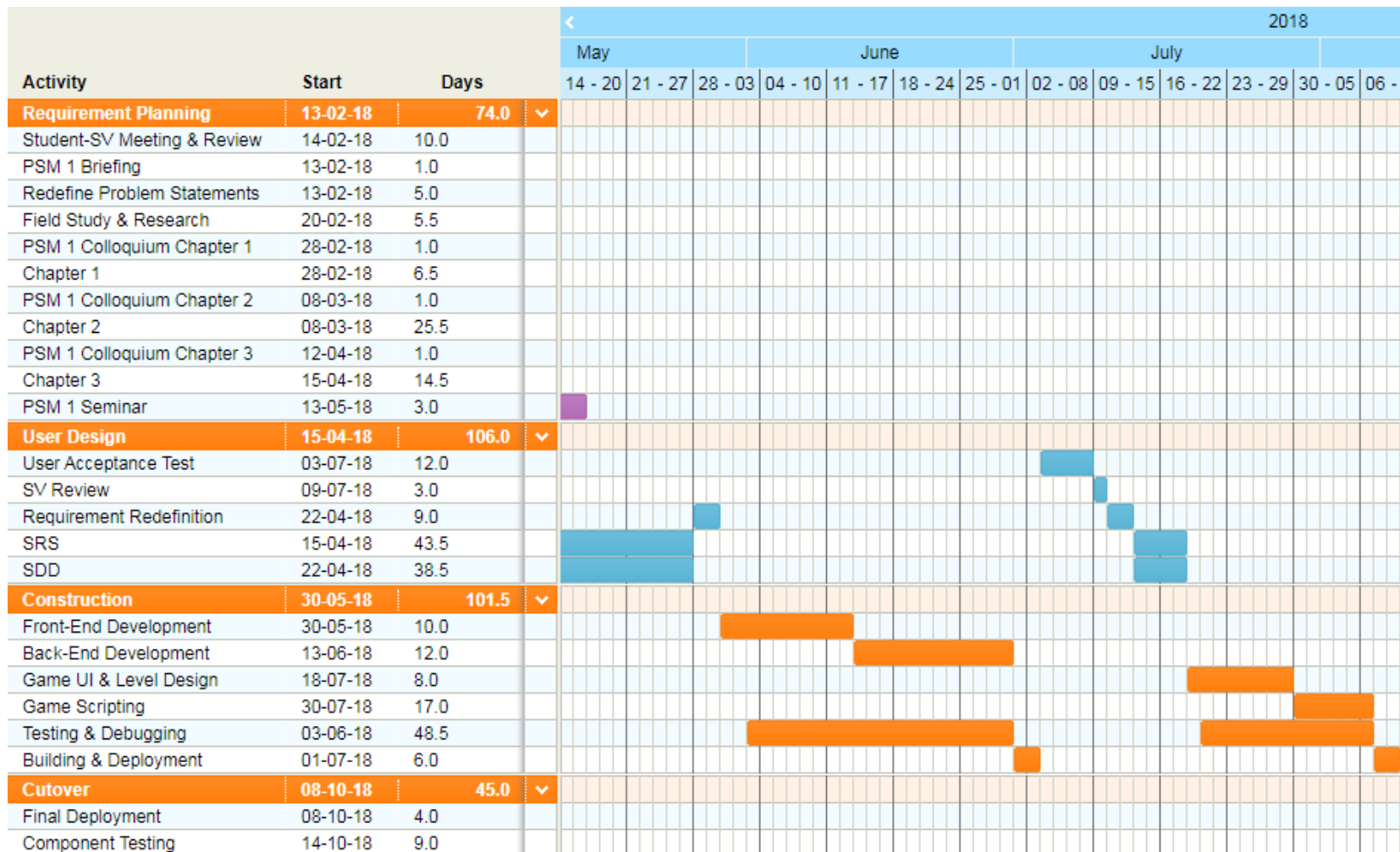


Figure B-2 Gantt Chart Part 2

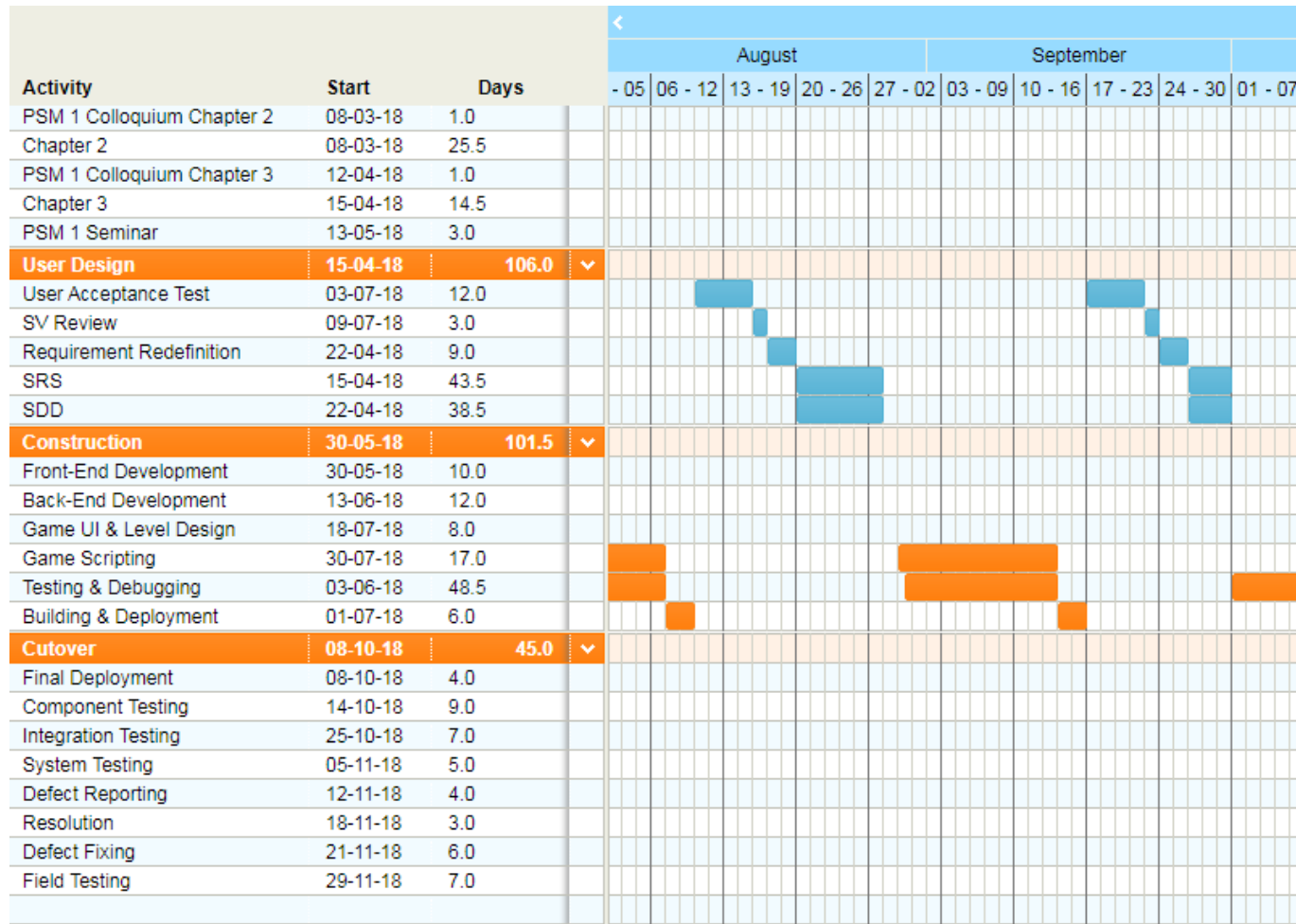


Figure B-3 Gantt Chart Part 3

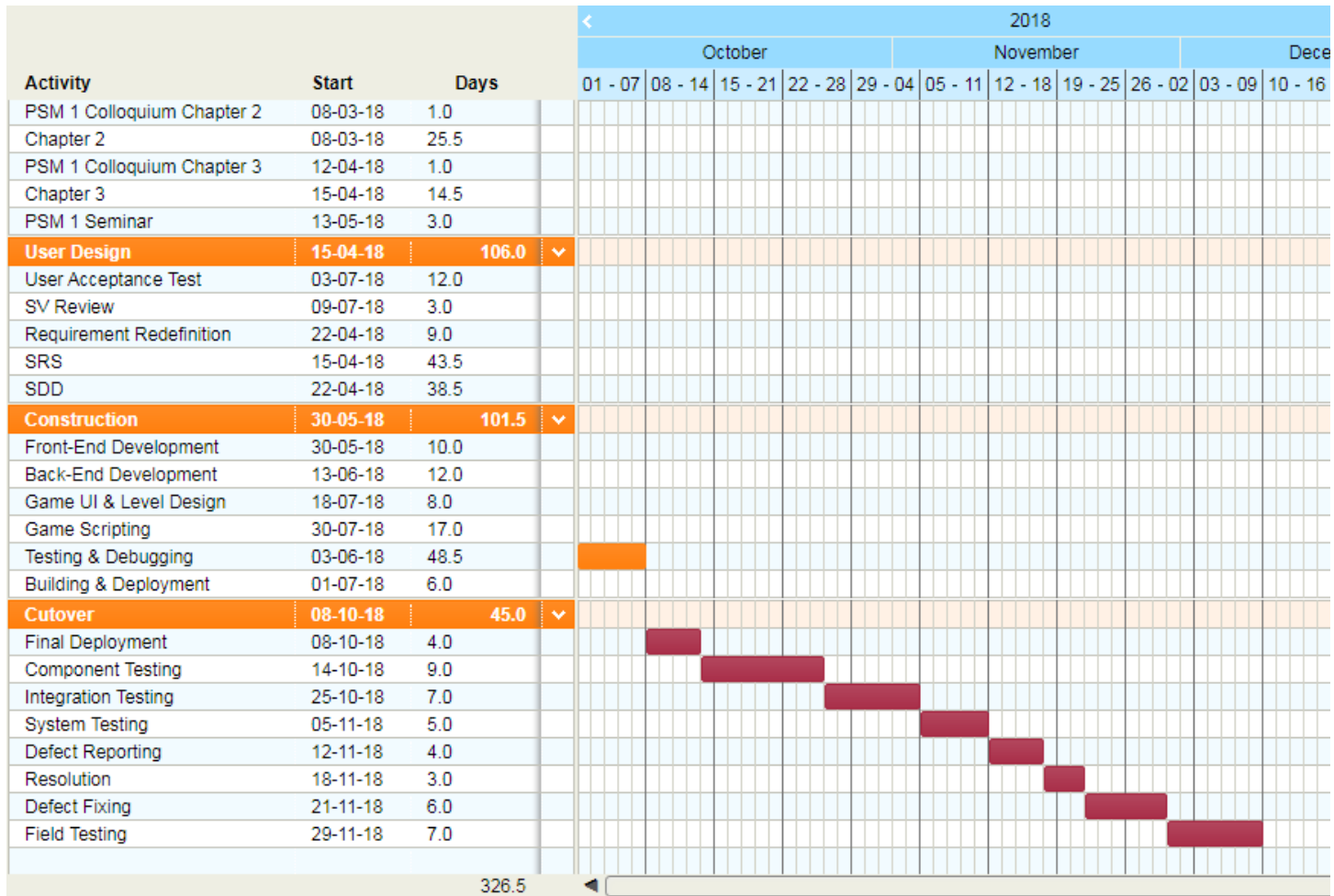


Figure B-4 Gantt Chart Part 4

APPENDIX C

SYSTEM REQUIREMENT SPECIFICATIONS

System Requirement Specifications (SRS) is very important document or artifact in a system development. SRS will state and explains the main function that the system to build. For example, how many modules are there in the system and how the flow of the system should be.

2019

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

ORANGECODE

NASRUL ARIF BIN ZAKRIA
UNIVERSITI MALAYSIA PAHANG
To be submitted to the Undergraduate Project
Bachelor of Computer Science (Software Engineering)



TABLE OF CONTENTS

TABLE OF CONTENTS	ii
LIST OF FIGURES	iii
LIST OF TABLES	iii
1. PRODUCT DESCRIPTION	1
1.1 USER CHARACTERISTIC.....	1
1.2 CONSTRAINTS.....	1
2. INTERFACE REQUIREMENTS	3
2.1 USER INTERFACE	3
2.2 HARDWARE INTERFACE	4
2.3 SOFTWARE INTERFACE.....	4
3. SOFTWARE PRODUCT FEATURES	6
3.1 SOFTWARE PRODUCT FEATURES	6
3.1.1 Create Account.....	6
3.1.2 Create Class.....	8
3.1.3 Play Games.....	12
3.1.4 Manage Profile.....	14
3.1.5 Operate Forum.....	16
3.1.6 Administrator Task.....	18
4. REQUIREMENT TRACEALIBTY	22
5. SYSTEM REQUIREMENT APPROVAL.....	25

LIST OF FIGURES

Figure 2.1	3
Figure 3.1	6
Figure 3.2	8
Figure 3.3	10
Figure 3.4	12
Figure 3.5	15
Figure 3.6	18

LIST OF TABLES

Table 1.1	1
Table 3.1	6 - 7
Table 3.2	8 - 9
Table 3.3	10 - 11
Table 3.4	12 - 14
Table 3.5	15 - 16
Table 3.6	18 - 21
Table 4.1	22
Table 4.2	22
Table 4.3	23
Table 4.4	24
Table 4.5	24
Table 4.6	25

LIST OF APPENDIXES

Appendix A - Sequence Diagrams

Appendix B - System Interfaces

1. PRODUCT DESCRIPTION

OrangeCode is a system where people can learn the fundamental of programming in different perspective, which is through gamification and visualization. OrangeCode helps users specifically students, to conduct their programming learning session in way that is more effective than lecture session.

1.1 USER CHARACTERISTIC

OrangeCode have its own target users that will be using it. Table 1.1 shows the list of user with their expected educational level along with their background experience to consider if they are eligible to use the system or not.

Table 1.1 User Characteristics

User	Education Level	Background Experience
Students	<ul style="list-style-type: none"> Primary School Secondary School Pre-University University Computer Related Course 	<ul style="list-style-type: none"> Know how to use computer or smartphone Was exposed to programming and their application
Teachers	<ul style="list-style-type: none"> Programming Experts Computer Science Experts Computer Science Lecturers, Consultants or Researchers 	<ul style="list-style-type: none"> Know how to program and write codes Understand and able to explain components in programming
Administrator	<ul style="list-style-type: none"> At least graduated from secondary school 	<ul style="list-style-type: none"> Know the structure of OrangeCode Know the rules and policies of OrangeCode

1.2 CONSTRAINTS

Below are the lists of constraints for OrangeCode:

- i. Budget Limitation; where the web server used will be the lowest tier offered by the provider.
- ii. Product File Size and RAM Consumption; the website might have some issues regarding the game loader depending on the current used computer specifications.
- iii. Software limitations
 - a. where the OrangeCode website might only able to run in particular browser that support it only

2. INTERFACE REQUIREMENTS

Below are the interface requirements for OrangeCode.

2.1 USER INTERFACE

Figure 2.1 shows the dialogue diagram of OrangeCode.

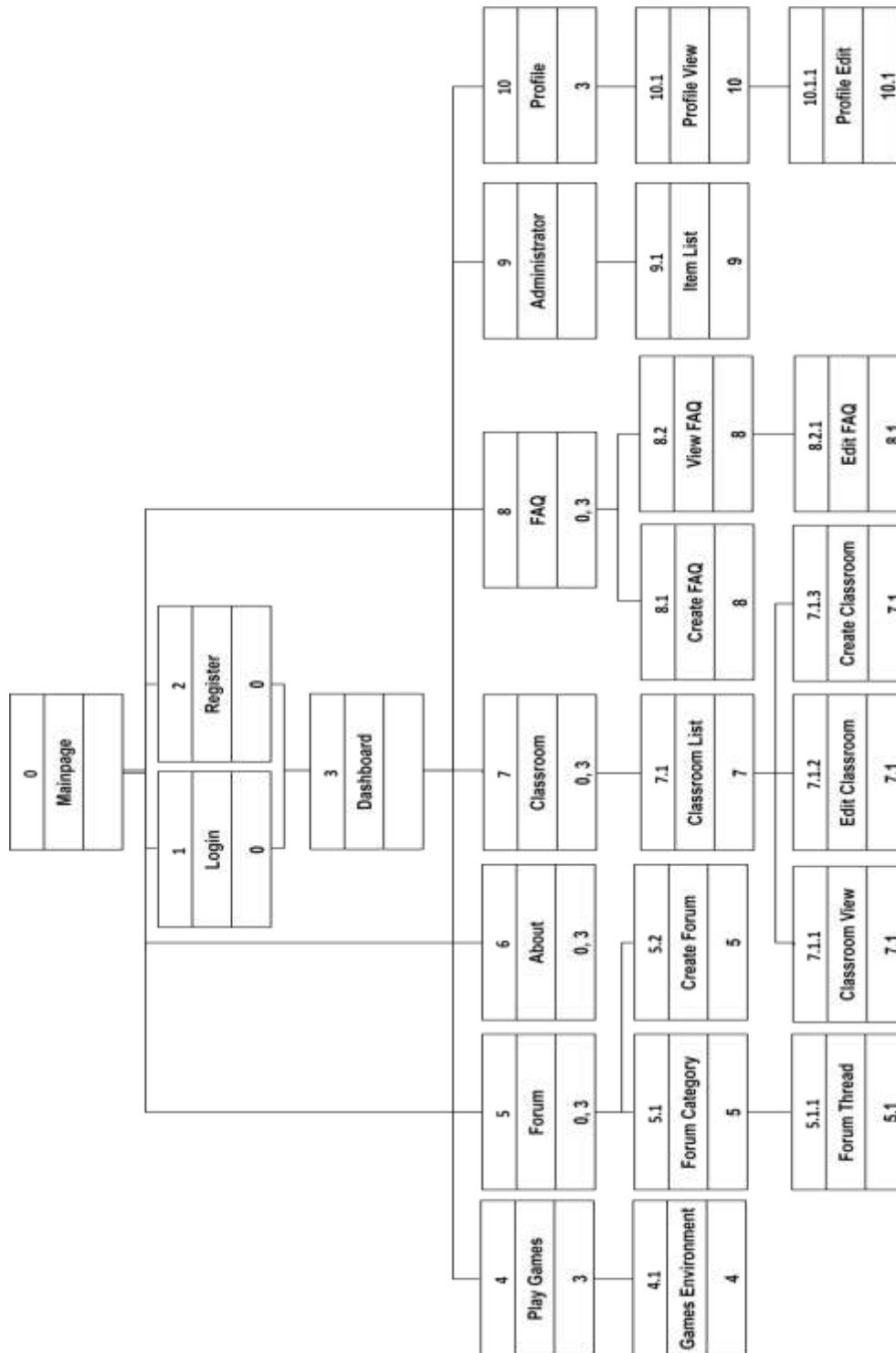


Figure 2.1 OrangeCode Dialogue Diagram

Main Page

- The page where user will be redirected on the first visit to the domain

Login Page

- Page to login into the website

Register Page

- The page where user will register for a new account

Dashboard

- The page where user being redirected after successfully logged in.

Profile

- The page where user can edit and update his/her profile information and displayed to the public.

Play Game

- Screen where the system displays the game environment, user interface and scene.

Forum

- The page where user able to interact with another user through forum-based communication.

Classroom Page

- The page where user able to view or manage the classroom.

FAQ

- The page where user able to read frequently asked question.

Administrator

- The page where admin will manage the website and perform administrator related tasks.

2.2 HARDWARE INTERFACE

- Complete Set of Desktop or Laptop Computer to browser OrangeCode web site.

2.3 SOFTWARE INTERFACE

- Unity3D Game Development Engine to develop games.
- PHP Laravel Framework and its dependencies (Composer, Form Collective, CKEditor, Bootstrap)

- Web Server to host a server for the community to access OrangeCode
- MySQL Server to store and retrieve data and related items from OrangeCode
- Visual Code IDE and Sublime Text to write most of the codes and scripts.
- Web Browsers (Chrome, Firefox, Internet Explorer, Edge) to browser OrangeCode web site.

3. SOFTWARE PRODUCT FEATURES

OrangeCode consists of six main functions, which are the Create Account, Create Class, Play Games, Generate Performance Report, Post Forum and Manage General Admin Task. The details of each functionality is explained below.

3.1 SOFTWARE PRODUCT FEATURES

3.1.1 Create Account

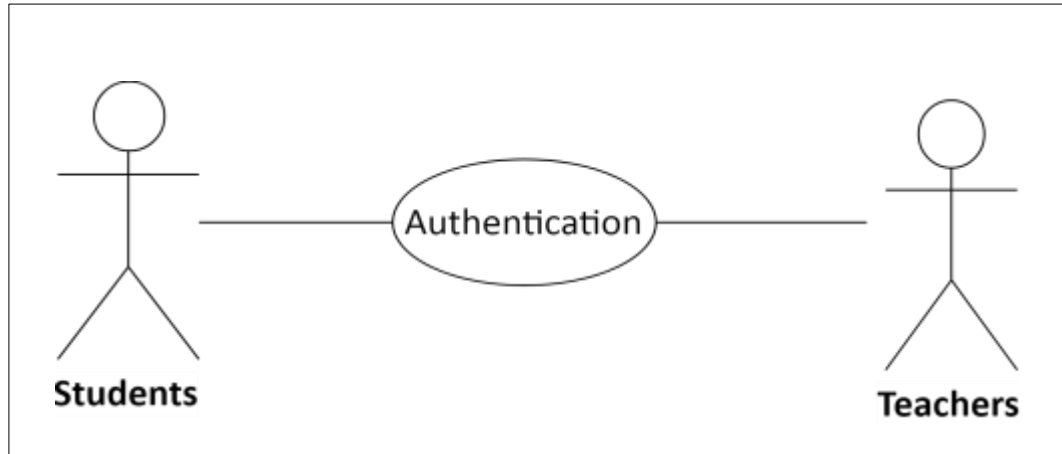


Figure 3.1 Authentication

Table 3.1 Authentication Use Case

Use Case ID	OC_UC01
Brief Description	This use case is about the creation of account and log in before able to access extra features of OrangeCode.
Actor	Students, Teachers
Pre-Conditions	<ul style="list-style-type: none"> OrangeCode server is online and available
Basic Flow	<p>B-1: Register</p> <ol style="list-style-type: none"> This use case begins when the user clicks on the “Register” button at the main screen. OrangeCode redirects user to the registration screen. OrangeCode displays registration form. User fills each of the form input. User clicks “Submit” button. OrangeCode validates form input. (C-1: Input Box Constraints) (E-1: Invalid or Empty Input Box)

	<ol style="list-style-type: none"> 7. OrangeCode displays success message. 8. OrangeCode redirect user to the dashboard. 9. The use case ends. <p>B-2: Login</p> <ol style="list-style-type: none"> 1. This use case begins when the user clicks on the “Login” button at the main screen. 2. OrangeCode redirects user to the login screen. 3. OrangeCode displays login form. 4. User fills each of the form input. 5. User clicks “Login” button. 6. OrangeCode validates form input. (C-1: Input Box Constraints) (E-1: Invalid or Empty Input Box) 7. OrangeCode displays success message. 8. OrangeCode redirect user to the dashboard. 9. The use case ends.
Alternative Flow	N/A
Exception Flow	<p>E-1: Invalid or Empty Input Box</p> <ol style="list-style-type: none"> 1. OrangeCode displays “Invalid Value” message. 2. OrangeCode prompts for correct value. 3. The use case continues.
Post-Conditions	<p>For B-1: Account successfully registered.</p> <p>For B-2: Account successfully logged in.</p>
Rules	N/A
Constraints	<p>C-1: Input Box Constraints</p> <ol style="list-style-type: none"> 1. Username must not contain special characters. 2. Username must be at least three characters long and not exceed 12 characters. 3. Password must be at least six characters long. 4. Email must follow standard email format.

Sequence Diagram	Refer Appendix A-1 1.1 : Basic Flow 1.2 : Exception Flow
-------------------------	--

3.1.2 Create Class

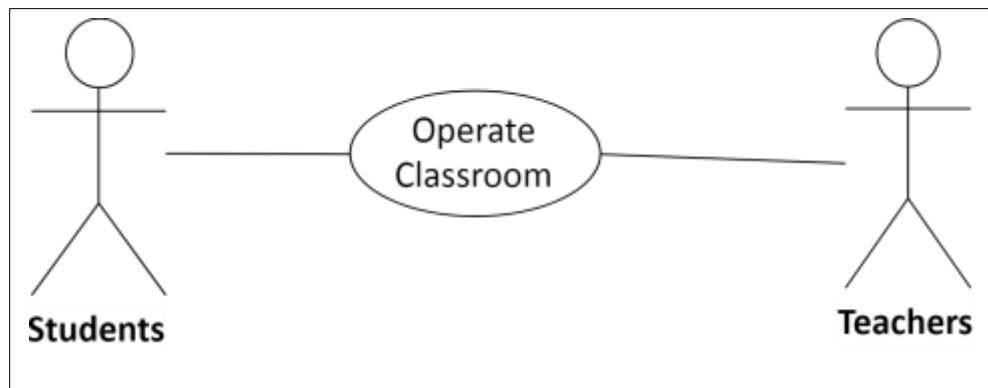


Figure 3.2 Operate Classroom Use Case Diagram

Table 3.2 Create Classroom Use Case

Use Case ID	OC_UC02
Brief Description	This use case is about the virtual classroom creation process.
Actor	Teachers, Students
Pre-Conditions	<ul style="list-style-type: none"> User must be logged in.
Basic Flow	<p>B-1: Create Classroom (Teachers)</p> <ol style="list-style-type: none"> The use case begins when the user clicks on “Create New Classroom” button in classroom home page. OrangeCode displays classroom registration form. User fills each of the form input. User clicks “Submit” button. OrangeCode validates form input. (C-1: Input Box Constraints) (E-1: Invalid or Empty Input Box) OrangeCode displays success message. OrangeCode redirects to classroom home screen. The use case ends.

B-2: Edit Classroom (Teachers)

1. The use case begins when the user clicks on “Edit” button in classroom home page.
2. OrangeCode displays classroom edit form.
3. User fills each of the form input.
4. User clicks “Update” button.
5. OrangeCode validates form input. **(C-1: Input Box Constraints) (E-1: Invalid or Empty Input Box)**
6. OrangeCode displays success message.
7. OrangeCode redirects to classroom home screen.
8. The use case ends.

B-3: Delete Classroom (Teachers)

1. The use case begins when the user clicks on “Delete” button in classroom home page.
2. OrangeCode displays warning message.
3. User clicks “Ok”.
4. OrangeCode deletes classroom.
5. OrangeCode refreshen page.
6. OrangeCode displays success message.
7. The use case ends.

B-4: Join Classroom (Students)

1. The use case begins when the user clicks on “Join Class” button in classroom view page.
2. OrangeCode pops-up a dialog message with secret code input box.
3. User enters secret code. **(E-1: Invalid or Empty Input Box)**
4. OrangeCode displays success message.

	<p>5. User is registered under chosen classroom.</p> <p>6. The use case ends.</p> <p>B-5: Leave Classroom (Students)</p> <p>1. The use case begins when the user clicks on “Leave Class” button in classroom view page.</p> <p>2. OrangeCode pops-up a warning message.</p> <p>3. User clicks “Ok”.</p> <p>4. OrangeCode displays success message.</p> <p>5. User is removed as student from the classroom.</p> <p>6. The use case ends.</p> <p>B-6: Calculate Classroom Performance</p> <p>1. The use case begins when any user view a classroom information.</p> <p>2. OrangeCode fetches GameData data of students in the classroom.</p> <p>3. OrangeCode perform calculation on performance. [R-1: Calculate User Account Performance Report] [R-2: Calculate Class Performance Report]</p> <p>4. OrangeCode displays performance.</p> <p>5. The use case ends.</p>
Alternative Flow	N/A
Exception Flow	<p>E-1: Invalid or Empty Input Box</p> <p>1. OrangeCode displays “Invalid Value” message.</p> <p>2. OrangeCode prompts for correct value.</p> <p>The use case continues.</p>
Post-Conditions	<p>For B-1: Classroom successfully created.</p> <p>For B-2: Classroom successfully updated.</p> <p>For B-3: Classroom successfully deleted.</p> <p>For B-4: User successfully joined classroom.</p>

	<p>For B-5: User successfully leave classroom.</p> <p>For B-6: Performance report displayed.</p>
Rules	<p>R-1: Calculate User Account Performance Report</p> <hr/> $\text{Game Performance, GP} = \frac{S}{\text{MaxS}} \times 100$ <p>Where</p> <p>S = Final score of current game</p> <p>MaxS = Maximum score achievable from current game</p> <hr/> $\text{Chapter Performance, ChP} = \frac{\sum_{i=1}^n \text{GP}}{n * \text{MaxS}} \times 100$ <p>Where</p> <p>GP = Game performance</p> <p>n = Number of games played</p> <p>Maxs = Maximum score achievable from the game</p> <hr/> $\text{Overall Performance, OP} = \frac{\sum_{i=1}^n \text{ChP}}{n} \times 100$ <p>Where</p> <p>ChP = Chapter performance</p> <p>n = Number of chapters</p> <hr/> <p>R-2: Calculate Class Performance Report</p>

	$\text{Class Performance} = \frac{\sum_{i=1}^n \text{OP}}{n} \times 100$ <p>Where</p> <p>OP = Overall performance of a student</p> <p>n = Number of students</p>
Constraints	C-1: Input Box Constraints <ol style="list-style-type: none"> 1. Class name must not contain any special character. 2. Class name must be at least five characters long and must not exceed twenty-four characters
Sequence Diagram	Refer Appendix A-2 2.1 - Basic Flow 2.2 - Alternative Flow 2.3 - Exception Flow

3.1.3 Play Games

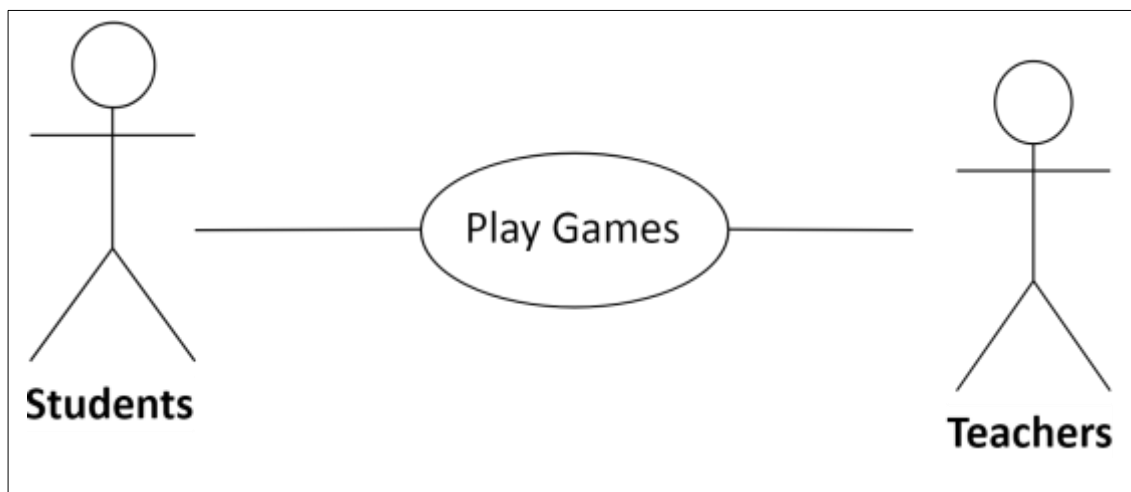


Figure 3.3 Play Games Use Case Diagram

Table 3.3 Play Games Use Case

Use Case ID	OC_UC03
Brief Description	This use case is about the flow process of users to play games.
Actor	Students, Teachers

Pre-Conditions	<ul style="list-style-type: none"> Users had successfully logged in
Basic Flow	<ol style="list-style-type: none"> The use case begins when the user click "Play Now" button. OrangeCode loads game environment. OrangeCode displays game interface. User selects the chapter to play. OrangeCode loads and displays the game scene. User interacts with specified inputs. (A-1: Pause and Exits) OrangeCode ends the game scene when user reached the losing condition or the winning condition. OrangeCode shows user current game performance. OrangeCode records user current game performance into account. OrangeCode displays the game interface and is idle until the next user action. The use case ends.
Alternative Flow	<p>A-1: Pause and Exits</p> <ol style="list-style-type: none"> User clicks in-game pause button. User clicks exits. OrangeCode displays confirmation message box. User clicks ok. Use case jumps to basic flow 10.
Exception Flow	N/A
Post-Conditions	OrangeCode load games and accept inputs.
Rules	N/A
Constraints	N/A
Sequence Diagram	<p>Refer Appendix A-3</p> <p>3.1: Basic Flow</p> <p>3.2: Alternative Flow</p>

3.1.4 Manage Profile

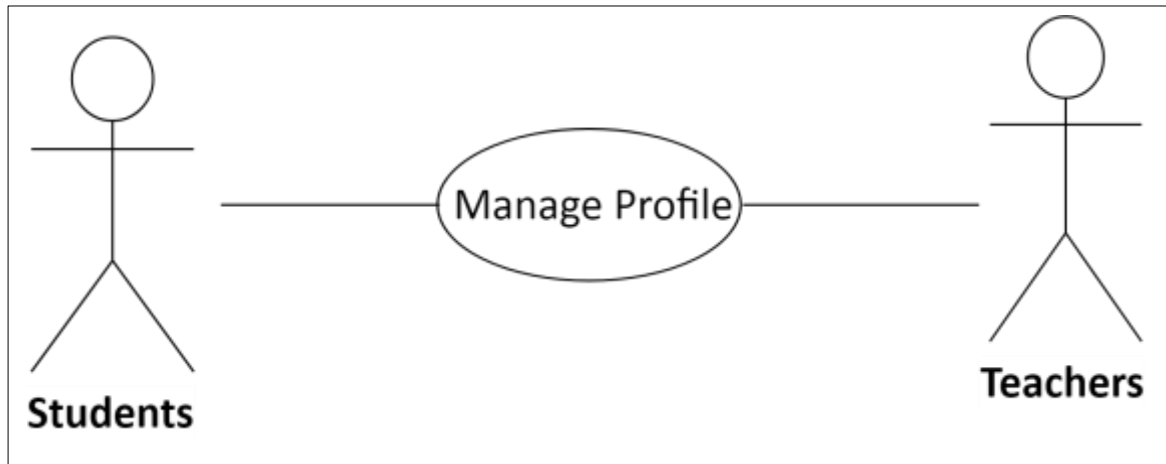


Figure 3.4 Manage Profile Use Case Diagram

Table 3.4 Manage Profile Use Case

Use Case ID	OC_UC04
Brief Description	This use case is about managing profile information.
Actor	Students, Teachers
Pre-Conditions	<ul style="list-style-type: none"> User had successfully logged in into OrangeCode
Basic Flow	<ol style="list-style-type: none"> The use case begins when user click on "Edit Profile" button in View Profile page. OrangeCode enable form inputs. User enters any input changes. User clicks "Update" button. OrangeCode updates database with any changes made to the user profile. OrangeCode displays success message. The use case ends.
Alternative Flow	N/A
Exception Flow	N/A
Post-Conditions	User's profile information updated.
Rules	N/A
Constraints	N/A

Sequence Diagram	Refer Appendix A-4 4.1: Basic Flow
-------------------------	---------------------------------------

3.1.5 Operate Forum

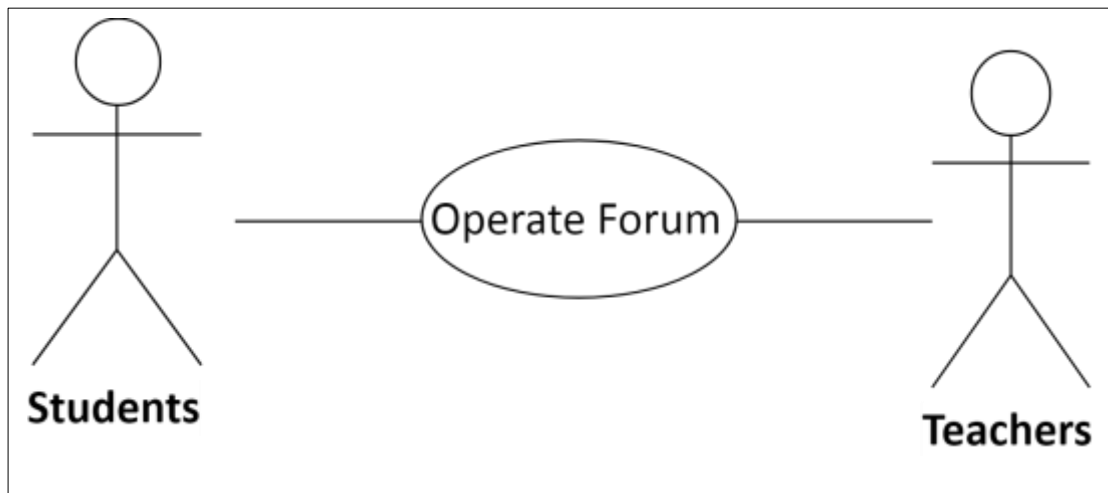


Figure 3.5 Operate Forum Use Case Diagram

Table 3.5 Operate Forum

Use Case ID	OC_UC05
Brief Description	This use case is about the flow of user in operating forum.
Actor	Students, Teachers
Pre-Conditions	<ul style="list-style-type: none"> Users must successfully logged in
Basic Flow	<ol style="list-style-type: none"> The use case begins when the user clicks "Forum" button. OrangeCode displays list of forum by categories:- <ol style="list-style-type: none"> Announcement Queries and Help Off Topics User selects any of the forum categories. OrangeCode redirects user to the specific forum page. Users can perform the following functions:- <ol style="list-style-type: none"> Create a new forum Reply forum post OrangeCode provide title box and body box for user to fills. User fills the textboxes.

	<p>8. User clicks submit.</p> <p>OrangeCode validates textboxes and conforms to the user. (C-1: Forum Post Constraints) (E-1: Invalid or Empty Input Box)</p> <p>9. User clicks ok.</p> <p>10. OrangeCode posts forum based on user input.</p> <p>11. OrangeCode refresh page.</p> <p>12. The use case ends.</p>
Alternative Flow	N/A
Exception Flow	<p>E-1: Invalid or Empty Input Box</p> <p>1. OrangeCode displays “Invalid Value” message.</p> <p>2. OrangeCode prompts for correct value.</p> <p>3. The use case continues.</p>
Post-Conditions	A post in forum successfully posted.
Rules	N/A
Constraints	<p>C-1: Forum Post Constraints</p> <p>1. Title must not exceed more than 100 characters.</p> <p>2. Contents must not exceed 1000 characters.</p>
Sequence Diagram	<p>Refer Appendix A-5</p> <p>5.1: Basic Flow</p>

3.1.6 Administrator Task

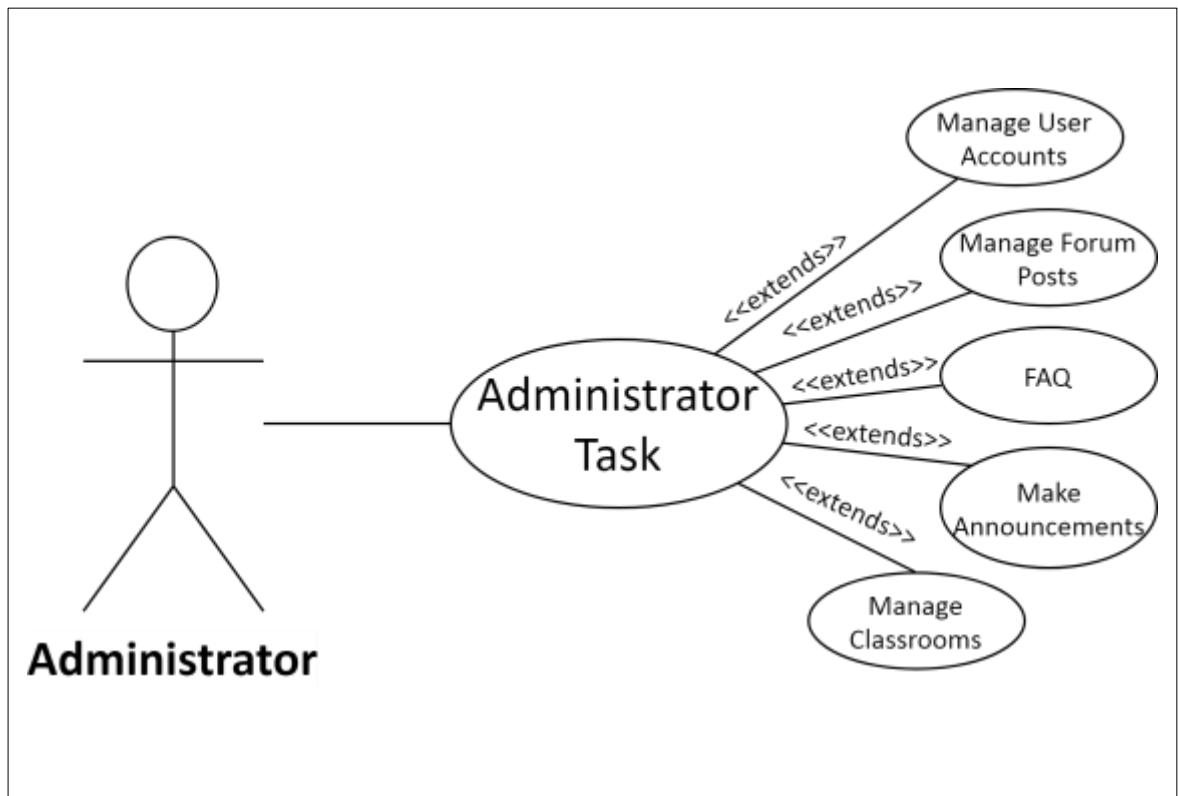


Figure 3.6 Administrator Task Use Case Diagram

Table 3.6 Use Case Name

Use Case ID	OC_UC06
Brief Description	This use case is about the flow of administrator tasks.
Actor	Administrator
Pre-Conditions	<ul style="list-style-type: none"> Administrator successfully logged in
Basic Flow	<ol style="list-style-type: none"> The use case begins when admin clicks on the “Management” button. OrangeCode displays the administrator functions screen. User are able to perform the following functions:- <ol style="list-style-type: none"> Search User Accounts (A-1: Manage User Accounts) Search Forum Posts (A-2: Manage Forum Posts) Search Classrooms (A-3: Manage

	<p style="text-align: center;">Classrooms)</p> <ol style="list-style-type: none"> 4. OrangeCode idle and awaits user action. 5. The use case ends.
Alternative Flow	<p>A-1: Manage User Accounts</p> <ol style="list-style-type: none"> 1. OrangeCode displays list of users account. 2. User clicks on any user name. 3. OrangeCode displays the user information. 4. User clicks on “Delete” button. 5. OrangeCode delete the user from database. 6. Use case continues. <p>A-2: Manage Forum Posts</p> <ol style="list-style-type: none"> 1. OrangeCode displays list of forum posts with information. 2. User clicks on any forum post. 3. OrangeCode displays the forum information. 4. User clicks on “Delete” button. 5. OrangeCode delete the forum from database. 6. Use case continues. <p>A-3: Manage Classrooms</p> <ol style="list-style-type: none"> 1. OrangeCode displays list of classrooms with information. 2. User clicks on any classroom. 3. OrangeCode displays the classroom information. 4. User clicks on “Delete” button. 5. OrangeCode delete the classroom from database. 6. Use case continues. <p>A-4: FAQ</p> <ol style="list-style-type: none"> 1. Admin click FAQ menu button. 2. OrangeCode displays types of FAQ. 3. Admin may perform the following functions:-

	<ol style="list-style-type: none"> a. Insert new FAQ (C-1: FAQ Constraints) b. Edit existing FAQ c. Remove existing FAQ d. View existing FAQ <ol style="list-style-type: none"> 4. Use case continues <p>A-4: Announcements</p> <ol style="list-style-type: none"> 1. Admin click Forum menu button. 2. OrangeCode displays forum page. 3. Admin clicks announcement forum. 4. Users can perform the following functions:- 5. Create a new forum 6. Reply forum post 7. OrangeCode provide title box and body box for admin to fills. 8. Admin fills the textboxes. 9. Admin clicks submit. 10. OrangeCode validates textboxes and conforms to the user. (C-2: Forum Post Constraints) (E-1: Invalid or Empty Input Box) 11. Admin clicks ok. 12. OrangeCode posts forum based on user input. 13. OrangeCode refreshen page. 14. OrangeCode displays success message. 15. Use case continues.
Exception Flow	<p>E-1: Invalid or Empty Input Box</p> <ol style="list-style-type: none"> 1. OrangeCode displays “Invalid Value” message. 2. OrangeCode prompts for correct value. 3. The use case continues.
Post-Conditions	<ul style="list-style-type: none"> • Users’ account status updated. • Forum posts successfully removed. • FAQ successfully updated and stored into database. • Announcement will successfully publish.

Rules	N/A
Constraints	<p>C-1: FAQ Constraints</p> <ol style="list-style-type: none">1. FAQ title must not exceed 50 characters.2. FAQ title must not contain any special characters.3. FAQ content must not exceed 1000 characters. <p>C-2: Forum Post Constraints</p> <ol style="list-style-type: none">1. Announcement title must not exceed 100 characters.2. Announcement title must not contain any special characters.3. Announcement description must not exceed 2000 characters.
Sequence Diagram	<p>Refer Appendix A-6</p> <p>6.1: Basic Flow</p> <p>6.2: Alternative Flow 1</p> <p>6.3: Alternative Flow 2</p> <p>6.4: Alternative Flow 3</p> <p>6.5: Alternative Flow 4</p>

4. REQUIREMENT TRACEABILITY

This section defines the list of requirements with their sources. This is to trace and filter whether the requirements really meets what the client is expecting.

Table 4.1 Requirement Traceability for Use Case Authentication (OC_UC01)

No.	Requirement No.	Description	Source
1.	OC_UC01_01	User shall click "Create Account" button to register an account.	Existing System
2.	OC_UC01_02	User able to perform the following functions:- a. Register as Student b. Register as Teacher	Existing System
3.	OC_UC01_03	User shall fill the account registration form.	Existing System
4.	OC_UC01_04	User shall click "Register" button to submit the registration form.	Existing System
5.	OC_UC01_05	OrangeCode shall validate the registration form.	Existing System

Table 4.2 Requirement Traceability for Use Case Create Classroom (OC_UC02)

No.	Requirement No.	Description	Source
1.	OC_UC02_01	Teacher shall click "Create Class" button to register a class.	Existing System
2.	OC_UC02_02	User shall fill the class registration form.	Existing System
3.	OC_UC02_03	OrangeCode shall validate the registration form.	Existing System
4.	OC_UC02_03	OC execute systematic guide function of navigating the classroom interface.	Existing System
5.	OC_UC02_04	Classroom name shall not contain any special character.	
6.	OC_UC02_05	Class name must be at least five characters	

		long and not exceed twenty-four characters.	
--	--	---	--

Table 4.3 Requirement Traceability for Use Case Play Games (OC_UC03)

No.	Requirement No.	Description	Source
1.	OC_UC03_01	OrangeCode redirects user to game environment when user clicks “Play Now” button.	
2.	OC_UC03_02	Students or teachers shall able to select game chapter.	
3.	OC_UC03_03	Students or teachers shall able to interact with the game using specific inputs.	
4.	OC_UC03_04	Students or teachers shall able to pause the game.	
5.	OC_UC03_05	Students or teachers shall able to exit the game anytime in the pause option.	
6.	OC_UC03_06	OrangeCode end the game when user reached losing or winning condition.	
7.	OC_UC03_07	OrangeCode displays the user scoring and performance.	
8.	OC_UC03_08	OrangeCode records the user scoring and performance	
9.	OC_UC03_09	OrangeCode games will cover chapters of Programming syllabus, which are the Variables, Operators and Expressions, Arrays, and Functions.	

Table 4.4 Requirement Traceability for Use Case Manage Profile (OC_UC04)

No.	Requirement No.	Description	Source
1.	OC_UC04_01	Students or teachers able to view others or own profile.	
2.	OC_UC04_02	Students or teachers able to edit and update own profile information.	



Table 4.5 Requirement Traceability for Use Case Operate Forum (OC_UC05)

No.	Requirement No.	Description	Source
1.	OC_UC05_01	User is able to view forum threads and their posts.	
2.	OC_UC05_02	Forum shall be divided by categories / topics.	
3.	OC_UC05_03	User can create new forum thread.	
4.	OC_UC05_04	User can reply to forum post.	
5.	OC_UC05_05	User can input various elements into the reply form such as images, tables and etc.	

Table 4.6 Requirement Traceability for Use Case Administrator Tasks (OC_UC05)

No.	Requirement No.	Description	Source
1.	OC_UC06_01	Administrator is able search for user account, forum post or classrooms.	
2.	OC_UC06_05	Adminstrator can delete existing users account, forums or classrooms.	
3.	OC_UC06_07	Administrator can view, add, remove or edit FAQ.	
4.	OC_UC06_08	Administrator can publish announcement.	

5. SYSTEM REQUIREMENT APPROVAL

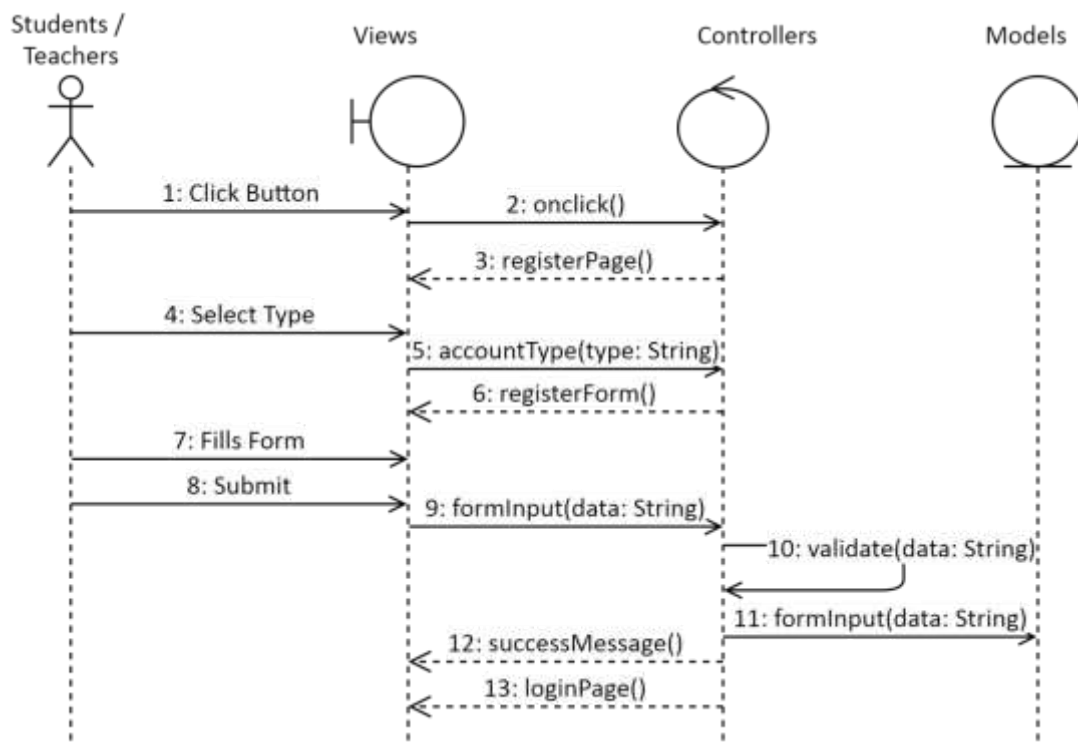
	Name	Date
Verified by:  Developer	NASRUL ARIFF BIN ZAKRIA	
Approved by:  Client	MUHAMMAD KHAIRIL AKMAL BIN MOHD KHAIRUDDIN	

APPENDIX A

Sequence Diagram

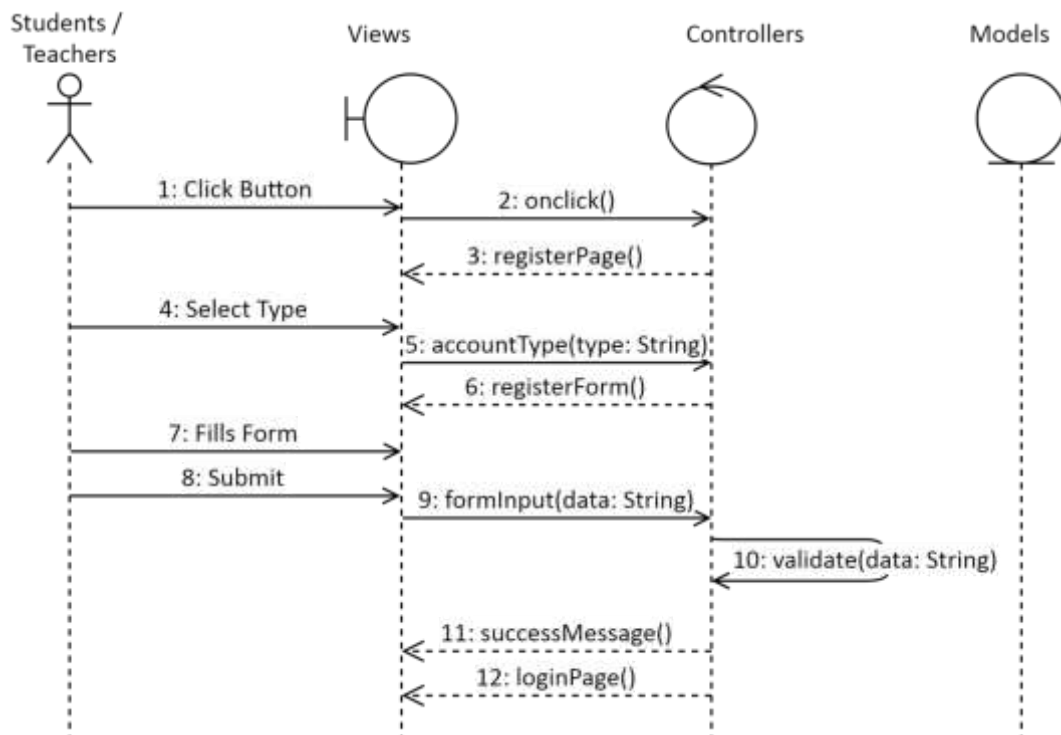
1. Authentication | Refer: OC_UC01

1.1. Basic Flow



Appendix A-1.1: Create Account Basic Flow Sequence Diagram

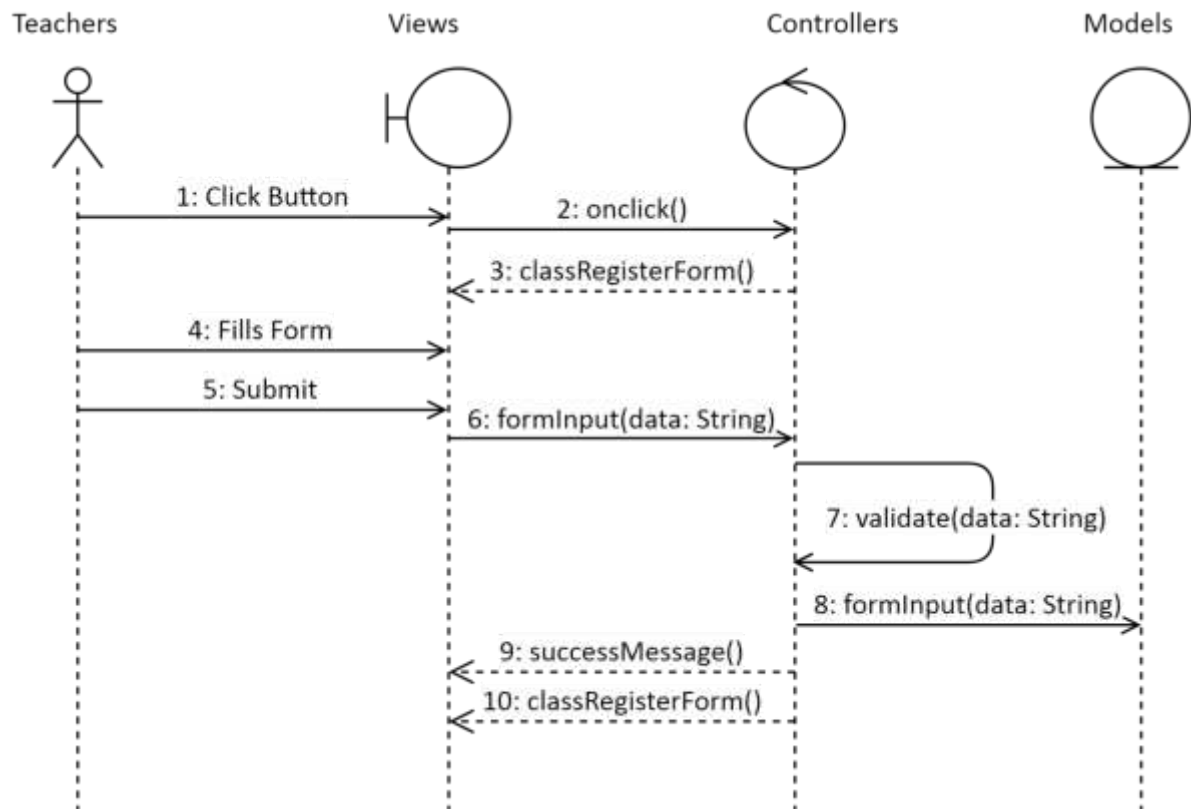
1.2. Exception Flow



Appendix A-1.2: Create Account Exception Flow Sequence Diagram

2. Create Classroom | Refer: OC_UC02

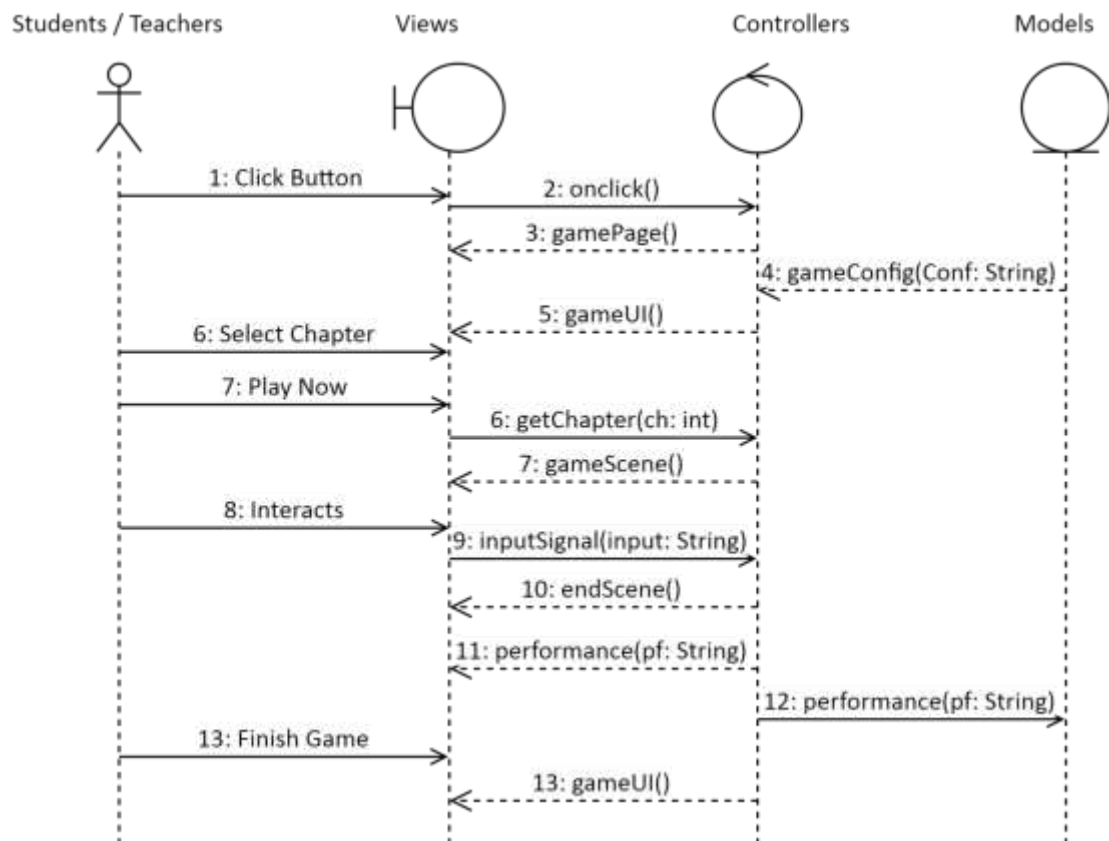
2.1. Basic Flow



Appendix A-2.1: Create Class Basic Flow Sequence Diagram

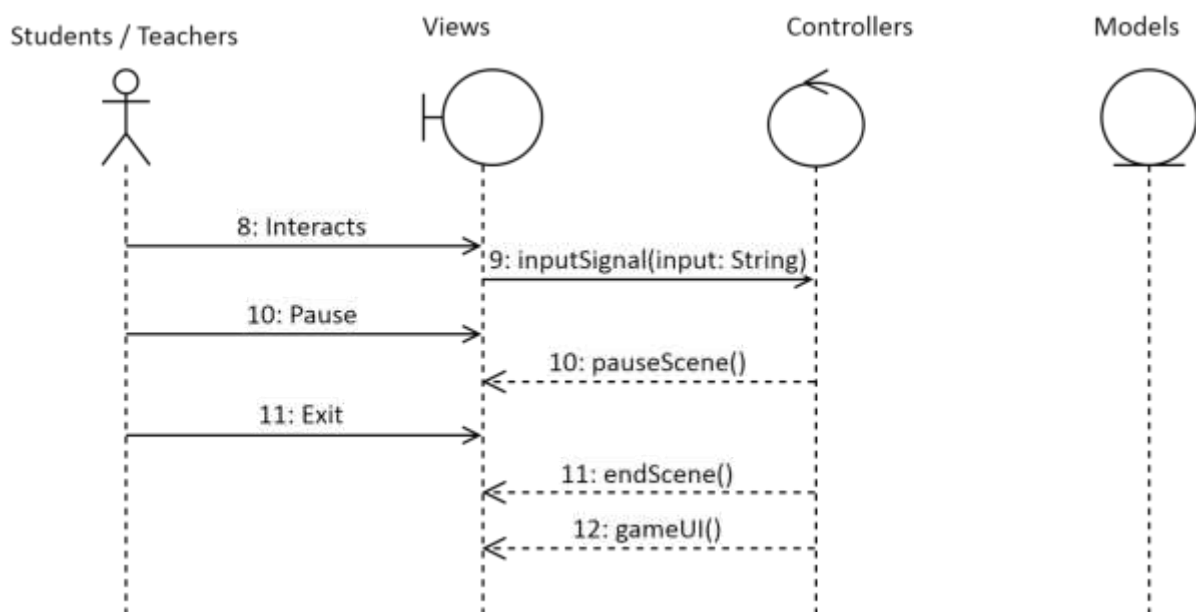
3. Play Games | Refer: OC_UC03

3.1. Basic Flow



Appendix A-3.1: Play Games Basic Flow Sequence Diagram

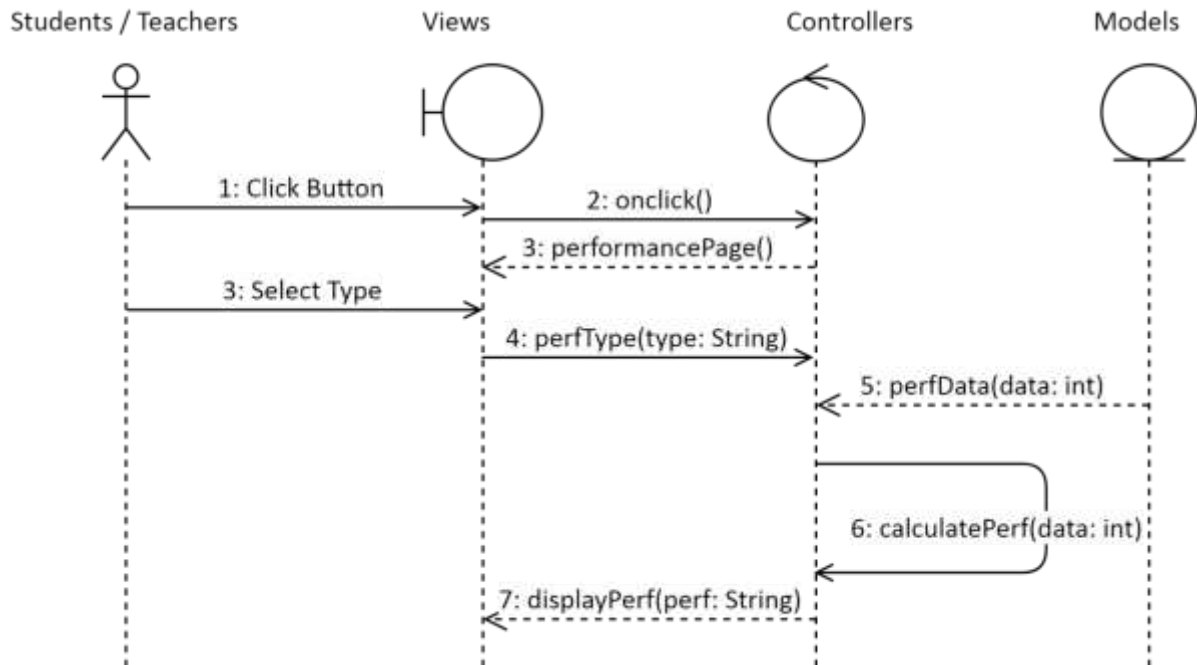
3.2. Alternative Flow



Appendix A-3.2: Play Games Alternative Flow Sequence Diagram

4. Generate Performance Report | Refer: OC_UC04

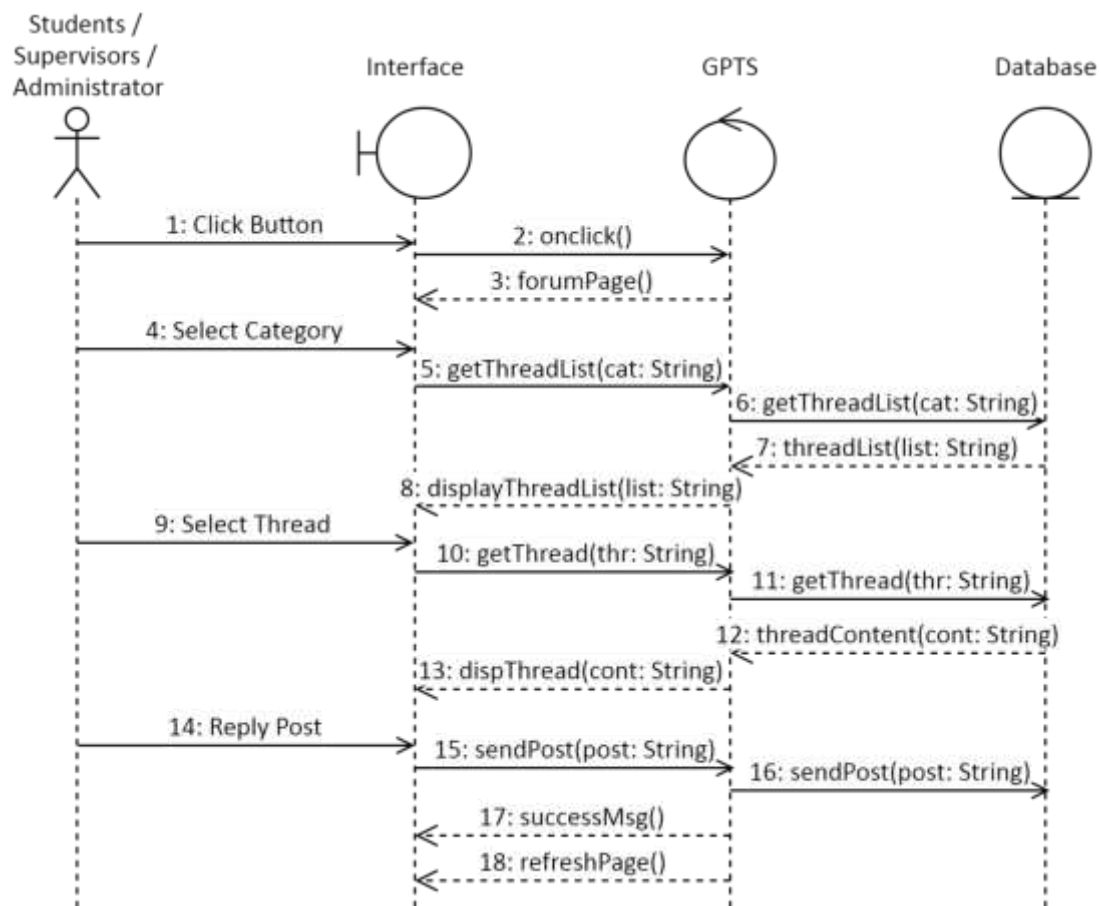
4.1. Basic Flow



Appendix A-4.1: Generate Performance Report Basic Flow Sequence Diagram

5. Operate Forum | Refer: OC_UC05

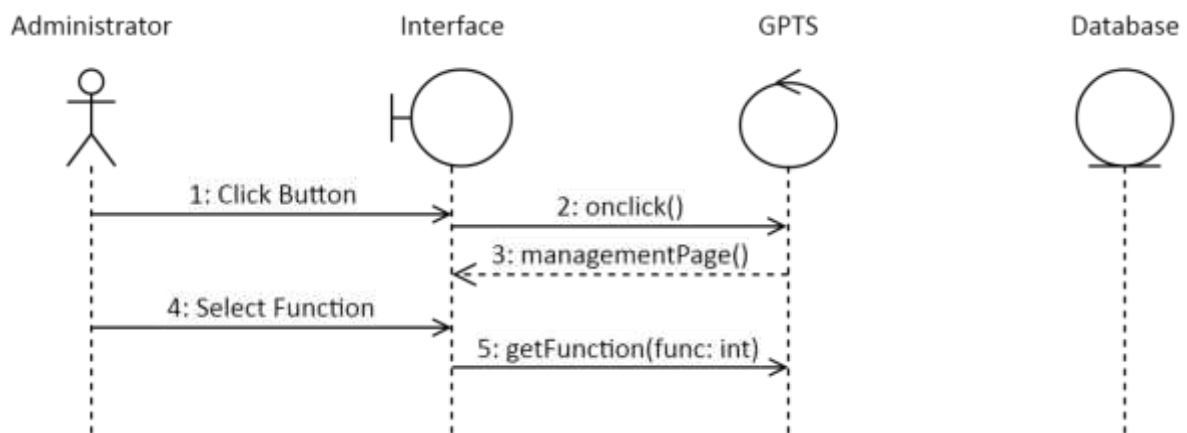
5.1. Basic Flow



Appendix A-5.1: Post Forum Basic Flow Sequence Diagram

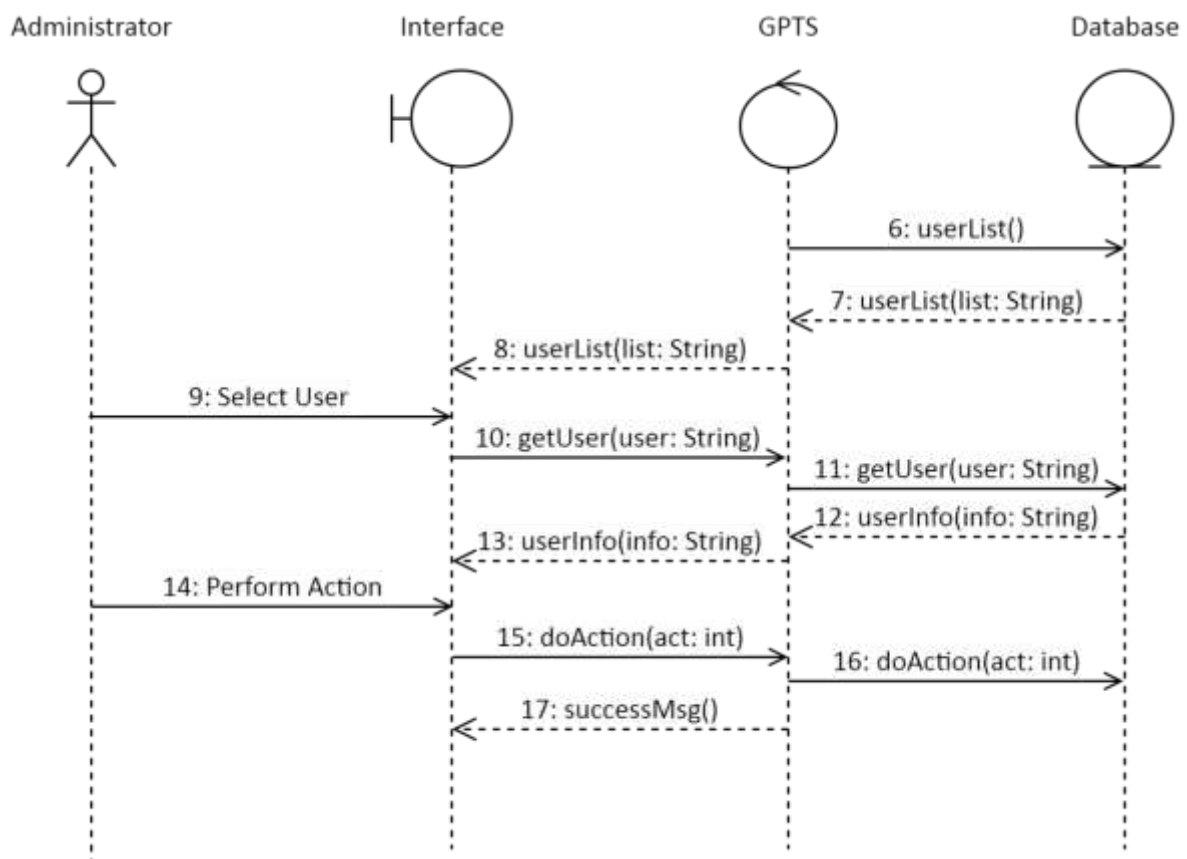
6. Administrator Task | Refer: OC_UC06

6.1. Basic Flow



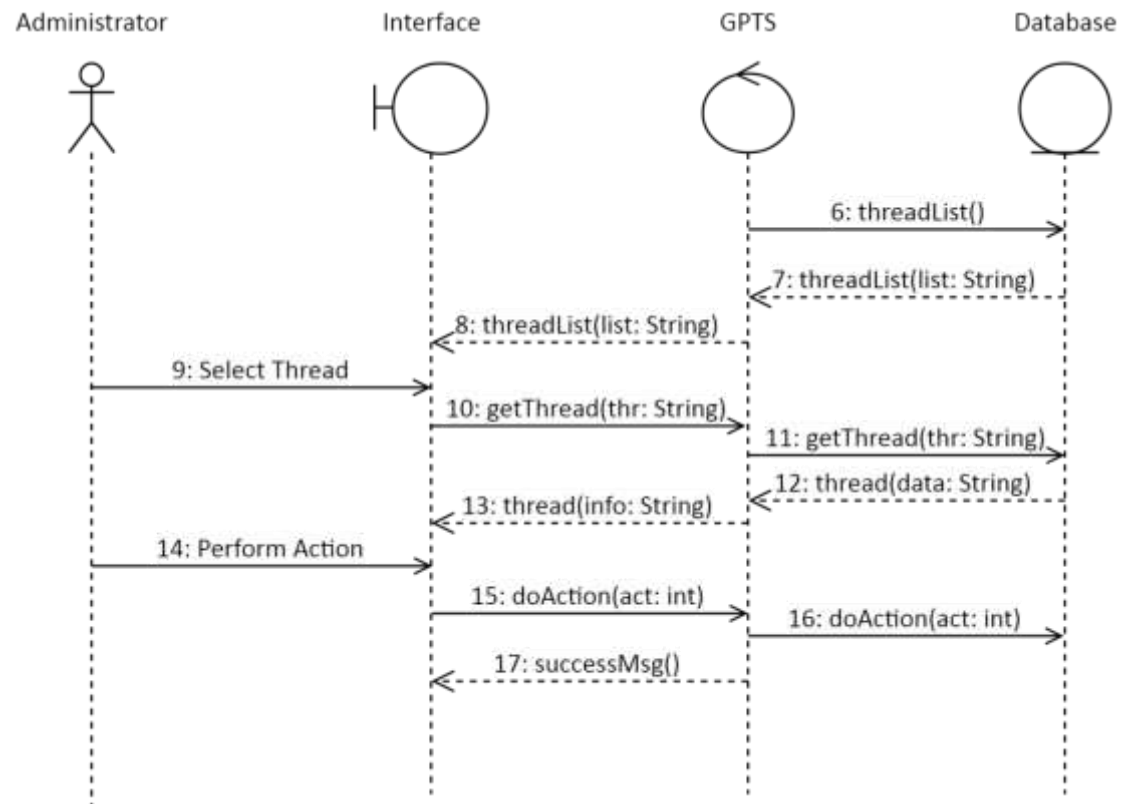
Appendix A-6.1: Manage General Admin Task Basic Flow Sequence Diagram

6.2. Alternative Flow 1 - User Manager



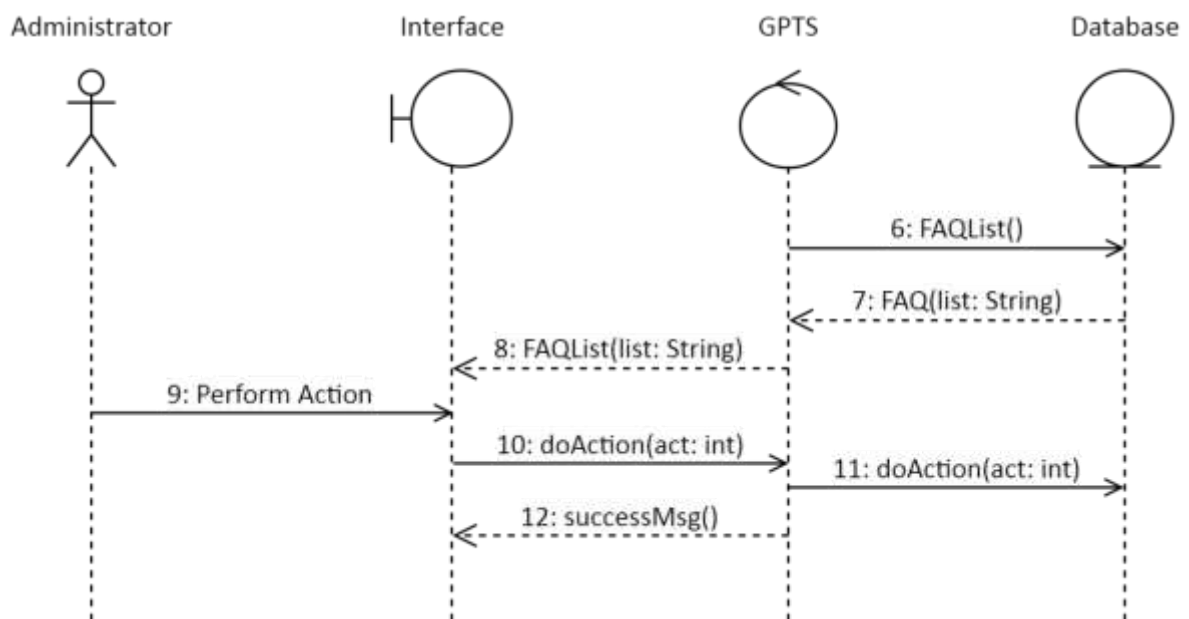
Appendix A-6.2: Manage General Admin Task Alternative Flow 1 Sequence Diagram

6.3. Alternative Flow 2 - Forum Action



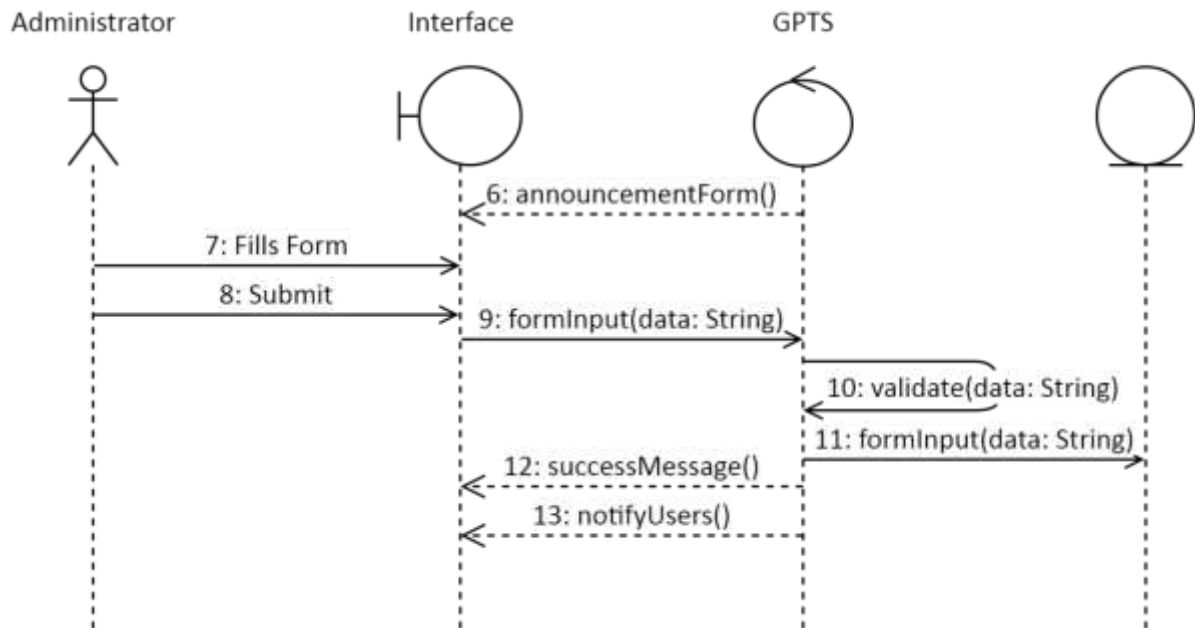
Appendix A-6.3: Manage General Admin Task Alternative Flow 2 Sequence Diagram

6.4. Alternative Flow 3 - FAQ



Appendix A-6.4: Manage General Admin Task Alternative Flow 3 Sequence Diagram

6.5. Alternative Flow 4 - Announcement

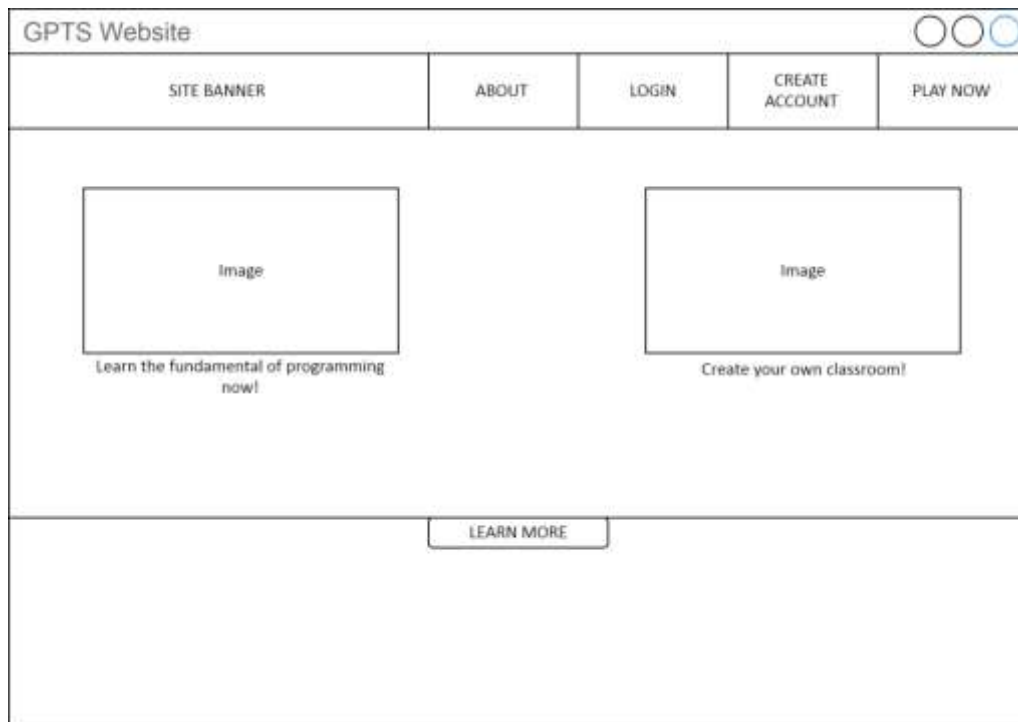


Appendix A-6.5: Manage General Admin Task Alternative Flow 4 Sequence Diagram

APPENDIX B

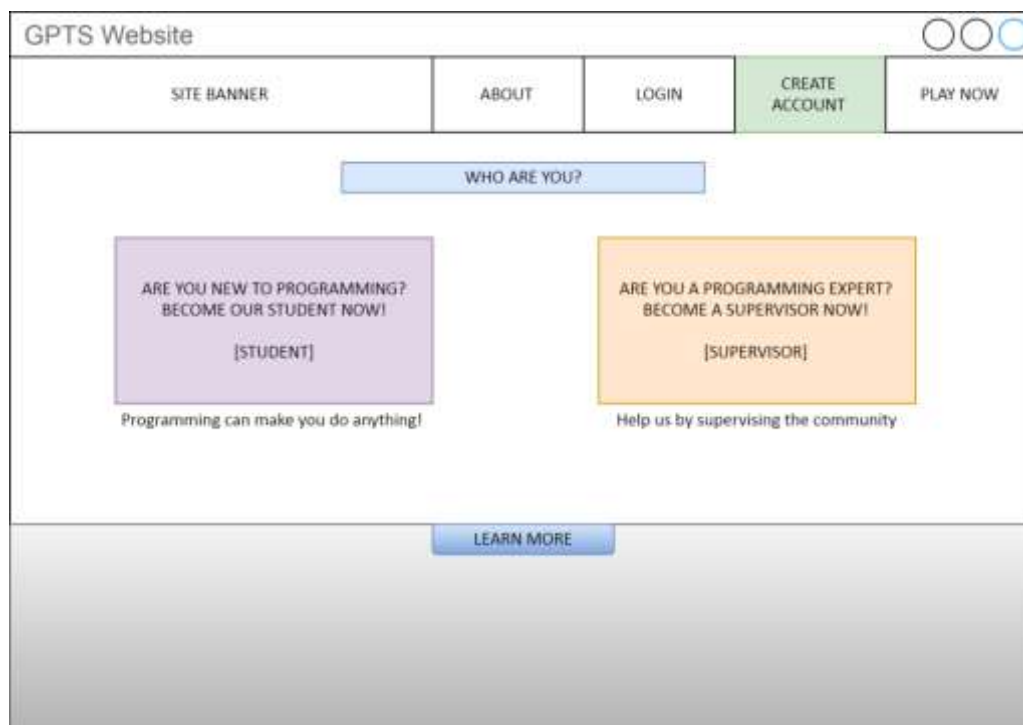
User Interfaces

1. Website Main Page



Appendix B-1: Main Website Page Interface

2. Create Account Page



Appendix B-2: Create Account Page

3. Create Account Page Student

The screenshot shows a web browser window titled "GPTS Website". The navigation bar includes "SITE BANNER", "ABOUT", "LOGIN", "CREATE ACCOUNT" (highlighted in green), and "PLAY NOW". Below the navigation bar is a blue button labeled "REGISTER AS STUDENT". The main content area contains a registration form for students, enclosed in a light orange rounded rectangle. The form fields are: USERNAME, PASSWORD, EMAIL, CONFIRM PASSWORD, CONFIRM EMAIL, MY AGE, and CLASS CODE. There is also a checkbox for "I agree with terms & policies" and a green "REGISTER" button at the bottom right of the form.

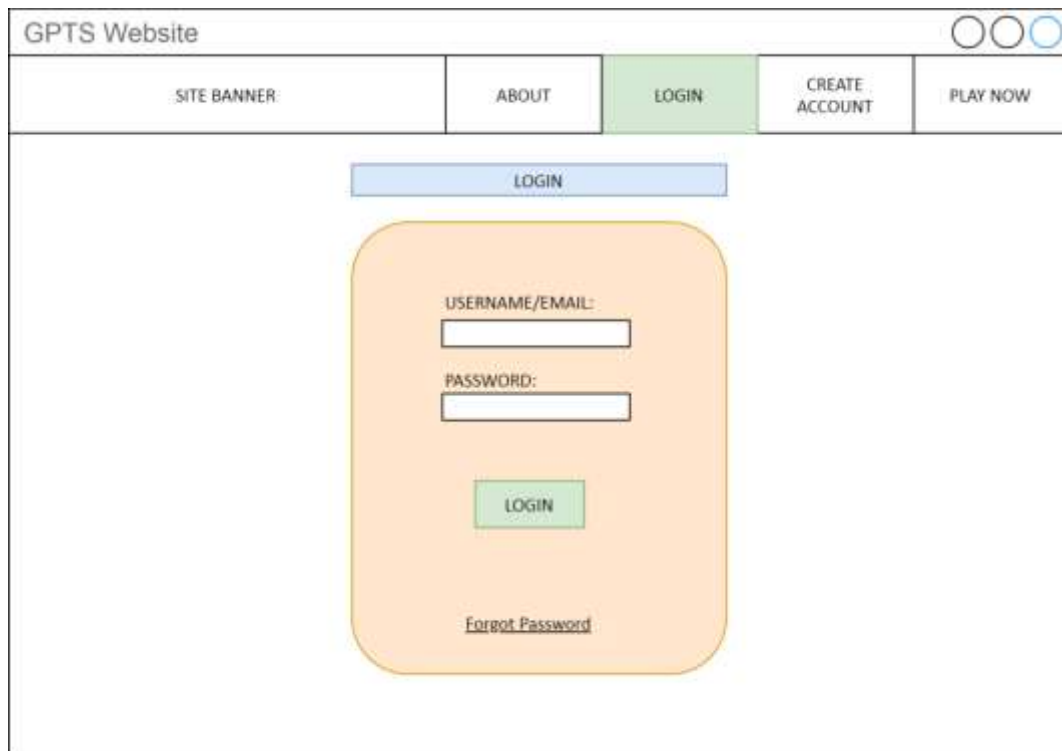
Appendix B-3: Create Account Page Student

4. Create Account Page Supervisor

The screenshot shows a web browser window titled "GPTS Website". The navigation bar includes "SITE BANNER", "ABOUT", "LOGIN", "CREATE ACCOUNT" (highlighted in green), and "PLAY NOW". Below the navigation bar is a blue button labeled "REGISTER AS SUPERVISOR". The main content area contains a registration form for supervisors, enclosed in a light orange rounded rectangle. The form fields are: USERNAME, PASSWORD, EMAIL, CONFIRM PASSWORD, CONFIRM EMAIL, MY AGE, and a checkbox for "I agree with terms & policies". There is a green "REGISTER" button at the bottom right of the form.

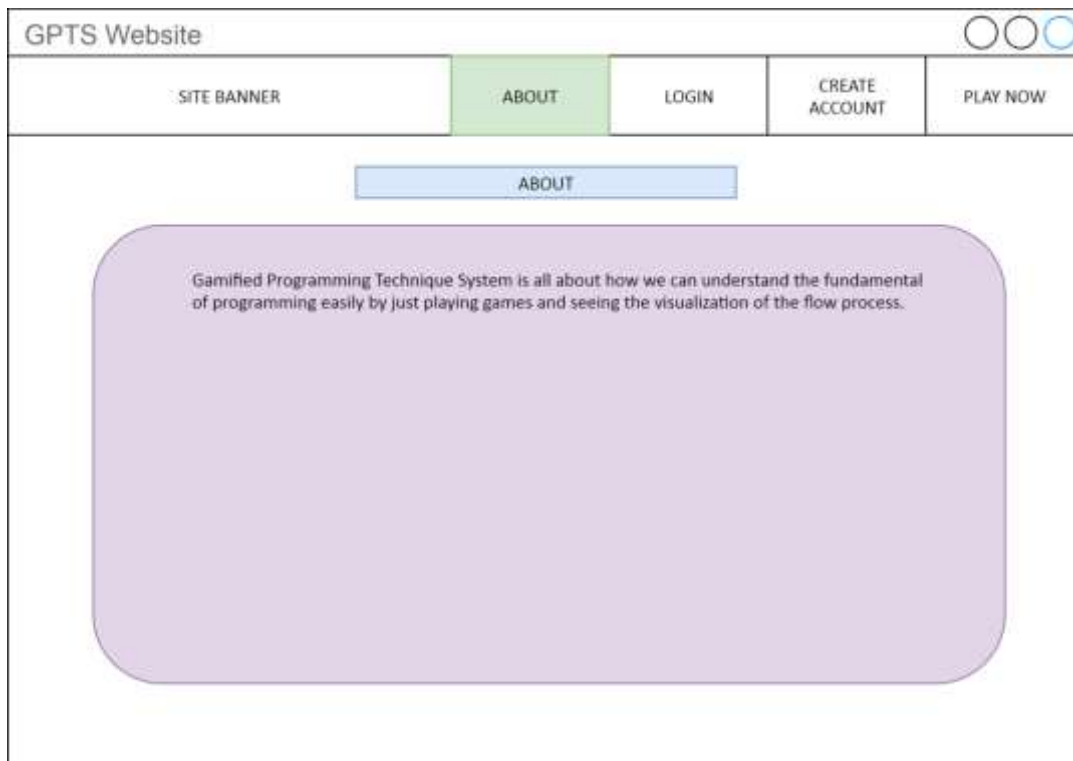
Appendix B-4: Create Account Page Supervisor

5. Login Screen



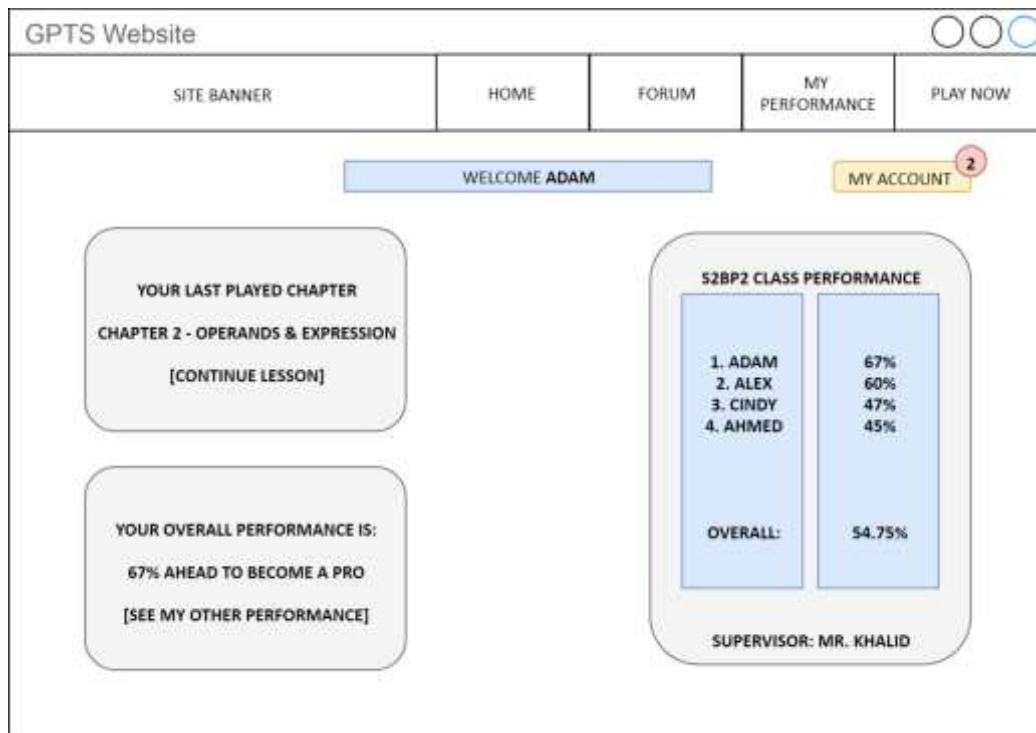
Appendix B-5: Login Screen

6. About



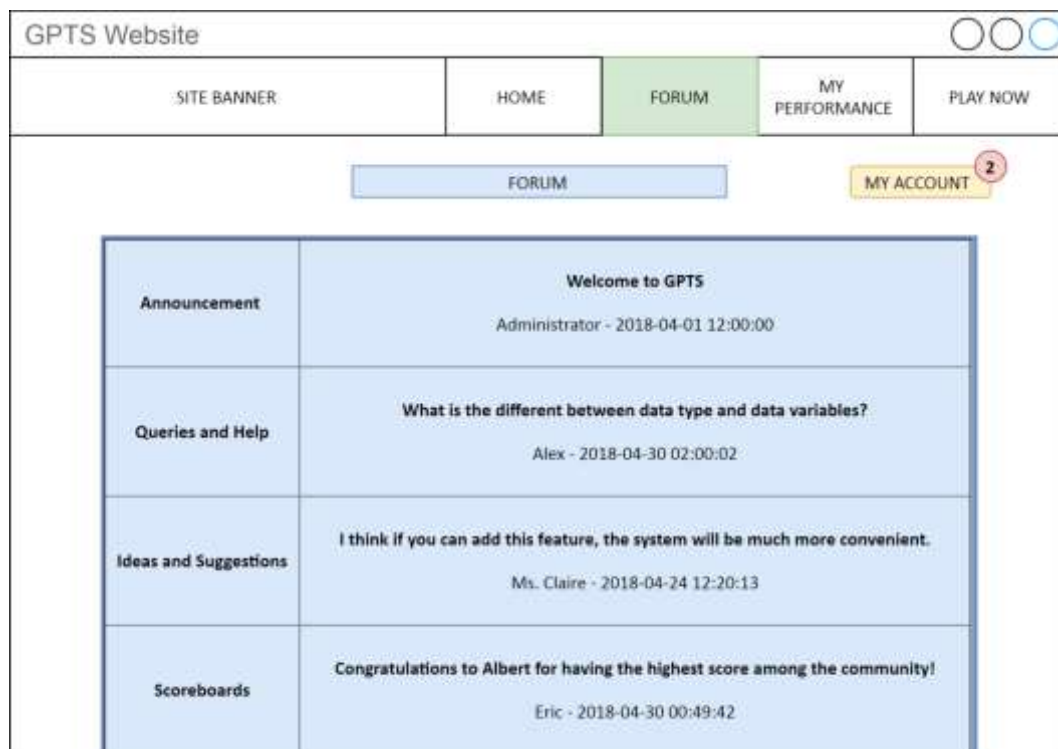
Appendix B-6: About

7. Homepage



Appendix B-7: Homepage (Students)

8. Forum



Appendix B-8: Forum

9. Forum Subcategories

User	Question Title	Timestamp
Alex [Student]	What is the different between data type and variables?	2018-04-30 02:00:02
Lina [Student]	What is the line "include<stdio.h> means?	2018-04-30 12:27:32
Albert [Student]	Expected ";" errors. Why I keep getting this?	2018-04-29 10:11:38
Jamal [Student]	Can I put If inside another If? Or while inside another while?	2018-04-29 09:20:57

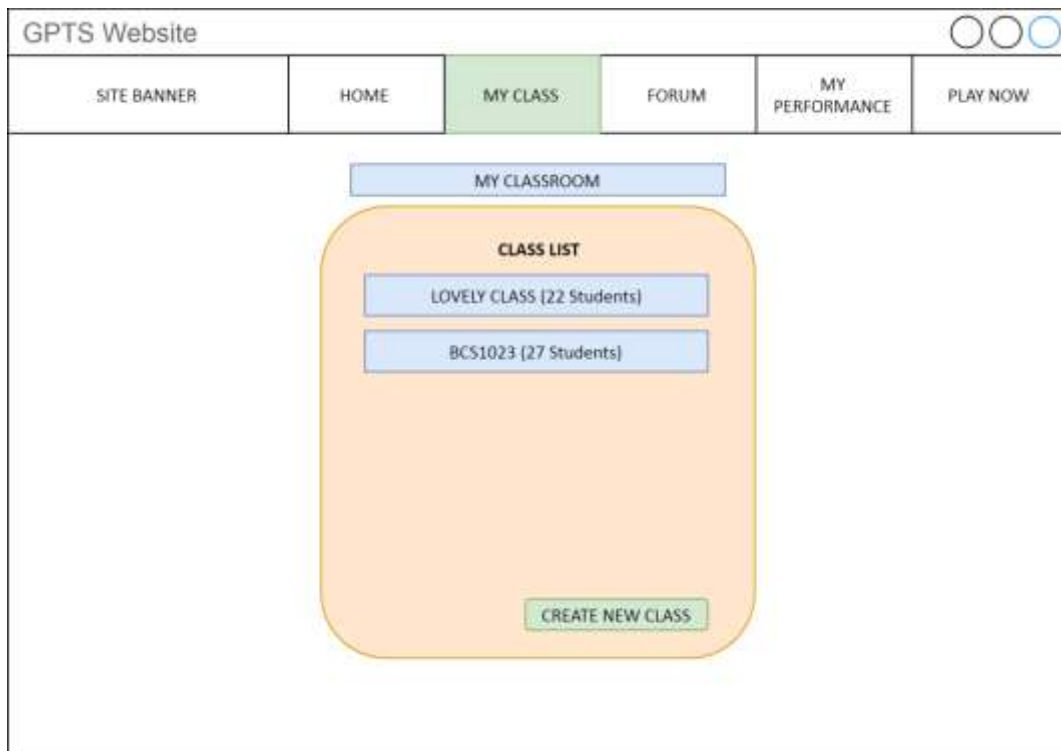
Appendix B-9: Forum Subcategories

10. Forum Thread

User	Message Content	Timestamp
Alex [Student]	Hey guys, what is the different between data type and variables? Because I cannot differentiate the different. When I declare a variable it is also called data type? Thank you.	2018-04-30 02:00:02
Mr. Khalid [Supervisor/Lecturer]	They are both different things but related. Let's take a coding example: int number; The data type will be the "int" and the variable will be "number" Data type basically means the type of data that the variable will be held. As in the example, "number" is a variable that hold "int" data type.	2018-04-30 02:01:31
Alex [Student]	Wow! Great explanation, now I understand what they are and how to differentiate them. Thank you Mr. Khalid.	2018-04-30 02:05:52
William [Student]	Hello, I am also new to programming and I was about to ask the same question. But thank you Mr. Khalid for the clear explanation.	

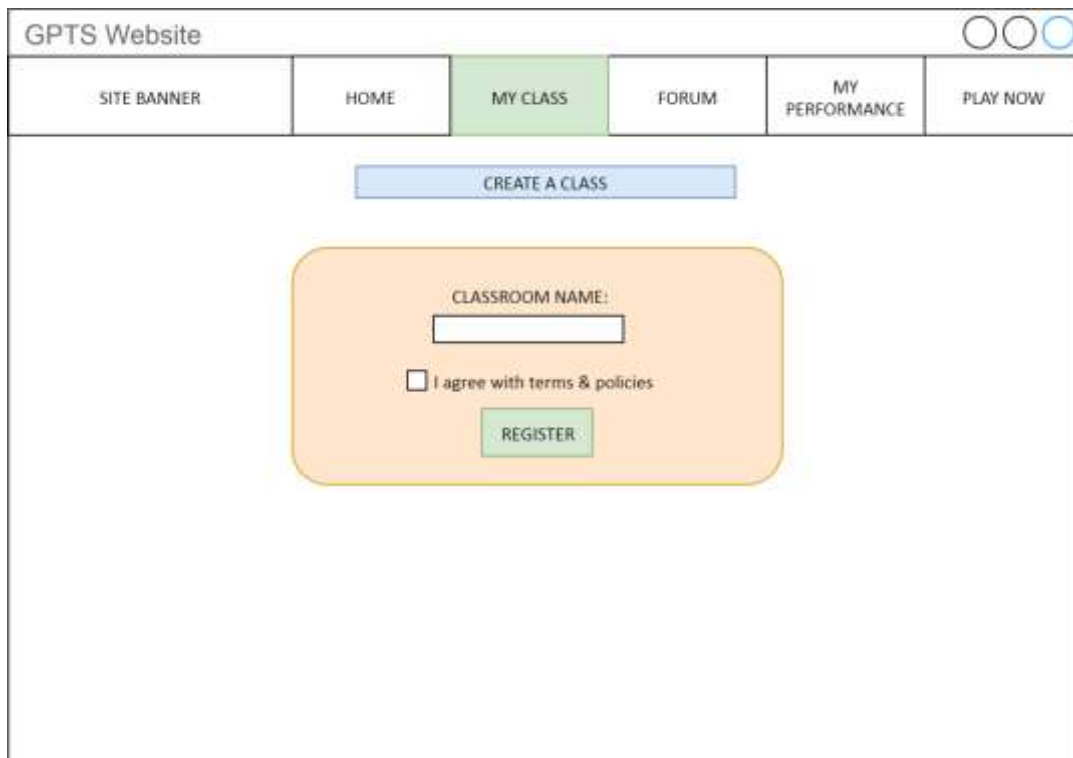
Appendix B-10: Forum Thread

11. Classroom Page



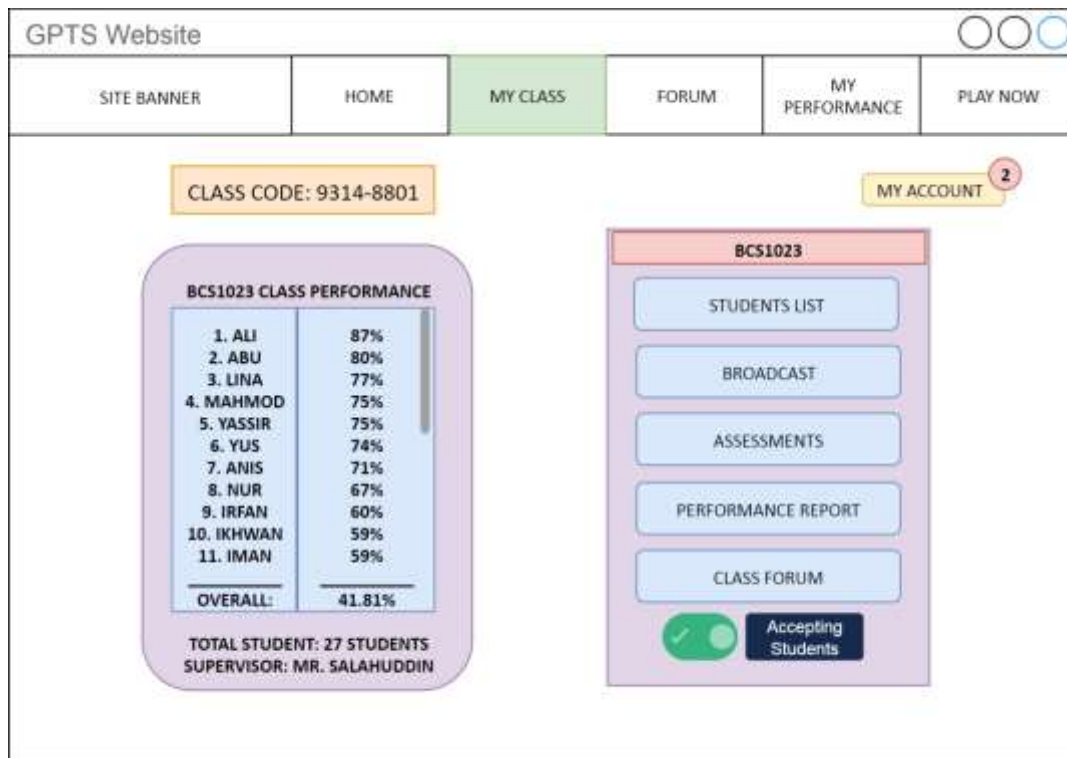
Appendix B-11: Classroom Page

12. Class Creation Page



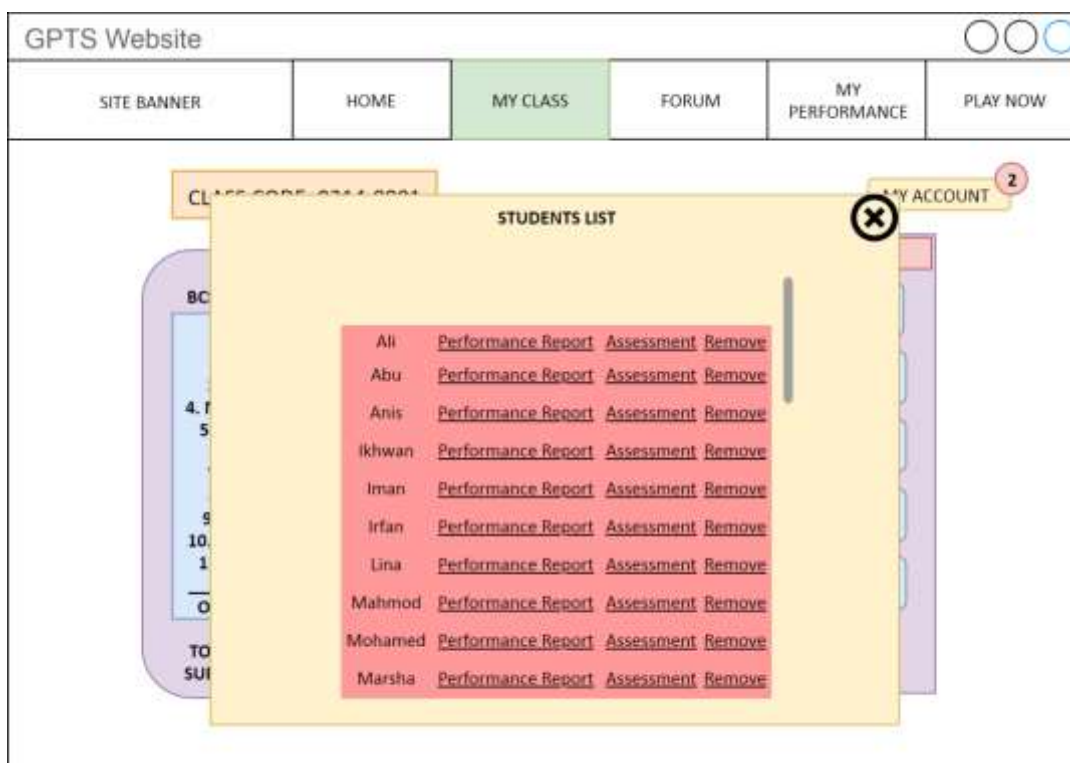
Appendix B-12: Class Creation Page

13. Class Manager Page



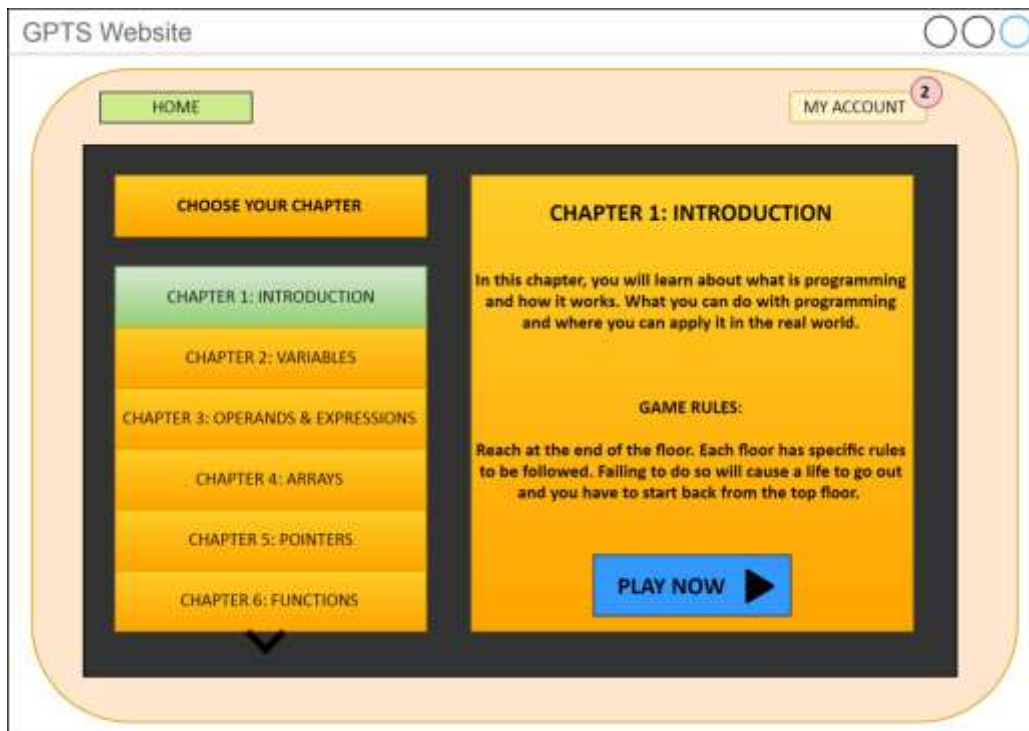
Appendix B-13: Class Manager Page

14. View Class Student List



Appendix B-14: View Class Student List

15. Game Interface



Appendix B-15: Game Interface

APPENDIX D

SOFTWARE DESIGN DOCUMENT

Software Design Document (SDD) is where the design of the system is being explained. Here is attached an SDD of the system in the next page to be used as a reference for the system design.

2019

SOFTWARE DESIGN DOCUMENT (SDD)

ORANGECODE

NASRUL ARIF BIN ZAKRIA

To be submitted to the Undergraduate Project 1
Bachelor of Computer Science (Software Engineering)



TABLE OF CONTENTS

TABLE OF CONTENTS	2
LIST OF FIGURES	3
LIST OF TABLES	3
1. DATA DICTIONARY	5
1.1 USERS TABLE	5
1.2 CLASSROOM TABLE	6
1.3 REPLIES TABLE	6
1.4 FAQ TABLE	7
1.5 FORUM THREAD TABLE	7
1.6 DISCUSSION TABLE	8
1.7 GAMEDATA TABLE	8
2. PRELIMINARY DESIGN	10
2.1 SYSTEM ARCHITECTURE	10
2.2.1 STATIC ORGANIZATION	10
2.2.2 DYNAMIC ORGANIZATION	13
3. DETAILED DESIGN	16
3.1 ProfileController	16
3.2 ForumController	16
3.3 FAQController	16
3.4 ClassroomController	17
3.5 AdminController	17
3.6 HomeController	18
4. SYSTEM DESIGN APPROVAL	19

LIST OF FIGURES

Figure 2.1	5
Figure 2.2	6
Figure 2.3	10
Figure 2.4	11
Figure 2.5	11
Figure 2.6	12

LIST OF TABLES

Table 1.1	1
Table 1.2	1
Table 1.3	2
Table 1.4	2
Table 1.5	3
Table 1.6	3
Table 1.7	4

1. DATA DICTIONARY

1.1 USERS TABLE

Field Name	Description	Data Type	Constraint
id	ID of the user	INT (Auto-Increment)	PK
name	Name of the user	VARCHAR(191)	
email	Email of the user	VARCHAR(191)	
password	Password of the user (Hashed)	VARCHAR(191)	
remember_token	Remember token for quick login (Hashed)	VARCHAR(100)	
created_at	User created at	TIMESTAMP	
updated_at	User last updated at	TIMESTAMP	
status	Status of the user (Banned, etc)	VARCHAR(191)	
institution	User's institution	VARCHAR(191)	
type	Type of user (Teacher, Student, Admin)	VARCHAR(191)	
nationality	User's nationality	VARCHAR(191)	
display_image	User's display image URL	VARCHAR(191)	
website	User's website	VARCHAR(191)	
interest	User's interest	MEDIUMTEXT	
biography	User's biography	MEDIUMTEXT	
birthdate	User's birthdate	DATE	
gender	User's gender	VARCHAR(191)	

classroomid	User's classroom joined ID	INT	FK
-------------	----------------------------	-----	----

1.2 CLASSROOM TABLE

Field Name	Description	Data Type	Constraint
classroomid	Classroom ID	INT (Auto-Increment)	PK
description	Class description	VARCHAR(191)	
objective	Class objective	VARCHAR(191)	
studentlimit	Student limit for the class	INT	
institution	Institution of the class from	VARCHAR(191)	
teacherid	Teacher ID who create the class	INT	FK
created_at	Class created at	TIMESTAMP	
updated_at	Class last updated at	TIMESTAMP	
classtitle	Class title	VARCHAR(191)	

1.3 REPLIES TABLE

Field Name	Description	Data Type	Constraint
replyid	Reply ID	INT (Auto-Increment)	PK
forumtype	The type of forum the reply is replying to	VARCHAR(191)	
userid	The ID of the sender	INT	FK
forumid	The ID of forum the reply refers to	INT	FK
body	The body of the reply	VARCHAR(191)	

created_at	Reply created at	TIMESTAMP	
updated_at	Reply last updated at	TIMESTAMP	

1.4 FAQ TABLE

Field Name	Description	Data Type	Constraint
faqid	FAQ ID	INT (Auto-Increment)	PK
faqtype	The type of FAQ (General, Account, Game, Forum)	VARCHAR(191)	
userid	The ID of the creator	INT	FK
title	The title of the FAQ	VARCHAR(191)	
body	The body of the FAQ	VARCHAR(191)	
created_at	FAQ created at	TIMESTAMP	
updated_at	FAQ last updated at	TIMESTAMP	

1.5 FORUM THREAD TABLE

Field Name	Description	Data Type	Constraint
forumid	FAQ ID	INT (Auto-Increment)	PK
forumtype	The type of Forum (Announcement, Queries and Helps, Off-Topics)	VARCHAR(191)	
userid	The ID of the creator	INT	FK
title	The title of the Forum	VARCHAR(191)	
body	The body of the Forum	VARCHAR(191)	
created_at	Forum created at	TIMESTAMP	

updated_at	Forum last updated at	TIMESTAMP	
------------	-----------------------	-----------	--

1.6 DISCUSSION TABLE

Field Name	Description	Data Type	Constraint
discussionid	Discussion ID	INT (Auto-Increment)	PK
senderid	Sender ID	INT	FK
message	Body of the discussion	VARCHAR(191)	
classroomid	The classroom that the discussion refers to	INT	FK
created_at	Discussion created at	TIMESTAMP	
updated_at	Discussion updated at	TIMESTAMP	

1.7 GAMEDATA TABLE

Field Name	Description	Data Type	Constraint
gameid	The game ID	INT (Auto-Increment)	PK
userid	The user ID of player	INT	FK
useremail	The email of player	VARCHAR(191)	
chapteronescore	The score gain from chapter one	INT	
chaptertwoscore	The score gain from chapter two	INT	
chapterthreescore	The score gain from chapter three	INT	
coins	Coins player	INT	

	obtained from the game		
health	Maximum health of the player	INT	
stringgunstr	Strength of String type gun	INT	
stringgunspeed	Speed of bullet of String type gun	INT	
stringgunrof	Rate of Fire of String type gun	INT	
integergunstr	Strength of Integer type gun	INT	
integergunspeed	Speed of bullet of Integer type gun	INT	
integergunrof	Rate of Fire of Integer type gun	INT	
boolgunstr	Strength of Boolean type gun	INT	
boolgunspeed	Speed of bullet of Boolean type gun	INT	
boolgunrof	Rate of Fire of Boolean type gun	INT	
created_at	Data created at	TIMESTAMP	
updated_at	Data last updated at	TIMESTAMP	

2. PRELIMINARY DESIGN

2.1 SYSTEM ARCHITECTURE

This section will define the internal organizational structure for OrangeCode. The integration between subsystems is being described in detail.

2.2.1 STATIC ORGANIZATION

Figure 2.1 shows the static organization for the OrangeCode. The organization consists of seven main packages.

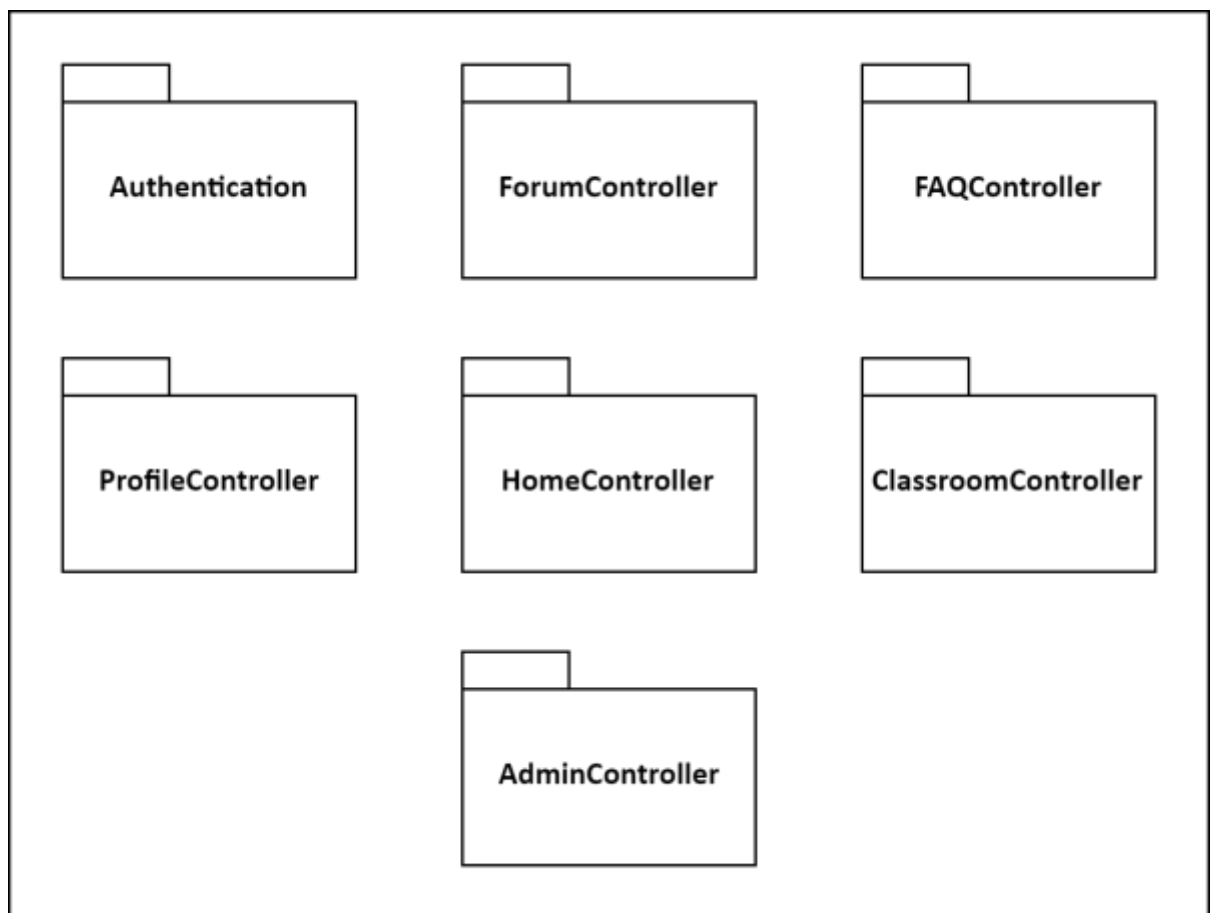


Figure 2.1 Static Organization of OrangeCode

1. Authentication

This package is to authenticate users. However, this is an automated feature in Laravel that we do not need to build by ourself.

2. ProfileController

This package is to edit and update the current user profile.

- a. Index Function
- b. Show Function
- c. Edit Function
- d. Update Function

3. ForumController

This package is to create, edit, update and view forum.

- a. index Function
- b. createPost Function
- c. createReplies Function
- d. store Function
- e. storeReply Function
- f. show Function

4. FAQController

This package is to create, edit, update and view FAQ.

- a. index Function
- b. create Function
- c. createByType Function
- d. store Function

- e. storeFAQByType Function
- f. show Function
- g. edit Function
- h. update Function
- i. destroy Function

5. ClassroomController

This package is to create, edit, update and view classroom.

- a. index Function
- b. create Function
- c. joinClass Function
- d. store Function
- e. sendDiscussion Function
- f. show Function
- g. edit Function
- h. update Function
- i. destroy Function

6. AdminController

This package is to create, edit, update and view FAQ.

- a. index Function
- b. search Function
- c. destroy Function

7. HomeController

This package is to display dashboard consist of only an index function.

2.2.2 DYNAMIC ORGANIZATION

Figure 2.2 until Figure 2.5 shows OrangeCode Model-View-Controller (MVC) pattern design with their relationships between the other components in the system.

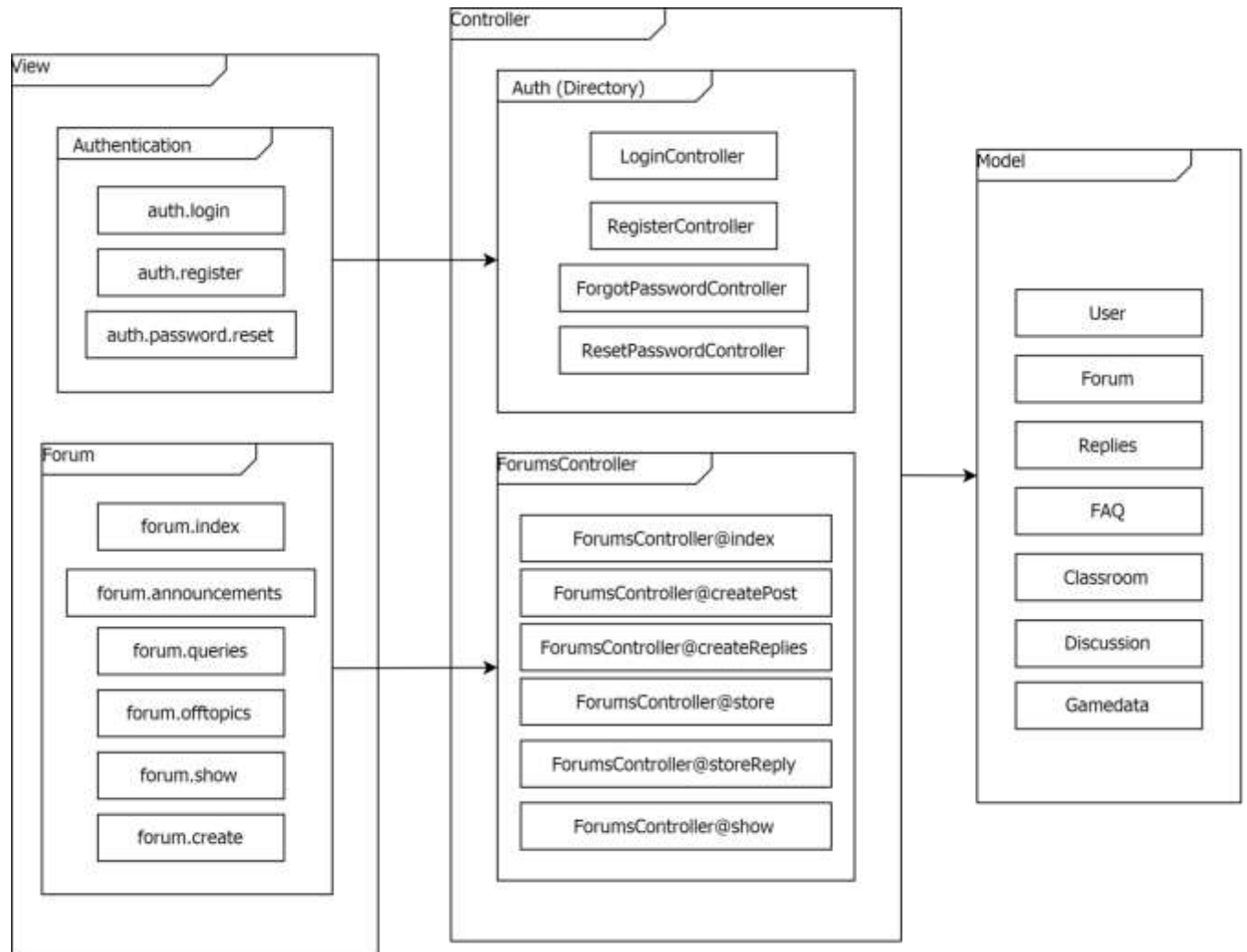


Figure 2.2 OrangeCode MVC Pattern Design Part 1

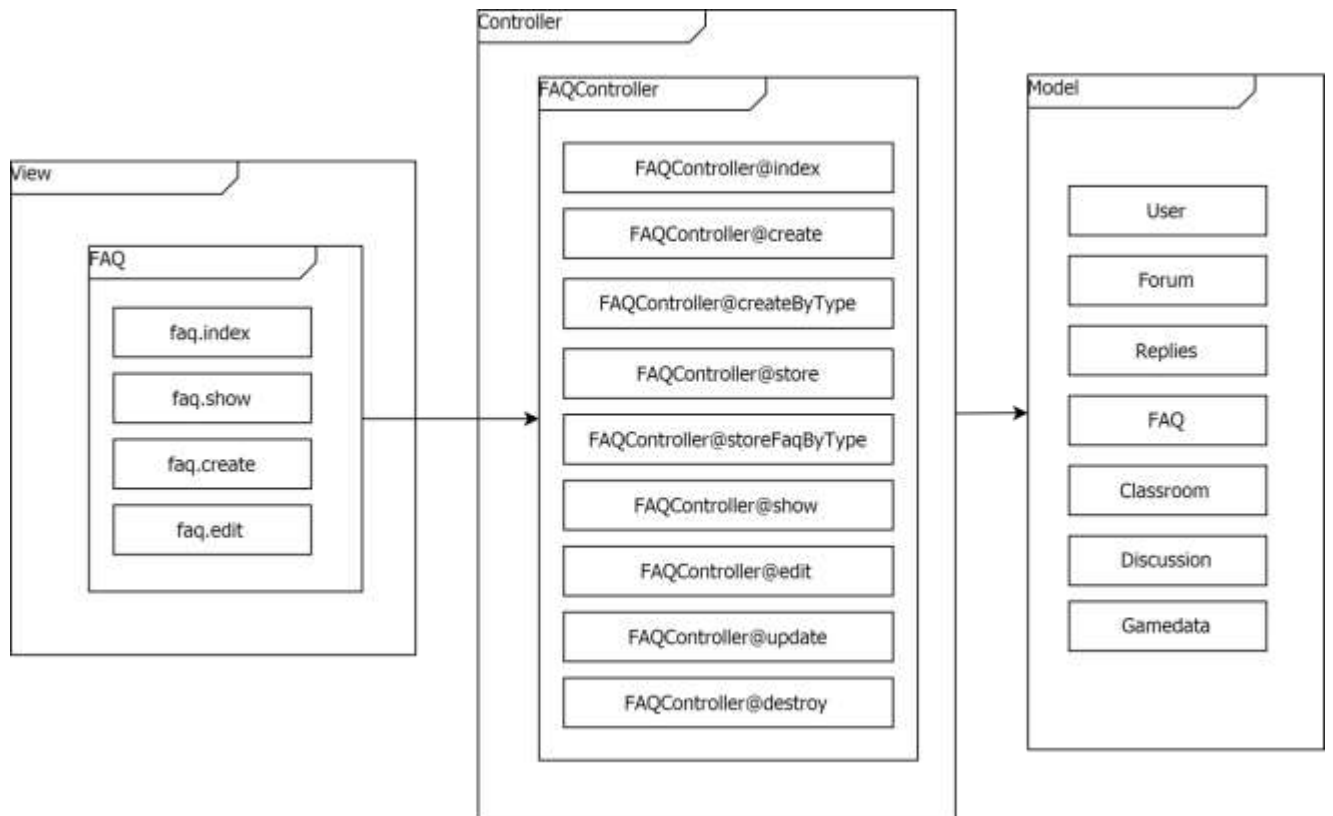


Figure 2.3 OrangeCode MVC Pattern Design Part 2

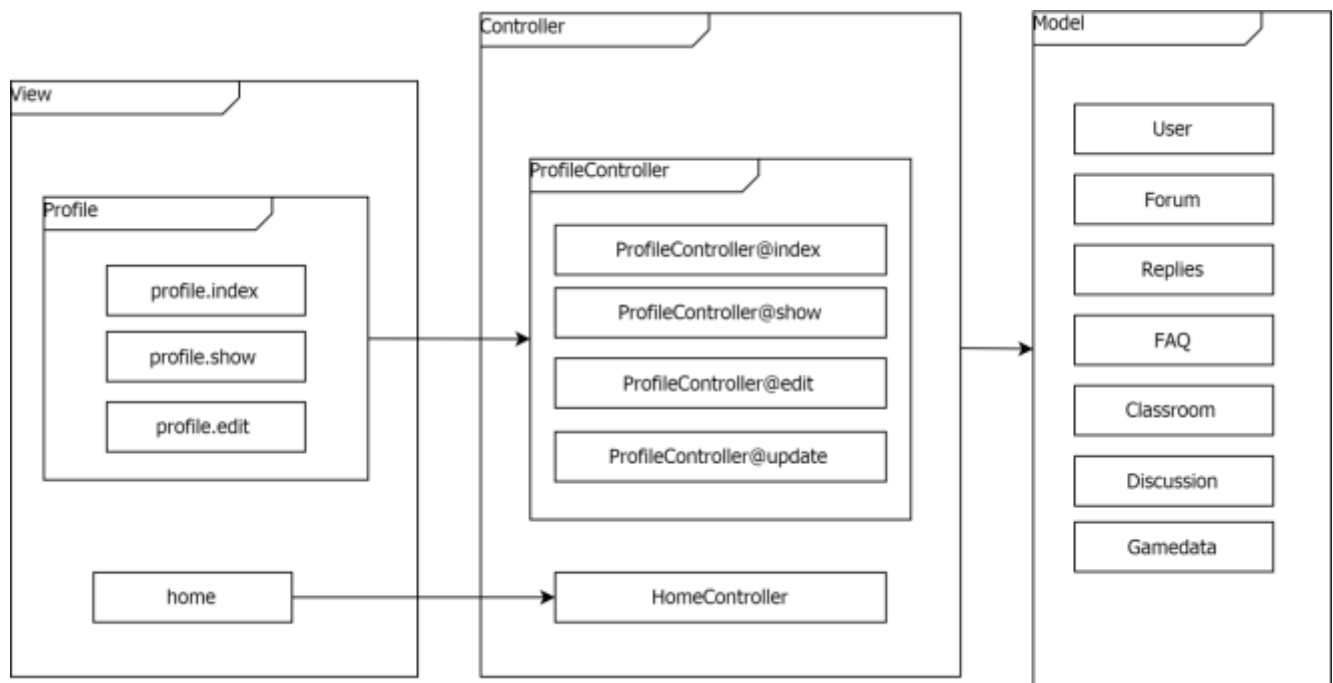


Figure 2.4 OrangeCode MVC Pattern Design Part 3

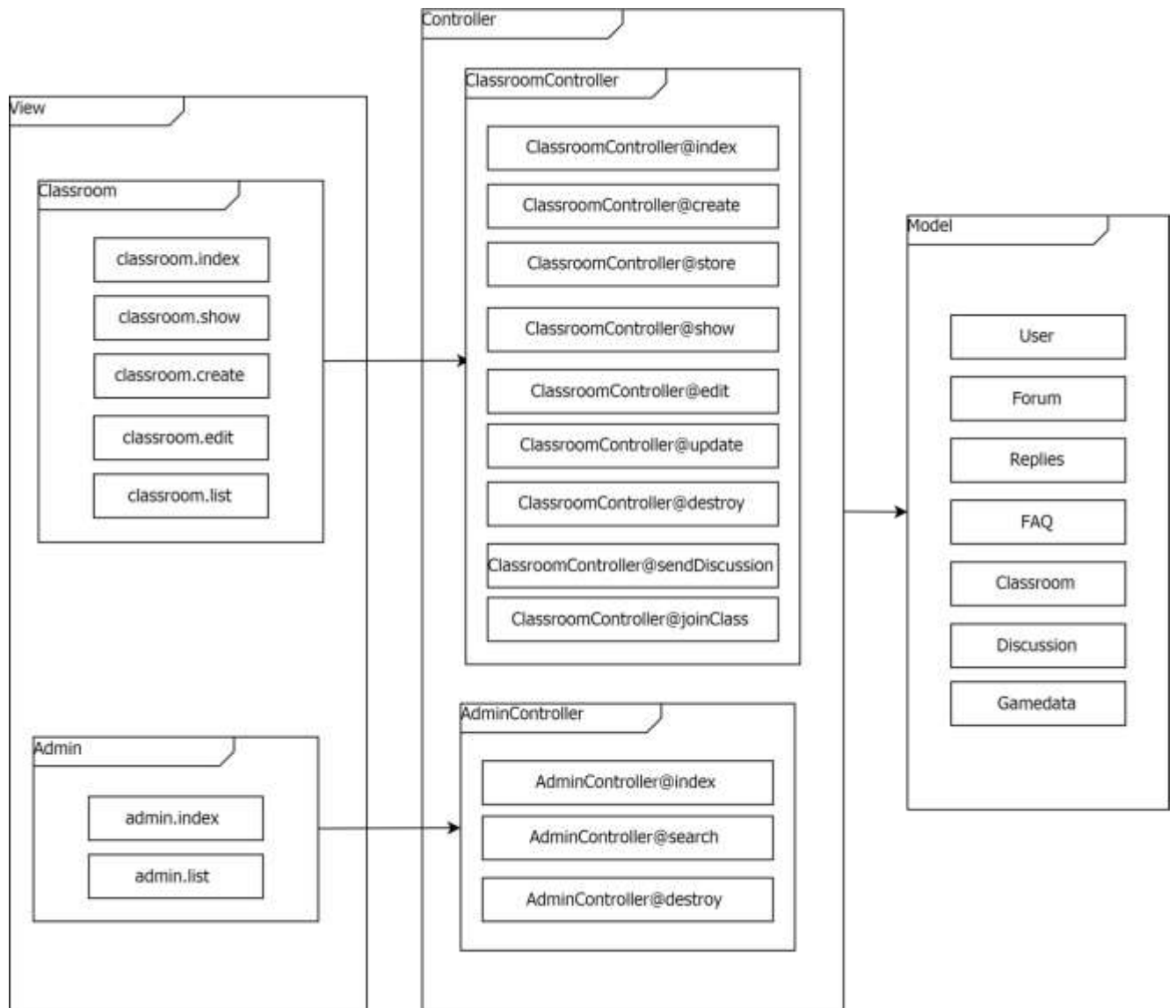


Figure 2.5 OrangeCode MVC Pattern Design Part 4

3. DETAILED DESIGN

3.1 ProfileController

Class Type:	Controller
Responsibility:	To route the system to display, edit, update profile
Attributes:	User Model, Replies Model, Carbon Class
Methods:	index() show(id) edit(id) update(request, id)

3.2 ForumController

Class Type:	Controller
Responsibility:	To route the system to create and view Forum
Attributes:	User Model, Replies Model, Forum Model
Methods:	index() createPost(type) createReplies(type) store(request) storeReply(request) show(id)

3.3 FAQController

Class Type:	Controller
Responsibility:	To route the system to create, edit, update, delete or view FAQ
Attributes:	User Model, FAQ Model
Methods:	index()

create()
createByType(type)
store(request)
storeFAQByType()
show(type)
edit(id)
update(request, id)
destroy(id)

3.4 ClassroomController

Class Type: Controller

Responsibility: To route the system to create, edit, delete, view classroom

Attributes: User Model, Discussion Model, Classroom Model

Methods: index()
create()
store(request)
joinClass(request, id)
show(id)
sendDiscussion(request)
edit(id)
update(request, id)
destroy(id)

3.5 AdminController

Class Type: Controller

Responsibility: To route the system to perform administrator tasks.

Attributes: User Model, Forum Model, Classroom Model

Methods: index()
search(request)
destroy(id)

3.6 HomeController


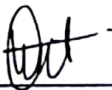
Class Type: Controller

Responsibility: To route the system to dashboard and display required item

Attributes: User Model, Replies Model

Methods: index()

4. SYSTEM DESIGN APPROVAL

	Name	Date
Verified by:  Developer	NASRUL ARIF BIN ZAKRIA	
Approved by:  Client	MUHAMMAD KITAIRIL ALMAL BIN MUHD KITAIRUPPIN	

APPENDIX E

USER ACCEPTANCE TESTING

User Acceptance Testing (UAT) is an important testing that should be conducted once a system is considered stable by the developer and ready to be used by the user. In the next page, a UAT sample from client has been attached as a proof that the system has been tested throughly.

1. TESTING REPORT

The purpose of this section is to outline the User Acceptance Testing (UAT) process for the system. Approval of this testing implies that reviewers are confident that following the execution of the test plan, the resulting system will be considered fully-tested and eligible for implementation.

1.1. Authentication

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Register	Khairi khairi@gmail.com 531736	Successfully registered, redirect to dashboard.	Same	Pass	
Login	khairi@gmail.com 531736	Successfully logged in, redirect to dashboard.	same	pass	

1.2. Profile

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Upload Display Image	513 kb image	Cannot upload image exceeding 2000KB. Cannot upload file other than images. Other than that will be successful.	success	Pass	
Edit Profile		Successfully			

	Add bio, interest	update if validation succeed.	Success	Pass	
--	----------------------	-------------------------------------	---------	------	--

1.3. Forum

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Create New Forum	Title: Debug? Body: How to debug code?	Successfully update if validation succeed.	Success	Pass	
Reply to a Forum	Body: Let me show you how.	Successfully update if validation succeed.	Success	Pass	

1.4. FAQ

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Create New FAQ	How to start playing? Click here	Successfully update if validation succeed.	Success	Pass	
Edit FAQ	"Click here" to "Click there"	Successfully update if validation succeed.	Success	Pass	
Delete FAQ	✓	Successfully if confirmed.	Success	Pass	



1.5. Classroom

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Create New Classroom	name Python code 12345678 desc Learn Python	Successfully update if validation succeed.	Success	Pass	
Edit Existing Classroom	name Basic Python code abcd1234 desc Learn Basic Python	Successfully update if validation succeed.	Success	Pass	
Delete Classroom	delete Basic Python	Successfully if confirmed.	Success	Pass	
Join Classroom	join Test Classroom code test1234	Successfully if secret code correct.	Success	Pass	
Leave Classroom	leave Test Classroom	Successfully if confirmed.	Success	Pass	

1.6. Administrator Tasks

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Search Item	user Khairil	Successfully update if validation succeed.	Success, name comes out	Pass	
Delete Item	user Khairil	Successfully if confirmed.	Success	Pass	

2. SYSTEM TESTING APPROVAL

	Name	Date
<p>Verified by:</p>  <hr/> <p>Developer</p>	NASRUL ARIF BIN ZAKRIA	
<p>Approved by:</p>  <hr/> <p>Client</p>	Muhammad Khairil Akmal Bin Mohd Khairuddin.	

APPENDIX F USER MANUAL

User Manual is to work as a guideline to the user who never used the system. The steps and flows of the system is explained in user manual so the user can follow it thoroughly.

ORANGECODE

Faculty of Computer System & Software Engineering

USER MANUAL

Contents

1. GENERAL INFORMATION	3
1.1. System Overview.....	3
2. SYSTEM SUMMARY	3
2.1. System Configuration.....	3
2.2. User Access Levels	3
2.3. Contingencies and Alternate Modes of Operation	4
3. GETTING STARTED.....	4
3.1. Opening the System.....	4
3.2. Registering or Logging In.....	4
3.3. Editing and Updating Your Profile.....	5
3.4. Start a Forum Thread	7
3.5. Reply to a Forum.....	7
3.6. Create a Classroom (Teacher only)	8
3.7. Joining and Leaving a Classroom (Student only).....	9
3.8. Play Games.....	10

1. GENERAL INFORMATION

1.1. System Overview

OrangeCode is a website that is mainly to learn about programming by playing games. OrangeCode also allows their users to do such thing:

- Register or Log In into OrangeCode
- Create a Forum post and replying to Forum post
- Create a virtual classroom that allows students to compete in the leaderboard
- Read Frequently Asked Questions (FAQs)
- Edit, Update and View profiles information
- Play Games and compete with other players

2. SYSTEM SUMMARY

2.1. System Configuration

OrangeCode is an online website that can be browse using any web browser by going to the domain “<https://www.theorange.com>”. Therefore, an internet connection is compulsory to use OrangeCode. It is recommended that the computer to have atleast 4.00GB of RAM for better performance when running the game since there have been lagging issues reported.

2.2. User Access Levels

OrangeCode have three access levels:

- Guest - Can only read forum and FAQs. Can only play the demo version of the game.
- Students/Teachers - Can do all the functionalities stated in the system overview as long as the user is logged in
- Administrator - Can perform administrator tasks such as managing the user account, forum management and classroom management.

2.3. Contingencies and Alternate Modes of Operation

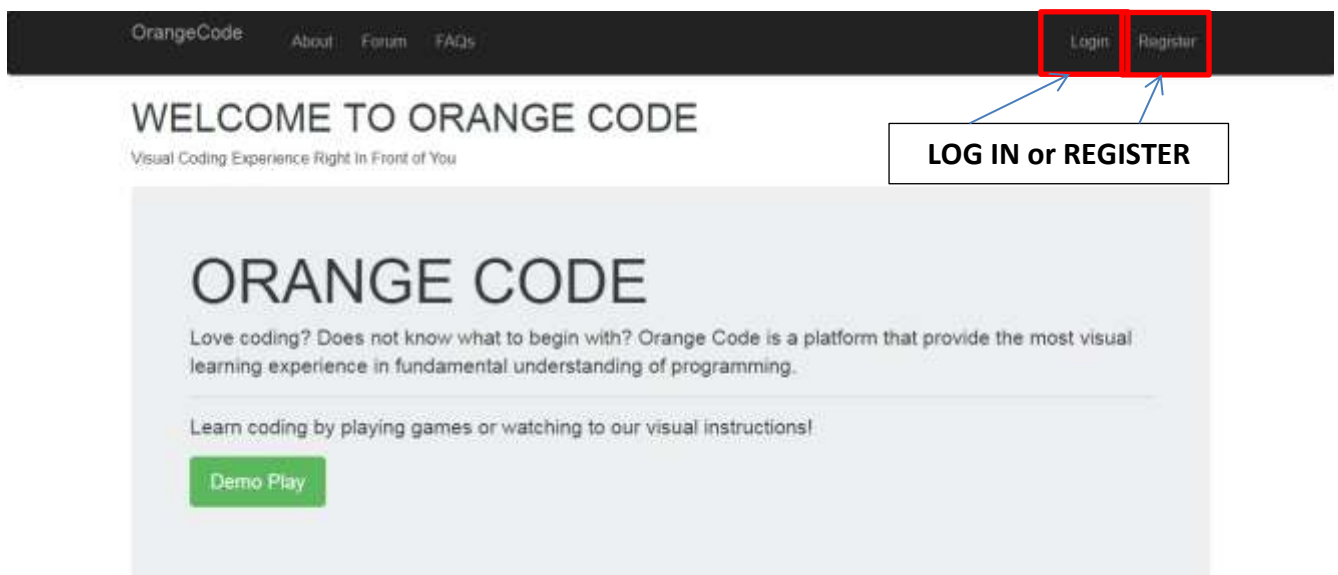
Since OrangeCode is an online system, there would not be any issues regarding data loss unless it is the users' mistake for not triggering the save event by the time of any issues happen. In case of the browser did not support the system, users can always try another browser to use the system. The most recommended browser would be Google Chrome and Mozilla Firefox since both of these browsers were used during the testing period of the system.

3. GETTING STARTED

3.1. Opening the System

- OrangeCode can be opened by using any web browser (Google Chrome recommended)
- Type in the URL box the following domain "https://www.theorangecode.com" (subject to change).
- If successfully, the OrangeCode website will load and ready to be used.

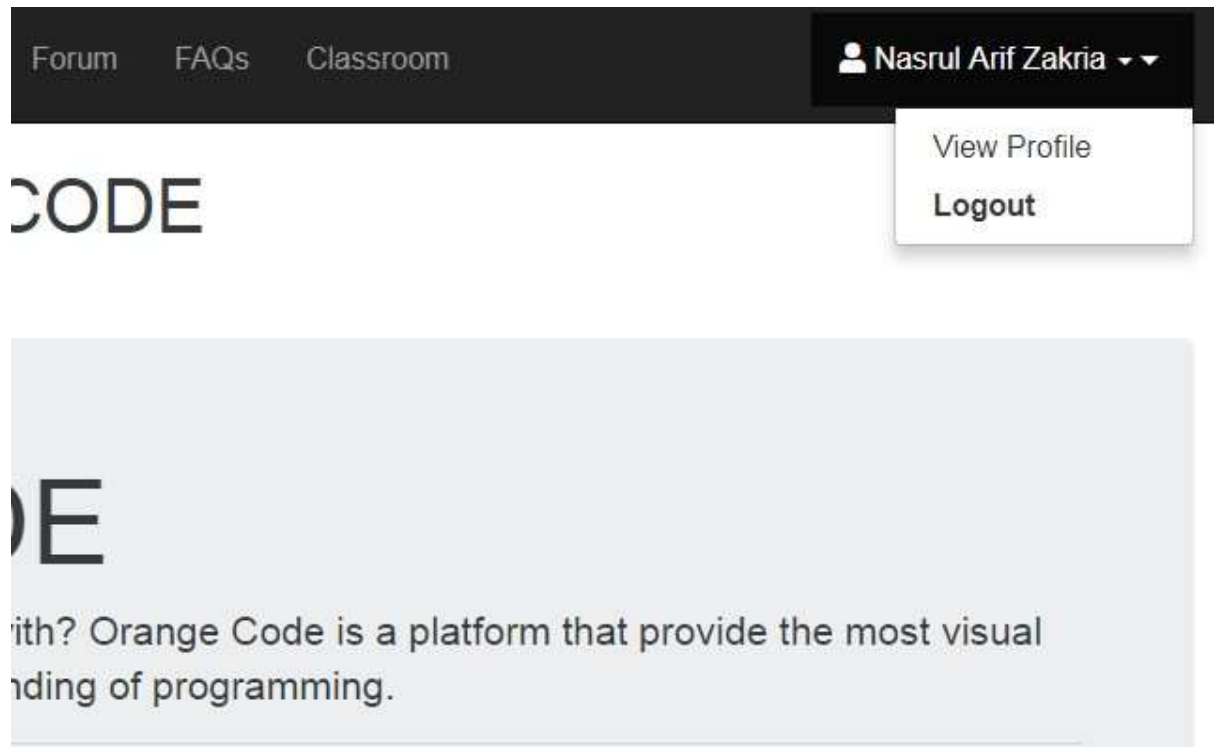
3.2. Registering or Logging In



- Once opened, any user with existing account can directly log in into the system and be redirected to the dashboard.

- b) If the user does not have an account then click “Register” on the navigation bar to create a new account.
- c) Enter required information and you will be redirected to the dashboard once the registration is completed.

3.3. Editing and Updating Your Profile



- a) Profile can be edit by clicking on your username on the top right corner of the navigation bar. Then press view profile, this will open the profile page.
- b) At the bottom of the profile page, there is a button called “Edit Profile” where you will be redirected to another page that allows you to edit or add your profile information.

The image shows a profile editing form with the following fields and content:

- Website:** OrangeCode build
- Company / Institution:** Universiti Malaysia Pahang
- Nationality:** Malaysia
- Interest:** (Empty text area)
- Biography:** I learnt to code since I am a children. It was fun and I love it so much!

An **Edit Profile** button is located at the bottom right of the form.

- c) Once you have done editing your profile, click the “Save Profile” button to update your profile as the one that you have input earlier.

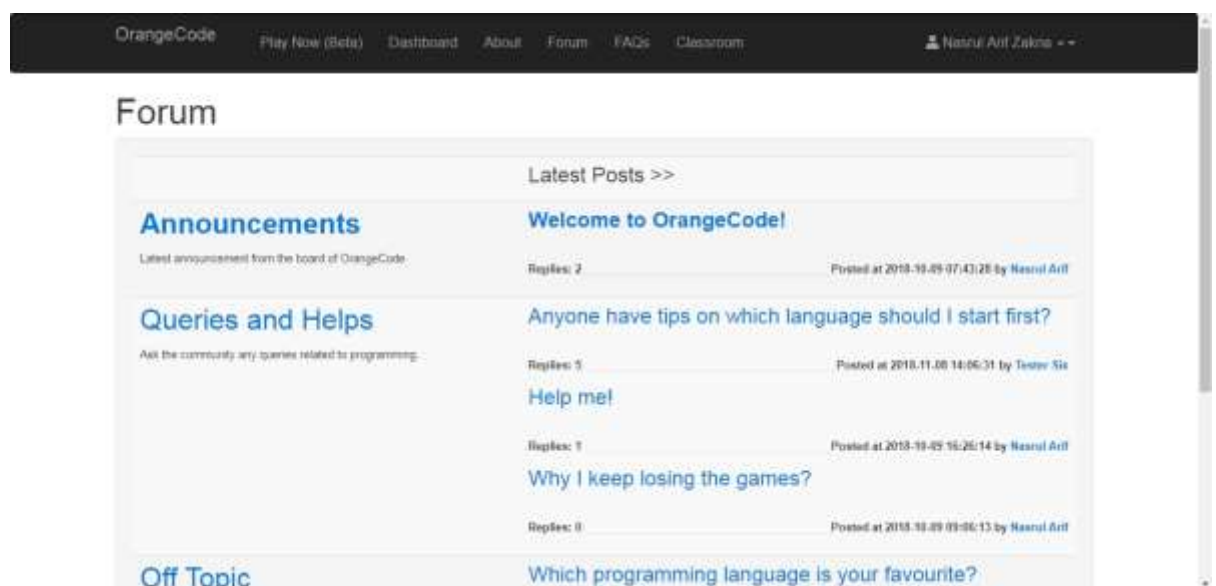
The image shows a profile editing form with the following fields and content:

- Nationality:** Malaysia (dropdown menu)
- Website:** OrangeCode.build
- Company / Institution:** Universiti Malaysia Pahang
- Interest:** Interest
- Biography:** I learnt to code since I am a children. It was fun and I love it so much!

A **Save Profile** button is located at the bottom of the form.

3.4. Start a Forum Thread

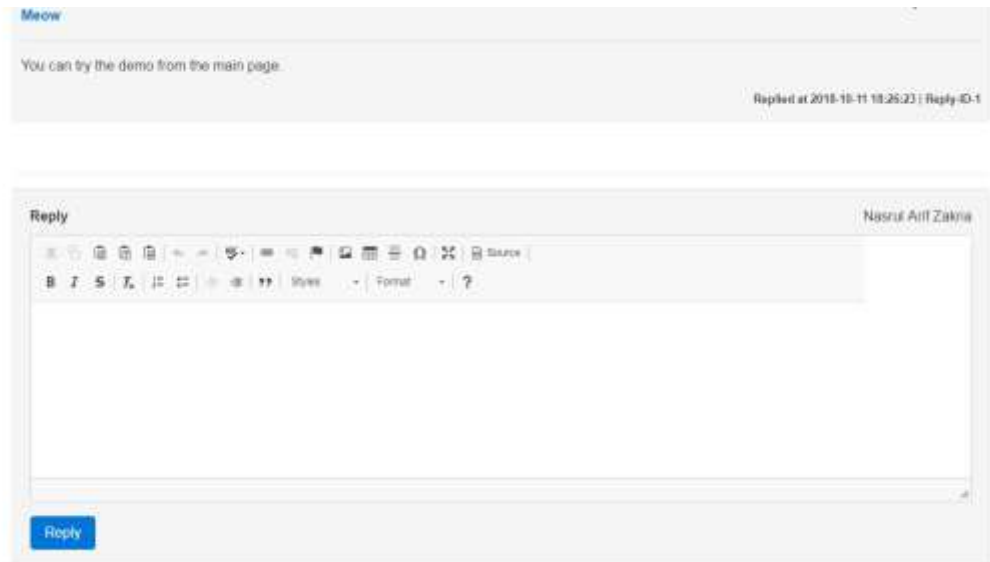
- a) A forum can be created in the forum page.
- b) In order to go to forum page, click the “Forum” menu button on top of the navigation bar.
- c) As you can see there are about three types of forum post, Announcements, Queries and Helps and Off Topics.
- d) Announcement is for Administrator to post any latest announcements.
- e) Queries and Helps and Off Topics are for daily users to open any issue or topic to be discussed with the community.



- f) Click on any of the forum type except Announcements until you are redirected to the next page.
- g) Scroll down to the bottom to find “Create Post” button.
- h) When clicked, you will be redirected to a page with form to be input.
- i) Click submit once you have input all fields and your post will be created.

3.5. Reply to a Forum

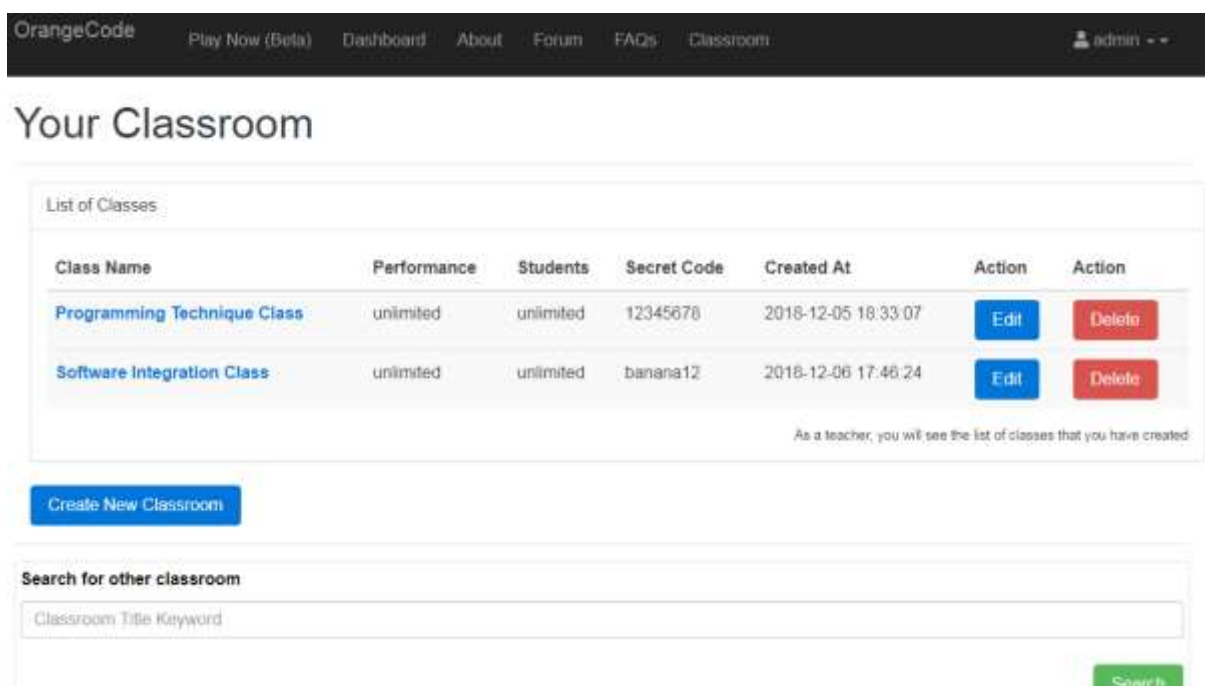
- a) To reply to a forum post, go to the specific forum post.
- b) Scroll down to the bottom to find an input box. Note that you must be logged in to see this.



- c) Fill the box with your reply and click the “Reply” button to submit your reply.
- d) You can always use the text enhancement tool to furnish your reply.
- e) You can also attach images using the tool.

3.6. Create a Classroom (Teacher only)

- a) Classroom can only be created by teacher.
- b) Navigate to the Classroom page (you must be logged in to do so).
- c) Notice that there is a list of classroom (empty or not depends on whether you have created any classroom before).
- d) At the bottom left of the panel, there is a “Create New Classroom” button.

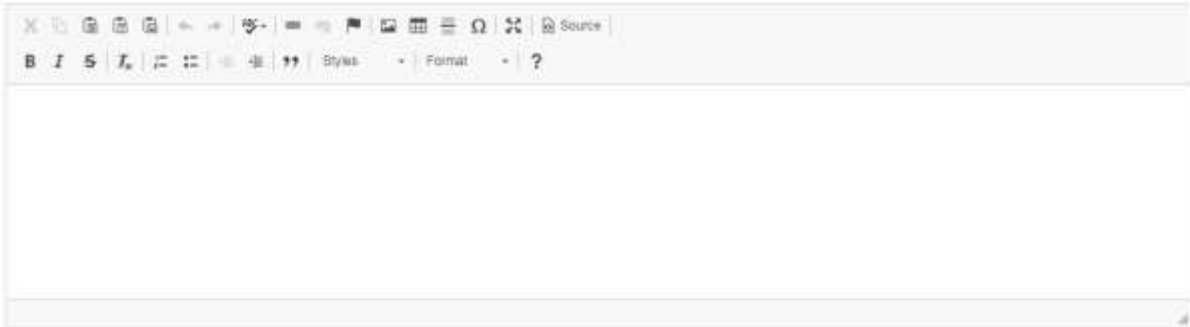


- e) Clicking that will open another page that requires your inputs.

Create New Classroom

Class Title

Class Description

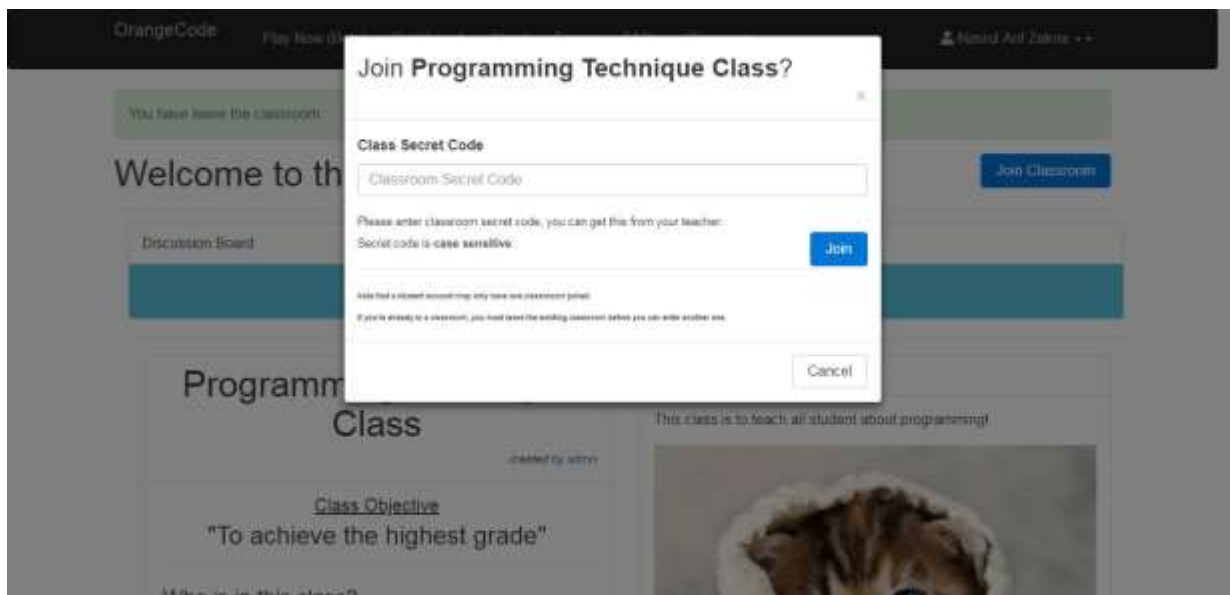
A rich text editor interface for entering a class description. It features a toolbar with various icons for text formatting (bold, italic, underline, strikethrough), alignment, bulleted and numbered lists, indentation, and link management. Below the toolbar is a large, empty text area for writing the description.

Objective

- f) Fill the input fields, and click “Submit” to confirm creating the classroom.
- g) Note that you can always edit the information later in the classroom page.
- h) **Secret Code** is a password that is required by student upon entering the class. If you have specific students to join the class, provide them with the code. Else, they will never be able to join the classroom.

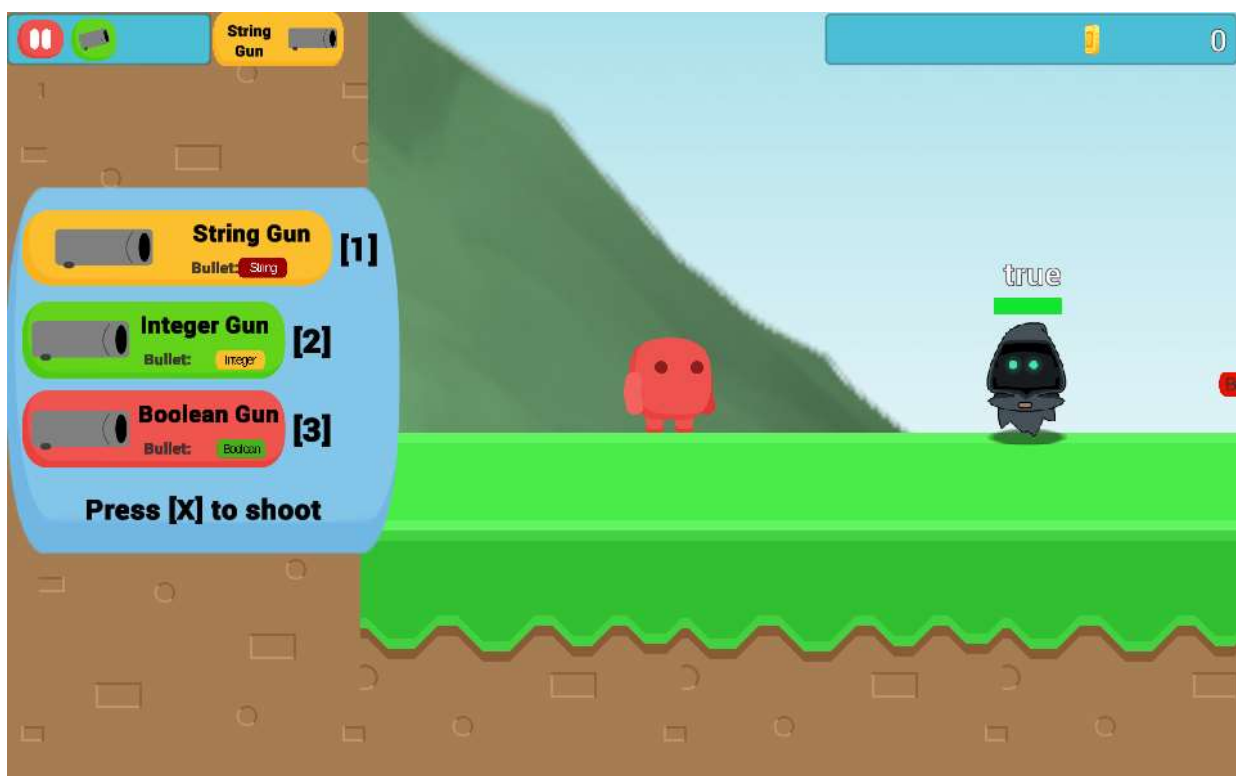
3.7. Joining and Leaving a Classroom (Student only)

- a) Only students can join or leave a classroom.
- b) Joining a classroom require secret code, this can be obtained from your teacher.
- c) To do so, open any classroom information page. If you fail to find it, you can use the search feature in the classroom main page to search for the classroom.
- d) At the classroom information page, click “Join Classroom” at the top right corner of the page to join the classroom.



- e) A pop-up will appear prompting the secret code. Enter the correct secret code and you will be inside the classroom.

3.8. Play Games



- a) You can start playing game by clicking the “Play Now” button in the navigation bar.

- b) The game comes with tutorial. It is the best to follow the tutorial there for more information.
- c) The basic game mechanic is that you are provided with three types of gun, String Gun, Integer Gun and Boolean Gun.
- d) In the game, monsters that appear will have name on top of it. For example, "189456" or "false" or "Monster".
- e) Depending on the name, you can destroy them using specific gun only.
- f) For instance, monster with numbering name can only be destroyed using Integer Gun.
- g) Reach at the end of the level and open the chest to proceed to the next level.
- h) You can always upgrade your gun to make it stronger, faster and have higher rate of fire.

REFERENCES

- Aki, O., Güllü, A., & Kaplanoğlu, E. (2015). An Application for Fundamental Computer Programming Learning. *Procedia - Social and Behavioral Sciences*, 176, 291-298. doi: <https://doi.org/10.1016/j.sbspro.2015.01.474>
- Budny, D., Lund, L., Vipperman, J., & Patzer, J. (2002). *Four steps to teaching C programming*. Paper presented at the Frontiers in Education, 2002. FIE 2002. 32nd Annual.
- Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141. doi: <https://doi.org/10.1016/j.compedu.2016.03.003>
- Sailer, M., Hense, J. U., Mayr, S. K., & Mandl, H. (2016). Computers in Human Behavior. *How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction*.
- Salleh, A. R. (2011). UKM Teaching and Learning Congress 2011. *How to make teaching interesting?*
- Shi, N., Min, Z., & Zhang, P. (2017). Effects of visualizing roles of variables with animation and IDE in novice program construction. *Telematics and Informatics*, 34(5), 743-754. doi: <https://doi.org/10.1016/j.tele.2017.02.005>
- Souza, Mauricio R. de A., Veado, L., Moreira, R. T., Figueiredo, E., & Costa, H. (2018). A systematic mapping study on game-related methods for software engineering education. *Information and Software Technology*, 95, 201-218. doi: <https://doi.org/10.1016/j.infsof.2017.09.014>
- Wang, M., Wu, B., Kinshuk, Chen, N.-S., & Spector, J. M. (2013). Connecting problem-solving and knowledge-construction processes in a visualization-based learning environment. *Computers & Education*, 68, 293-306. doi: <https://doi.org/10.1016/j.compedu.2013.05.004>