# MERIT POINT ATTENDANCE SYSTEM UTEM

CHON YAO JUN



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

# MERIT POINT ATTENDANCE SYSTEM UTeM

CHON YAO JUN



This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Software Development) with Honours.



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

# DECLARATION

I hereby declare that this project report entitled

**Merit Point Attendance System UTeM**

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT :  Date : 04/09/2021

(CHON YAO JUN)



I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Software Development) with Honours.

SUPERVISOR :  Date : 10 September 2021

(PROF. MADYA TS. DR. MOHD SANUSI AZMI)

**DEDICATION**

I would like to dedicated my work to my loving parents, supervisor and friends. A indescribable expression to them whose always encourage and support me along with my work. And special thank to my supervisor, Dr. Mohd Sanusi Azmi who supports me throughout the process. I always appreciate all the things that given by them in helping me finish my works.

## ACKNOWLEDGEMENTS

In this project, the guidance from many people such as my supervisor, Dr. Mohd Sanusi Azmi play the important role. I would like to thank you to all these people.

I have to start by thanking Dr. Mohd Sanusi Azmi for giving assistant to complete this project successfully, provided me with the golden opportunity to do this wonderful project, who guided me in the direction to take for this project, provided me with the materials, and the necessary hardware for me to develop this system. For this, I am thankful.

Next, I would also like to thank my friends who helped me a lot in finalizing this project within the limited time frame, and also for all the moral support that they provided that finally allowed me to push on and finish this project.

# ABSTRACT

This project is designed to provide a merit point based attendance system in UTeM, named as U-Tem Merit . The U-Tem Merit is a website for the student, organizer and management team to use to manage programs in UTeM or outside programs. This system will provide all the services along the program from start to the end. U-Tem Merit provides easy and friendly use system for them when managing the programs.

# ABSTRAK

Project ini adalah untuk membuat satu system nilai merit kehadiran di UTeM, dinamakan sebagai U-Tem Merit. U-Tem Merit adalah satu laman web bagi pelajar, pengajur dan jabatan pengurusan menguruskan program di UTeM atau program luar UTeM. Sistem ini akan membekalkan semua perkhidmatan yang diperlukan sepanjang program diadakan. U-Tem Merit membekalkan mudah and mesra bagi penguna menggunakan sistem ini apabila menguruskan program.

# TABLE OF CONTENTS

**PAGE**

**DECLARATION** ..... **II**

**DEDICATION** ..... **III**

**ACKNOWLEDGEMENTS** ..... **IV**

**ABSTRACT** ..... **V**

**ABSTRAK** ..... **VI**

**TABLE OF CONTENTS** ..... **VII**

**LIST OF TABLES** ..... **XII**

**LIST OF FIGURES** ..... **XIII**

**LIST OF ABBREVIATIONS** ..... **XV**

**LIST OF ATTACHMENTS** ..... **XVI**

**CHAPTER 1: INTRODUCTION** ..... **1**

1.1 Introduction ..... 1

1.2 Problem Statement ..... 1

1.3 Objective ..... 2

1.4 Scope ..... 2

1.4.1 Target User ..... 2

1.4.2 Platform ..... 2

1.4.3 Modules to be developed ..... 3

1.4.3.1 Student Module ..... 3

1.4.3.2 Organizer Module ..... 3

1.4.3.3 Management Team Module ..... 4

1.5 Deliverable ..... 6

1.6 Project Significance ..... 6

1.7 Expected Output ..... 6

1.8 Conclusion ..... 6

**CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY . 7**

2.1 Introduction ..... 7

2.2 Facts and findings ..... 7

2.2.1 Domain ..... 7

2.2.2 Existing System ..... 8

2.2.3 Technique ..... 9

2.3 Project Methodology ..... 9

2.3.1 System Development Life Cycle ..... 9

2.3.2 Database Life Cycle ..... 11

2.4 Project Requirements ..... 12

2.4.1 Software Requirement ..... 12

2.4.2 Hardware Requirement ..... 13

2.4.3 Other Requirements ..... 13

2.5 Project Schedule and Milestones ..... 13

2.6 Conclusion ..... 14

**CHAPTER 3: ANALYSIS ..... 15**

3.1 Introduction ..... 15

3.2 Problem Analysis ..... 15

3.3 Requirement Analysis ..... 16

3.3.1 Date Requirement ..... 17

3.3.2 Functional Requirement ..... 21

3.3.2.1 Student Module ..... 21

3.3.3 Non-Functional Requirement ..... 23

3.4 Conclusion ..... 24

**CHAPTER 4: DESIGN ..... 25**

4.1 Introduction ..... 25

4.2 High-Level Design ..... 25

4.2.1 System Architecture ..... 25

4.2.2 Use Case Diagram ..... 27

4.2.3 User Interface Design ..... 27

4.2.3.1 Interfaces for All Roles ..... 27

4.2.3.2 Interfaces for Students ..... 30

4.2.3.3 Interfaces for Organizer ..... 34

4.2.3.4 Interfaces for Admin ..... 38

4.3 Database Design ..... 44

4.3.1 Conceptual and Logical Database Design ..... 44

4.3.2 Business Rule: ..... 44

4.4 Detailed Design ..... 45

4.4.1 Module Integration and Interface Design ..... 45

4.4.1.1 Student Modules ..... 45

4.4.1.2 Organizer Modules ..... 46

4.4.1.3 Management Team ..... 47

4.4.2 Software Design ..... 48
4.4.2.1 Students ..... 49
4.4.2.2 Organizer ..... 52
4.4.2.3 Management Team ..... 55
4.4.3 Physical Database Design ..... 59
4.5 Conclusion ..... 59
**CHAPTER 5: IMPLEMENTATION ..... 60**
5.1 Introduction ..... 60
5.2 Software Development Environment Setup ..... 60
5.3 Software Configuration Management ..... 61
5.3.1 Configuration Environment Setup ..... 61
5.3.2 Version Control Procedure ..... 61
5.3.2.1 Web Application ..... 62
5.3.2.2 Web Server ..... 63
5.4 Implementation Status ..... 64
5.5 Conclusion ..... 65
**CHAPTER 6: TESTING ..... 66**
6.1 Introduction ..... 66
6.2 Test Plan ..... 66
6.2.1 Test Organization ..... 66
6.2.2 Test Environment ..... 67
6.2.3 Test Schedule ..... 67
6.3 Test Strategy ..... 68

6.3.1 Classes of tests ..... 68
6.3.1.1 Unit Testing ..... 68
6.3.1.2 Integration Testing ..... 68
6.4 Test Design ..... 68
6.4.1 Unit Testing ..... 69
6.4.1.1 Student ..... 69
6.4.1.2 Organizer ..... 72
6.4.1.3 Management Team ..... 75
6.4.2 Integration Testing ..... 80
6.5 Test Result and Analysis ..... 87
6.6 Conclusion ..... 88
**CHAPTER 7: PROJECT CONCLUSION ..... 89**
7.1 Observation on Weakness and Strengths ..... 89
7.1.1 Weakness ..... 89
7.1.2 Strengths ..... 89
7.2 Propositions for Improvement ..... 90
7.3 Project Contribution ..... 90
7.4 Conclusion ..... 91
**REFERENCES ..... 92**
**APPENDIX A ..... ERROR! BOOKMARK NOT DEFINED.**

# LIST OF TABLES

**PAGE**

**Table 2.1: List of Software** ..... **12**
**Table 2.2: List of Hardware** ..... **13**
**Table 2.3: Gantt Chart** ..... **13**
**Table 3.1 Table of Attendance** ..... **18**
**Table 3.2: Table of Location** ..... **18**
**Table 3.3: Table of PointList** ..... **19**
**Table 3.4: Table of Program** ..... **19**
**Table 3.5: Table of User** ..... **20**
**Table 3.6: Student Module Requirement** ..... **21**
**Table 3.7: Organizer Module Requirement** ..... **22**
**Table 3.8: Management Team Module Requirement** ..... **22**
**Table 3.9: Non-Functional Requirement** ..... **23**
**Table 5.1: Website** ..... **62**
**Table 5.2: Web Server** ..... **63**
**Table 5.3: Module Implementation Progress** ..... **64**
**Table 6.1: Test Environment** ..... **67**
**Table 6.2: Test Schedule** ..... **67**
**Table 6.3: Test Case for Student** ..... **69**
**Table 6.4: Test Case for Organizer** ..... **72**
**Table 6.5: Test Case for Management Team** ..... **75**

# LIST OF FIGURES

**PAGE**

**Figure 2.1: Waterfall Model** ..... **11**
**Figure 3.1: Current Flow** ..... **16**
**Figure 3.2: U-Tem Merit Flow** ..... **17**
**Figure 4.1: System Architecture Visualized** ..... **26**
**Figure 4.2: Use Case Diagram of U-Tem Merit** ..... **27**
**Figure 4.3: Login** ..... **27**
**Figure 4.4: Forgot Password** ..... **28**
**Figure 4.5: Personal Information** ..... **29**
**Figure 4.6: Update Personal Information** ..... **29**
**Figure 4.7: Register as Student** ..... **30**
**Figure 4.8: Participated Programs with More Columns** ..... **31**
**Figure 4.9: Participated Programs with Lesser Columns** ..... **31**
**Figure 4.10: Claim UTeM Program's Attendance** ..... **32**
**Figure 4.11: Claim Outsire Program's Attendance** ..... **32**
**Figure 4.12: Select Attendance** ..... **32**
**Figure 4.13: Unable to Take Attendance** ..... **33**
**Figure 4.14: Allow to Take Attendance** ..... **33**
**Figure 4.15: All Programs for Organizer** ..... **34**
**Figure 4.16: Register Program** ..... **35**
**Figure 4.17: Update Program** ..... **35**
**Figure 4.18: View Program with QR Code** ..... **36**
**Figure 4.19: Manage Attendance** ..... **36**
**Figure 4.20: Generate Report** ..... **37**
**Figure 4.21: Register for Admin** ..... **38**

**Figure 4.22: View All Students Detail** ..... **38**
**Figure 4.23: View More Columns of Programs** ..... **39**
**Figure 4.24: View Lesser Columns of Programs** ..... **39**
**Figure 4.25: Register Program for Admin** ..... **40**
**Figure 4.26: Update Program for Admin** ..... **40**
**Figure 4.27: Manage Other's Program** ..... **41**
**Figure 4.28: View Program with QR Code** ..... **42**
**Figure 4.29: Manage Attendance** ..... **42**
**Figure 4.30: Generate Report** ..... **43**
**Figure 4.31: Manage Outside Programs** ..... **43**
**Figure 4.32: Conceptual and Logical Database Design** ..... **44**
**Figure 4.33: Authentication for Student** ..... **49**
**Figure 4.34: Participated Programs** ..... **50**
**Figure 4.35: Take Attendance** ..... **50**
**Figure 4.36: View All Programs** ..... **51**
**Figure 4.37: Authentication for Organizer** ..... **52**
**Figure 4.38: Manage Own Create Program** ..... **53**
**Figure 4.39: Manage Attedance for Organizer** ..... **54**
**Figure 4.40: Authentication for Management Team** ..... **55**
**Figure 4.41: Student List** ..... **56**
**Figure 4.42: Manage Programs for Management Team** ..... **57**
**Figure 4.43: Manage All Program Attendances** ..... **58**
**Figure 4.44: Manage Outside UTeM Program Attendances** ..... **58**
**Figure 4.45: Entity Relational Diagram** ..... **59**

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| **FTMK** | **-** | **Fakulti Teknologi Maklumat & Komunikasi** |
| **UTeM** | **-** | **Universiti Teknikal Malaysia Melaka** |
| **SDLC** | **-** | **System Development Life Cycle** |
| **DBLC** | **-** | **Database Life Cycle** |
| **ERD** | **-** | **Entity Relationship Diagram** |
| **SQL** | **-** | **Structured Query Language** |
| **RDBMS** | **-** | **Relational Database Management System** |
| **IDE** | **-** | **Integrated Development Environment** |
| **UML** | **-** | **Unified Modelling Language** |
| **FYP** | **-** | **Final Year Project** |

# LIST OF ATTACHMENTS


| | | PAGE |
|---|---|---|
| **Appendix A** | **Sample of source code** | **93** |

# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

Merit Point Attendance System UTeM is an website that helps students, organizer of programs and management team to manage programs which are merit point based. UTeM has lot of students and programs, but students did not able to know their performance in UTeM. They can only know by the certificates received. UTeM also does not have measurement given to the students who have contributions in the participated program. The students will have motivation to take part in program if some measurements are given such as merit point. Merit point is the point given to the participants when they take charge of some position or make achievement in the program. Depend on the performance of the participants, different merit point will be given. Organizer and management team also will get profit from it. Organizer can register programs with few easy steps and managements team can easily manage the systematically listed programs request from organizer. The organizer can be anyone who is assigned by the management team. Merit Point Attendance System UTeM is designed to solve the problems and provide convenience to all three of them.



## 1.2 Problem Statement

- The students do not know their own performance in university life by participating programs of university.

- The organizers are facing problems when registering and managing attendance of students in university.

- The management team spends lot of time for doing authentic programs and generate report.

## 1.3 Objective

This project embarks on the following objects:

- To digitalize the attendance system.
- To provide a merit point attendance system
- To increase the efficiency for the students, organizer and management team when managing a program.

## 1.4 Scope

### 1.4.1 Target User

This project is target for students, organizer of programs and management team in UTeM.

### 1.4.2 Platform

The platform to be used for this project is web browser for the frontend system as the interface to the user and Node.js for the backend system as the server provides service to the frontend system. The framework used for the frontend system is Vue.js which is open-source model-view-viewmodel. The backend framework used for the system is Express.js. Express.js is also free and open-source software for the Node.js which is suitable for this system. It is used for developing web application and acting as an application programming interface to help in communication between the frontend and backend system.

### 1.4.3 Modules to be developed

#### 1.4.3.1 Student Module

- Participated Programs Module
  - U-Tem Merit will display total point achieved from all the programs and a list of participated programs with calculated marks by adding up the achievement, position and program point in the program.
  - U-Tem Merit will display the status of the program requested by the student to indicate whether it is approved by the organizer.
- Claim Module
  - U-Tem Merit will provide a form for the student to claim the points for the achievement, position and program point in the program which can be UTeM program or out of UTeM program by providing certificate through uploading it.
- Attendance Module
  - U-Tem Merit will display processed programs which will be conducted on that day and student only can take attendance during the program is conducting.
- Programs Module
  - U-Tem Merit will display all the conducted programs and all participated students with their achievement and position.



#### 1.4.3.2 Organizer Module

- Manage Programs Modules

- U-Tem Merit will display a form for the organizer to fill in with the information of program which is going to be conduct.

- U-Tem Merit will display the status of the program for the creator of program for the organizer.

- U-Tem Merit will inform the creator of program any update of the program by the management team through email.

o Programs Module

- U-Tem Merit will display all the conducted programs which are created by the current organizer account only and all participated students with their achievement and position.

- U-Tem Merit will provide service for the organizer to download the file in .csv format which containing the list of the students of a program.

- U-Tem Merit will provide QR Code of the program which will be conducted on that day and approved.



- U-Tem Merit will provide service for the organizer to manage the qualification of the participated students of the programs which are created by the organizer.

#### 1.4.3.3 Management Team Module

o Student List Module

- U-Tem Merit will display all the students and process participated programs by each student.

- U-Tem Merit will calculate the total point achieved in all participated programs by each student.

- Manage Programs Modules

  - U-Tem Merit will display a form for the management team to fill in with the information of program which is going to be conduct.

  - U-Tem Merit will display the detail of the program requested by the organizers for the management team to approve or reject.

  - U-Tem Merit will inform the creator of program any update of the program by the management team through email.

- Programs Module

  - U-Tem Merit will display all the conducted programs and participated students with their achievement and position.

  - U-Tem Merit will provide service for the organizer to download the file in .csv format which containing the list of the students of a program.

  - U-Tem Merit will provide QR Code of the program which will be conducted on that day and approved.



  - U-Tem Merit will provide service for the management team to manage the qualification of the participated students.

- Other Programs Module

  - U-Tem Merit will display the students who request the point achieved in program which is out of UTeM programs for management teams to approve or reject the qualification of the students to get the points.

## 1.5 Deliverable

The deliverables of this project are the literature review, project methodology, analysis, design, implementation, testing and finally the complete U-Tem Merit.

## 1.6 Project Significance

This project is built especially for students, organizer and management teams in UTeM. By using this system, this will definitely decrease the time consume on managing programs, taking attendance and collecting data of each program. As a result, it saves the time for all of three target users.

## 1.7 Expected Output

There are some of the expected outputs in this project. Firstly, U-Tem Merit is built perfectly at the end of this project, all of the modules stated above are built accurately. Secondly, the website is widely accepted by students, organizer and management team, it is easy to use and learn to use. Thirdly, the website helps students, organizer and management team of reducing time consume in managing merit point based program.



## 1.8 Conclusion

In conclusion, this chapter does the overview of this project which include problem statements, objectives, scope and so on. In following section, it will preview to the literature review of this project.

# CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

## 2.1 Introduction

This chapter aims to confer on the related works regarding merit point attendance system. The literature will describe in details on the types of merit point attendance system. It will act as the conclusion of the related topic of published researched. This will give the clear description on what has just been done, what is commonly acknowledged, what is emerging and what is the present state of thinking on the topic.

## 2.2 Facts and findings

In software engineering, facts and findings are the truth and information collected based on techniques which contain sampling of existing documents, research, observation and so on (*Definition Of Fact Finding Techniques Information Technology Essay*, no date). Facts and findings are important, they helps software development team to design a tailor-made solution, identify objections before they come up and so on (*10 benefits of fact-finding | ThinkAdvisor*, no date). Section below describes the facts and findings related to this project.



### 2.2.1 Domain

Time and attendance systems (TNA) are used to track and monitor when employees start and stop work. A time and attendance system enables an employer to monitor their employees working hours and late arrivals, early departures, time taken on breaks and absenteeism(*How time and attendance technology can benefit your small business clients*, 2017). Based on the statement above, we can know that attendance system is used to monitor a group of people in an organization. U-Tem

Merit based on this concept to track and monitor the student's participant a program until the end of the program. Attendance system with merit point will be another concept. This system monitoring the attendance of the students and also the performance of the students in a program. Different points will be assigned to each student based on their role and achievement in the program. Different type and level of programs also will manipulate the points given to each student. The points will act as the reward and also the prove of the students having active performance in university. It is useful in their future when getting a interview or a job.

### 2.2.2 Existing System

Till date, there are many applications that utilize attendance system to perform collecting attendance of the student when conducting a lecture, speech, program and etc.

One of the example is the myUTeM(*myUTeM - Apps on Google Play*, no date) which is implemented in this few year for students to check the grades for all the semester, take attendance by scanning the QR code, make a complaint, detail of the students and list of questions and answers relating to the application. The attendance system used in the application by scanning the QR Code created on one of UTeM website by using the lecturer account with the camera. The QR Code will be refreshed in every specified second to ensure only the present student able to scan the QR Code. There has another way to take attendance which is manually entering the matric number of the student.

Another example is the Time Attendance Software(*Time Attendance System Malaysia | Attendance App | Attendance System*, no date) which is a cloud-based software which captures employees TIME IN & TIMEOUT from biometric machines / Face Recognition readers / Mobile Apps. The attendance data will real time sync to the Time Attendance Software. Time Attendance Software has the customizable options to create multiple shift patters with different Overtime & Lateness rules. Attendance data can be linked to info-tech Payroll Software or can be export to Excel, CSV or real time sync to other applications.

Next, StaffAny(*Track time from clock-in to timesheets with StaffAny*, no date) is mobile application to track the employee who is in or late from everywhere. The application provides timesheets to record the time employee clock-in. Once someone clocks-in, StaffAny reflects this instantly on the timesheets. Don't wait till the end of the month to process the records. Every user sees the same records. All edits recorded for 100% transparency. With StaffAny's mobile app, users can easily see who is in or late.

### 2.2.3 Technique

This section discusses about the other techniques or approaches which are also applicable and related. This system is able to apply the geolocation which refers to the identification of the geographic location of a user or computing device via a variety of data collection mechanisms. Typically, most geolocation services use network routing addresses or internal GPS devices to determine this location(*Geolocation: Displaying User or Device Position on Maps*, no date). This technology accords with this system because most of the students will take attendance by mobile devices which are built-in with GPS services. By using this technology, system can identify the location of user when the attendance taken by the user to prevent any use of deception.



## 2.3 Project Methodology

Project methodology including system development life cycle (SDLC) and database life cycle (DBLC) is important because they help to convert the requirements of a project into a purposeful and fully operational structure and defines the steps involved for applying a database respectively.

### 2.3.1 System Development Life Cycle

In this project, waterfall model is used as the system development life cycle approach. Waterfall model is the earliest SDLC approach that was used for software development. The waterfall model illustrates the software development process in a linear sequential flow (*SDLC - Waterfall Model - Tutorialspoint*, no date). Waterfall model is divided into separate phases. The sequential phases in waterfall model are:

- Requirement Gathering and Analysis – Capturing all the possible requirements which are going to be developed in this step and will be recorded in a requirement specification document.

- System Design – In this step, the documented requirement specifications will be analyzed for preparing the design of the system. The hardware such as laptop or microprocessor and system requirements will be stated in this step to define the overview system architecture.

- Implementation – For this step, the system design will be input. The small part of the programs is developed and called as units, which will be integrated in the next step. The developed units will be tested for the functionality to ensure it is matched with the requirement specification. The tests carried out are also called as Unit Testing.

- Integration and Testing – All the small developed parts of programs are integrated into a system after unit testing. If there have any faults or failures, they will be tested out by the whole system post integration.

- Deployment of system – The system could be deployed in the customer environment when the functional and non-functional testing are done. The product could be released into the market.

- Maintenance – The product will be improved after released into the market if any changes of the system. If there are some bugs or problems which come up in the client environment, the improvement will take place to solve the issues. All the works will be delivered in the customer environment.

The following illustration is a representation of the different steps of the waterfall model:

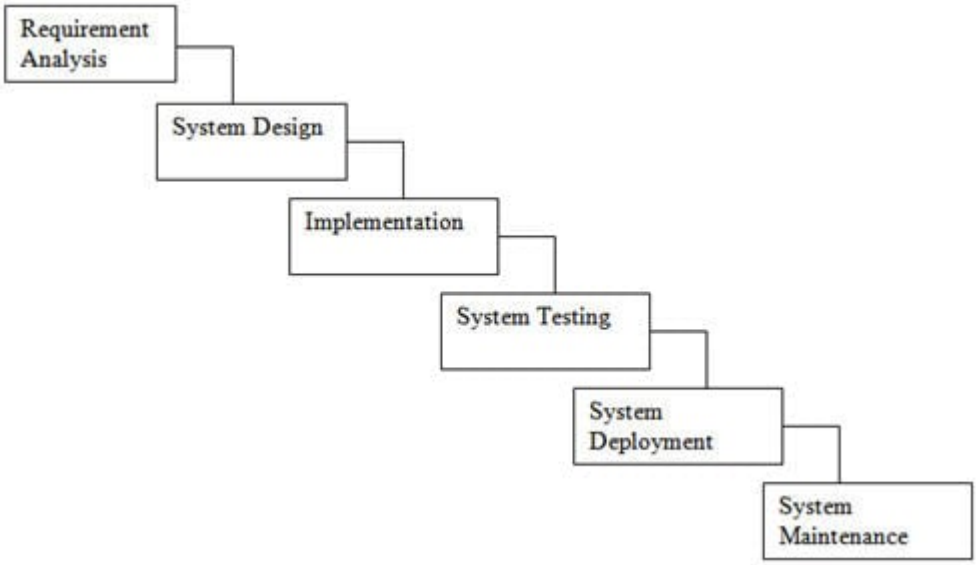



**Figure 2.1: Waterfall Model**

### 2.3.2 Database Life Cycle

Database life cycle defines the stages involved for implementing a database, starting with requirements analysis and ending with monitoring and modification. There are five stages in the database life cycle which are:

- Requirement Analysis - Requirement analysis is the first and most important stage in the life cycle database phase. This stage involves assessing the informational needs of an organization so that a database can be designed to meet those needs.

- Logical Design – During this stage, a conceptual model is created based on the needs assessment performed in stage one. The conceptual model is typically an entity-relationship diagram (ERD) that shows the tables, fields, primary keys of the database and so on.

- Physical Design – This stage is all about maximize database efficiency. In simple word, this stage involves finding ways to speed up the performance of the RDBMS.

- Implementation – During this stage, the tables developed in the ERD are converted into SQL statements. The SQL statements are then executed in the RDBMS to create a database.

- Monitoring, Modification, and Maintenance – A successfully implemented database must be carefully monitored to ensure that it is functioning properly and that it is secure from unauthorized access.

## 2.4 Project Requirements

Project requirements such as software requirement and hardware requirement are important to get the U-Tem Merit done. Section below lists out the software and hardware requirement for this project.

### 2.4.1 Software Requirement

**Table 2.1: List of Software**

| | |
|---|---|
| Visual Studio Code | A source-code editor which is used to code the front end and backend system. |
| Star UML | An open-source software modelling tool that supports the Unified Modelling Language (UML) framework for system and software modelling. |
| Microsoft Word | A word processor published by Microsoft. It is used to type and save document in this project. |
| MongoDB Compass | An open-source monitoring tool for the MongoDB database. |
| NoSQLBooster | A shell-centric cross-platform graphical user interface tool for MongoDB. |

### 2.4.2 Hardware Requirement

**Table 2.2: List of Hardware**

| | |
|---|---|
| Laptop/Desktop | Laptop or desktop is required to build the U-Tem Merit. |
| Smartphone with internet connection | Smartphone with internet connection is used to access the website to use the feature of the system. |
| Smartphone with camera | Camera of smartphone is used to scan the prepared QR Code to take attendance |

### 2.4.3 Other Requirements

None

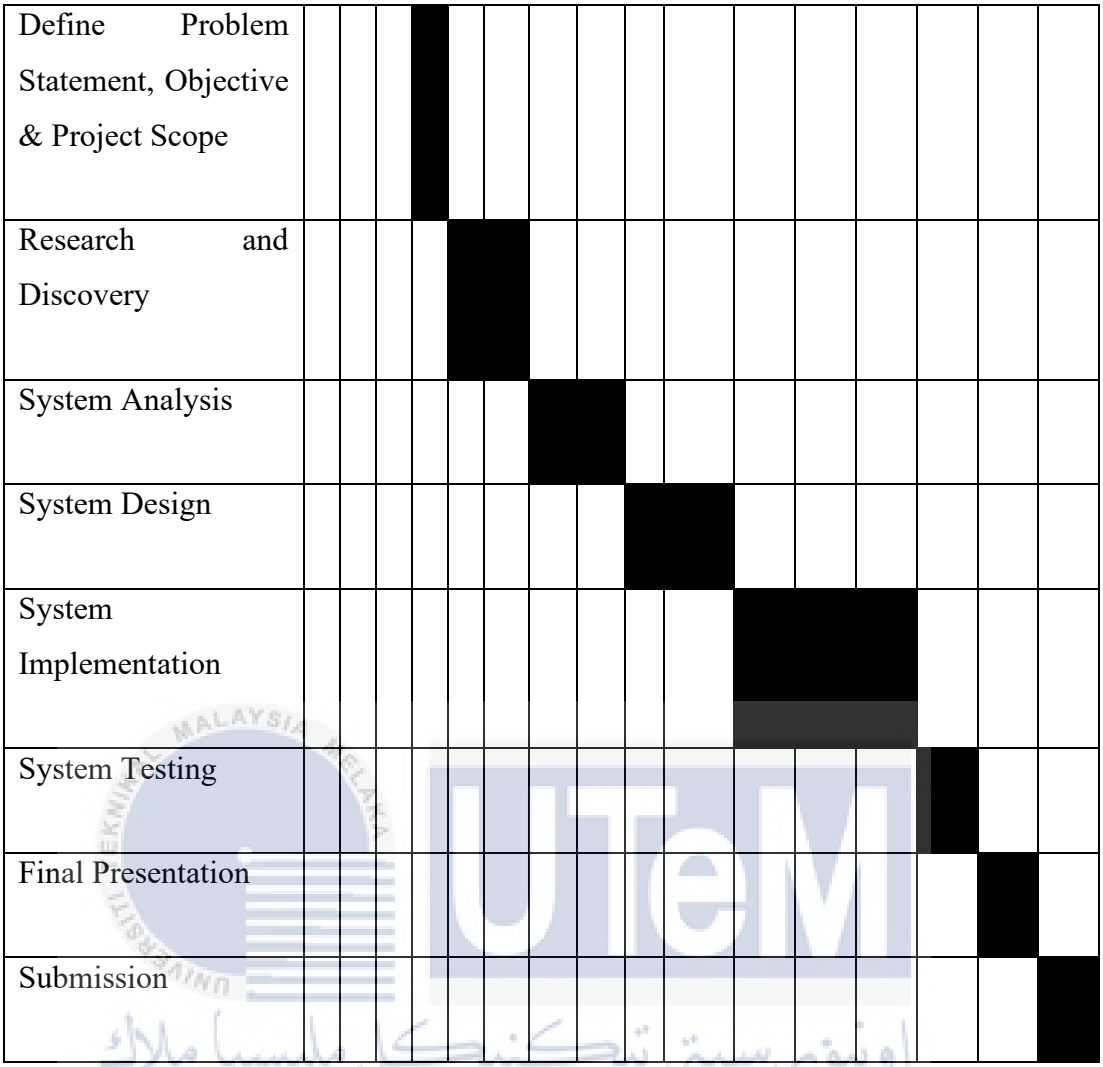
## 2.5 Project Schedule and Milestones

A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes, although they are particularly useful for simplifying complex projects. Table below shows the Gantt chart for this project.



**Table 2.3: Gantt Chart**

| Week / Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Proposal Preparation | ■ | | | | | | | | | | | | | | | |
| Proposal Correction/Improvement & Approval | | ■ | | | | | | | | | | | | | | |
| Proposal Submission | | | ■ | | | | | | | | | | | | | |






## 2.6 Conclusion

In conclusion, this chapter have reviewed some of the previous product that have been completed by other researchers. In next chapter, it will have the further describe about the approach on the proposing a solution to the U-Tem Merit.

# CHAPTER 3: ANALYSIS

## 3.1 Introduction

This chapter details the analysis that was done on this system. Analysis was done by analyzing the existing system, finding out the problems and implementing new ideas and innovation as solutions. This is done to define the goals and objectives.

## 3.2 Problem Analysis

The current system is the myUTeM implemented by Universiti Teknikal Malaysia Melaka. It does not provide the information about the points achieved by the students. The programs record the attendance of the student which does not including the role and achievement of the student and the type of the program. The system also does not provide service to generate report for management team further use.

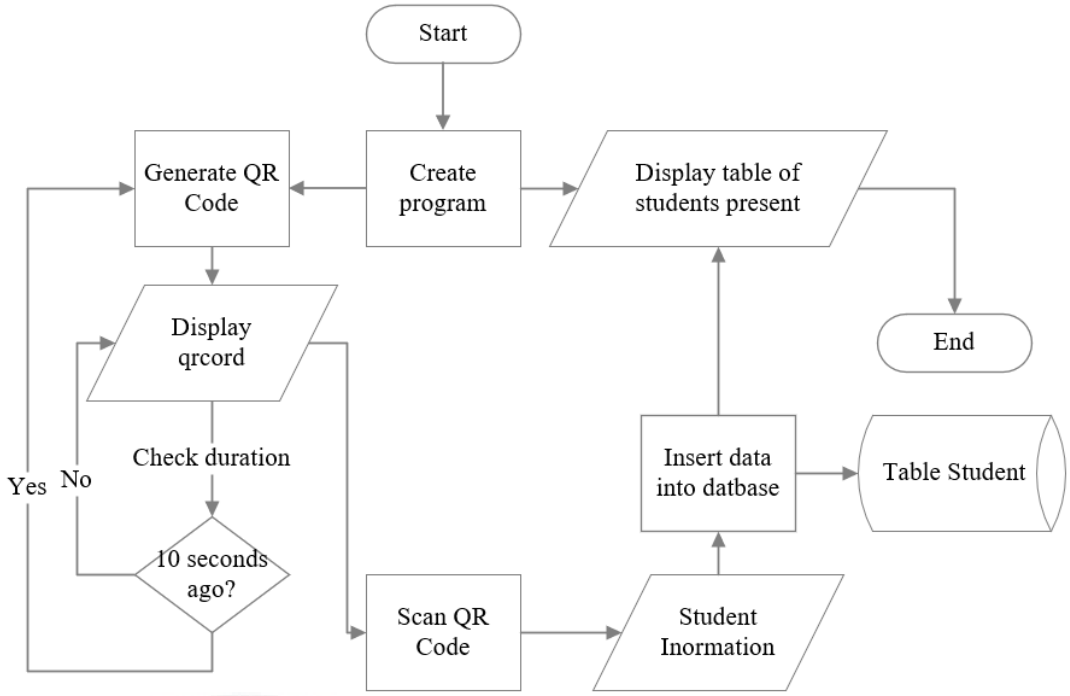



**Figure 3.1: Current Flow**

Figure 3.1 describes the current flow which are including some steps such as creating a program, generating QR code, taking attendance by student and recording the attendance. Without the system, the program details do not be checked by the management teams and only record the attendance of students, there do not have merit point concept implemented to record the role and achievement of student. The attendance of the program has to export manually by the management teams for further use.

## 3.3 Requirement Analysis

The requirement analysis for this chapter comprises of functional, software, hardware, and network requirements. Below is the proposed flow of patrons after implementation and usage of application.

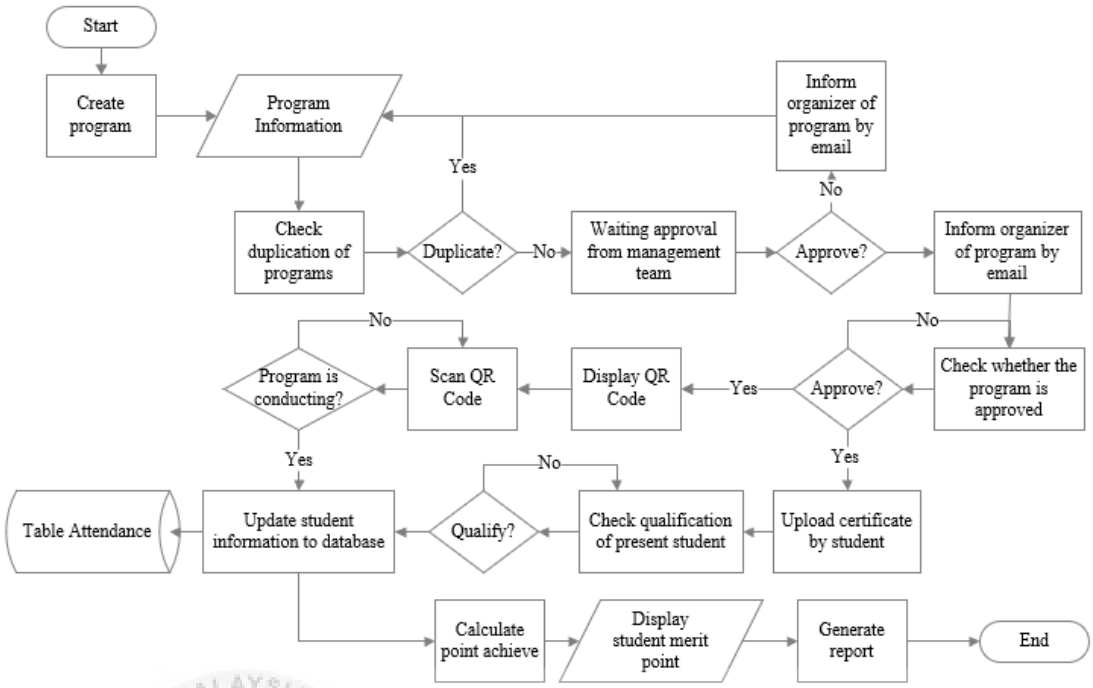



**Figure 3.2: U-Tem Merit Flow**

Figure 3.2 describes the proposed flow after U-Tem Merit implementation. The system will record the attendance with the merit point set based on the role of the student and achievement made by the student and also the grade of the program. In the process, the management team can review the program detail to decide whether the program submitted by the organizer meets the standard compliant. The organizer will be informed if any update from the management team. For the organizer, the organizer can check the qualification of the student to get the points for the role and achievement based on the certificate uploaded by the student. Student and management team able to review the merit point given to the students for each program participated. Management team able to generate the report which included the participated student information.

### 3.3.1 Date Requirement

This section will describe the data that the system should input and output, as well as the data that is needed to be stored in the database. This will also function as a data dictionary.

**Table 3.1 Table of Attendance**

| Attribute | Description | Data Type | Sample Data | Constraint |
|---|---|---|---|---|
| programId | The student participates program | String | 60b31de3d3c7b03e70a2c1cd | Primary key |
| userId | The participant of program | String | 60a20fb754e86947f8076bf4 | Primary key |
| achievementPointId | The id for the achievement point | String | 608931016c4e1c97a2a71e31 | Not null |
| achievementStatus | The status for the qualification of the achievement | Boolean | True, false | Not null |
| positionPointId | The id for position point | String | 608938246c4e1c97a2a71e32 | Not null |
| positionStatus | The status for the qualification of the position | Boolean | True, false | Not null |
| levelPointId | The id for the level point | String | 6089308bb4314c393c5e0c6e | Not null |
| levelStatus | The status for the qualification of the program | Boolean | True, false | Not null |
| date | The date of the outside program | Date | 2021-06-03 | |
| time | The time of the outside program | Time | 11:00 | |
| location | The location of the outside program | String | 103, Jalan Bukit Baru, Taman Melaka, 75450, Melaka, Malaysia | |



**Table 3.2: Table of Location**

| Attribute | Description | Data Type | Sample Data | Constraint |
|---|---|---|---|---|
| locationId | The id for each location | String | 60a1249e08ce5720c43b0e08 | Primary key |
| room | The room of building UteM | String | BK 1 | Not null |
| building | The building of UteM | String | FTMK | Not null |
| campus | The campus of UteM | String | Main Campus | Not null |

**Table 3.3: Table of PointList**

| Attribute | Description | Data Type | Sample Data | Constraint |
|---|---|---|---|---|
| pointId | The id for each point | String | 6089308bb4314c393c5e0c6e | Primary key |
| type | The type of the point | String | Level, position, achievement | Not null |
| name | The name of the point | String | Faculty, participant, T10 | Not null |
| point | The merit point | Integer | 2 | Not null |



**Table 3.4: Table of Program**

| Attribute | Description | Data Type | Sample Data | Constraint |
|---|---|---|---|---|
| programId | The id of program | String | 60b31de3d3c7b03e70a2c1cd | Not null |
| name | The name of program | String | Qing Yi | Not null |
| levelId | The id of the program level | String | 6089308bb4314c393c5e0c6e | Not null |

| | | | | |
|---|---|---|---|---|
| locationId | The id of the program location | String | 60a1249e08ce5720c43b0e08 | Not null |
| date | The date of program conducted | Date | 2021-06-10 | Not null |
| time | The time start of program | Time | 11:00 | Not null |
| duration | The duration of the program | Integer | 5 | Not null |
| status | The indicator of the program to show whether is approved | Boolean | True, false | Not null |
| remark | The message for the creator from management team | String | The level is too high | Not null |
| creator | The id for the creator of program | String | 60a20fb754e86947f8076bf4 | Not null |

**Table 3.5: Table of User**

| Attribute | Description | Data Type | Sample Data | Constraint |
|---|---|---|---|---|
| userId | The id of user | String | 60b31de3d3c7b03e70a2c1cd | Not null |
| firstName | The first name of user | String | Chon Yao Jun | Not null |
| lastName | The last name of user | String | 6089308bb4314c393c5e0c6e | Not null |
| contactNumber | The contact number of users | Integer | 60197710427 | Not null |
| matricNumber | The matric number of users | String | B031810223 | Not null |
| email | The email of user account | String | lesliecyj@gmail.com | Not null |

| | | | | |
|---|---|---|---|---|
| password | The password of the account | String | $2b$10$Od5oPU4wi2oRJsHNawOiJ.eDmHE6wRuBUYGRw7slZ3Ez8dAjLPAHe | Not null |
| role | The role of the user | String | Student, organizer, admin | Not null |

### 3.3.2 Functional Requirement

This section defines and describes the functional requirements of the U-Tem Merit. These requirements will be split into three sections to account for the role of user.

#### 3.3.2.1 Student Module

**Table 3.6: Student Module Requirement**

| Requirement | Description |
|---|---|
| View participated program | Enable student to review their participated program detail and the point given for each program. |
| Claim attendance by certificate | Enable student to claim the attendance to get the points by providing the certificate. The organizer of program can be UTeM or other organization. |
| Take attendance | Enable student to take attendance when the program is conducting by scanning QR Code or manually select the program. |
| View all conducted programs | Enable student to review all the UTeM program conducted |

**Table 3.7: Organizer Module Requirement**

| Requirement | Description |
|---|---|
| Register program | Enable organizer to register program and edit program if any requirement from management team. |
| Manage participants | Enable organizer to manage the participants of the program by reviewing the certificate provided by participants. |
| QR code generator | Enable organizer to get the QR Code of the program for student to take attendance |
| View all own programs | Enable organizer to review the all own conducted programs |
| Generate program report | Enable student to take attendance when the program is conducting by scanning QR Code or manually select the program. |
| View all conducted programs | Enable student to review all the UTeM program conducted |



**Table 3.8: Management Team Module Requirement**

| Requirement | Description |
|---|---|
| View all students information | Enable management team to view all the student information including the merit point achieved and participated programs. |
| Register program | Enable management team to register program and edit program. |
| Manage requested program9s by other organizers | Enable management team to approve or reject the requested programs. Inform the organizer with email if any update of program. |
| Manage participants | Enable management team to manage the participants of the program by reviewing the certificate provided by participants. |

| | |
|---|---|
| QR code generator | Enable management team to get the QR Code of the program for student to take attendance |
| View all programs | Enable management team to review the all conducted programs |
| Generate program report | Enable student to take attendance when the program is conducting by scanning QR Code or manually select the program. |
| Manage students who request the attendance of outside programs | Enable management team to approve or reject the qualification of student to get the merit point for outside programs. |

### 3.3.3 Non-Functional Requirement

Non-functional requirements considered important. They ensure that the system functions as expected and quality and performs in standard level without any error.

**Table 3.9: Non-Functional Requirement**

| Requirement | Description |
|---|---|
| Standard | Standard coding for interface designing, implementing and data coding should be used.<br>Interface designing, implementing and data coding based on standard use for website and programming. |
| Security Requirements | Unauthorized user should not access the website.<br>All data inside database could be accessed by authorized user with assigned valid token. |
| Performance Requirements | Perform under any condition and without errors.<br>Database should be online 24 hours.<br>Server should be running 24 hours. |
| Usability | This system is easy to use and user easy to understand the flow of system |

| | |
|---|---|
| Reliability | The data storing and update process should be consistent. |
| Simple GUI | The interface should be user-friendly and easy to understand for the user.<br>The navigation through the system interface is indicated. |

## 3.4 Conclusion

System requirements is collected through this analysis phase in order to ensure the functionality of the system before the next stages take over and in order to fit the project scope. This chapter has provided several flow of business to visualize the quick system. The system process can be seen clearly with visualizing diagram.

# CHAPTER 4: DESIGN

## 4.1 Introduction

System design and the activities involved of the project is discussed in this chapter. The activities involved includes converting information, functional and non-functional requirements that was identified in the previous chapter into the design specification. This chapter will also discuss about defining the result of the analysis of the preliminary and detailed design.

## 4.2 High-Level Design

High-level design explains the architecture that would be used to develop a system. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.



### 4.2.1 System Architecture

Below is the visual representation of the system architecture. The interaction of all three different roles of user and the system when conducting a program from start to the end:

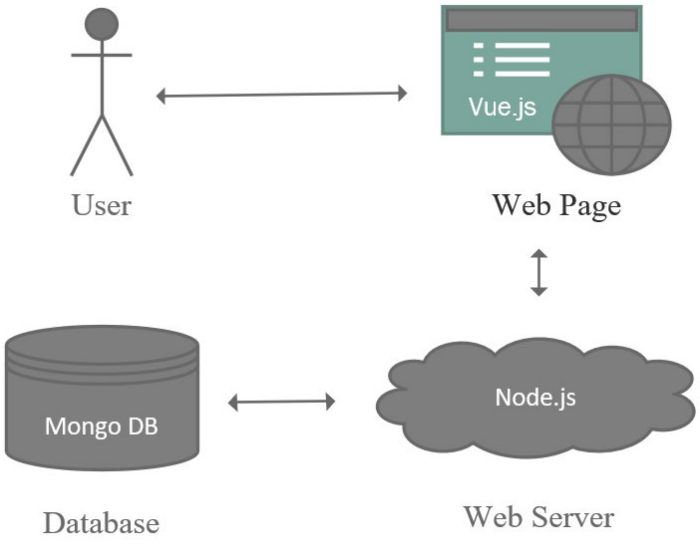



**Figure 4.1: System Architecture Visualized**

The system uses a centralized, cloud and NoSQL database. The database used is Mongo DB. All the data is stored in a single location which is managed by the Node.js server to provide web service to the user. The cloud database makes user easy to access the data from anywhere, as long as the user have an internet connection. MongoDB is one of the NoSQL databases. System is allowed to process larger amounts of data at speed because of it lack of structure and it is easier to be expanded in the future. NoSQL databases regularly use cloud computing.

### 4.2.2 Use Case Diagram

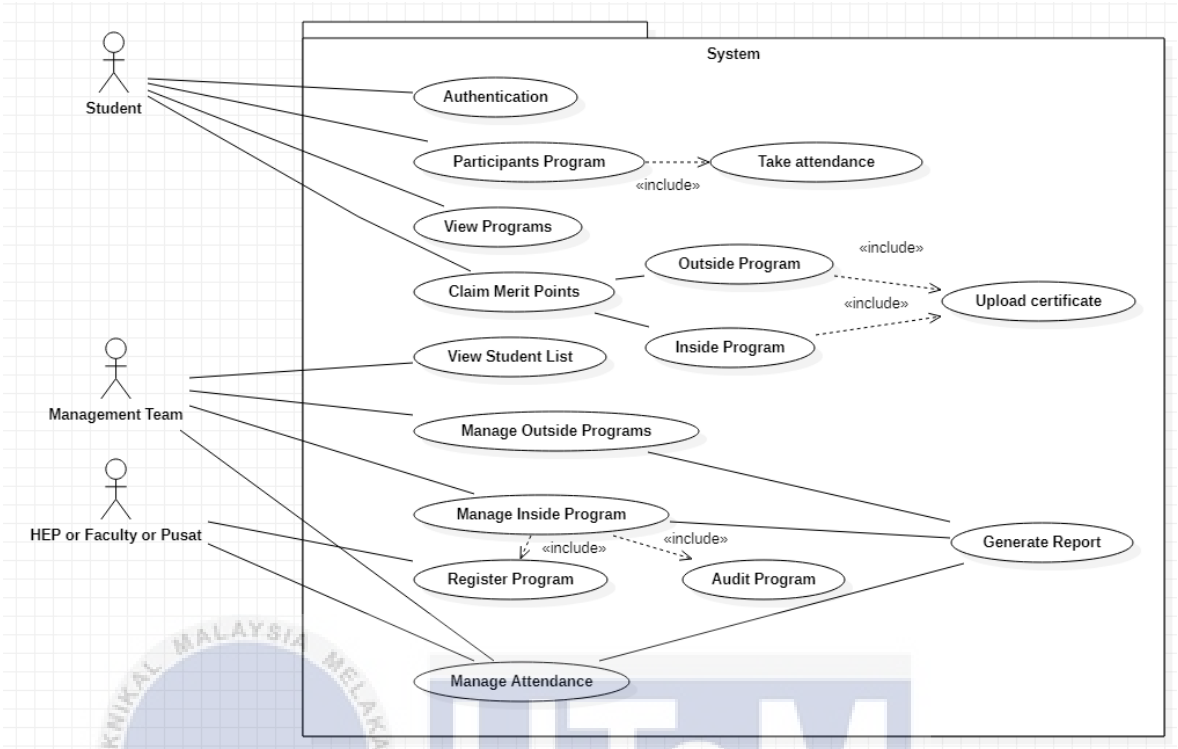



**Figure 4.2: Use Case Diagram of U-Tem Merit**

### 4.2.3 User Interface Design

#### 4.2.3.1 Interfaces for All Roles

*(a) Login*

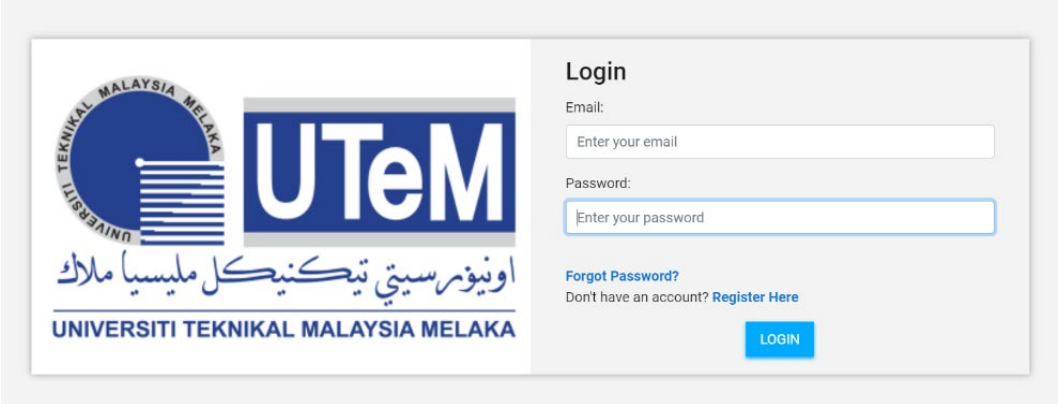



**Figure 4.3: Login**

All users are required to be authorized before using all the features of system. If the user do not have an account, the user has to register as student to able to log in into system. The organizer and new member of management team has to request the management team to help them register as organizer or management team member.

*(b) Forgot Password*

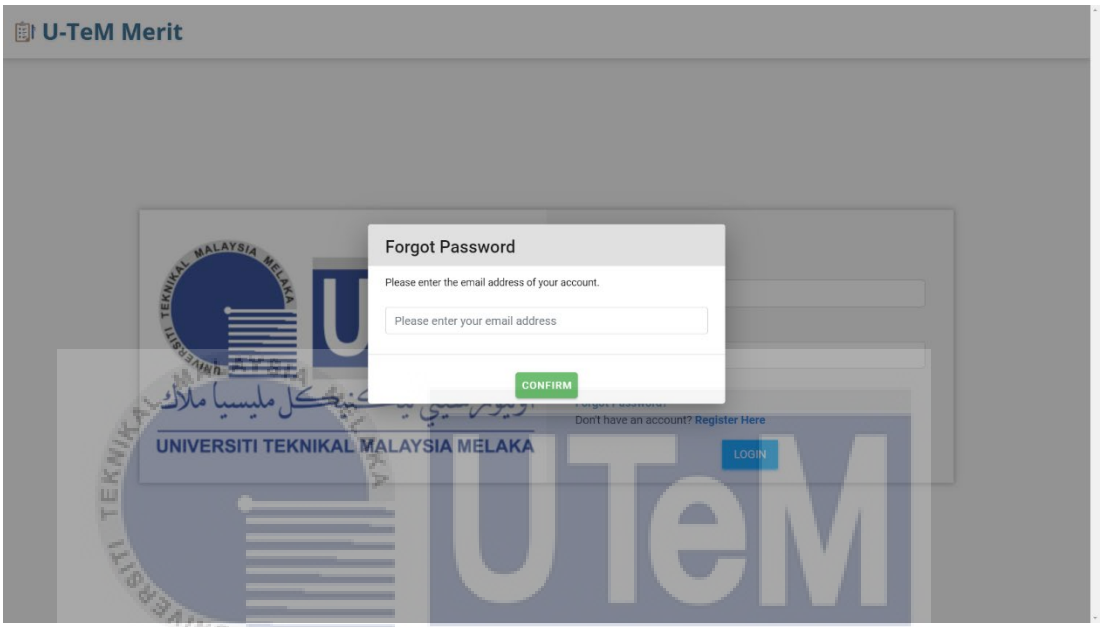



**Figure 4.4: Forgot Password**

User is able to get a new password if the user forgot the password. The user is required to enter the email using for the account to get the new password by email which sent by this system.

*(c) Personal Information*

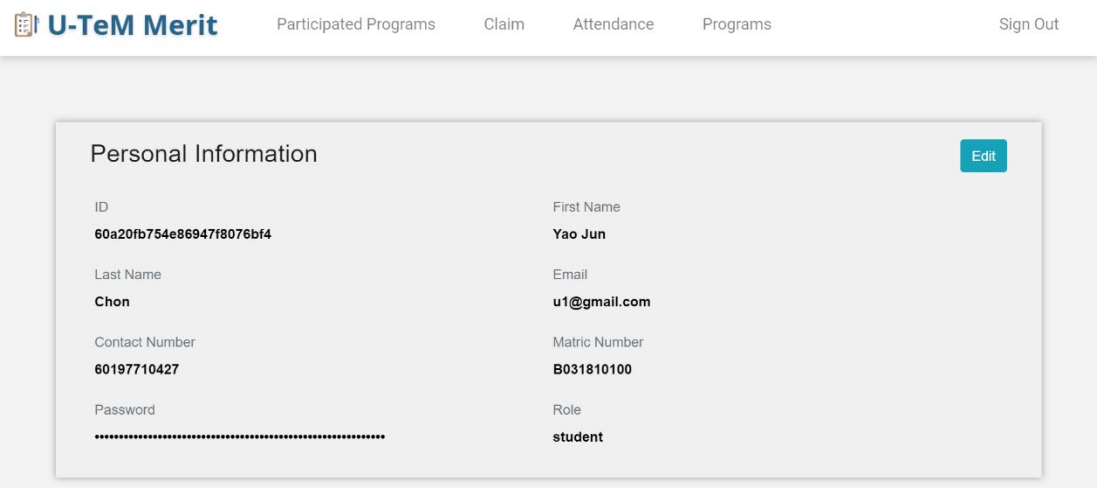



**Figure 4.5: Personal Information**

*(d) Update Personal Information*




**Figure 4.6: Update Personal Information**

User is able to update certain fields of personal information. System is disable the noneditable fields to avoid any problems. The email will be checked to prevent any duplication of email with others when the user try to change as the email is unique for every users.

#### 4.2.3.2 Interfaces for Students

*(e) Register as Student*

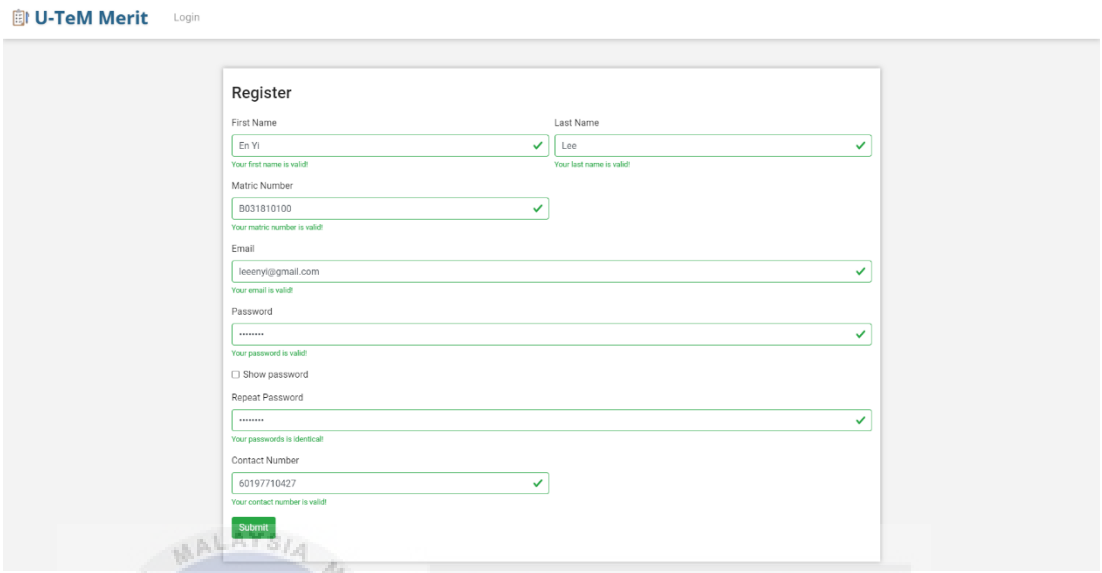



**Figure 4.7: Register as Student**

If student does not have an account, student has to register with the system. The matric number and email is unique. The email and matric number of new account will be checked before register the student to prevent any duplication of them with students' account.

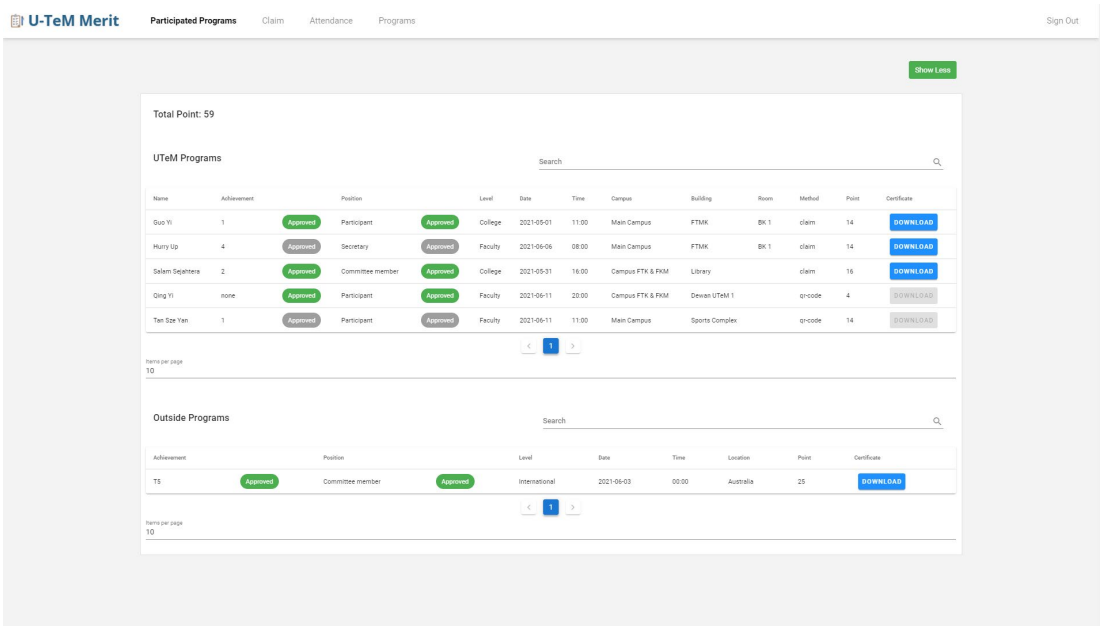



**Figure 4.8: Participated Programs with More Columns**

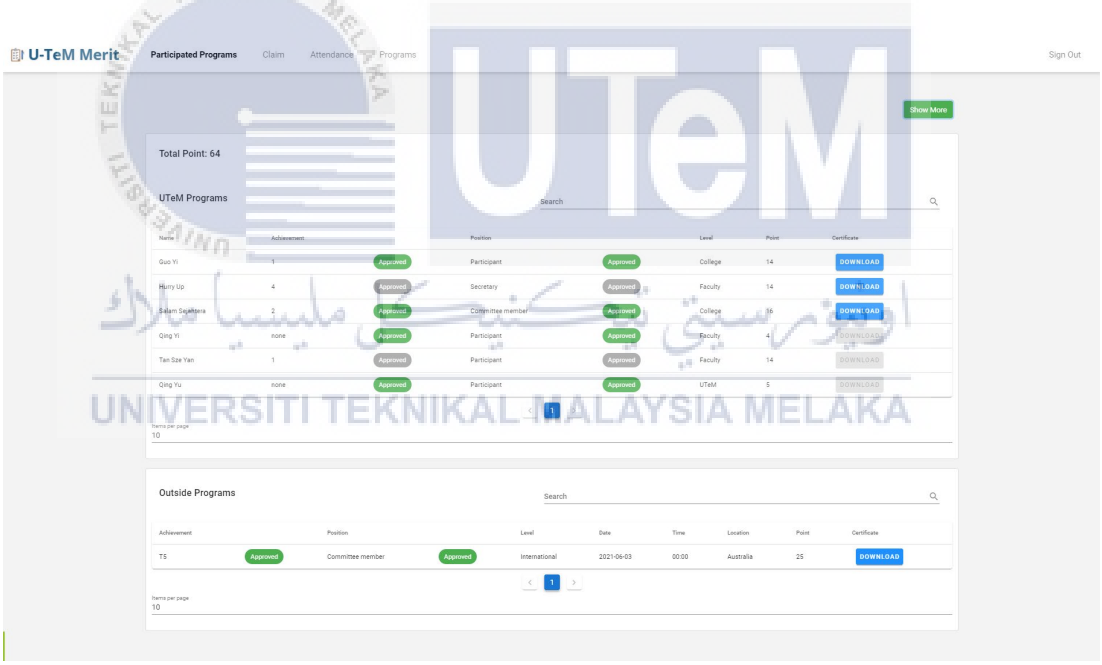





**Figure 4.9: Participated Programs with Lesser Columns**

List of participated programs by the student will be listed. Each programs has the point for the level of the program, position and achievement of the participant. The total point of all participated programs will be shown at top left corner. The top right corner has a button to show or hide some of the columns of table UTeM programs.

*(f) Claim Attendance*

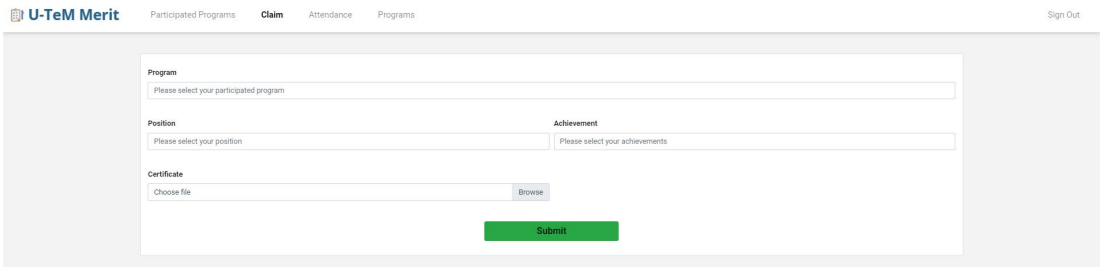



**Figure 4.10: Claim UTeM Program's Attendance**

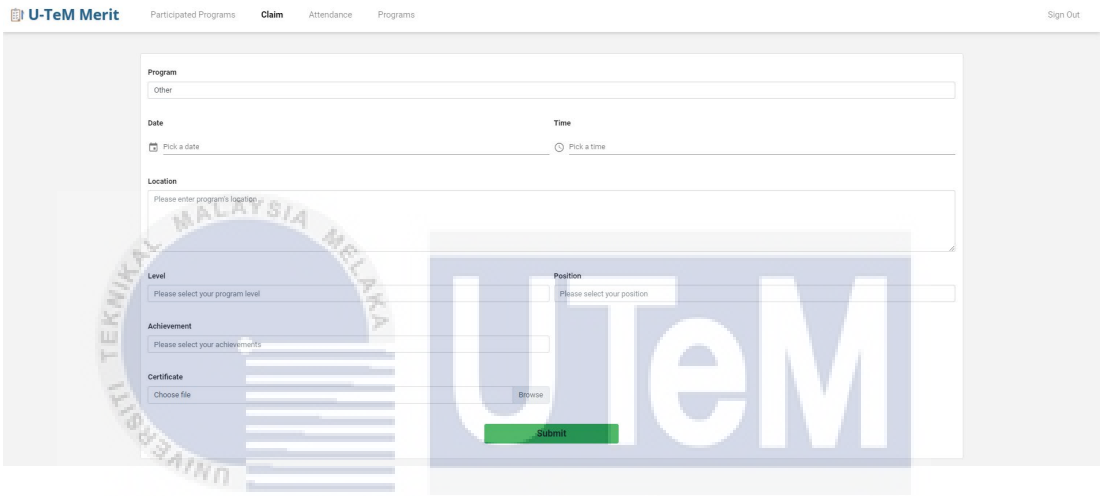



**Figure 4.11: Claim Outsire Program's Attendance**

Student is able to claim attendance of inside or outside program to get the point from the participated program after application form is approved by organizer or management team. Student has to provide certificate to prove the participation. For the outside program's attendance, student has to provide more detail about the program.

*(g) Take Attendance*

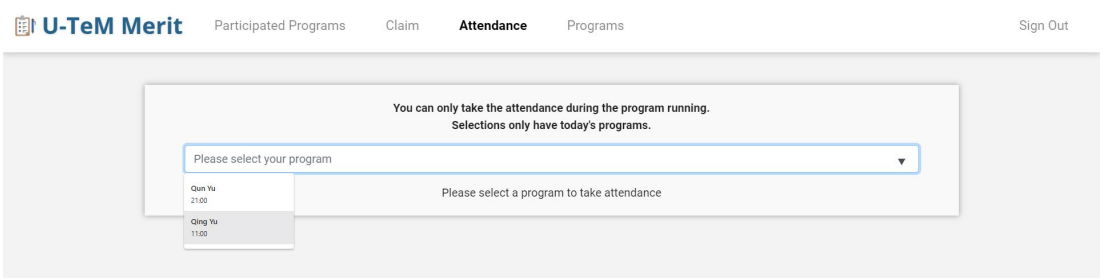



**Figure 4.12: Select Attendance**

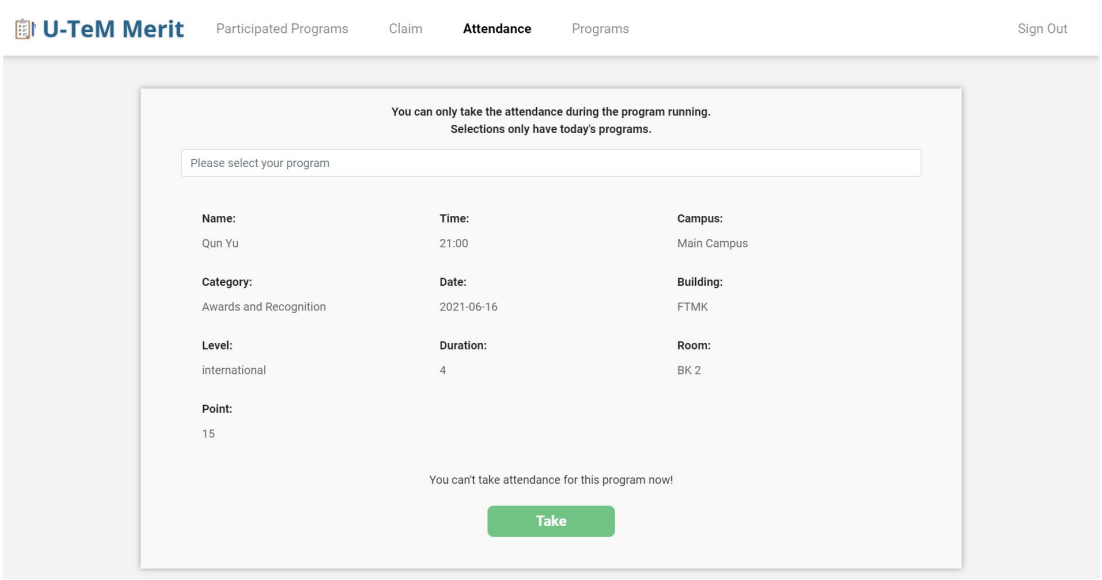



**Figure 4.13: Unable to Take Attendance**

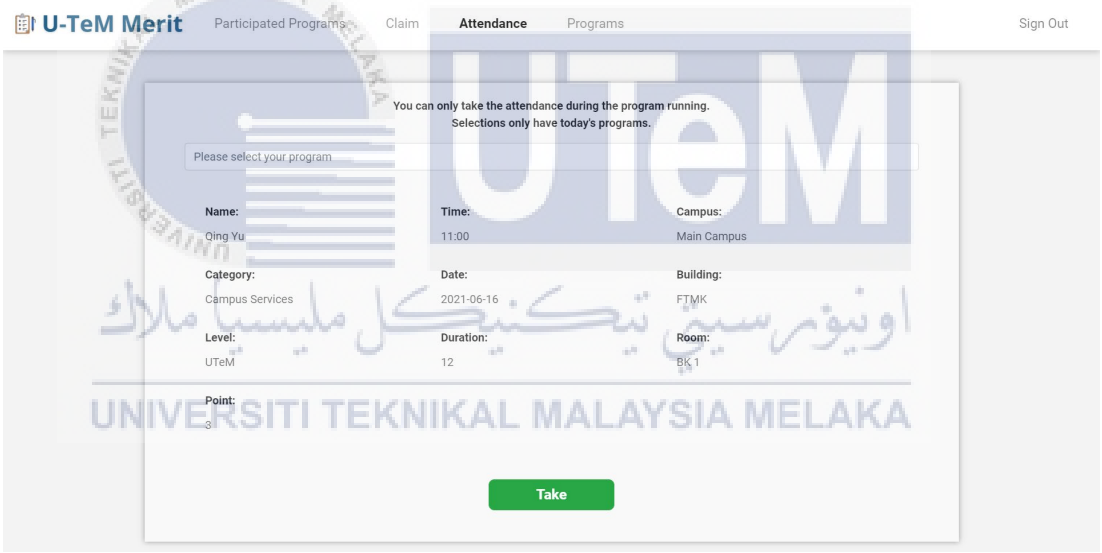



**Figure 4.14: Allow to Take Attendance**

The system will list the programs which are available to take attendance by the day. The system will disable the button of take attendance to disable student to take attendance not in the program time.

*(h) View All Programs*

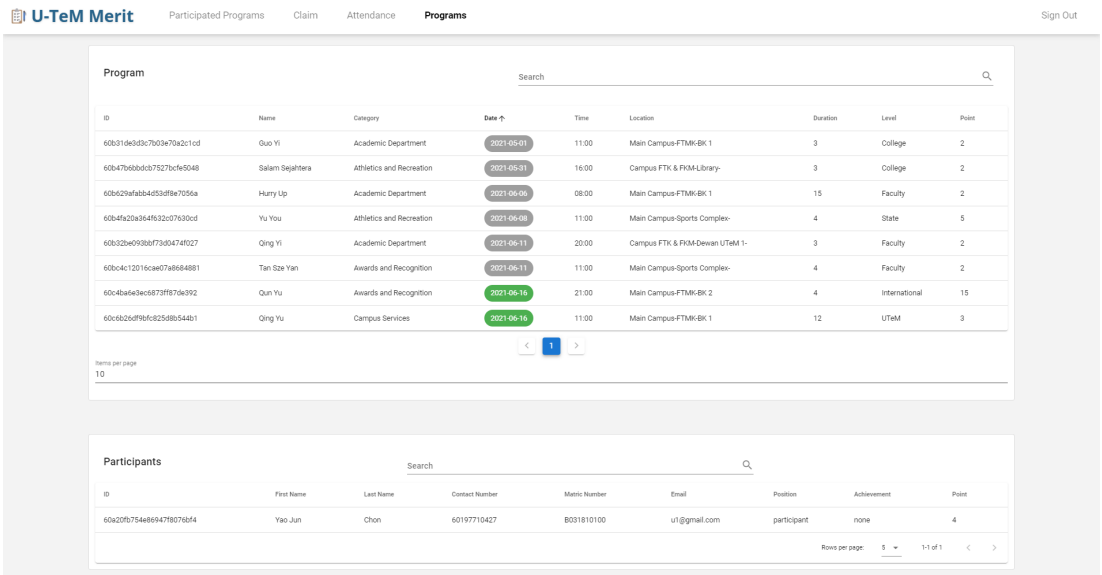



**Figure 4.15: All Programs for Organizer**

The system lists all the approved programs in UTeM for student. The column of date will be highlighted by different colors to indicate the programs are conducted or not. The participants also will be listed with their achievement and position which are approved by organizer or management team for each program.



### 4.2.3.3 Interfaces for Organizer

*(a) Register Program*

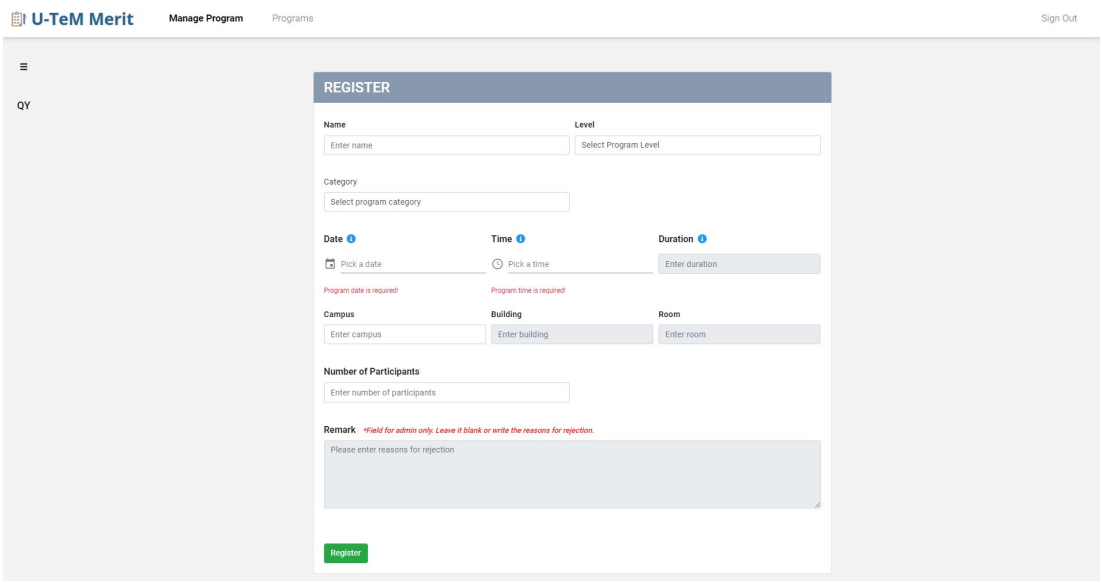


**Figure 4.16: Register Program**

*(i) Update Program*

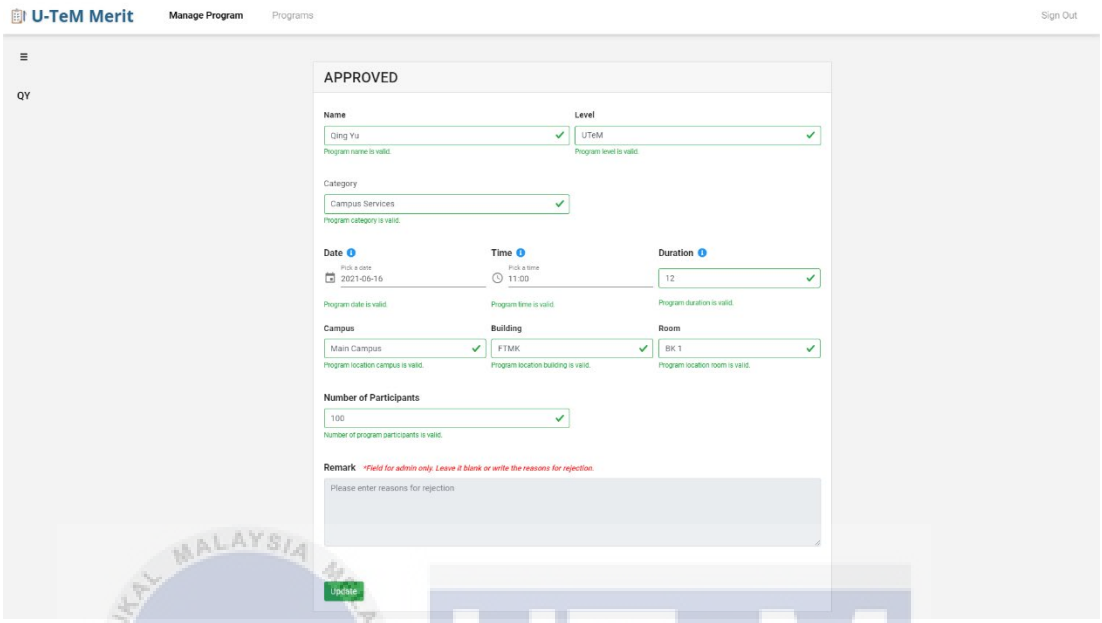



**Figure 4.17: Update Program**

Organizer is able to register new program by this system by providing the detail of the program. The requested unapproved programs created by the organizer will be shown at left hand side for the organizer to update the detail. In the form, there will have a remark area for the management team to left the message for organizer about the reason of rejection.

*(j) Manage Attendance*

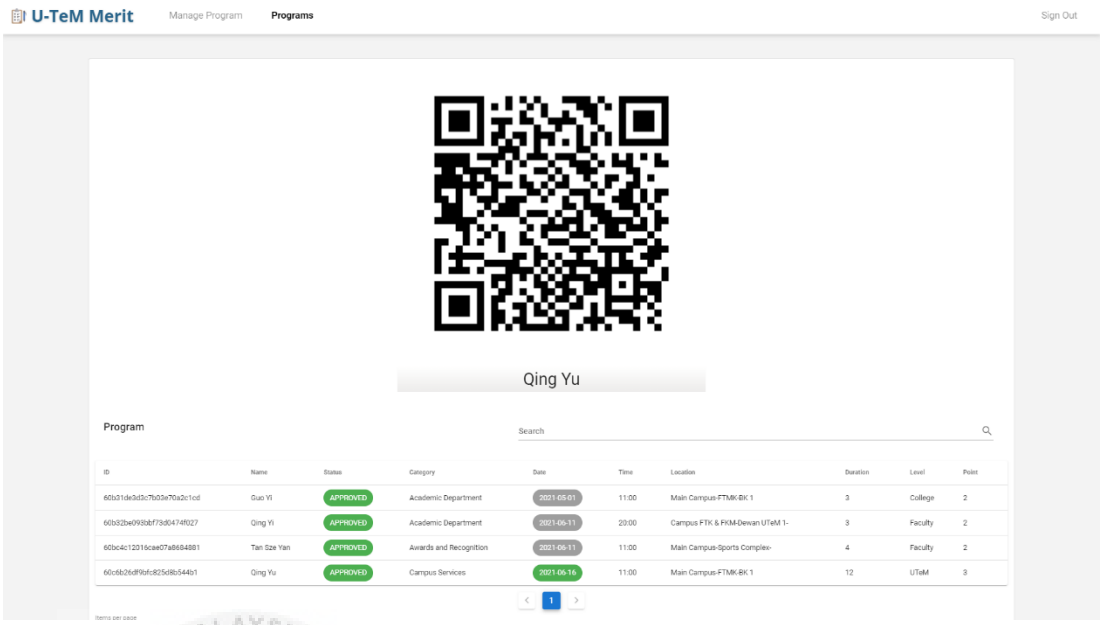



**Figure 4.18: View Program with QR Code**

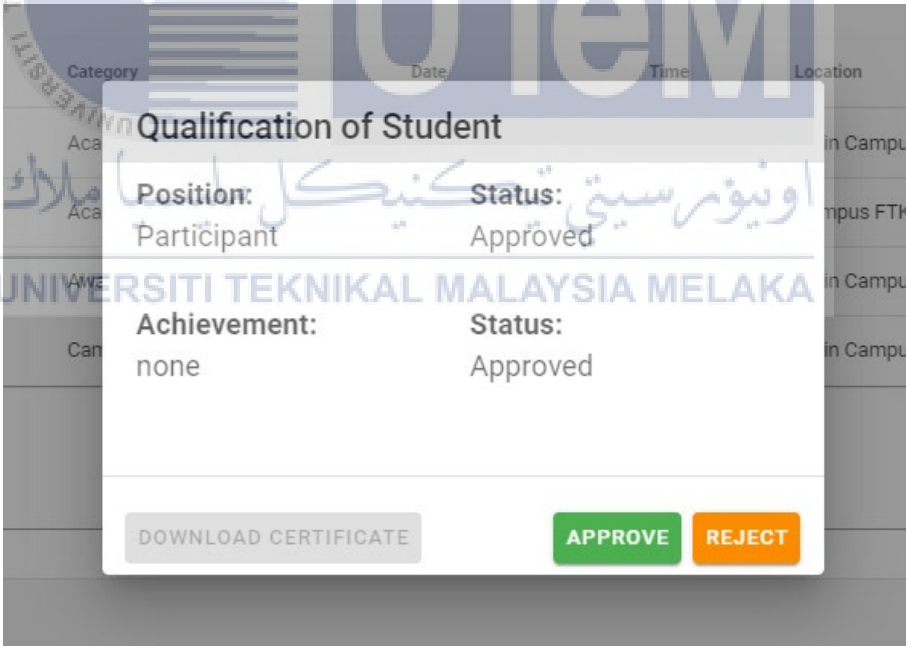



**Figure 4.19: Manage Attendance**

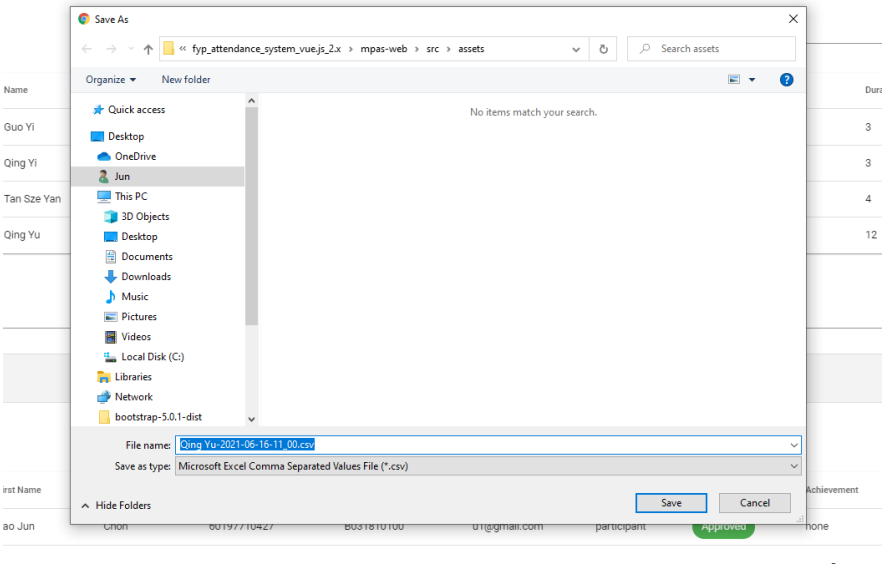



**Figure 4.20: Generate Report**

If the program is approved and conducted, the program will be shown at the table in Figure 4.18. If the program is going to be conducted on the day or after the day, the QR Code will be generated for organizer to save. The student can scan the QR Code to take attendance when the program is conducting. Figure 4.19 shows the organizer can approve and reject the qualification of participants. Organizer also can download the certificate of the participants if the certificate is provided. Organizer also can generate the program report which contains the information of participants for each program.

#### 4.2.3.4 Interfaces for Admin

*(a) Register Organizer or Admin*

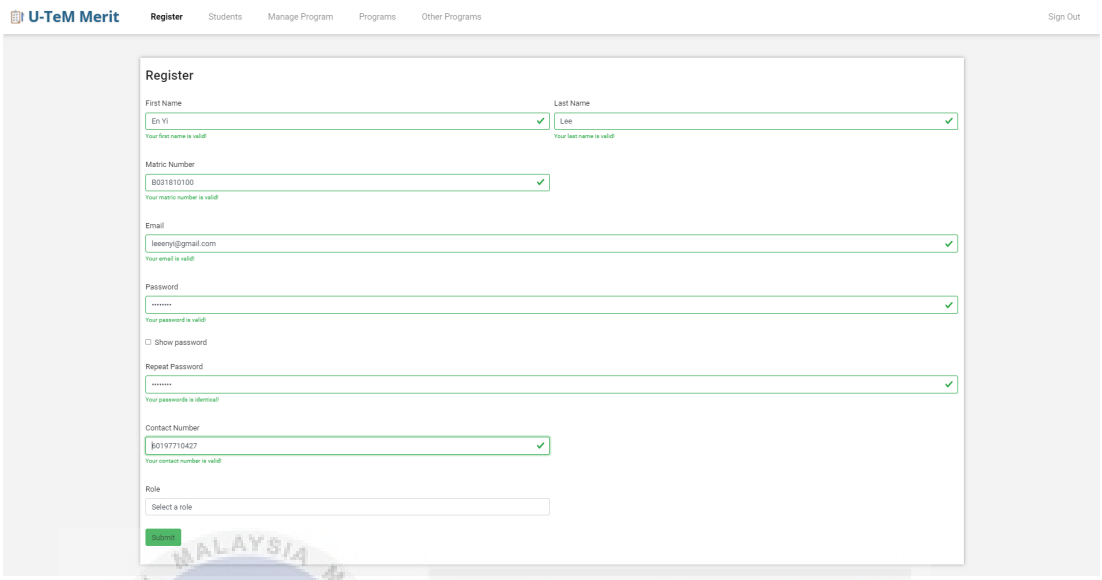



**Figure 4.21: Register for Admin**

Management team can register new management team member or organizer into system.

*(k) Manage students*

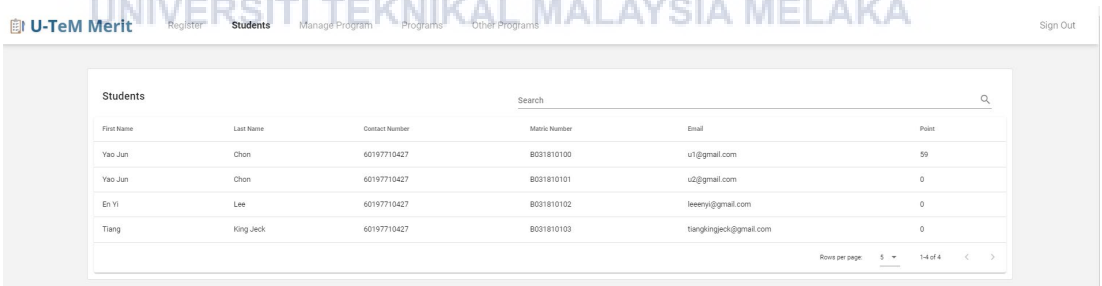



**Figure 4.22: View All Students Detail**

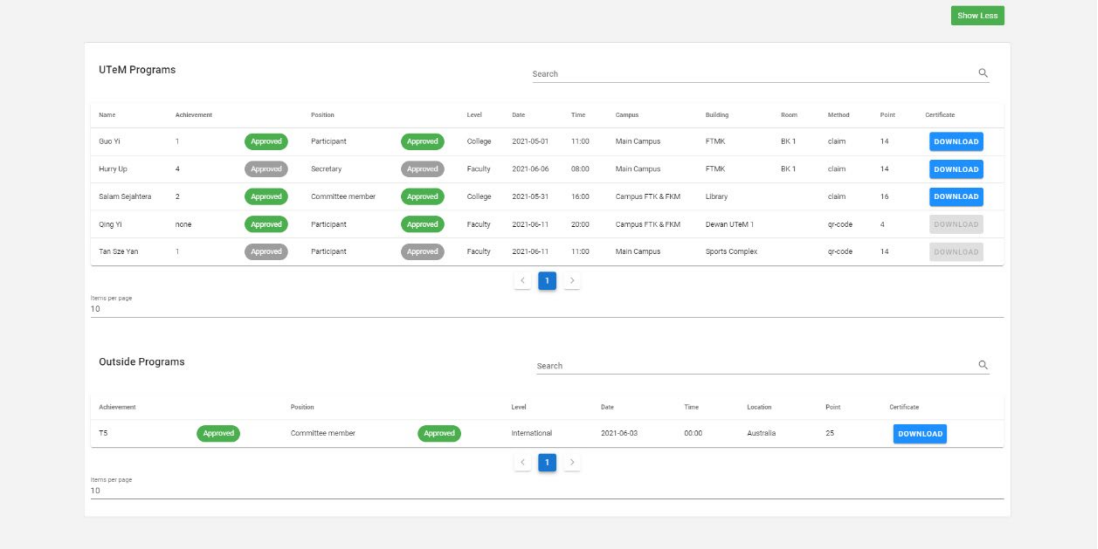



**Figure 4.23: View More Columns of Programs**


**Figure 4.24: View Lesser Columns of Programs**

Management team is able to view all students with the personal information and the merit points achieved by participating programs in figure 4.22. For figure 4.23 and 4.24, all the participated programs which are included the UTeM's programs and outside programs for the selected student in table of figure 4.22. The difference between them is the column for the table UTeM programs. Figure 4.23 has more columns that the table in figure 4.24. Management team can change the number of columns by clicking the button 'Show More' or 'Show Less'

*(l) Manage Program*

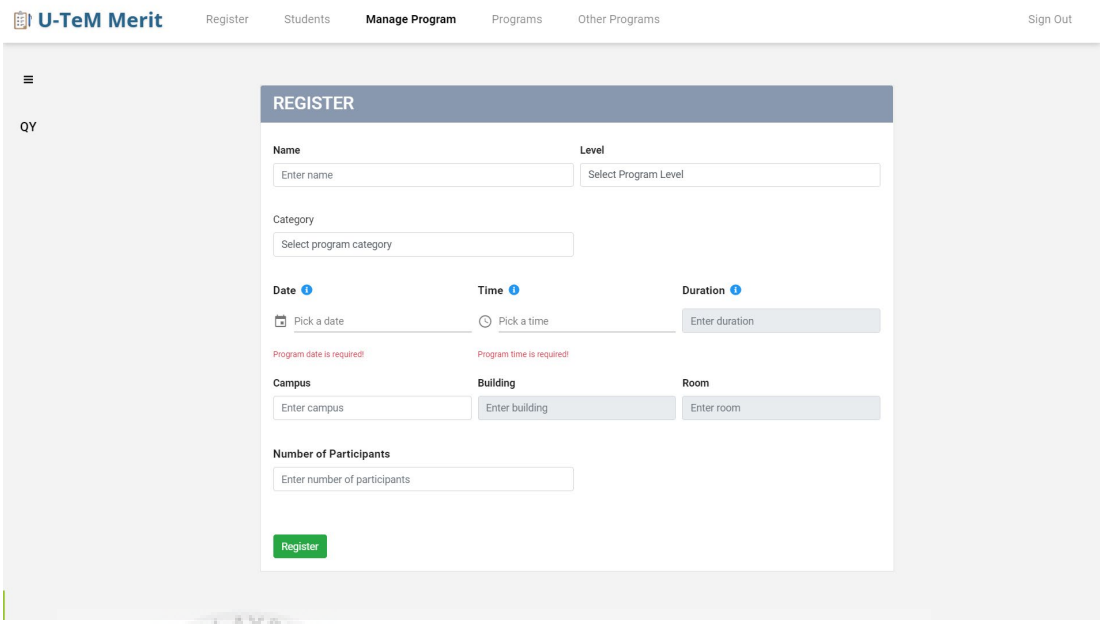



**Figure 4.25: Register Program for Admin**

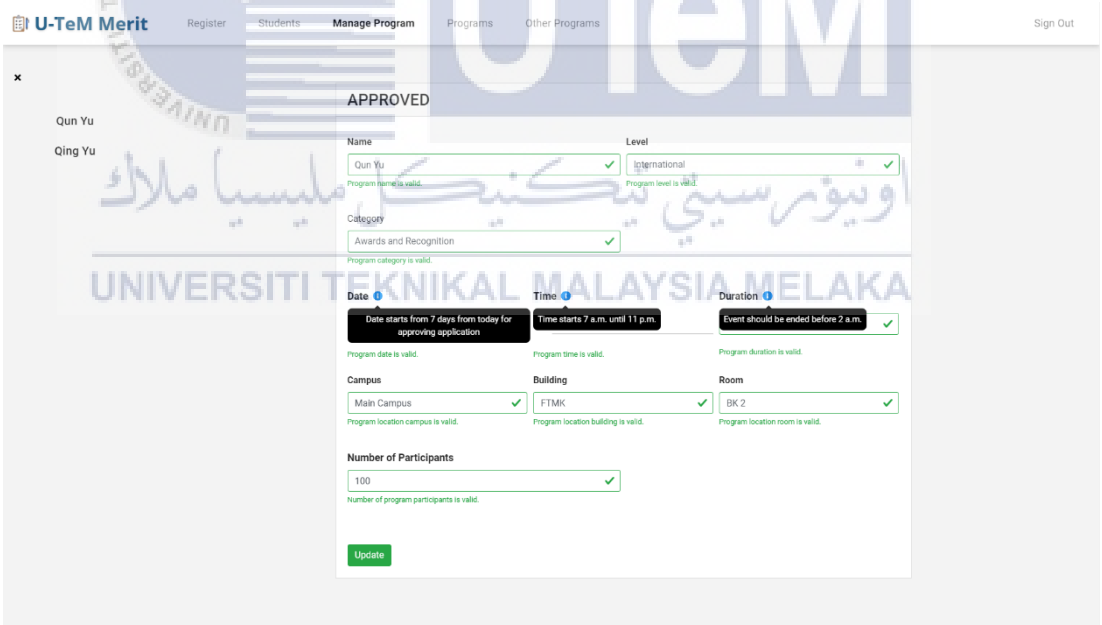



**Figure 4.26: Update Program for Admin**

Management team can register new program. The system hides the remark area because it is unnecessary for management team. Management team can also update program.

*(m) Manage Programs Created by Other Organizer*

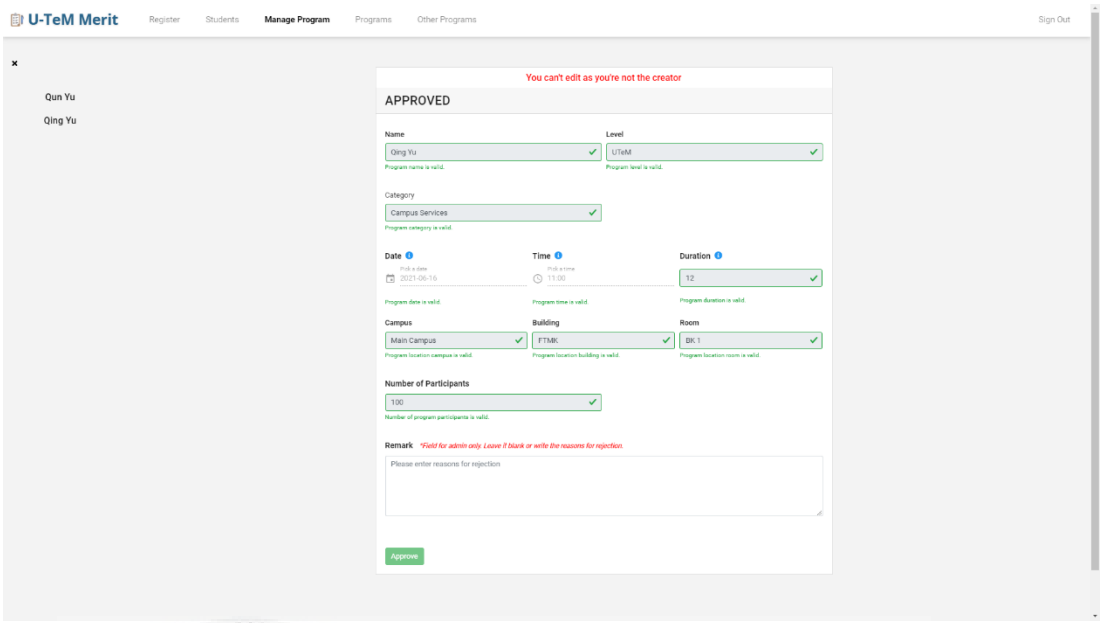



**Figure 4.27: Manage Other's Program**

Management team can update other's program by left the message at remark area. The system will inform the owner of program by sending email if any update from the management team.

*(n) Manage Attendance*

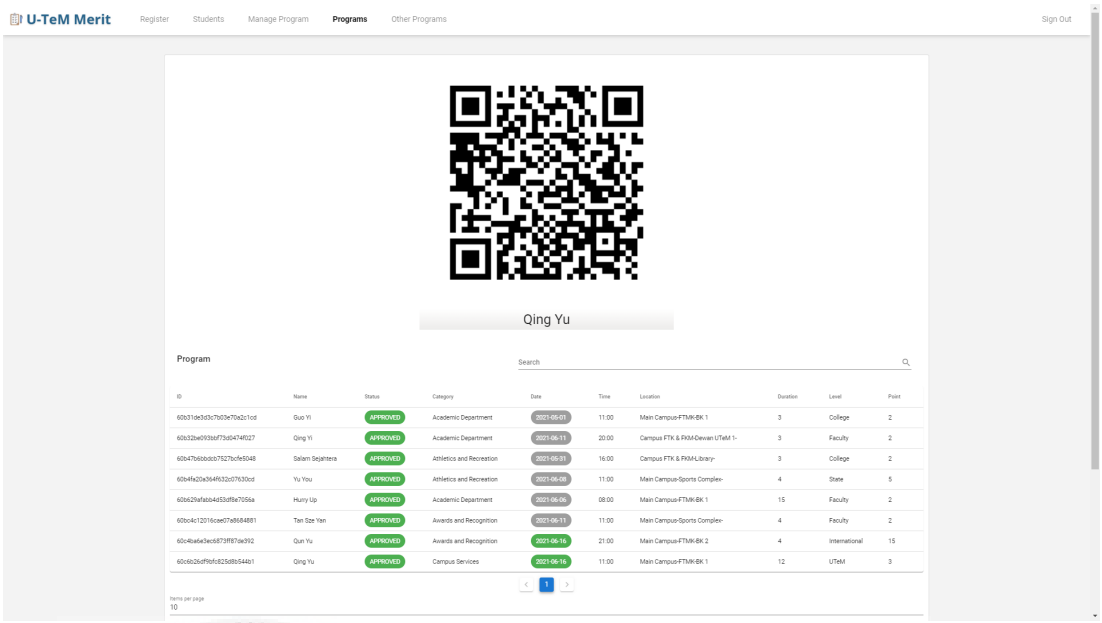



**Figure 4.28: View Program with QR Code**


**Figure 4.29: Manage Attendance**

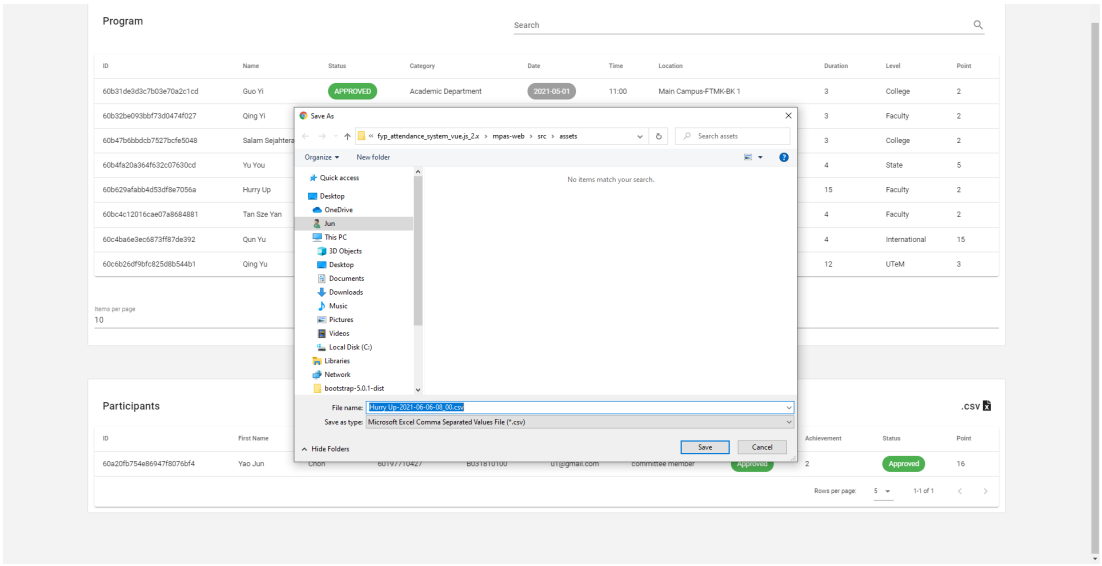



**Figure 4.30: Generate Report**

Management team has similar access with the organizer for this part which is explained at the 4.2.3.3(b) part. The difference between them is the management team is able to view all the programs but the organizer only able to view the programs created by their own.

*(o) Manage Outside Program*

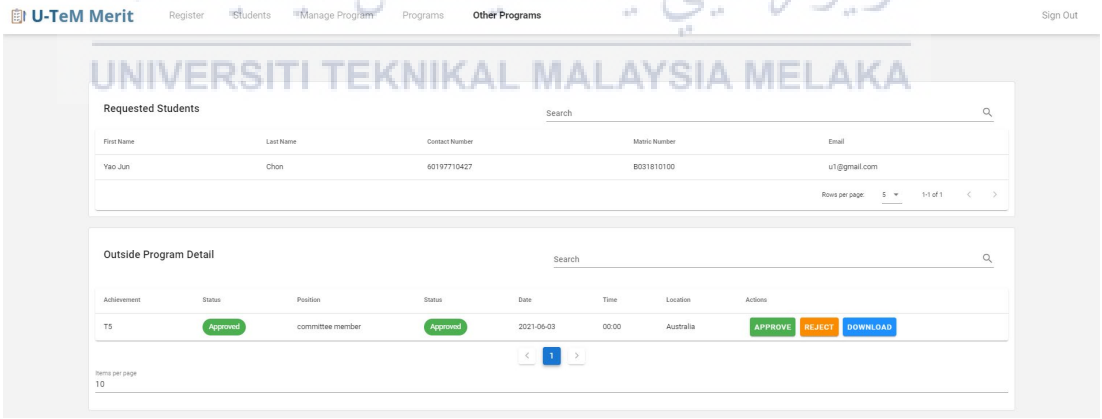



**Figure 4.31: Manage Outside Programs**

Management team will able to view all the request by the students for the attendance of outside programs. Management team can approve and reject the qualification of student and also download the certificate of the requested program.

The table of outside program detail in figure 4.31 will provide two columns to show the status of qualification by the grey and green colors.

## 4.3 Database Design

Database design is the organization and management of data according to the database model. The data is designed and determined to be stored into database and the interrelation between data elements. Entity Relationship Diagram (ERD) with UML notation is used to visualize the data models of system. The data models is non-relational as the database used is MongoDB which is NoSQL database. The ERD is just only the visualization of the data models for the documentation purpose.

### 4.3.1 Conceptual and Logical Database Design

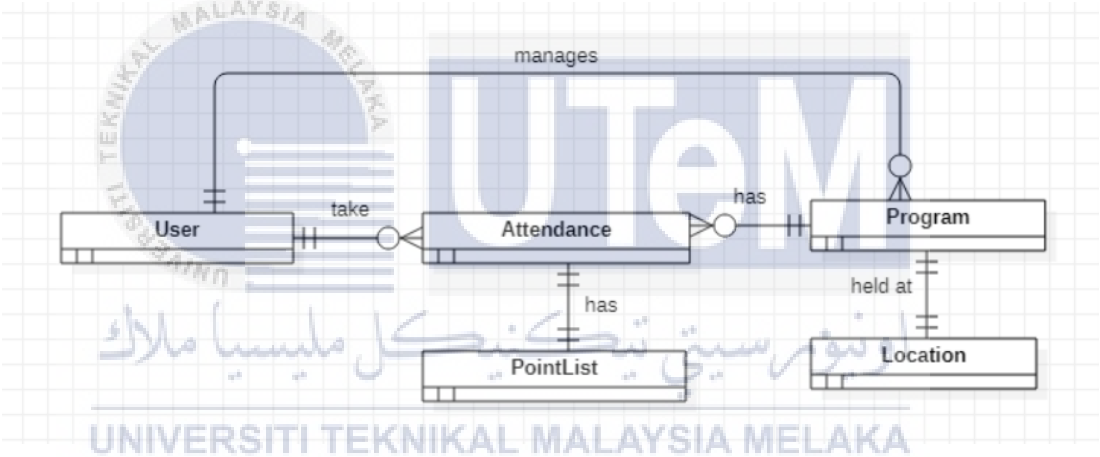





**Figure 4.32: Conceptual and Logical Database Design**

### 4.3.2 Business Rule:

1. User may participate zero or many programs.
2. User may manage or participates zero or many programs.
3. Program can be managed or participated by zero or many users.
4. Program may be held at a location.
5. Attendance may have one point information.

The data dictionary has already been discussed in system analysis (3.3.1 Data Requirement).

## 4.4 Detailed Design

### 4.4.1 Module Integration and Interface Design

Specification and diagrams may be further elaborated. Emphasis should be on the logic of the design and the approach to satisfying the requirements – How will the system function?

Below discuss about the modules which will be used in system U-Tem Merit. There will three sub chapter according to the roles of the system which are student, organizer and management team.

#### 4.4.1.1 Student Modules

*(a) Authentication*

In this module, students are required to enter the email and password which is assumed in the database of UTeM. The website will validate the student's input to check whether it is in correct format. If the input is valid, the website of system will send student's information to the server of system to verify the existent of email and verify the password whether it is matched with email. If all is matched, the server of system will assign a token to the student and it will be stored in the local storage of website in browser.



*(b) Participated Program*

In this module, student can view all the participated programs which are processed at the server of system. Behind the scene, the website of system will send a http request with token assigned to represent the identification of student to the server of system. The server checks the validity of token and get the student information from it. The server uses the identification of student to track the participated programs and send back response to the student. The student able to get some information with detail which are the list of participated programs, including inside and outside programs, the points of each program and the total points achieved from the programs.

*(c) Claim Attendance*

In this module, student can claim the attendance of program which is conducted before. The program type includes inside and outside of UTeM. The server will filter out the programs which meets the conditions and send to students for selection. The student should provide the information of program if the program type is outside of UTeM. Student also should provide certificate of program to prove the participation of program for both type of program.

*(d) Take Attendance*

In this module, server will filter the programs which are going to be conducted on the day for student to take attendance. The website of system only allow student to take the attendance when the program is conducting. If the time of student take attendance is not in the time of program conducting, the attendance is rejected. Student has two ways to take attendance which are selecting the programs listed by system or scanning the QR Code provided by the organizer.

*(e) View All Programs*

In this module, student able to review all the programs which are approved. Behind the scene, the server will receive the http request from student and filter the approved programs which does not matter the programs were conducted or will be conducted in the future. The list of students also will be retrieved according to each conducted program. The role and achievement of student for each program also will be displayed.

#### 4.4.1.2 Organizer Modules

*(a) Authentication*

In this module, organizer is required to enter the email and password which is assumed in the database of UTeM. The website will validate the organizer's input to check whether it is in correct format. If the input is valid, the website of system will send organizer's information to the server of system to verify the existent of email and verify the password whether it is matched with email. If all is matched, the server of

system will assign a token to the student and it will be stored in the local storage of website in browser.

*(b) Manage Own Create Programs*

In this module, organizer able to register program. After registered program, organizer able to get update email from management team if management team has done any update about the organizer's program. Organizer still able to edit the detail of program as long as the program is not conducted yet. The location and date of program will be validated before registering program to avoid any duplication of program at the same time and location.

*(c) Manage Attendance*

In this module, server only retrieve the programs which are finish conducted and created by the organizer. The students who participated the organizer's programs also will be listed in with their role and achievement. Organizer able to manage the attendance of students who participate own program by approving or rejecting their attendance. Download certificate of student service also provided for organizer to check.



#### 4.4.1.3 Management Team

*(a) Authentication*

In this module, management team are required to enter the email and password which is assumed in the database of UTeM. The website will validate the manage team's input to check whether it is in correct format. If the input is valid, the website of system will send organizer's information to the server of system to verify the existent of email and verify the password whether it is matched with email. If all is matched, the server of system will assign a token to the student and it will be stored in the local storage of website in browser.

*(b) Student List*

In this module, management team able to get all the students and the participated programs from the server. The participated programs are including the

inside and outside of UTeM. The total merit point achieved for each student also will be automatically calculated and displayed.

*(d) Manage Programs*

In this module, management team able to register program. After registered program, management team able to approve or reject the program requested and inform the program's organizer if any update by email. Management team also able to create program. The location and date of program will be validated before creating program to avoid any duplication of program at the same time and location.

*(e) Manage All Program Attendances*

In this module, server only retrieve the programs which are finish conducted. The students who participated programs also will be listed in with their role and achievement. Management team able to manage the attendance of students by approving or rejecting their attendance. Download certificate of student service also provided for management team to check.

*(f) Manage Outside UTeM Attendances*

In this module, server only retrieve the programs which are outside of UTeM program. Management team able to download the certificate uploaded by students to check.



### 4.4.2 Software Design

Software design is the process by which an agent creates a specification of a software artifact intended to accomplish goals, using a set of primitive. Below are the sequence diagrams representing the main activities and actions for each role in U-Tem Merit:

#### 4.4.2.1 Students

*(a) Authentication*

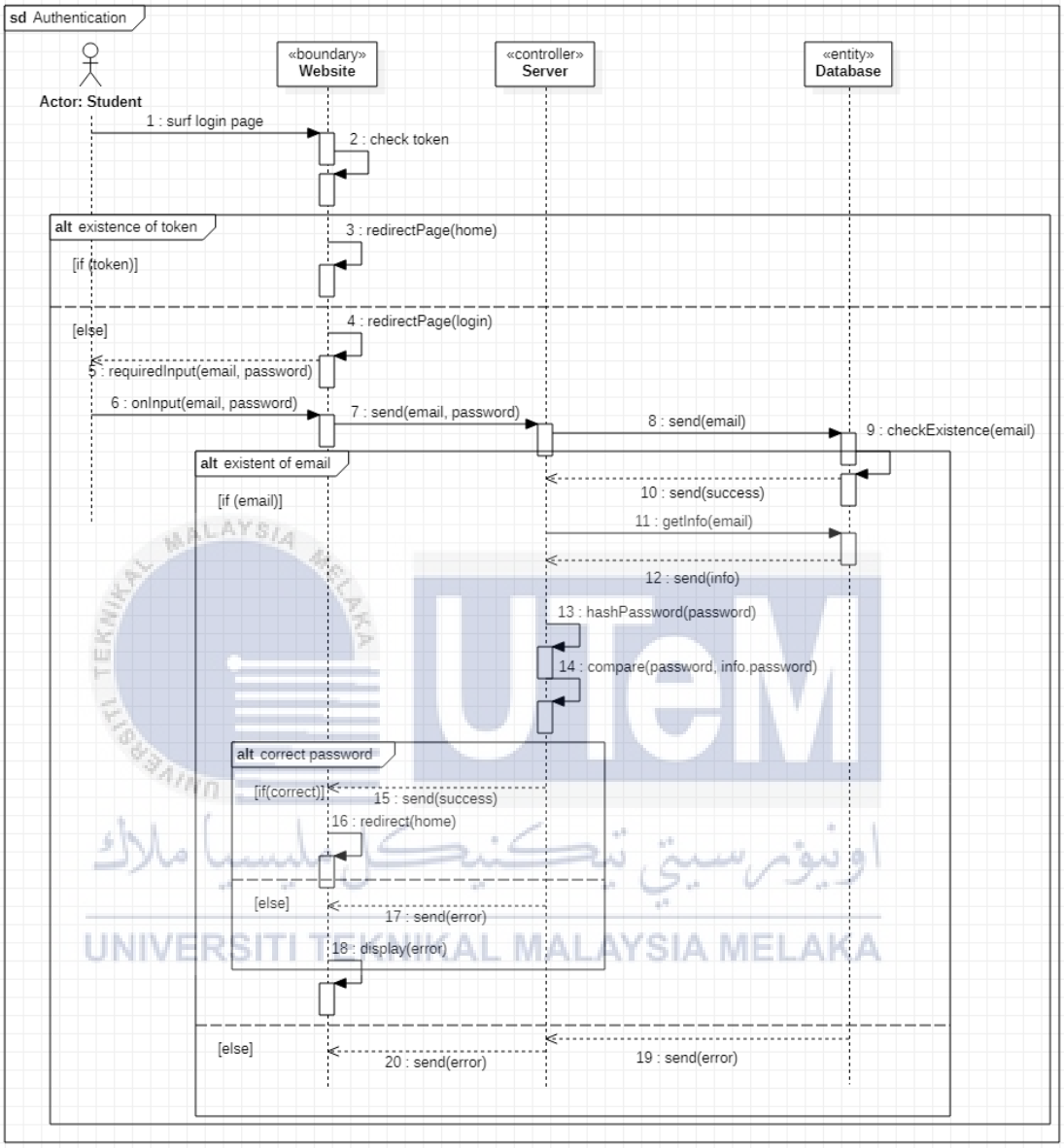



**Figure 4.33: Authentication for Student**

*(b) Participated Program*

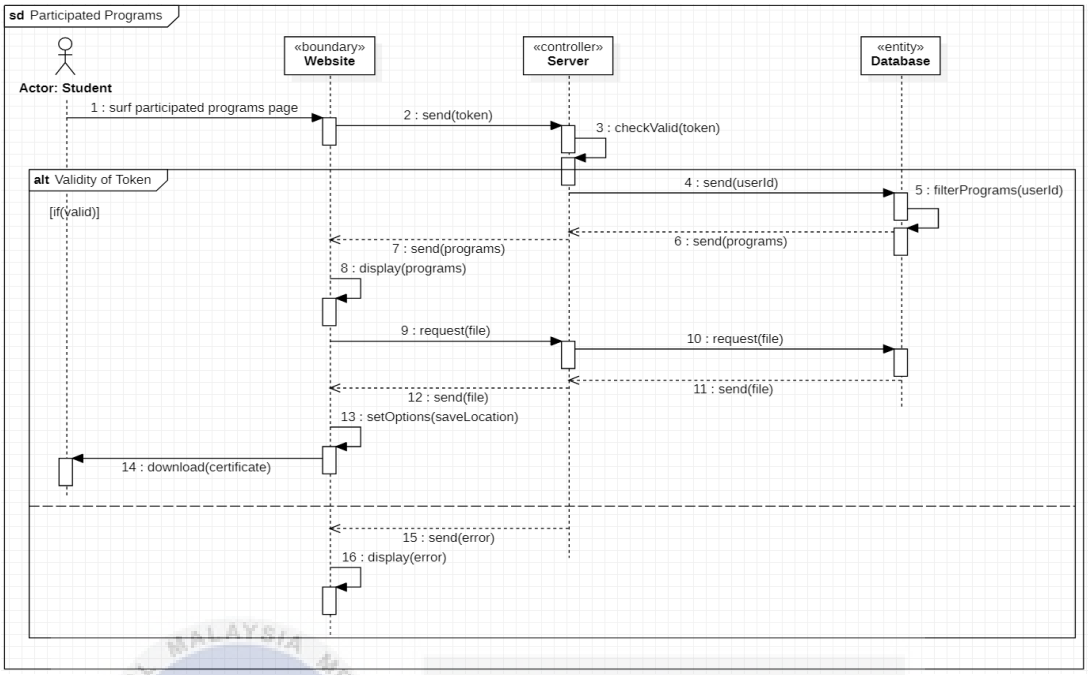



**Figure 4.34: Participated Programs**

*(c) Take Attendance*

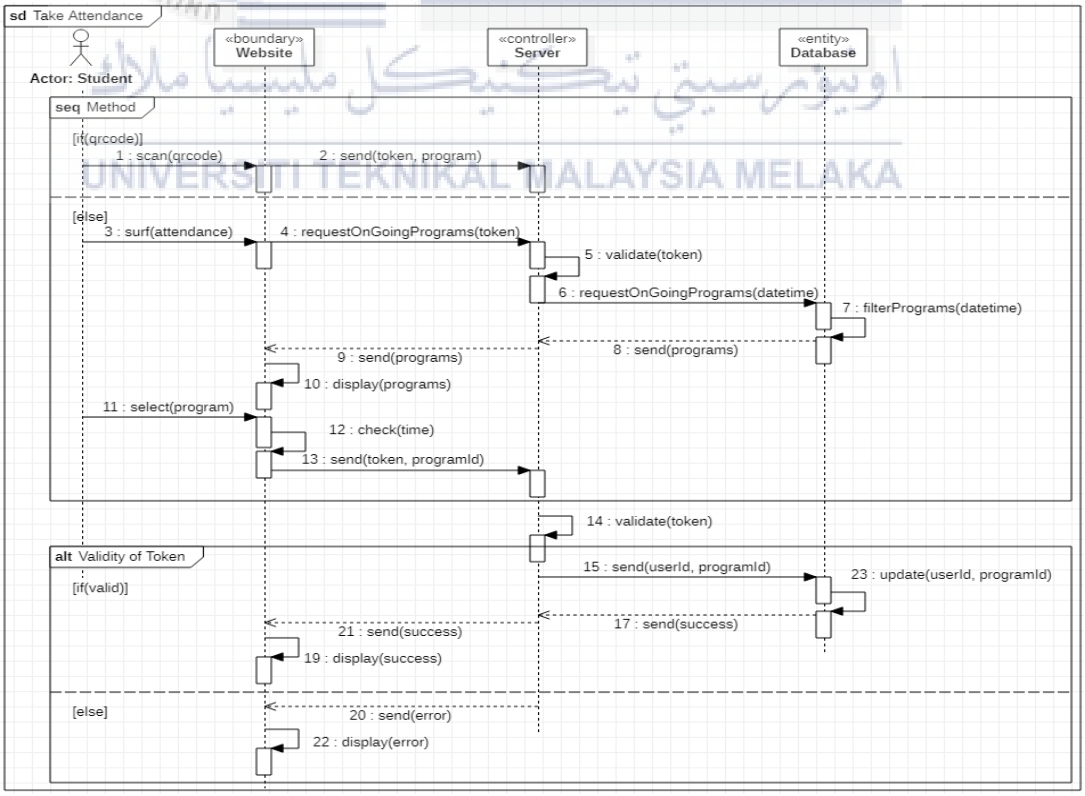



**Figure 4.35: Take Attendance**

*(d) Claim Attendance*

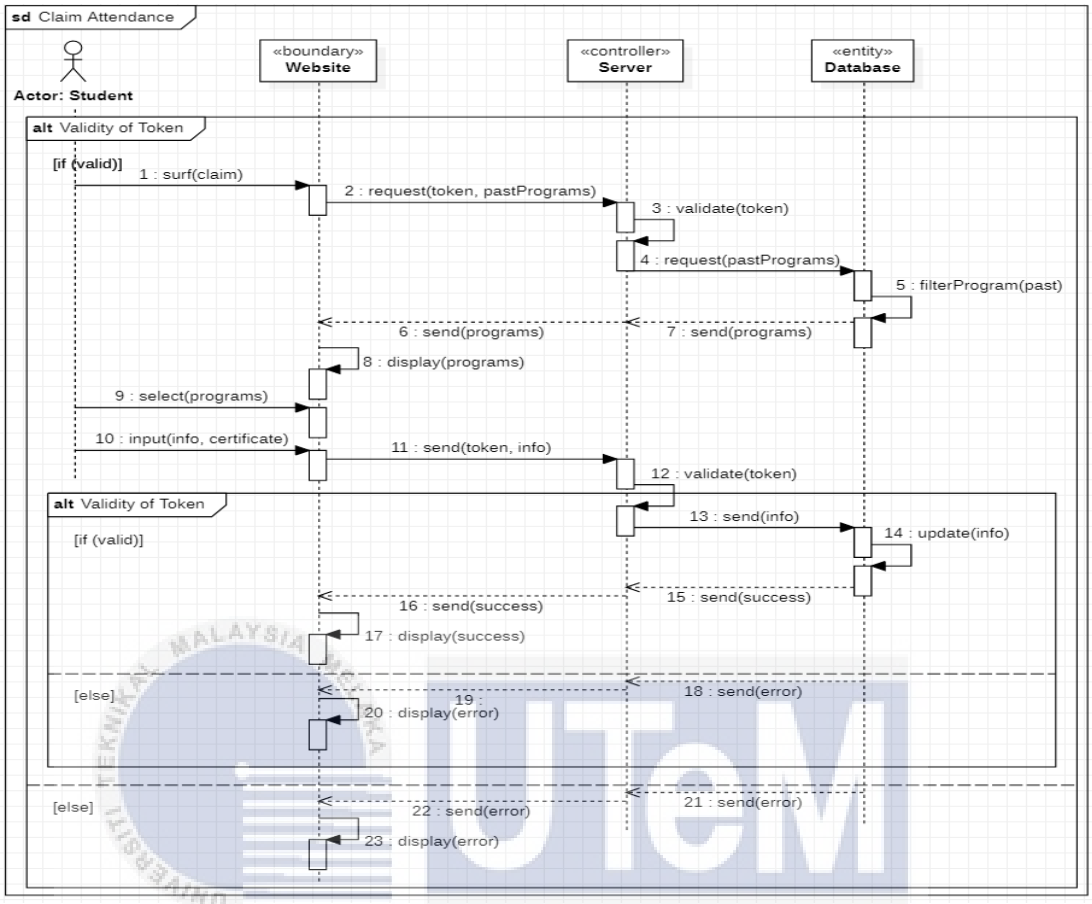



**Figure 4.36: Claim Attendance**



*(e) View All Programs*

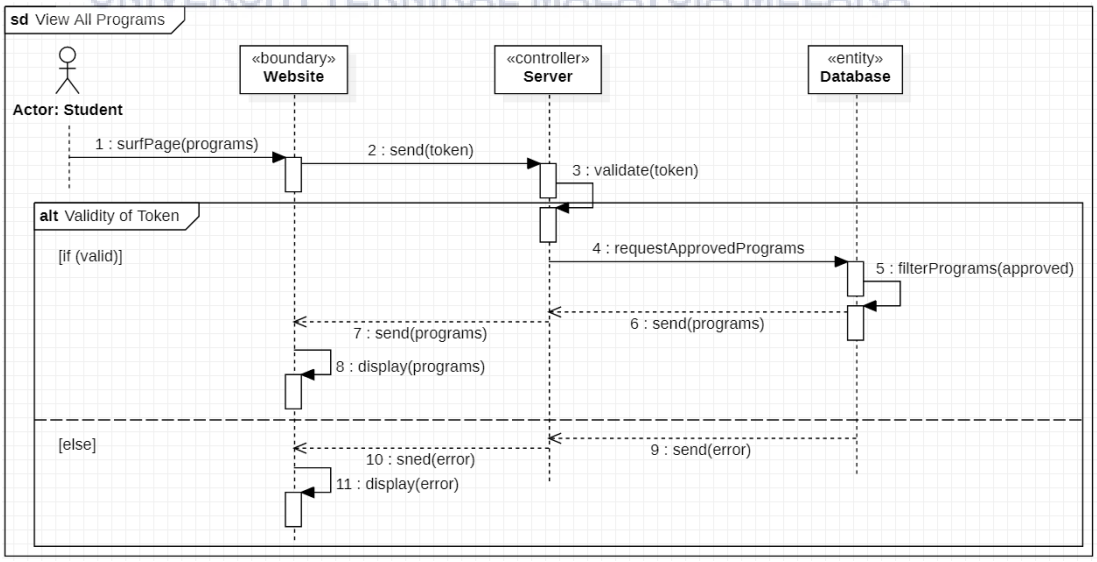



**Figure 4.36: View All Programs**

#### 4.4.2.2 Organizer

*(a) Authentication*

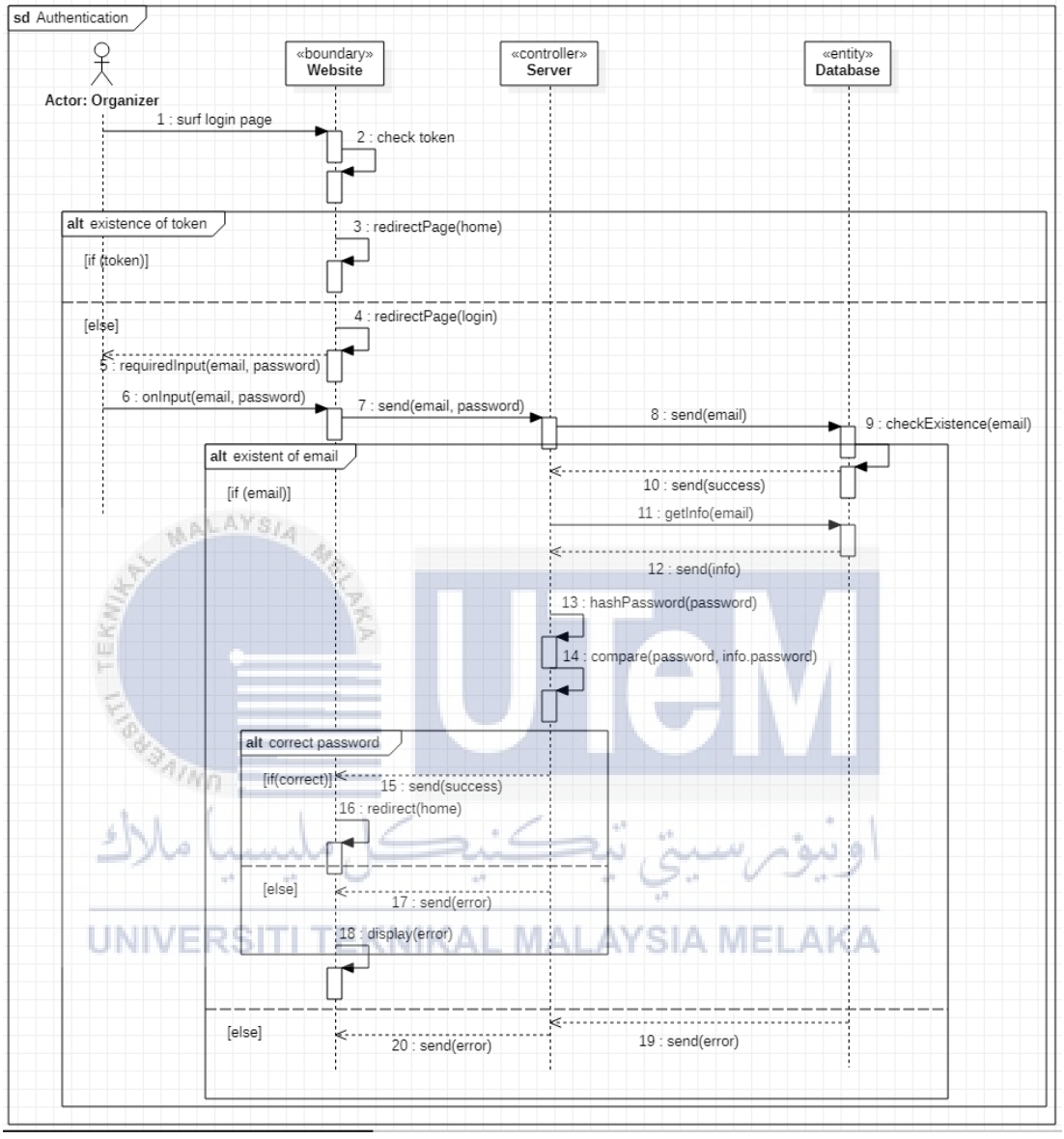



**Figure 4.37: Authentication for Organizer**

*(b) Manage Own Create Programs*

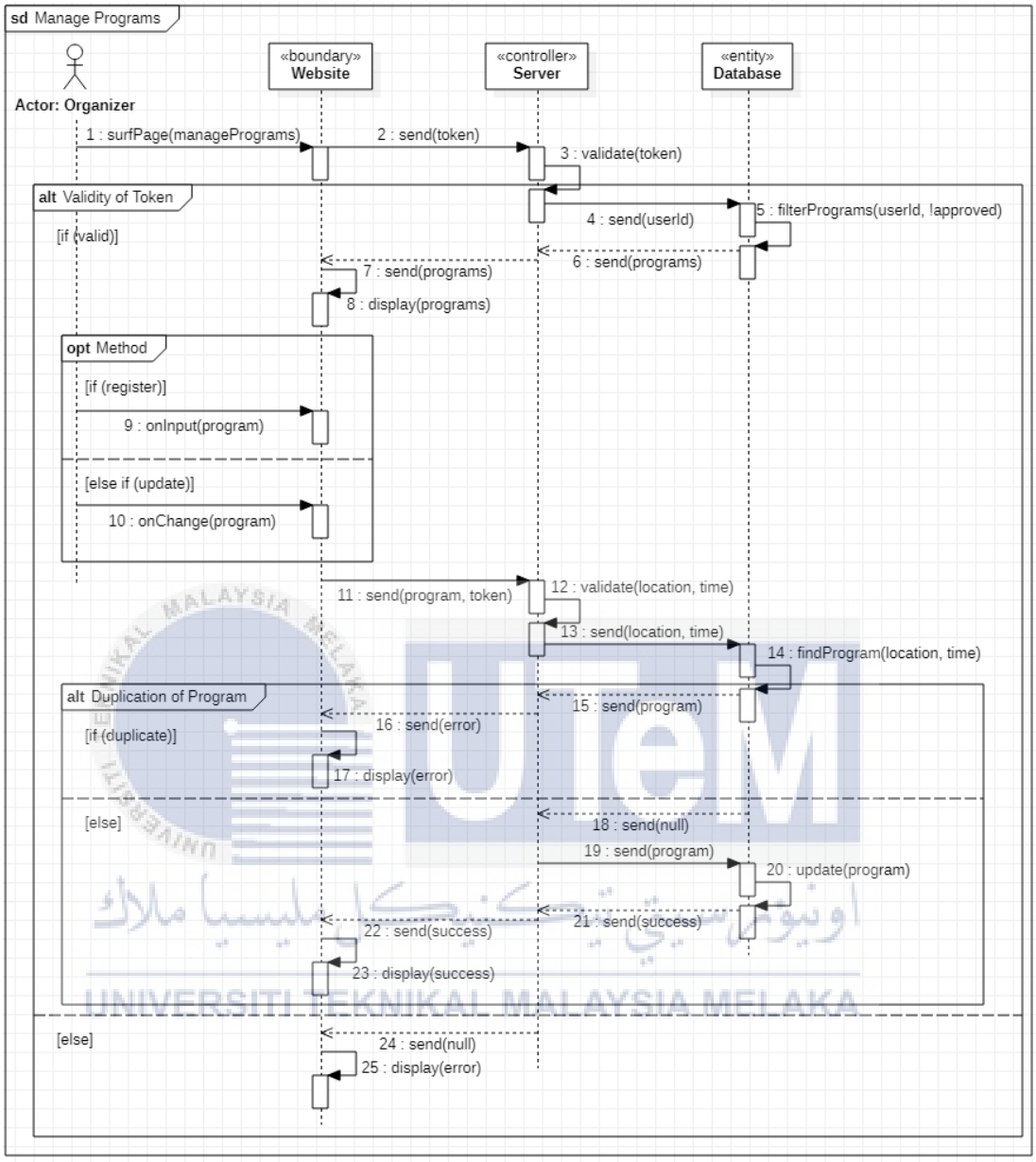



**Figure 4.38: Manage Own Create Program**

*(c) Manage Attendance*

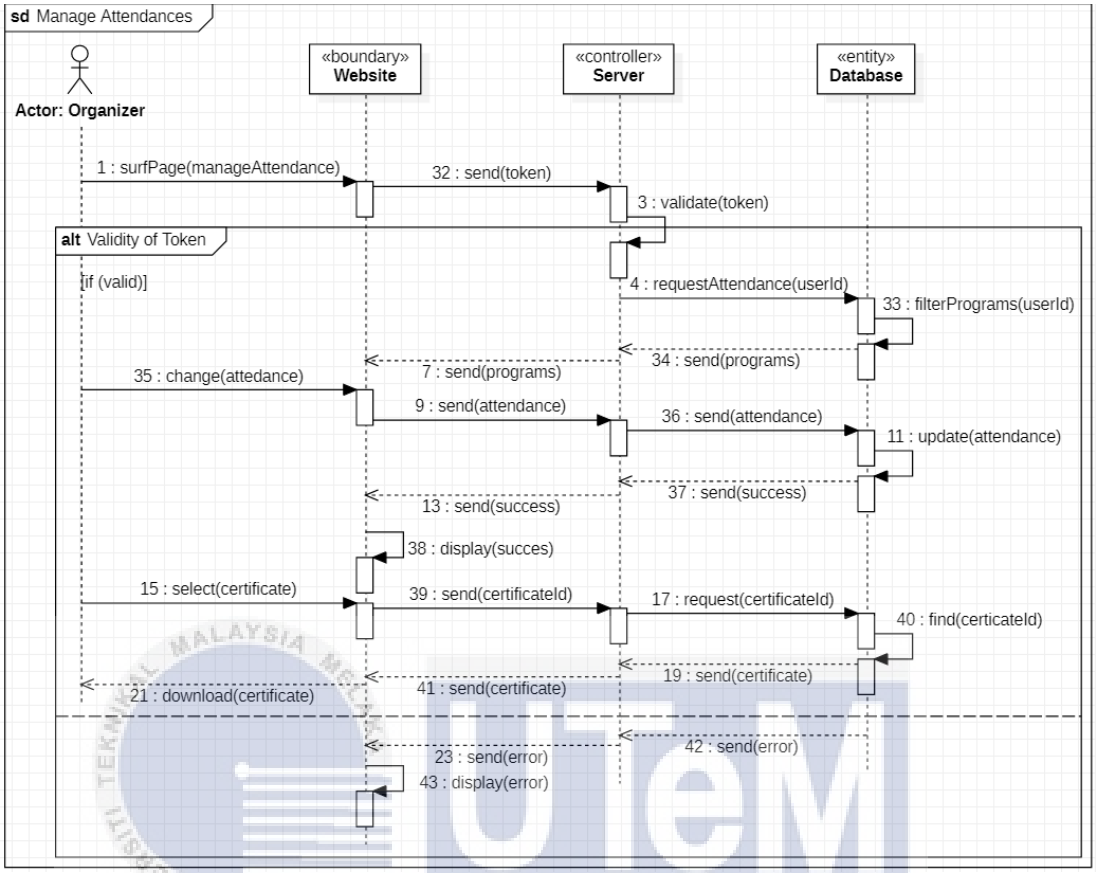



**Figure 4.39: Manage Attedance for Organizer**

#### 4.4.2.3 Management Team

*(a) Authentication*

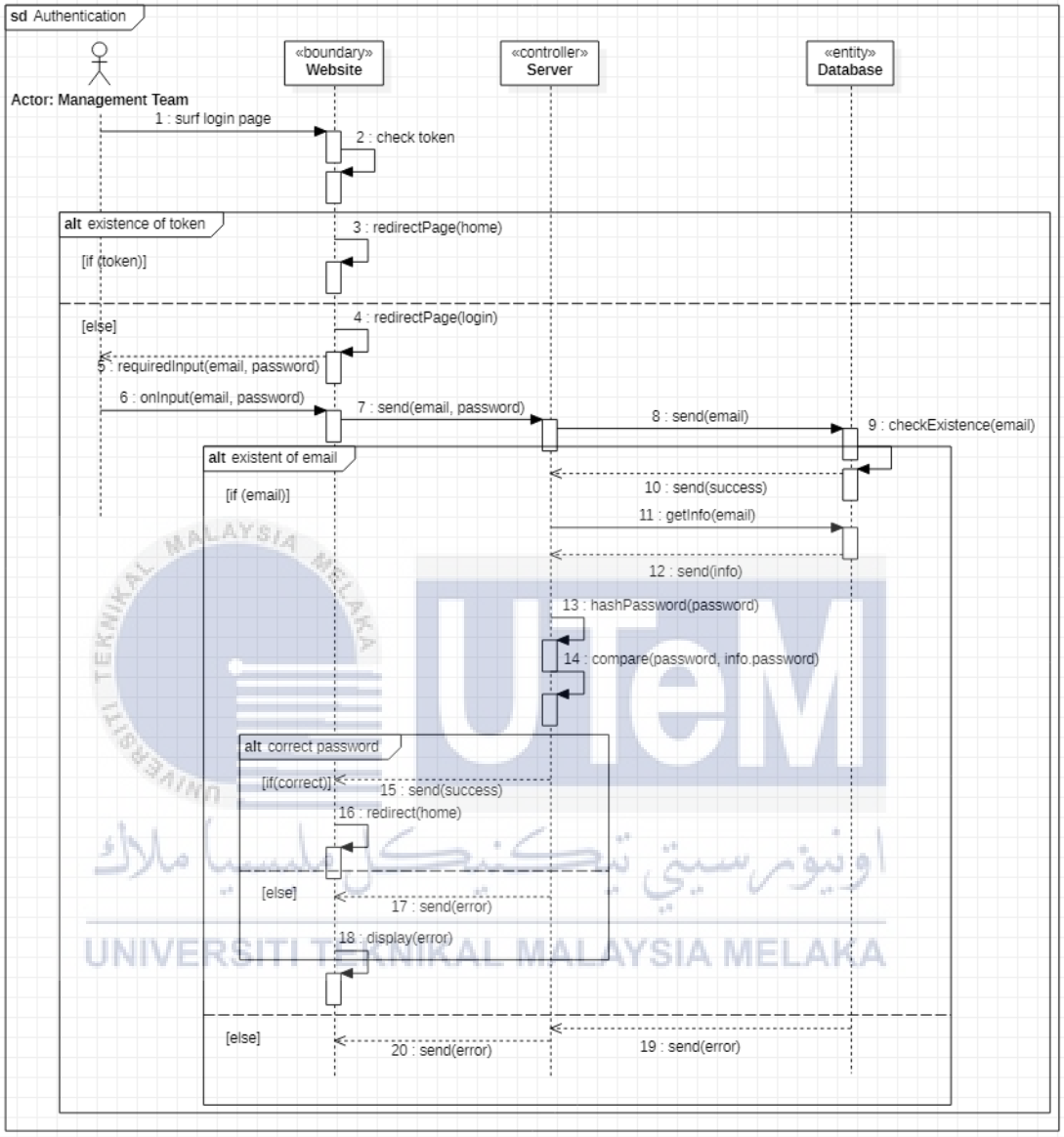



**Figure 4.40: Authentication for Management Team**

*(b) Student List*

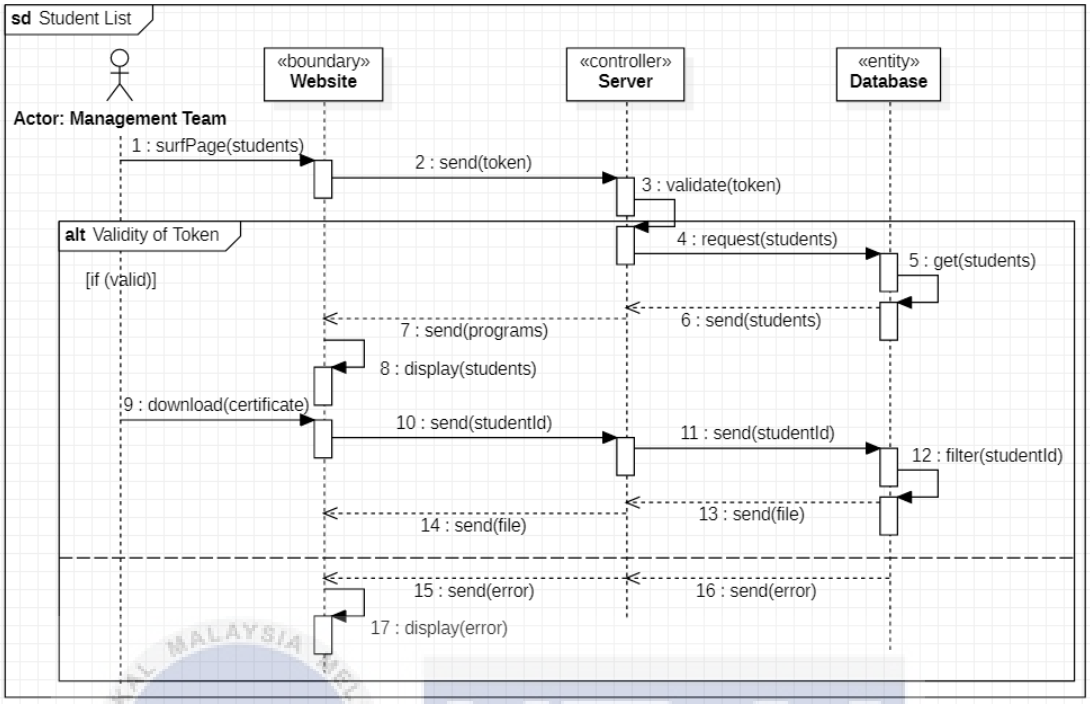



**Figure 4.41: Student List**

*(c) Manage Programs*

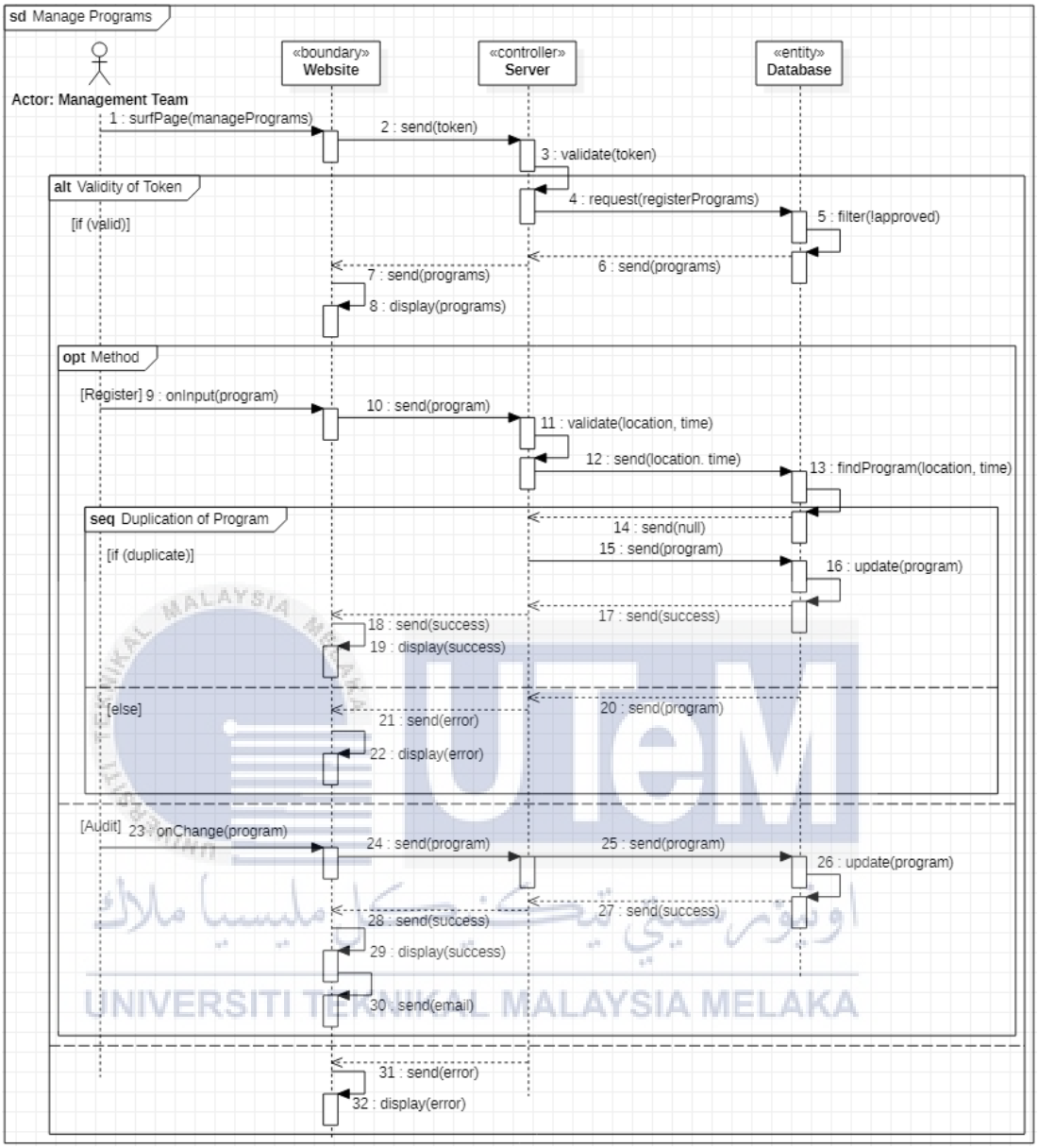



**Figure 4.42: Manage Programs for Management Team**

*(d) Manage All Program Attendances*

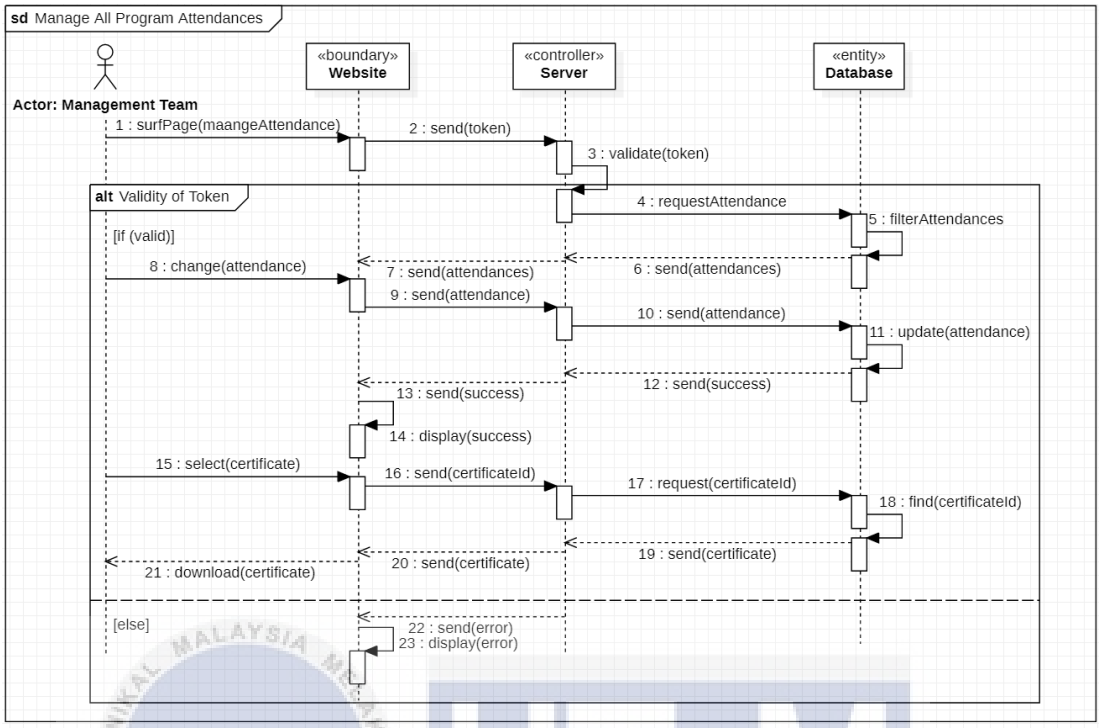



**Figure 4.43: Manage All Program Attendances**

*(e) Manage Outside UTeM Program Attendances*

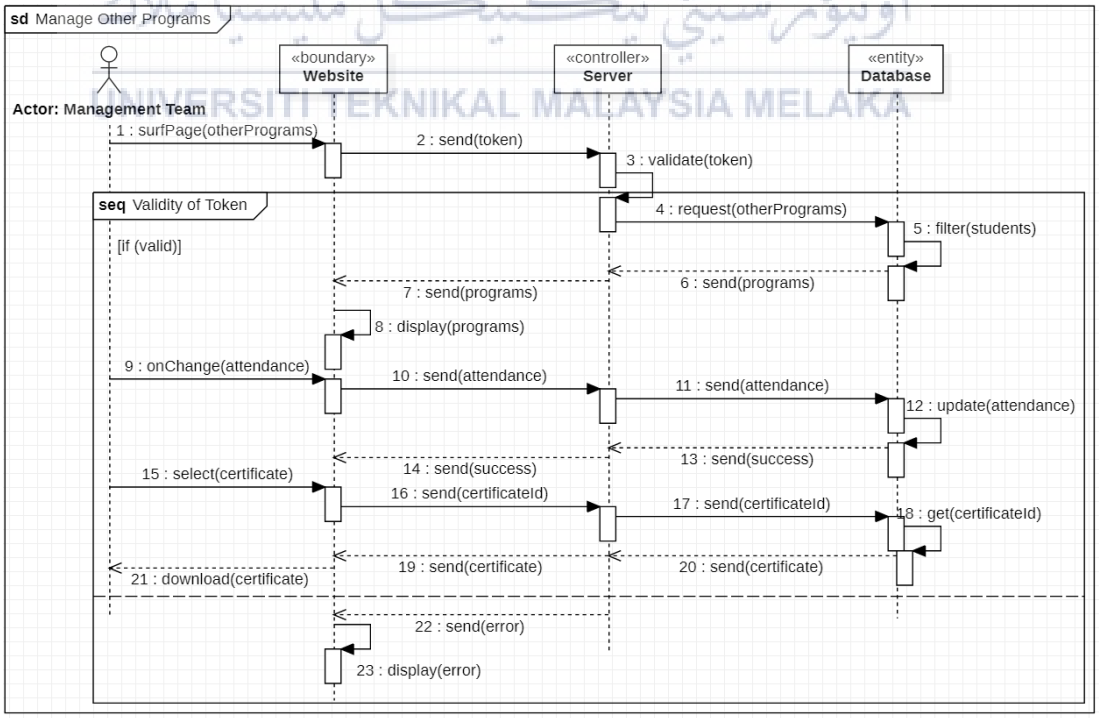



**Figure 4.44: Manage Outside UTeM Program Attendances**

### 4.4.3 Physical Database Design

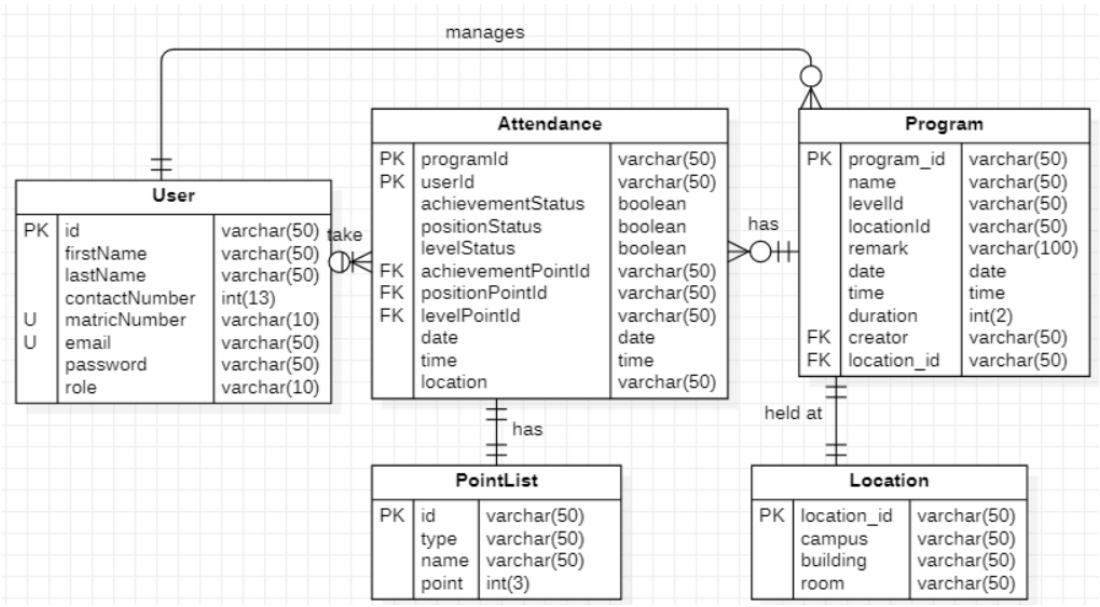



**Figure 4.45: Entity Relational Diagram**

## 4.5 Conclusion

The system will need to be designed as discussed in this chapter to ensure proper integration, as the system will interact with multiple tiers which are Vue.js as front end, Node.js as back end and Cloud MongoDB as database. The user interfaces are also designed to be as simple as possible while still remaining intuitive.

# CHAPTER 5: IMPLEMENTATION

## 5.1 Introduction

System implementation defines how the information system should be configured (i.e., the physical system design), ensures that the information system is managed and run and guarantees an information technology quality standard (i.e., quality assurance). This ensures the new system can be made available to prepared users (the deployment) and the system's continuous support and maintenance within the Performing Organization. The implementation of the system includes executing every step required to educate customers about the use of the new system, bringing into newly developed system, making sure that all the information needed at operational start-up is accessible and reliable, and validating the correct workings of business functioning that interacts with the systems.



## 5.2 Software Development Environment Setup

The development environment is the collection of software or software product processes and programming tools sed to construct the application. Often the word can mean the physical environment as well. An integrated development environment is one in which processes and instruments are organized to provide developers with an orderly interface and a convenient view of the process of development (or at least the processes of writing code, testing it, and packaging it for use).

- Node.js

  Node.js is an open source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code

outside a web browser. It is used as the back-end server to manage the request from the users.

- MongoDB

  MongoDB is source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas which are suitable to use for the web development.

## 5.3 Software Configuration Management

System configuration is a term in systems engineering that defines the computer hardware, the processes as well as the various devices that comprise the entire system and its boundaries. This term also refers to the settings or the hardware and software arrangement and how each device and software or process interact with each other based on a system settings file created automatically by the system or defined by the user.

### 5.3.1 Configuration Environment Setup

The environment configuration is done to ensure software runs correctly and ensuring all needed package are loaded.



**Table 5-1: Environment Setup**

| Visual Studio Code (extension) | Operating System |
|---|---|
| Debugger for Chrome<br>ESLint<br>Vetur<br>Vue | Node JS Platform v14.15.4 |

### 5.3.2 Version Control Procedure

Version control is the process by which different drafts and versions of a document or record are managed. It is a tool which tracks a series of draft document, culminating in a final version. It provides an audit trail for the revision and update of

these finalized versions. The version control is done in GitHub. Projects are committed using either Android Studio or Git, with audit trails auto generated and readily accessible on the website itself. There are 4 separate repositories, each accounting for different applications.

#### 5.3.2.1 Web Application

**Table 5.1: Website**

| Version | Description |
|---|---|
| 1.0 | Initial project structure |
| 1.1 | Finished interfaces design for student |
| 1.2 | Finished interfaces design for organizer |
| 1.3 | Finished interfaces design for management team |
| 2.0 | Finished authentication function for all types of users |
| 2.1 | Finished participated program list function for student |
| 2.2 | Finished claim and take attendance function for student |
| 2.3 | Finished program list function for all types of users |
| 2.4 | Finished student list function for management team |
| 2.5 | Finished manage program function for organizer and management team |
| 2.6 | Finished program list function for organizer and management team |

| | |
|---|---|
| 2.7 | Finished manage outside program function for management team |

#### 5.3.2.2 Web Server

**Table 5.2: Web Server**

| Version | Description |
|---|---|
| 1.0 | Initial project structure |
| 1.1 | Finished application programming interface of the authentication |
| 1.2 | Finished application programming interface of the participated program module for the students |
| 1.3 | Finished application programming interface of the claim attendance module for the students |
| 1.4 | Finished application programming interface of the take attendance module for the students |
| 1.5 | Finished application programming interface of the view all programs module for the students |
| 1.6 | Finished application programming interface of the manage program module for the organizer and management team |

| | |
|---|---|
| 1.7 | Finished application programming interface of the manage attendance module for the organizer and management team |
| 1.8 | Finished application programming interface of the student list module for the management team |
| 1.9 | Finished application programming interface of the manage outside program module for the management team |

## 5.4 Implementation Status

The table below shows the implementation progress of the specified module at the time of writing. Some modules which were scrapped are omitted from the list as it is no longer considered to be added.

**Table 5.3: Module Implementation Progress**

| Platform | User Type | Module | Progress |
|---|---|---|---|
| Web application | All | Authentication | 100% |
| | | Program list | 100% |
| | Student | Participated program | 100% |
| | | Claim attendance | 100% |
| | | Take attendance | 100% |
| | Organizer & Management team | Manage program | 100% |
| | | Inside program list | 100% |
| | Management team | Student list | 100% |
| | | Outside program list | 100% |
| Web server | All | Authentication | 100% |
| | | Mange program | 100% |
| | | Manage student | 100% |

| | | | |
|---|---|---|---|
| | | Manage attendance | 100% |
| | - | Schema for database | 100% |

## 5.5 Conclusion

The implementation phase is the building phase of the system, something which is crucial to be kept track of. This chapter presents the environment and the configuration required to build the system to ensure smooth development.

# CHAPTER 6: TESTING

## 6.1 Introduction

System testing is a method to check whether the actual system product matches expected requirements and to ensure that system product is defect free. There are various type of testing such as alpha testing, acceptance testing, back-end testing, black box testing and etc. The selected testing type is Black Box Testing to test the system. This type of testing is carried out based on the requirements and functionality. The term 'Behavioral Testing' is also used for Black Box Testing. Black Box Testing can be applied to three main types of tests which are non-functional, functional and regression testing and, in this chapter, the functional testing is mainly focused. Testers are needed to perform the testing and the testers are required to follow the test cases created from the requirement specification document.



## 6.2 Test Plan

The test will mainly focus on the detection and counting part of the project, with secondary focus on the interaction between the modules and roles. The input of the test will be recorded for each testing. The steps for the testing will be provided for the testers to ensure the testers know how to carry out the testing for the particulate test case.

### 6.2.1 Test Organization

Test organization describes the personnel that will perform the testing. In this case, the organization is only made up of a two person, one of the testers is myself and another is my friend, Koh Kok Sheng. Mr. Koh is in charged for the integration testing

after all unit testing is done by myself to ensure the system at least free of the bugs at the first stage of testing.

### 6.2.2 Test Environment

**Table 6.1: Test Environment**

| Component | Requirement |
|---|---|
| Hardware | Laptop<br>Mobile devices |
| Operating system | Node.js installed<br>Internet connection |

### 6.2.3 Test Schedule

Test schedule describes about the schedule for testing is planned. With cycle describing how many times the test is run.

**Table 6.2: Test Schedule**

| Platform | Module | Duration (minutes) | Test Cycle |
|---|---|---|---|
| Website | Authentication | 5 | 3 |
| | Participate programs for student | 5 | 3 |
| | Claim attendance for students | 5 | 3 |
| | Take attendance for students | 5 | 3 |
| | Manage programs for organizer | 5 | 3 |
| | Manage attendance for organizer | 5 | 3 |
| | Manage programs for management team | 5 | 3 |
| | Manage attendance for management team | 5 | 3 |
| | Manage outside program's attendance for management team | 5 | 3 |
| | Student list | 5 | 3 |

## 6.3 Test Strategy

The test strategy that was used is the black box testing and bottom-up approach. Black box testing is a method where the internal structure of the module is not known to the tester or is not actively utilized in testing, thus making it easier to discover unexpected behavior, rather than following the pre-determined flow. Meanwhile, bottom-up approach is an approach to integrated testing where lower-level components are tested first and then used to facilitate testing of higher-level components.

### 6.3.1 Classes of tests

This section will describe the classes of test that will be applied to the testing phase. Below are the classes and its description.

#### 6.3.1.1 Unit Testing

Unit testing is the stage in software testing in which individual software functions testing. It is carried out by testing each function in the system. The unit testing will be categorized by the roles of user. In another word, each role represents each category. Each role will have some modules and each module will be subdivided into some unit testing. The detail of unit testing will be described at the next section test design below.



#### 6.3.1.2 Integration Testing

Integration testing normally is the next stage of software testing after unit testing which is the combination of unit testing to test the interaction between the target units.

## 6.4 Test Design

This section describes the test case identification, test cases and expected result for each module are designed and documented. The test cases are carried with some data which are documented as well. The test results also documented.

### 6.4.1 Unit Testing

#### 6.4.1.1 Student

**Table 6.3: Test Case for Student**

| Test Case ID | Test Case | Procedure | Expected Result | Test Data |
|---|---|---|---|---|
| u_s_01_01 | Login to system | 1. Login to system with the email and password provided | The system redirects the user to the home page with the information of user. | Email: u1@gmail.com Password: 12345678 |
| u_s_02_01 | Display all the participated programs by the user | 1. Select the "Participated Program" tab in the navigation bar | The system redirects the user to the participated program page with the information of all the programs participated by the user. | - |
| u_s_03_01 | Display the status of each participated program by the user | 1. Observe the status of the participated program in both inside and outside program | The approved indicator for the status is green while not approved | - |

| | | | | |
|---|---|---|---|---|
| | | | indicator for the status is grey. | |
| u_s_04_01 | Display the claim attendance page | 1. Select the "Claim" tab in the navigation bar | The system redirects the user to the claim page with a form which is for the user to apply to claim the conducted program's attendance. | - |
| u_s_04_02 | Claim conducted UTeM program's attendance | 1. Select a program from the list<br>Select the position and achievement from the list<br>Upload the program certificate | The system displays a successful message to the user. | Program: Yao<br>Position: Committee member<br>Achievement: 1<br>Certificate: Resume_2.pdf |
| u_s_04_02 | Claim conducted UTeM program's attendance | 1. Select a program from the list<br>2. Fill in the date, time and location<br>3. Select the level, position and achievement from the list<br>4. Upload the program certificate | The system displays a successful message to the user. | Program: Yao<br>Date: 2021-08-01<br>Time: 11:00<br>Location: 103, Jalan Bukit Gemilang 6, 81000 Kulai, Johor, Malaysia<br>Level: State |

| | | | | |
|---|---|---|---|---|
| | | | | Position: Committee member<br>Achievement: 1<br>Certificate: Resume_2.pdf |
| u_s_05_01 | Display the take attendance page | 1. Select the "Attendance" tab in the navigation bar | The system redirects the user to the take attendance page with a form which is for the user to apply to take currently conducting program's attendance. | - |
| u_s_05_02 | Take attendance by manually select program | 1. Select the conducting program from the list<br>2. Click "Take" button to the attendance | The system displays a successful message to the user. | Program: Yao |
| u_s_05_03 | Take attendance by scan QR code of the program | 1. Scan the QR code of the program | The system displays a successful message to the user. |  |

| | | | | |
|---|---|---|---|---|
| u_s_06_01 | Display the program list page | 1. Select the "Programs" tab in the navigation bar | The system redirects the user to the program list page with all the UTeM programs. | - |
| u_s_06_02 | Display the participants of the selected program from the program list | 1. Select a program from the program list | The system displays all the participants of the selected program. | Program: Junny |

#### 6.4.1.2 Organizer

**Table 6.4: Test Case for Organizer**

| Test Case ID | Test Case | Procedure | Expected Result | Test Data |
|---|---|---|---|---|
| u_o_01_01 | Display creates program page | 1. Select the "Manage Program" tab in the navigation bar | The system displays a form of the application of program and a list of programs created and not | - |

| | | | | |
|---|---|---|---|---|
| | | | conducted yet by the user at the left if any. | |
| u_o_01_02 | Register program | 1. Fill in the required information<br>2. Click “Register” button | The system displays a successful message to the user. The program status is changed to be pending instead of register | Name: Oracle Campaign<br>Level: UTeM<br>Category: University Governance<br>Date: 2021-08-20<br>Time: 11:00<br>Duration: 15<br>Campus: Main Campus<br>Building: FTMK<br>Room:BK 1<br>Number of Participants: 100 |
| u_o_01_03 | Update program | 1. Click a program from the program | The system displays a successful | Duration : 14 |

| | | | | |
|---|---|---|---|---|
| | | list at the left to edit.<br><br>2. Edit the information of program<br><br>3. Click “Update” button | message to the user. | |
| u_o_02_01 | View programs | 1. Select the “Programs” tab in the navigation bar | The system redirects the user to programs list page with all the programs created by the user. | - |
| u_o_02_02 | Generate QR code of the program | 1. Select a program with a date is same with the current date from the program list | The system generates a QR code | Program: Junny |
| u_o_02_03 | Display the participants of the selected program from the program list | 1. Select a program from the program list | The system displays all the participants of the selected program. | Program: Junny |

| | | | | |
|---|---|---|---|---|
| u_o_02_04 | Approve the qualification of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click “Approve” button | The system displays a successful message to the user. | Program: Guo Yi<br>Matric Number: B031810100 |
| u_o_02_05 | Reject the qualification of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click “Reject” button | The system displays a successful message to the user. | Program: Guo Yi<br>Matric Number: B031810100 |
| u_o_02_06 | Download the certificate of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click “Download” button | The system displays a dialog for the user to select the location of the file to download. | Salam Sejahtera-B031810100.pdf |



#### 6.4.1.3 Management Team

**Table 6.5: Test Case for Management Team**

| Test Case ID | Test Case | Procedure | Expected Result | Test Data |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| u_m_01_01 | Display creates program page | 1. Select the "Manage Program" tab in the navigation bar | The system displays a form of the application of program and a list of programs created and not conducted yet at the left if any. | - |
| u_m_01_02 | Register program | 1. Fill in the required information<br>2. Click "Register" button | The system displays a successful message to the user. The program status is changed to be approved instead of register. | Name: Oracle Campaign<br>Level: UTeM<br>Category: University Governance<br>Date: 2021-08-20<br>Time: 11:00<br>Duration: 15<br>Campus: Main Campus<br>Building: FTMK<br>Room:BK 1<br>Number of Participants: 100 |
| u_m_01_03 | Update program | 1. Click a program from the program list at the left to edit. | The system displays a successful | Duration : 14 |

| | | | | |
|---|---|---|---|---|
| | | 2. Edit the information of program<br>3. Click “Update” button | message to the user. | |
| u_m_01_04 | Reject program | 1. Select a program which is not created by the user<br>2. Fill in the remark field with the reason of rejection<br>3. Click “Reject” button | The system displays a successful message to user. A email will be sent to the creator of the program. | Remark: The participants are too many |
| u_m_01_05 | Approve program | 1. Select a program which is not created by the user<br>2. Click “Approve” button | The system displays a successful message to user. A email will be sent to the creator of the program. | - |
| u_m_02_01 | View programs | 1. Select the “Programs” tab in the navigation bar | The system redirects the user to programs list page with all the programs. | - |

| | | | | |
|---|---|---|---|---|
| u_m_02_02 | Generate QR code of the program | 1. Select a program with a date is same with the current date from the program list | The system generates a QR code | Program: Junny |
| u_m_02_03 | Display the participants of the selected program from the program list | 1. Select a program from the program list | The system displays all the participants of the selected program. | Program: Junny |
| u_m_02_04 | Approve the qualification of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click “Approve” button | The system displays a successful message to the user. | Program: Guo Yi<br>Matric Number: B031810100 |
| u_m_02_05 | Reject the qualification of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click “Reject” button | The system displays a successful message to the user. | Program: Guo Yi<br>Matric Number: B031810100 |
| u_m_02_06 | Download the certificate | 1. Select a participant from the participant list after a | The system displays a dialog for the user to | Salam Sejahtera-B031810100.pdf |

| | | | | |
|---|---|---|---|---|
| | of the participant | program was selected<br>2. Click "Download" button | select the location of the file to download. | |
| u_m_03_01 | View outside programs | 1. Select the "Programs" tab in the navigation bar | The system redirects the user to programs list page with all the programs. | - |
| u_m_03_02 | Approve the qualification of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click "Approve" button | The system displays a successful message to the user. | Program: Guo Yi<br>Matric Number: B031810100 |
| u_m_03_03 | Reject the qualification of the participant | 1. Select a participant from the participant list after a program was selected<br>2. Click "Reject" button | The system displays a successful message to the user. | Matric Number: B031810100<br>Achievement: T5<br>Position: committee member<br>Level: international |
| u_m_04_01 | Display student list | 1. Select the "Students" tab in | The system redirects the user to | - |

| | | | | |
|---|---|---|---|---|
| | | the navigation bar | the page with the two tables which are students table and participated programs table for the selected student. | |

### 6.4.2 Integration Testing

| Test Case ID | Test Case | Procedure | Expected Result | Test Data |
|---|---|---|---|---|
| i_01_01 | Display correct total marks of participants | 1. Login to system with admin account<br>2. Select the "Programs" tab in the navigation bar<br>3. Select a program from the program table<br>4. Select a student from the student | The manually calculated marks will be same with the shown marks in the student table. | email: a1@gmail.com<br>password: 12345678<br>matric number of the student: B031810100<br>program name: OTA |

| | | | | |
|---|---|---|---|---|
| | | table and remember the matric number of then student<br>5. Approve or reject the qualification of the participant of the student<br>6. Select the "Students" from the navigation bar<br>7. Select the selected student from the student table<br>8. Calculate the total mark of all the program by sum up the marks of program, position and achievement<br>9. Compare the | | |

| | | | | |
|---|---|---|---|---|
| | | calculated total mark and the total mark shown at the student table<br>10. Repeat the steps from step two until step nine but replace the step two until four with the step 11 until 13<br>11. Select the "Other Programs" tab in the navigation bar<br>12. Select a student from the student table and remember the matric number of the student<br>13. Select a program from the | | |

| | | | | |
|---|---|---|---|---|
| | | program table | | |
| i_02_01 | Student claims inside attendance and attendance recorded and displayed in "Programs" page | 1. Login to system with student account<br>2. Select the "Claim" tab in the navigation bar<br>3. Select a program from the program selection except the "Other" option and remember the selected program detail and provide some the information required<br>4. Logout and login to system with management team account. | The attendance in the "Programs" page is matched with the application at the "Claim" page | student email: b031810111@gmail.com<br>student password: 12345678<br>management team email: b031810104@gmail.com<br>management team password: 12345678<br>program: OTC<br>position: committee member<br>achievement: none<br>certificate name: OTC_B031810111.pdf |

| | | | | |
|---|---|---|---|---|
| | | 5. Select the "Programs" tab in the navigation bar<br>6. Find and select the program just now<br>7. Check the attendances information with the application done at step three | | |
| i_02_02 | Student claims inside attendance and the admin account displays the application at "Other Programs" page | 1. Login to system with student account<br>2. Select the "Claim" tab in the navigation bar<br>3. Select "Other Program" option and provide some the information required | The attendance in the "Programs" page is matched with the application by the student account | email: u1@gmail.com<br>password: 12345678<br>program: Olympic<br>position: committee member<br>achievement: none<br>location: London<br>level: international<br>certificate name: Olympic_B031810100.pdf |

| | | | | |
|---|---|---|---|---|
| | | 4. Logout and login to system with admin account<br>5. Select the "Other Programs" tab in the navigation bar<br>6. Find and select the student just now<br>7. Check the attendances information with the application done at step three | | |
| i_03_01 | Student take inside attendance and attendance recorded and displayed in "Programs" page | 1. Login to system with student account<br>2. Select the "Attendance" tab in the navigation bar<br>3. Select a program | The attendance in the "Programs" page is matched with the application at "Attendance" page | email: u1@gmail.com<br>password: 12345678<br>program: OTC |

| | | | | |
|---|---|---|---|---|
| | | from the program selection and remember the selected program detail<br>4. Select the "Programs" tab in the navigation bar<br>5. Find and select the program just now<br>6. Check the attendances information with the application done at step three | | |
| i_04_01 | Check program is created correctly and student can take attendance | 1. Login to system with admin account<br>2. Select "Manage Program" tab in the | Student able to takes attendance successfully | email: u1@gmail.com<br>password: 12345678<br>program: OTB |

| | | | | |
|---|---|---|---|---|
| | of the program | navigation bar<br>3. Fill the form and select today as the date of program<br>4. Submit the form<br>5. Logout and login to system with student account<br>6. Select the "Attendance" tab in the navigation bar<br>7. Select the program just now and take the attendance | | |



## 6.5 Test Result and Analysis

Test results and analysis is the chapter to discuss about results of test case and analysis the test results. The test result for each test case will be pass or fail. All of the test results are recorded. All of the test cases for unit testing are recorded as pass 100%. In another word, 35 test cases are passed and this indicates each function of the system can be functioned well. While, for the integrating system, one of five test cases fail and the other four pass. This means only 75% of integration testing pass. The failure test case is i_01_01 test case which cannot function well when the size of all the

certificates is greater than 16 megabytes. MongoDB limits and thresholds for the BSON documents which is the type of certificate stored is 16 megabytes. The system is corrected by retrieving the data with the identification of certificates only to ensure the data is not exceed 16 megabytes and only retrieve the document from the database when the user wants to download the certificate by using the identification of certificate. After fixing the problem, both unit and integration testing are 100% pass.

## 6.6 Conclusion

This chapter is to ensure that the system fulfilled the specifications which are stated in the requirements. In conclusion, the system is performed well and as expected.

# CHAPTER 7: PROJECT CONCLUSION

## 7.1 Observation on Weakness and Strengths

After the development of this system and testing, it can be seen that the project has multiple weakness it could be improved in the future. However, there has many strengths that are provided by the system.

### 7.1.1 Weakness

The weakness of this system is the system could not match the location of user with the avenue of the program because the system does not get the location of user. User can deceive the system to get the attendance of the program. Furthermore, the user who take the attendance may not the actual owner of the user account because the system checks the identification of the user by email and password only.



### 7.1.2 Strengths

The system aims to provide a platform which combines some services into one. All the students, organizer and management team use the same platform which enables them to communicate each other. Before this, they difficult to get the information from each other. Indirectly, all the progress is delay because the time is wasted in unnecessary steps.

All the attendances is recorded and displayed in the system. In another word, the attendance is updated from time to time and visible to all the users. So, if any changes, the user able to know immediately. The organizer and management team can

easily to track and export the accurate attendance of a program with the system if needed. They do not need to do unnecessary paper works to get the attendance. The system provides the service to generate a report which contains the information of the program and the list of the attendance. The attendance also will be calculated and limit the student to take. This service is to ensure the number of took attendance is in the range of the number of participants stated by the organizer.

For the organizer, they can easily get the notification about the latest changes from the management teams. The system will send an email to inform the creator of the program whether the program is approved or rejected. The reason of the reject also will be included to ease the organizer to make adjustment by read through the email.

## 7.2 Propositions for Improvement

To prevent any fake attendance, some techniques such as face recognition can be implemented into the system. Face recognition is one of the high accuracy biometric methods to detect the identification of the student.

Attendance with location could also be implemented. The system able to track the location of the student to check whether the tracked location is matched with the avenue of the program.



To provide multiple platforms, the system could be implemented into different platform such as desktop and mobile (IOS and Android operating system). Multiple platforms provide convenience to the user to choose as the most suitable platform to use the system.

## 7.3 Project Contribution

This project on the university level will contribute to a reduced amount of time and human resources when a organizer conducting a program. For the management team, the project eases the works such as generating report.

## 7.4 Conclusion

The project has been completed successfully while there have some improvements. But, as a prototype, it is considered as a finished product. It is really can be helpful and useful system to the student, organizer and management team.

## REFERENCES


*10 benefits of fact-finding | ThinkAdvisor* (no date).

*Definition Of Fact Finding Techniques Information Technology Essay* (no date).

*Geolocation: Displaying User or Device Position on Maps* (no date). Available at: https://developers.google.com/maps/documentation/javascript/geolocation (Accessed: 2 August 2021).

*How time and attendance technology can benefit your small business clients* (2017). Accounting Today. Available at: https://www.accountingtoday.com/opinion/how-time-and-attendance-technology-can-benefit-your-small-business-clients (Accessed: 11 June 2021).

*myUTeM - Apps on Google Play* (no date). Available at: https://play.google.com/store/apps/details?id=myutem2.utem.edu.my(Accessed: 11 June 2021).

*SDLC - Waterfall Model - Tutorialspoint* (no date).

*Time Attendance System Malaysia | Attendance App | Attendance System* (no date). Available at: https://www.info-tech.com.my/time-attendance-software-pro.htm (Accessed: 30 July 2021).

*Track time from clock-in to timesheets with StaffAny* (no date). Available at: https://www.staffany.com/cico-with-us/ (Accessed: 30 July 2021).

# APPENDIX A

Sample Source Code

UserController.js

Web service used to manage the request from and give response back to the web app.

```
const dotenv = require('dotenv')
dotenv.config({ path: './config/config.env' })
const User = require('../models/user')
const { genSaltSync, hashSync, compareSync } = require('bc
rypt')
const { generateToken, sendEmailbyNodeMailer } = require('
../service/service')

/**
 * @desc Login to system
 * @route POST /login
 * @access Public
 * @body { email, password } req
 * @json { token } res
 */
exports.login = (req, res) => {
   let filter = { email: req.body.email }

   console.log(new Date().toDateString(), req.body.email)

   User.findOne(filter, (err, user) => {
      if (err)
         return res.status(500).json({
            success: 0,
            message: err,
         })
      if (!user) {
         return res.status(401).json({
            success: 0,
            message: 'Student Not Found',
         })
      }
      // incorrect password
      if (!compareSync(req.body.password, user.password))
{
```

```
            return res.status(401).json({
                success: 0,
                message: 'Invalid Credentials',
            })
        }

        return res.status(200).json({
            success: 1,
            message: 'Login Successfully',
            token: generateToken(user._id, 'userId'),
        })
    })
}

exports.registerUser = async (req, res) => {
    const body = req.body
    console.log(body)
    body.password = hashSync(body.password, genSaltSync(10)
)

    let user = {
        first_name: body.firstName,
        last_name: body.lastName,
        contact_number: body.contactNumber,
        matric_number: body.matricNumber,
        email: body.email,
        password: body.password,
        role: body.role === '' ? 'student' : body.role,
    }

    try {
        let response = await User.create(user)
        return res.status(201).json({ response })
    } catch (err) {
        console.log(err)
        return res.status(200).json({ error: handlerError(er
r) })
    }
```

```
}

exports.updateUser = (req, res) => {
   let data = req.body

   let user = {
      first_name: data.firstName,
      last_name: data.lastName,
      contact_number: data.contactNumber,
      email: data.email,
      password: hashSync(data.password, genSaltSync(10)),
   }

   User.updateOne({ _id: req.params.id }, user, (err, doc)
 => {
      if (err?.code === 11000) {
         return fail(res, 404, { message: 'Email already e
xists' })
      }

      if (err) {
         return fail(res, 500, 'Internal Server error')
      }

      if (doc) {
         return res.status(200).json({ doc })
      }
   })
}

function handlerError(err) {
   let arrErr = []
   Object.keys(err.errors).forEach((errorKey) => {
      arrErr.push({
         key: errorKey,
         message: err.errors[errorKey].message,
      })
   })
```

```
    return arrErr
}

/** Get user by decode token to get user._id
 * todo: Rewrite aggregation
 * @param {*} req
 * @param {*} res
 * @returns
 */
exports.getUser = (req, res) => {
    let userId = req.user.id
    User.findOne({ _id: userId }, (err, user) => {
        if (err) res.status(401).json({ success: 0, message:
 err })
        if (user)
            return res.status(200).json({
                success: 1,
                message: 'Get User Info Successfully',
                user: {
                    id: user._id,
                    firstName: user.first_name,
                    lastName: user.last_name,
                    contactNumber: user.contact_number,
                    matricNumber: user.matric_number,
                    email: user.email,
                    role: user.role,
                    password: user.password,
                    country: user.country,
                    point: user.point,
                },
            })
        return res.status(204).json({ success: 0, message: e
rr })
    })
}

exports.forgotPassword = async (req, res) => {
    let email = req.query.email
```

```
   let exist = await User.findOne({ email }).catch((err) =
> {
      return res.status(400).json({ err })
   })
   console.log(exist)
   if (exist) {
      try {
         let password = await sendEmailbyNodeMailer(email)
         password = hashSync(password.toString(), genSaltS
ync(10))

         if (password) {
            return User.updateOne({ email }, { password })
               .then(function (response) {
                  return res.status(200).json({
                     message: 'New Password Sent to Your E
mail Address',
                  })
               })
               .catch(function (err) {
                  return res.status(400).json({ err })
               })
         }
      } catch (err) {
         return res.status(400).json({ err })
      }
   } else return res.status(200).json({ message: 'Email no
t found' })
}

const success = (res, data) => res.status(200).json({ succ
ess: 1, data })

const fail = (res, code, error) => res.status(code).json({
 success: 0, error })
```