

**PRIMCHAT SYSTEM****UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

## PRIMCHAT SYSTEM

AHMAD RAZIQ DANISH BIN AMIRRUDDIN



This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Database Management)] with Honours.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY  
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

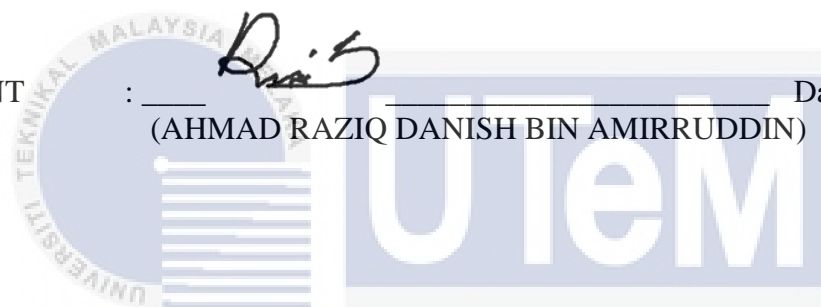
**DECLARATION**

I hereby declare that this project report entitled

**[PRIMCHAT SYSTEM]**

is written by me and is my own effort and that no part has been plagiarized  
without citations.

STUDENT :  \_\_\_\_\_ Date : 17/6/2021  
(AHMAD RAZIQ DANISH BIN AMIRRUDDIN)



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

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 (Database Management)] with Honours.

SUPERVISOR :  \_\_\_\_\_ Date : 8 September 2021  
(YAHYA BIN IBRAHIM)

## DEDICATION

First of all, I would like to dedicate my hard work and passion in this work towards my parents, siblings and every person who non-stop supporting my journey up until right now and made my education a priority in life.



## ACKNOWLEDGEMENTS

Firstly, a praise towards Almighty God, Allah SWT that bless me with good health and life to do and complete the given task, report of Final Year Project 1 & 2 (FYP 1 & FYP 2).

Also, a hundred of appreciations towards my supervisor, Sir Yahya Bin Ibrahim for his guidance, supportive and professionalism throughout this pandemic we are going through currently to complete this task.

A token of appreciation for my parents too for supporting every decision that correcting any mistake that I made.

Lastly, not to forget my dear seniors and course mates who had and are going through their own student's life.



## ABSTRACT

PRIMChat System is a web-based application system that can be surfed via a web browser such as Chrome, Safari, IE, etc. With a proper internet connection, users can access it easily without any inconvenience. PRIMChat System is developed to fulfill the needs of chatting module for the existing system named PRIM (Parent-Teacher Relation Information System), which is currently being used by various schools in Malacca to handle and manage school-related things such as online school fees payment, class management, etc. In my module, it consists of 2 types of users which are the Parent and School Teacher. The teacher can create a class group chat to broadcast messages to all their student's parents. Both the teacher and parent also can create a personal chatroom with each other to inquire about children's progress and status in school.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

## ABSTRAK

PRIMChat System adalah sistem aplikasi berasaskan web yang dapat dilayari melalui pelayar web seperti Chrome, Safari, IE, dan lain-lain. Dengan koneksi internet yang baik, pengguna dapat mengaksesnya dengan mudah tanpa kesulitan. PRIMChat System dibangunkan untuk memenuhi keperluan modul mesej dan chatting untuk sistem yang sedia ada dikenali sebagai PRIM (Parent-Teacher Relation Information System), yang kini digunakan oleh pelbagai sekolah di Melaka untuk menangani dan menguruskan hal-hal yang berkaitan dengan sekolah seperti pembayaran yuran sekolah secara atas talian, pengurusan kelas dan lain-lain. Dalam modul saya, ia terdiri daripada 2 jenis pengguna iaitu Ibu Bapa dan Guru Sekolah. Guru boleh membuat bilik pengumuman bagi kelas mereka untuk menyiarkan mesej kepada semua ibu bapa pelajar mereka. Guru dan ibu bapa juga dapat membuat bilik mesej peribadi antara satu sama lain untuk bertanya tentang kemajuan dan status anak-anak di sekolah.

## TABLE OF CONTENTS

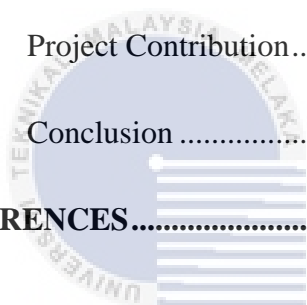
	<b>PAGE</b>
<b>DECLARATION.....</b>	<b>II</b>
<b>DEDICATION.....</b>	<b>III</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>IV</b>
<b>ABSTRACT .....</b>	<b>V</b>
<b>ABSTRAK .....</b>	<b>VI</b>
<b>TABLE OF CONTENTS.....</b>	<b>VII</b>
<b>LIST OF TABLES .....</b>	<b>XI</b>
<b>LIST OF FIGURES .....</b>	<b>XIII</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>XV</b>
<b>LIST OF ATTACHMENTS.....</b>	<b>XVI</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1    Introduction.....	1
1.2    Problem Statement .....	1
1.3    Objectives .....	2
1.5    Project Significance .....	3
1.6    Expected Output.....	3
1.7    Conclusion .....	4
<b>CHAPTER 2: PROJECT METHODOLOGY AND PLANNING .....</b>	<b>5</b>



2.1	Introduction.....	5
2.2	Project Methodology.....	5
2.2.1	Methodology in Developing Mythology .....	5
2.3	Project Planning and Milestone .....	7
<b>CHAPTER 3: ANALYSIS.....</b>		<b>8</b>
3.1	Introduction.....	8
3.2	Problem Analysis .....	8
3.3	Proposed Solution .....	9
3.4	Requirement Analysis.....	10
3.4.1	Functional Requirement.....	10
3.4.1.1	Context Diagram.....	11
3.4.1.2	Data Flow Diagram.....	11
3.4.2	Non-Functional Requirement .....	13
3.4.3	Non-Functional Requirement .....	15
3.4.3.1	Software Requirement .....	15
3.4.3.2	Hardware Requirement.....	16
3.5	Conclusion .....	16
<b>CHAPTER 4: DESIGN .....</b>		<b>17</b>
4.1	Introduction.....	17
4.2	Database Design.....	17
4.2.1	Conceptual Design.....	17
4.2.2	Logical Design.....	20
4.2.2.1	Data Validation .....	29

4.2.2.2	Query Design .....	29
4.3	Graphical User Interface (GUI) .....	30
4.3.1	Navigation Design .....	31
4.3.2	Input/Output Design .....	32
4.4	Conclusion .....	40
<b>CHAPTER 5: IMPLEMENTATION.....</b>		<b>41</b>
5.1	Introduction.....	41
5.2	Software Development Environment Setup.....	41
5.2.1	Step of Installation .....	41
5.3	Database Implementation.....	47
5.3.1	Data Definition Language (DDL).....	47
5.3.2	Stored Procedures .....	53
5.3.3	Triggers.....	53
5.4	Conclusion .....	54
<b>CHAPTER 6: TESTING .....</b>		<b>55</b>
6.1	Introduction.....	55
6.2	Test Plan.....	55
6.2.1	Test Organization.....	56
6.2.2	Test Environment.....	56
6.2.3	Test Schedule .....	57
6.3	Test Strategy .....	58
6.3.1	Classes of Tests.....	59
6.4	Test Design .....	59
6.4.1	Test Description.....	59

6.4.2	Test Data.....	68
6.5	Test Result and Analysis.....	77
6.6	Conclusion .....	81
<b>CHAPTER 7: PROJECT CONCLUSION .....</b>		<b>82</b>
7.1	Introduction.....	82
7.2	Observation on Strengths and Weaknesses.....	82
7.2.1	Strengths .....	82
7.2.2	Weaknesses .....	83
7.3	Propositions for Improvement .....	83
7.4	Project Contribution.....	84
7.5	Conclusion .....	84
<b>REFERENCES.....</b>		<b>85</b>



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

## LIST OF TABLES

	PAGE
<b>Table 3.1: Non-Functional Requirement .....</b>	<b>14</b>
<b>Table 3.2: Software Requirement for PRIMChat.....</b>	<b>15</b>
<b>Table 3.3: Hardware Requirement for PRIMChat .....</b>	<b>16</b>
<b>Table 4.1: Business Rules of PRIMChat System.....</b>	<b>19</b>
<b>Table 4.2: User Table.....</b>	<b>20</b>
<b>Table 4.3: Chat Table .....</b>	<b>21</b>
<b>Table 4.4: Grouproom Table .....</b>	<b>22</b>
<b>Table 4.5: Organization Table .....</b>	<b>22</b>
<b>Table 4.6: Organization_Role Table .....</b>	<b>23</b>
<b>Table 4.7: Organization_User Table.....</b>	<b>23</b>
<b>Table 4.8: Message Table .....</b>	<b>24</b>
<b>Table 4.9: Attachment Table.....</b>	<b>25</b>
<b>Table 4.10: Student Table .....</b>	<b>25</b>
<b>Table 4.11: Parent_Student Table.....</b>	<b>26</b>
<b>Table 4.12: Organization_Classes Table.....</b>	<b>26</b>
<b>Table 4.13: Class_Student Table .....</b>	<b>27</b>
<b>Table 4.14: BroadcastRooms Table.....</b>	<b>27</b>
<b>Table 4.15: BroadcastMessage Table .....</b>	<b>28</b>
<b>Table 6.1: Roles and Responsibilities .....</b>	<b>56</b>
<b>Table 6.2: Test Environment for Hardware Component .....</b>	<b>57</b>
<b>Table 6.3: Test Environment for Software .....</b>	<b>57</b>
<b>Table 6.4: Test Schedule in PRIMChat System .....</b>	<b>58</b>
<b>Table 6.5: Test Description for Registration .....</b>	<b>60</b>
<b>Table 6.6: Test Description for Login .....</b>	<b>61</b>

<b>Table 6.7: Test Description for Update Profile .....</b>	<b>62</b>
<b>Table 6.8: Test Description for Chatting .....</b>	<b>64</b>
<b>Table 6.9: Test Description for Broadcast Message.....</b>	<b>67</b>
<b>Table 6.10: Test Data for Registration.....</b>	<b>68</b>
<b>Table 6.11: Test Data for Login .....</b>	<b>70</b>
<b>Table 6.12: Test Data for Update Profile.....</b>	<b>71</b>
<b>Table 6.13: Test Data for Chatting.....</b>	<b>73</b>
<b>Table 6.14: Test Data for Broadcast Message .....</b>	<b>76</b>
<b>Table 6.15: Test Result for Registration .....</b>	<b>77</b>
<b>Table 6.16: Test Result for Login .....</b>	<b>78</b>
<b>Table 6.17: Test Result for Update Profile .....</b>	<b>78</b>
<b>Table 6.18: Test Result for Chatting .....</b>	<b>79</b>
<b>Table 6.19: Test Result for Broadcast Message.....</b>	<b>80</b>



## LIST OF FIGURES

	PAGE
Figure 2.1: Project Milestone Gantt Chart.....	7
Figure 3.1: Flow Chart of PRIMChat .....	9
Figure 3.2: PRIMChat System Context Diagram .....	11
Figure 3.3: Parent DFD Level 1 .....	12
Figure 3.4: Teacher DFD Level 1 .....	13
Figure 4.1: ERD for PRIMChat System .....	18
Figure 4.2: Data Validation for table User, GroupRooms, Chat and Message..	29
Figure 4.3: User Navigation (Teacher).....	31
Figure 4.4: User Navigation (Parent) .....	32
Figure 4.5: Landing Page .....	33
Figure 4.6: Register Page.....	33
Figure 4.7: Login Page.....	34
Figure 4.8: Dashboard (System Usage Reporting).....	34
Figure 4.9: Profile Page .....	35
Figure 4.10: Organization Page .....	35
Figure 4.11: Children Page.....	36
Figure 4.12: Main Messaging Page.....	36
Figure 4.13: Open New Group chat .....	36
Figure 4.14: Voice/Video Calling Page.....	37
Figure 4.15: Chatting page .....	37
Figure 4.16: Attachment Page (e.g. document) .....	38
Figure 4.17: Main Announcement Page (Display Broadcast Room).....	38
Figure 4.18: Create new broadcast message (Teacher) .....	39
Figure 4.19: Broadcast Message for Class .....	39

<b>Figure 4.20: Broadcast message (for Parent)</b> .....	<b>40</b>
<b>Figure 5.1: Download Page</b> .....	<b>42</b>
<b>Figure 5.2: First Installation page</b> .....	<b>42</b>
<b>Figure 5.3: Installation Option page</b> .....	<b>43</b>
<b>Figure 5.4: Installation Setup Summary</b> .....	<b>44</b>
<b>Figure 5.5: Laragon main page</b> .....	<b>45</b>
<b>Figure 5.6: Preferences page</b> .....	<b>46</b>
<b>Figure 5.7: Laragon services running</b> .....	<b>47</b>
<b>Figure 5.8: User Table DDL</b> .....	<b>48</b>
<b>Figure 5.9: Organization Table DDL</b> .....	<b>48</b>
<b>Figure 5.10: Organization Role Table DDL</b> .....	<b>48</b>
<b>Figure 5.11: Organization Class Table DDL</b> .....	<b>49</b>
<b>Figure 5.12: Organization User Table DDL</b> .....	<b>49</b>
<b>Figure 5.13: Student Table DDL</b> .....	<b>49</b>
<b>Figure 5.14: Parent Student Table DDL</b> .....	<b>50</b>
<b>Figure 5.15: Chat Table DDL</b> .....	<b>50</b>
<b>Figure 5.16: Grouproom Table DDL</b> .....	<b>50</b>
<b>Figure 5.17: Broadcast Room Table DDL</b> .....	<b>51</b>
<b>Figure 5.18: Attachment Table DDL</b> .....	<b>51</b>
<b>Figure 5.19: Broadcast Message Table DDL</b> .....	<b>51</b>
<b>Figure 5.20: Class Student Table DDL</b> .....	<b>52</b>
<b>Figure 5.21: Message Table DDL</b> .....	<b>52</b>
<b>Figure 5.22: Stored procedure for CreateChatroom</b> .....	<b>53</b>
<b>Figure 5.23: Stored procedure for CreateUser</b> .....	<b>53</b>
<b>Figure 5.24: Trigger before insert users</b> .....	<b>53</b>
<b>Figure 5.25: Trigger after update users</b> .....	<b>54</b>

**LIST OF ABBREVIATIONS**

<b>FYP</b>	-	<b>Final Year Project</b>
<b>PRIM</b>	-	<b>Parent-Teacher Relation Information</b>
<b>IE</b>	-	<b>Internet Explorer</b>
<b>JS</b>	-	<b>Javascript</b>
<b>SQL</b>	-	<b>Structured Query Language</b>
<b>JSON</b>	-	<b>Javascript Object Notation</b>
<b>IDE</b>	-	<b>Integrated Development Environment</b>



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



## LIST OF ATTACHMENTS

		<b>PAGE</b>
<b>Appendix A</b>	<b>Sample of data</b>	<b>19</b>
<b>Appendix B</b>	<b>Analysis of data collection</b>	<b>78</b>
.....	.....	
.....	.....	



## **CHAPTER 1: INTRODUCTION**

### **1.1 Introduction**

With the pandemic that is happening around us, the demands for an online system that can be used to manage and maintaining schools' data from home are increasing quickly. In this pandemic, people are advised to stay at home and avoid crowded place that can occur physical contact. The consequences of it, it is harder for teacher and parent to communicate with parents, as there are no physical activities being held at school including standard teaching and learning.

Both teacher and parent can use this system to communicate with each other to discuss school matters such as children's progress and position in academic. Teacher also can use it to remind parents of important matters or events such as examination week, school fees payment, etc.

### **1.2 Problem Statement**

The problem statements of the existing system are:

- i. No chatting module in the current system that allows user to communicate.
- ii. Difficult for teacher to search for parent's contact information.

- iii. Teacher asks student to remind their parents of important matters such as school fees.
- iv. School fees payment still by cash with no payment gateway integration.

### 1.3 Objectives

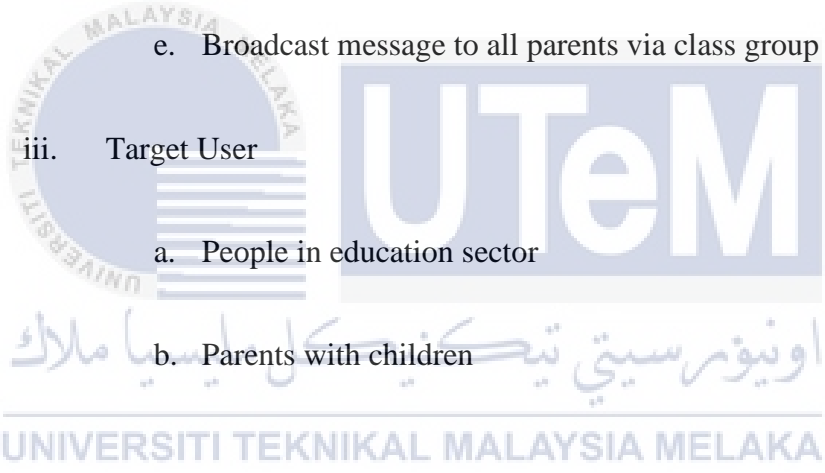
This project embarks on the following objectives:

- i. Provide a medium for users (parent and teacher) to communicate with each other more easily.
- ii. Develop an interface for teacher to easily identify and find parent's information and their children.
- iii. Provide functionality for teacher to broadcast message to all parents that are related to their class students.
- iv. Integrate current system with live payment gateway from PayNet to enable online transaction.

### 1.4 Scope

Module to be developed:

- i. Parent
  - a. Register
  - b. Login
  - c. View children's information

- d. Create chat room with class teacher
  - e. Online payment for school fees
- ii. School Teacher
    - a. Register
    - b. Login
    - c. View parent's information
    - d. Create chat room with parents
    - e. Broadcast message to all parents via class group
  - iii. Target User
    - a. People in education sector
    - b. Parents with children
- 

## 1.5 Project Significance

PRIMChat System is developed to provide chatting and broadcasting module for users (teacher and parent). There are several popular services that was used to develop this system such as Cloud Firebase from google, which provides real-time data management with NoSQL that allows rapid development.

## 1.6 Expected Output

Output 1: Display Children/Student information.

Output 2: Display parent/class teacher contact information.

Output 3: Option to create chat room between the teacher and parent.

Output 4: Option to broadcast message to all parents for teacher.

Output 5: Link current system with live payment gateway environment.

## 1.7 Conclusion

As a conclusion, this chapter is an overview of how and what is the purpose of the system to be developed. It consists of current system problem statements, objectives, and scope of this project.



## **CHAPTER 2: PROJECT METHODOLOGY AND PLANNING**

### **2.1 Introduction**

The methodology approach that is being used is database life cycle. Database life cycle consists of 5 stages which are planning, analysis, database design, implementation, and maintenance.

### **2.2 Project Methodology**

System Development Life Cycle (SDLC) methodology is being used as the approach to develop the web-application system. The database where we used to store and retrieve data is MySQL and Cloud Firestore. Laravel framework is used to develop the back-end of the system while the front end are mostly developed in Vanilla JS, jQuery and Bootstrap 4.

#### **2.2.1 Methodology in Developing Mythology**

The activities carried out in each phase are:

##### **a. Planning**

In planning phase, the first thing to do is always to determine what kind of project to develop and figure out what is the purpose of the proposed

project. The proposal must be created and deliver to supervisor to get approval. All hardware and software requirements must be listed to make sure all of it are ready before starting the project.

#### **b. Analysis**

Analysis phase is where the process of gathering data and information from various sources is initiated. For example, sources from online or interviewing public users to get requirement are the ways to get better understanding about the current problems and ideas on how to bring the solution.

#### **c. Design**

In this phase, blueprint or interface prototyping will be created after several data requirement gathered in analysis phase has been completed. It is used to deliver a design that suitable for requirement given from end users.

#### **d. Implementation**

Development coding process are initiated in this phase. After designing process has been completed and flows of the system are clear, software engineer translates the designing into source code.

#### **e. Maintenance**

Maintenance phase starts after implementation is completed. During this phase, various people like system analysts, testers, several end users will try out the system and give feedback to the developer regarding bugs, adding features etc. before publishing to the internet.

## 2.3 Project Planning and Milestone

Figure below shows the project milestone.

	Date Start	Date End	Time Period by Week and Month																			
			February				March				April				May				June			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Sending proposal	15/2	23/2																				
Approval proposal	24/2	28/2																				
System Planning to be developed	1/3	7/3																				
Project Identification, System Development & Analysis Planning	10/3	19/3																				
System Design and Delivery System Design	20/3	26/3																				
System Implementation and Delivery of System Implementation	1/4	30/5																				
Final Report Delivery	1/6	16/6																				
System Presentation and Assessment																						
Final Report Correction																						
Final Report Completion																						

Figure 2.1: Project Milestone Gantt Chart



## CHAPTER 3: ANALYSIS

### 3.1 Introduction

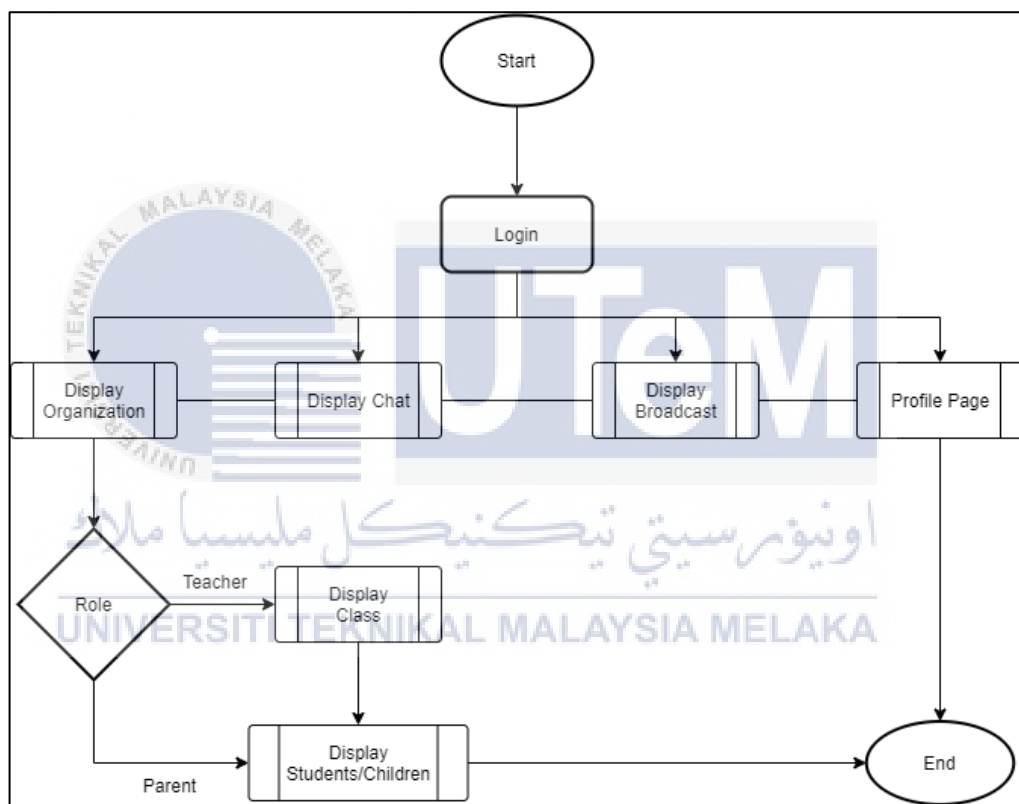
In Analysis Chapter, various of system requirement diagram will be shown to analyse data gathered by creating Flow Charts, Data Flow Diagram (DFD) and Context Diagram. It is also including the process of analysing the requirements gathered from earlier process in order to gain better understanding of what the client/user wants from the system to be developed, before development starts.

### 3.2 Problem Analysis

Problem analysis is the process of dissecting and meticulously evaluating an issue in order to comprehend how it arose and grew to its current size. With the pandemic happens today, many school activities are being held back and students and teachers are not allowed to continue the normal teaching and learning process. With existing system like PRIM, school staffs were able to control and manage school matters, such as student registration, school fees management etc. The only problem is there are no payment gateway integration for online payment and no communication medium between parents and teachers.

### 3.3 Proposed Solution

The proposed system is called PRIMChat which is a web-based application with responsive interface and can be accessed via mobile browser such as safari, chrome mobile. It has real-time data transaction for the chatting module, which means any changes towards data related to chatting is processed in real-time. Besides chatting, it provides feature for teacher/parent to easily identify each other based on their children/student's information. Teachers also able to broadcast important message for the parents to notify with important matters.



**Figure 3.1: Flow Chart of PRIMChat**

Figure 3.1 above shows the flow chart of PRIMChat system. Flow chart displays/visualizes the core process of the system including the business logic for the system.

### 3.4 Requirement Analysis

Requirement analysis is a task in systems engineering and software engineering that focuses on determining the needs or conditions for a new or altered product or project, considering the possibly conflicting requirements of various stakeholders, analysing, documenting, validating, and managing software or system requirements. This section aims to improve the system's overall performance in terms of efficiency, faster query and logical transaction.

#### 3.4.1 Functional Requirement

A functional requirement is a definition of behaviour between outputs and inputs that describes a function of a system or its component. The functional requirement for PRIMChat system includes users parent and teachers.

##### 1. Parent

- Register/Login

- Display/manage profile

- Display/create chatroom & message

- View schools and children

- View broadcast room & message

##### 2. Teacher

- Register/Login

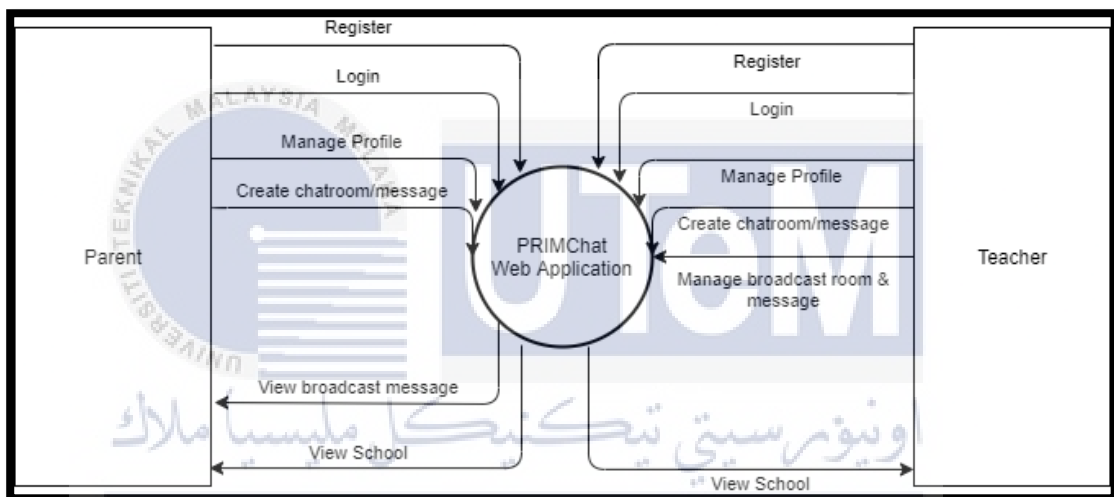
- Display/manage profile

- Display/create chatroom & message

- View school and student
- Create/Send broadcast message

### 3.4.1.1 Context Diagram

A context diagram is a diagram that shows the entities that interact with a system, or a section of a system, and specifies the border between the system and its environment. This diagram depicts a system at a high level, which is also known as 'Level 0' in Data Flow Diagram.



UNIVE **Figure 3.2: PRIMChat System Context Diagram A**

### 3.4.1.2 Data Flow Diagram

The Data Flow Diagram (DFD) is a graphical tool for depicting the processes or activities that are carried out as well as how data moves between each function. The first step in this project is to create a context diagram that depicts the external entities involved as well as the data flows that begin and terminate in them. Following that, a DFD fragment is generated based on the project requirements to demonstrate how data flows via all processes, external entities, and data stores.

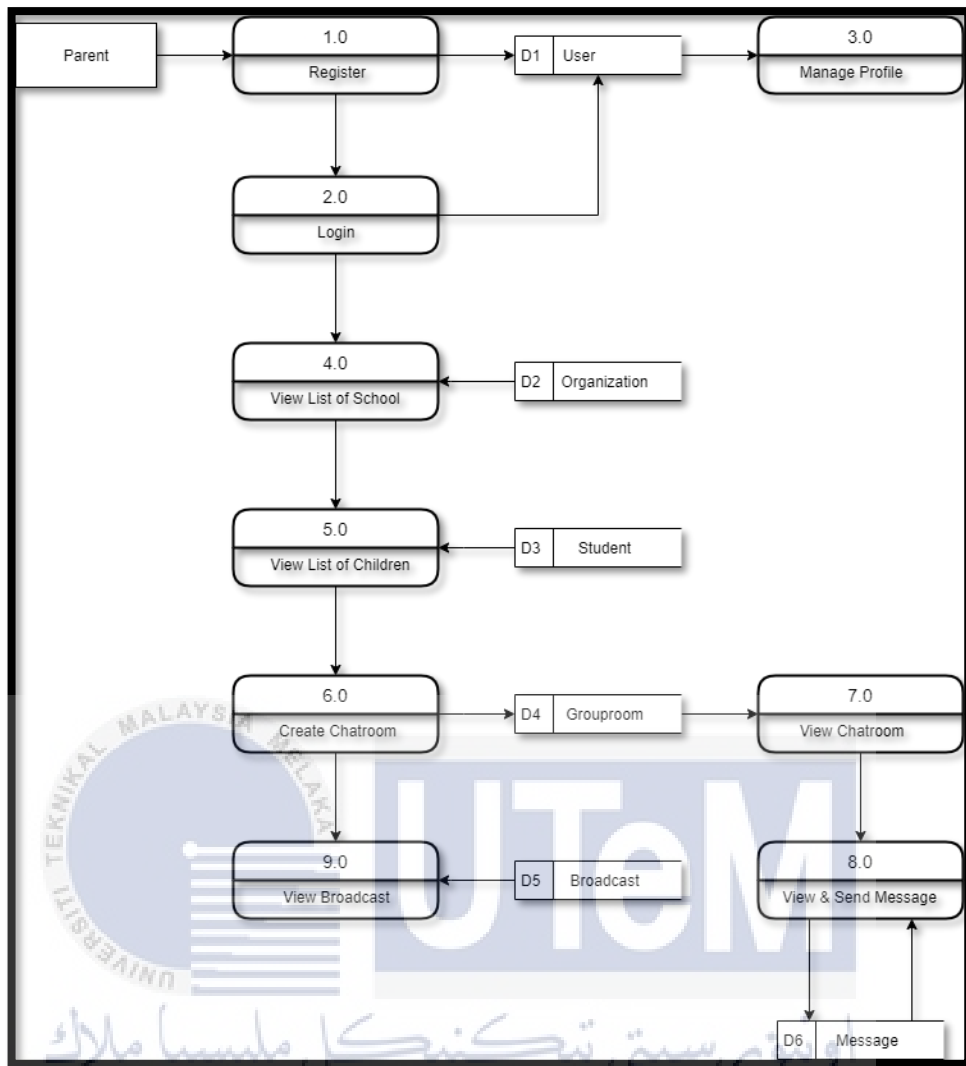
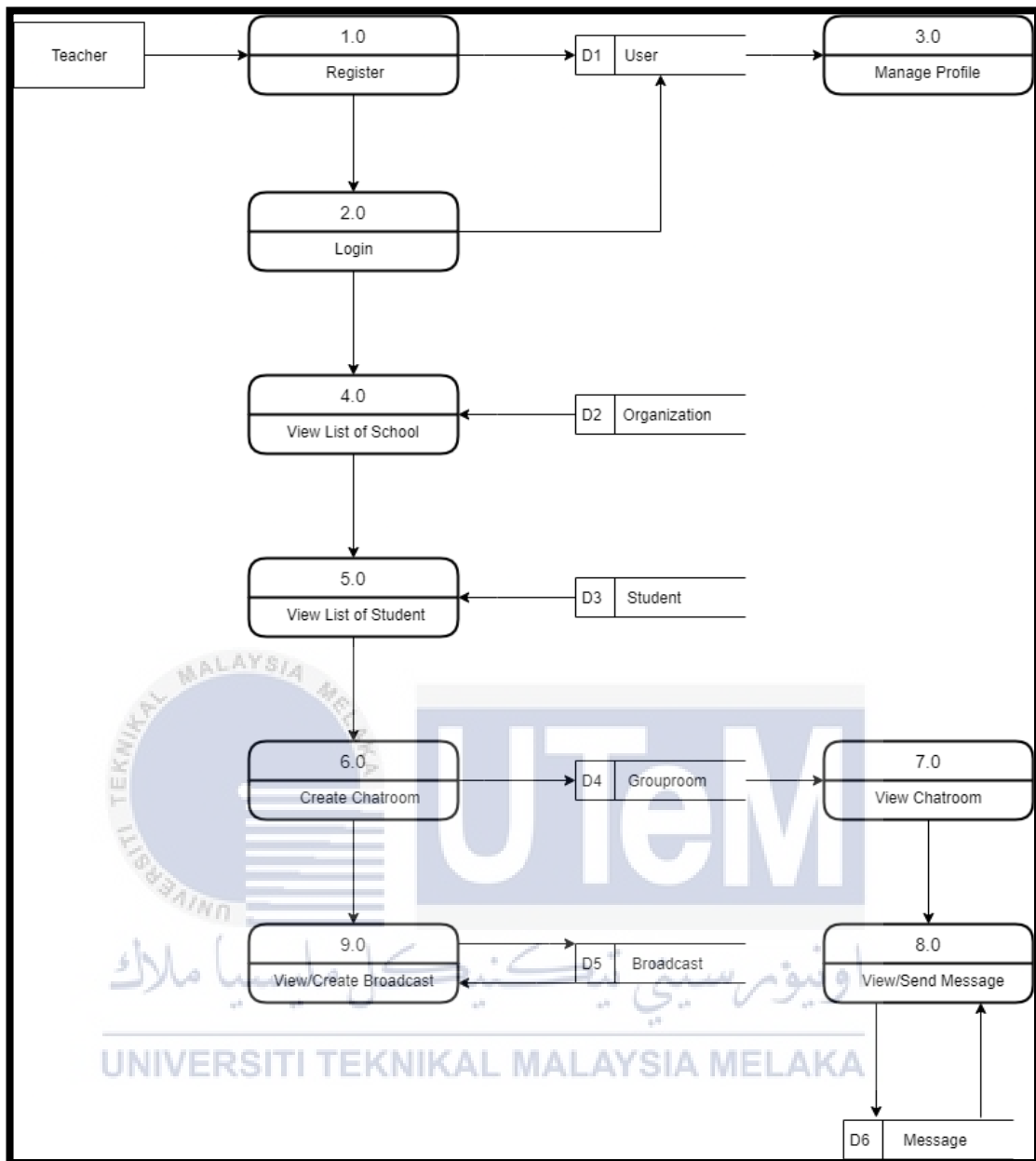


Figure 3.3: Parent DFD Level 1



**Figure 3.4: Teacher DFD Level 1**

### 3.4.2 Non-Functional Requirement

A non-functional requirement is a collection of requirements that describe a variety of system properties in order to increase user satisfaction with the system. **Table 3.1** contains all of the project's non-functional criteria and their descriptions.

**Table 3.1: Non-Functional Requirement**

Requirement	Description
Security	The system is equipped with various of validations which only several users can access specific data.
Reliability & Availability	The data stored in the system are hosted in a 24/7 dedicated server, thus is available for access if there is an internet connection.
Usability	The system works as many other online messaging applications, which provides real-time service.  Firebase (Cloud Firestore)
Capacity	The system database is subscribed to a pay-as-you-go bill where if there are increasing demand of capacity, the charges will apply to taken resource.  Gbnetwork (MySQL)
System Programming Language	The system used various of architecture and language to develop such as PHP (back-end), HTML, JS, CSS (front-end) and MySQL (database).

### 3.4.3 Non-Functional Requirement

The requirements for database system development can be divided into two categories: hardware requirements and software requirements. The software requirement outlines the software required to produce the system, whereas the hardware need explains the hardware used to run the software defined to build the system.

#### 3.4.3.1 Software Requirement

The requirements and specifications of software components needed to construct the PRIMChat System are listed in **Table 3.2**.

**Table 3.2: Software Requirement for PRIMChat**

No	Software Component Name	Function
1	Heidi SQL	Used to access MySQL database.
2	Windows 10	Operating System from Microsoft used as an environment to develop the system.
3	Visual Studio Code	VS Code is an IDE used to implement programming language such as PHP, HTML, JS and CSS.
4	Draw.io	Draw.io used to create various of system requirement and design diagram for the system.
5	Microsoft Word 2019	Used to create proposal and final report.



6	Laragon	Laragon used to serve the website in local environment for development & testing
7	Github	Github used to store the coding over the internet, which can be used to backup the source code and discuss with the community on the internet.

### 3.4.3.2 Hardware Requirement

**Table 3.3** shows the hardware components that will be used in the PRIMChat System.

**Table 3.3: Hardware Requirement for PRIMChat**

No	Hardware	Description
1	Desktop PC	The workstation used to develop the system.
2	Router	Devices that were used to connect the PC to the internet via LAN cable.
3	Dual Monitor	Display the coding workstation on first monitor and finding references on second.

### 3.5 Conclusion

In this chapter has covered most of the requirement analysis parts of the system that need to be implemented in the system. In the next chapter, an overview of the system design including interfaces, ERD, data dictionary and so on will be described.

## **CHAPTER 4: DESIGN**

### **4.1 Introduction**

In design phase, it is regarded as one of the most important phases in development because the output in this phase will affect the later phases. The logical system design resulted from system analysis and will be developed into the physical system design. Database design, scheme, input and output will be specified in this phase. In this stage also important because data structure, control processes and interface will be determined.

Various of techniques such as ERD, business rules, data dictionary, data normalization, selection of Database Management System (DBMS) and Graphical User Interface (GUI) will be drawn to design system.

### **4.2 Database Design**

Database design is meant for organize data according to database model. Database designer decides which data should be saved and how data should be used.

#### **4.2.1 Conceptual Design**

The first phase of database design methodology is conceptual database design. The ER-Model is created to represent the structure of the database. The entity relationship model (or ER model) is a graphical representation of the logical relationships between entities (or objects) in order to establish a database. During this

step, we identify entity types and relationship types. Identify and connect properties with entity or relationship types. It also determines attribute domains and primary key characteristics and considers the application of advanced modelling concepts.

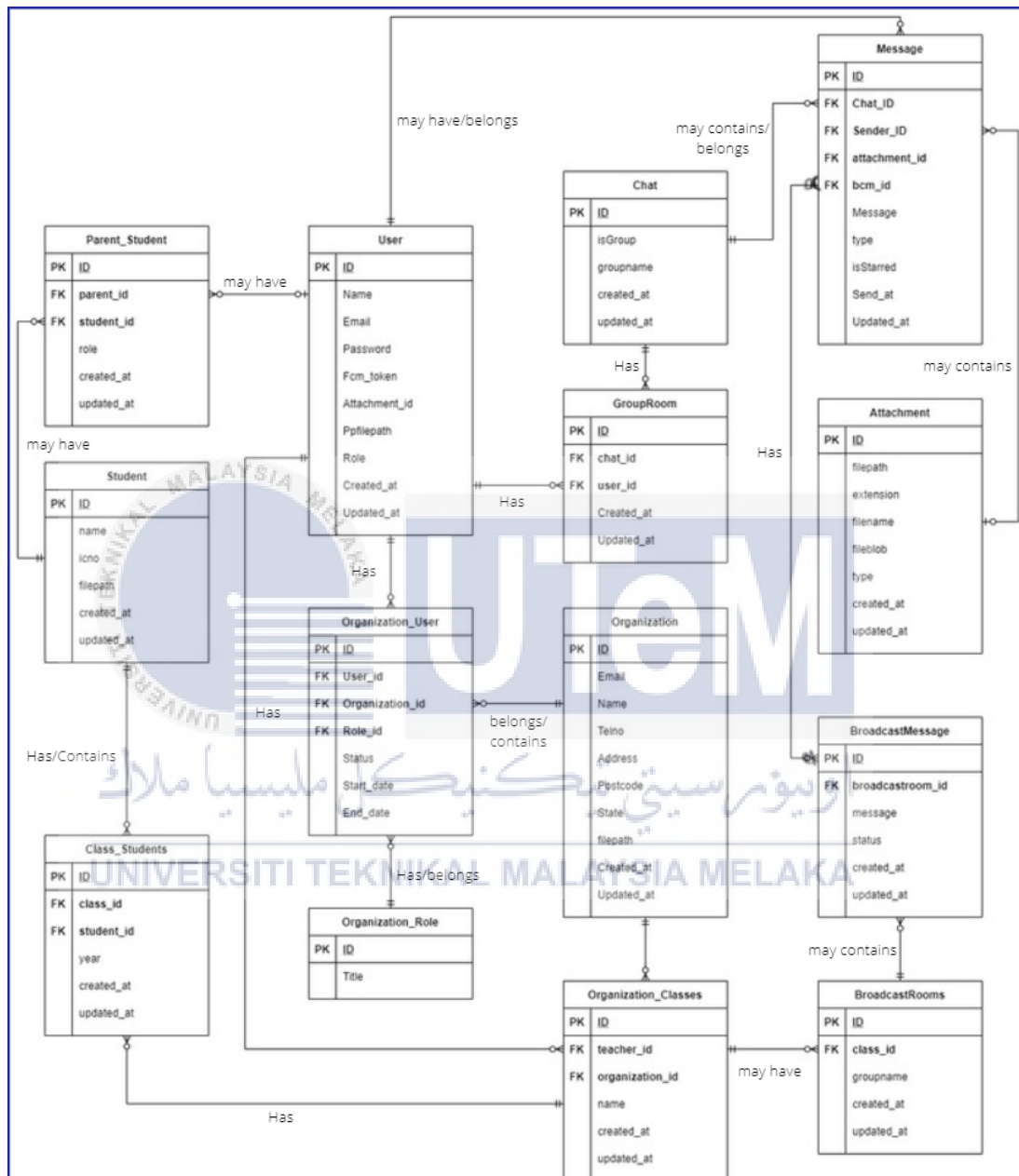


Figure 4.1: ERD for PRIMChat System

Table 4.1 is PRIMChat System's business rules that provide explanations for every relationship among entities based on Figure 4.1.

**Table 4.1: Business Rules of PRIMChat System**

Business Rule	Description
1	1 user may not have an organization or belongs to many, while one organization must contain at least one or many users.
2	1 user must have 1 and only 1 role in 1 organization, while a role can be holds by 0 or many users.
3	1 user should have 0 or many chatrooms in the system, while 1 chat room must contain at least 2 user or many.
4	1 chatroom may contain 0 or many messages, while a message must belong to 1 and only 1 chatroom.
5	1 message may contain 0 or 1 attachment, and 1 attachment may contain 0 or many messages.
6	1 message may be a reply 0 or 1 broadcast message, while a broadcast message may be referred by 0 or many messages.
7	1 organization may have 0 or many classes, and 1 class belongs to 1 and only 1 organization.
8	A class has 1 and only 1 class teacher, and 1 user may handle 0 or many classes.

9	A student may have names in 1 or many classes, and 1 may contain 1 or many students.
10	1 user may be a parent to 0 or many students, and a student may have a 0 or many parent users.

#### 4.2.2 Logical Design

Below shows the data dictionary for PRIMChat System that contains attributes for each entity and explanations of data type, constraint, foreign key and description.

**Table 4.2: User Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	User ID	int	yes	yes	PK	
Name	User's Name	Varchar(100)		yes		
Email	User's Email	Varchar(100)	yes	yes		
Password	Account Password	Varchar(100)		yes		
Fcm_token	Firestore Device Token	Varchar(255)	yes			
Attachment_id	Attachment File ID	Int			FK	Attachment(ID)

Ppfilepath	Profile Picture Filepath	Varchar(255)				
Role	User Role	Int		yes		
Created_at	Account Created	Datetime		yes		
Updated_at	Account Updated	Datetime		yes		

**Table 4.3: Chat Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Chat ID	int	yes	yes	PK	
isGroup	Attribute for determining whether group chat or not	bool		yes		
Groupname	Group Name	Varchar(50)				
Created_at	Chatroom created	Datetime		Yes		
Updated_at	Chatroom updated	Datetime		yes		

**Table 4.4: Grouproom Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Group Room ID	Int	Yes	Yes	PK	
Chat_ID	Chat ID	Int		Yes	FK	Chat(ID)
User_ID	User ID	Int		Yes	FK	User(ID)
Created_at	Group created	Datetime		Yes		
Updated_at	Group updated	Datetime		Yes		

**Table 4.5: Organization Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Organization ID	int	yes	yes	PK	
Email	Organization Email	Varchar(100)	yes	yes		
Name	Organization Name	Varchar(100)		yes		
Telno	Organization Telephone No	Varchar(12)	yes			

Address	Organization Address	Varchar(100)				
Postcode	Organization postcode	int				
State	Organization state	Varchar(30)		yes		
Filepath	Organization picture path	Varchar(255)		yes		
Created_at	Organization date created	Datetime		yes		
Updated_at	Organization updated date	Datetime		yes		

**Table 4.6: Organization\_Role Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Role ID	int	yes	yes	PK	
Title	Role title	Varchar(50)	yes	yes		

**Table 4.7: Organization\_User Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Organization User ID	Int	yes	yes	PK	
User_id	User ID	Int		Yes	FK	User(ID)
Organization_id	Organization ID	Int		Yes	FK	Organization(ID)



Role_id	Organization Role ID	Int		yes	FK	Organization_Role(ID)
Status	User Status	varchar		yes		
Start_date	Date of joining organization	Date		Yes		
End_date	Expiry date of member	date		yes		

**Table 4.8: Message Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Message ID	int	yes	yes	PK	
Chat_id	Chat ID	int		yes	FK	Chat(ID)
Sender_id	Sender (User ID)	Int		yes	FK	User(ID)
Attachment_id	Attachment ID	Int		yes	FK	Attachment(ID)
Bcm_id	Broadcast Message ID	Int			FK	BroadcastMessage (ID)
Message	Text message	Varchar(255)		Yes		
type	Message type	Varchar(20)		Yes		
isStarred	Starred Message	Boolean		Yes		
Send_at	Send date	Datetime		Yes		

Updated_at	Updated date	datetime		yes		
------------	--------------	----------	--	-----	--	--

**Table 4.9: Attachment Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Attachment ID	Int	yes	yes	PK	
Filepath	File path	Varchar(255)		Yes		
Extension	File extension	Varchar(5)		Yes		
Filename	File name	Varchar(30)		Yes		
Fileblob	Blob binary	Big int				
Type	Media type	Varchar(20)		Yes		
Created_at	Attachment stored date	Datetime		Yes		
Updated_at	Attachment updated date	datetime		yes		

**Table 4.10: Student Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Student ID	Int	yes	yes	PK	
Name	Student Name	Varchar(100)		Yes		
Icno	Student IC No	Varchar(12)	yes	Yes		

Filepath	Student image path	Varchar(255)				
Created_at	Student registered date	Datetime		Yes		
Updated_at	Student updated date	Datetime		Yes		

**Table 4.11: Parent\_Student Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Parent Student ID	Int	yes	yes	PK	
Parent_id	User ID (parent)	Int		yes		
Student_id	Student ID	Int		Yes		
Role	Parent Role	Varchar(20)		Yes		
Created_at		Datetime		Yes		
Updated_at		Datetime		Yes		

**Table 4.12: Organization\_Classes Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Class ID	Int	yes	yes	PK	
Teacher_id	User ID (Teacher)	Int		Yes	FK	User(ID)

Organization_id	Organization ID	Int		Yes	FK	Organization(ID)
Name	Class Name	Varchar(50)		Yes		
Created_at		Datetime		Yes		
Updated_at		Datetime		Yes		

**Table 4.13: Class\_Student Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Class Student ID	Int	yes	yes	PK	
Class_id	Class ID	Int		Yes	FK	Organization_Classes(ID)
Student_id	Student ID	Int		Yes	FK	Student(ID)
Year	Current year	int		Yes		
Created_at		Datetime		Yes		
Updated_at		datetime		yes		

**Table 4.14: BroadcastRooms Table**

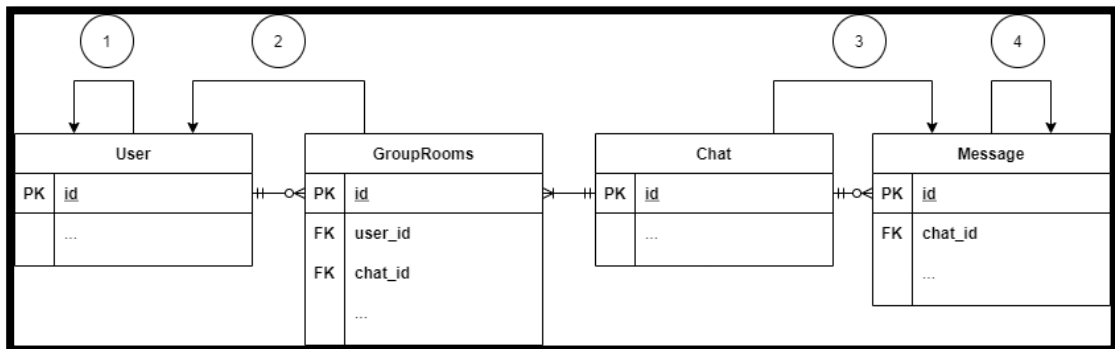
Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Broadcast Room ID	Int	yes	yes	PK	

Class_id	Class ID	int		yes	FK	Organization_Classes (ID)
Groupname	Class Group Name	Varchar(50)		Yes		
Created_at		Datetime		Yes		
Updated_at		Datetime		yes		

**Table 4.15: BroadcastMessage Table**

Attribute	Description	Data Type	Unique	Not Null	PK/FK	FK Reference Table
ID	Broadcast Message ID	Int	yes	yes	PK	
Broadcastroom_id	Broadcast room ID	int		yes	FK	BroadcastRooms (ID)
Message	Broadcast Message	Varchar(255)		Yes		
Status	Message status	boolean		Yes		
Created_at		Datetime		Yes		
Updated_at		datetime		yes		

#### 4.2.2.1 Data Validation



**Figure 4.2: Data Validation for table User, GroupRooms, Chat and Message**

1. Register new account in the system.
2. Retrieve group chat for the logged-on user correspond to the chat\_id inside GroupRooms table.
3. Fetch all messages based on chat\_id that user clicked to.
4. Send or modify messages in the chatroom.

Figure 4.2 presents data retrieval, insertion and manipulation of data. This has made every table in the system are required to fulfil the requirements from user.

#### 4.2.2.2 Query Design

Many queries have been created to produce various types of outputs. This query design is based on the data validation described previously. Table 4.3 displays various query design examples.

N o	Figur e	User Transacti on	DML Statement
1	4.2	Register user	<pre>\$user = new User([   'name' =&gt; 'Raziq'   'email' =&gt; 'raziq@test.com'   'password' =&gt; 'test123'</pre>

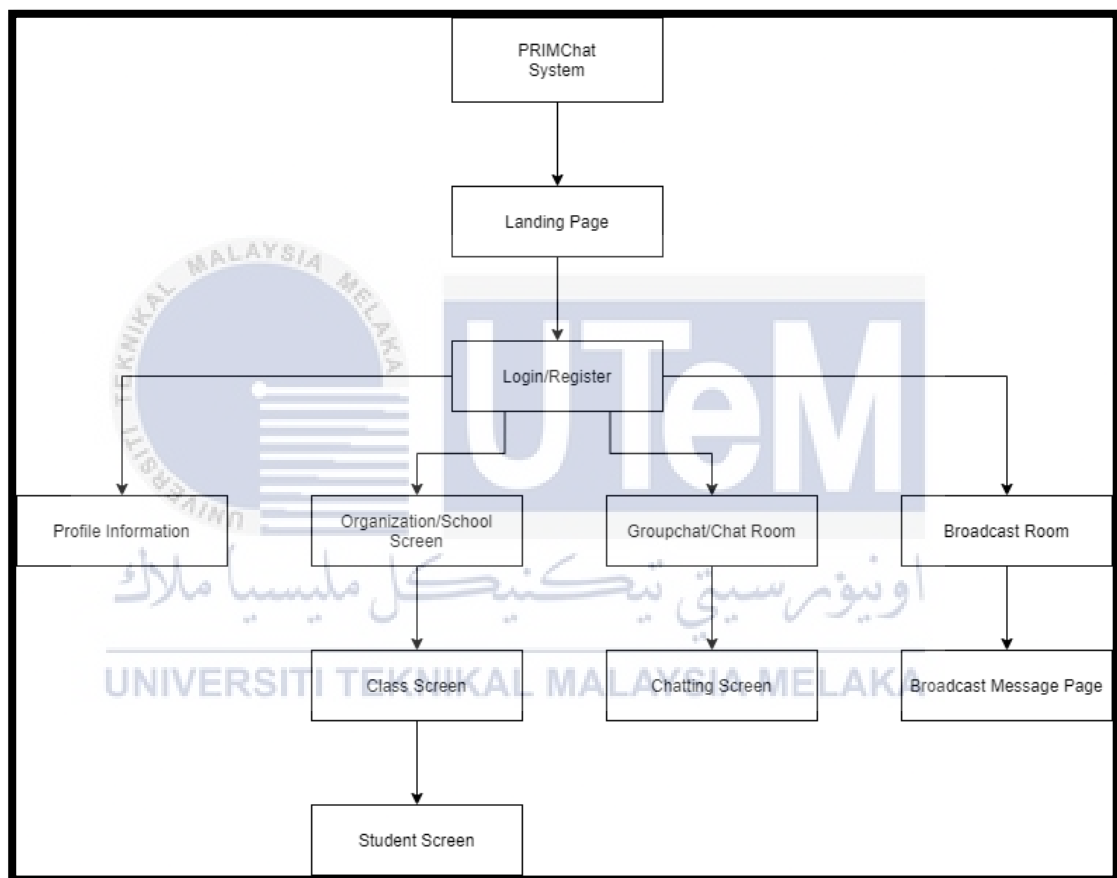
			<code>'role' =&gt; 1 ]);</code>
2	4.2	Retrieve group chat for user after login	<code>Chat::join('grouprooms','chats.id','=', 'grouprooms.cha t_id')-&gt;where('user_id',Auth::id())-&gt;get();</code>
3	4.2	Fetch messages based on chatroom	<code>Messages::join('users','messages.sender_id','=', 'users. id')-&gt; leftJoin('attachments','users.attachment_id','=', 'attachme nts.id')-&gt; orderBy('messages.created_at','desc')-&gt; where('chat_id',\$cid)-&gt;get();</code>
4	4.2	Insert message into database	<code>\$message = new Messages([ 'sender_id' =&gt; \$input['sender_id'], 'chat_id' =&gt; \$input['chat_id'], 'message' =&gt; \$input['message'] ]);</code>
5	4.2	Insert message with attachment	<code>\$message = new Message([ 'sender_id' =&gt; Auth::id(), 'message' =&gt; \$input['message'], 'chat_id' =&gt; \$input['chat_id'],  'isFile' =&gt; true, 'type' =&gt; \$type, 'attachment_id' =&gt; \$attachment-&gt;id ]);</code>

### 4.3 Graphical User Interface (GUI)

The process of developing the aesthetics and user interaction of a system is known as graphical user interface design. The design frequently focuses on usability in order to ensure that users can interact with the software quickly and naturally. Graphical user interface design is divided into three parts: navigation design, output design, and input design.

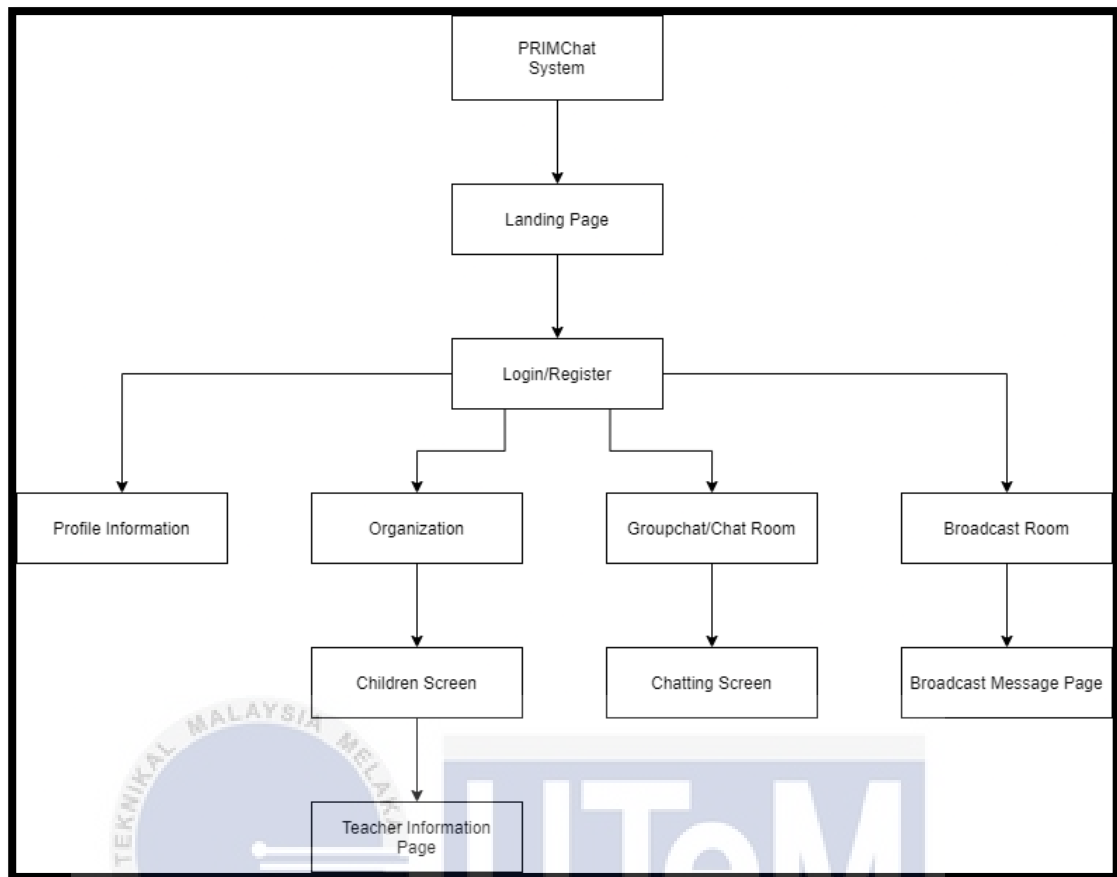
### 4.3.1 Navigation Design

The purpose of navigation design is to create the system navigation and how users can navigate between displays. Navigation is the act of navigating from one screen to another and can be image-based or text-based. A smart navigation design allows users to spend as little time as possible on the right interactions.



**Figure 4.3: User Navigation (Teacher)**





**Figure 4.4: User Navigation (Parent)**

#### 4.3.2 Input/Output Design

The process of designing inputs for the user is known as input design. It should be simple to use and understand. It should also validate user input by employing appropriate input controls and feedback design.

## i) Landing Page

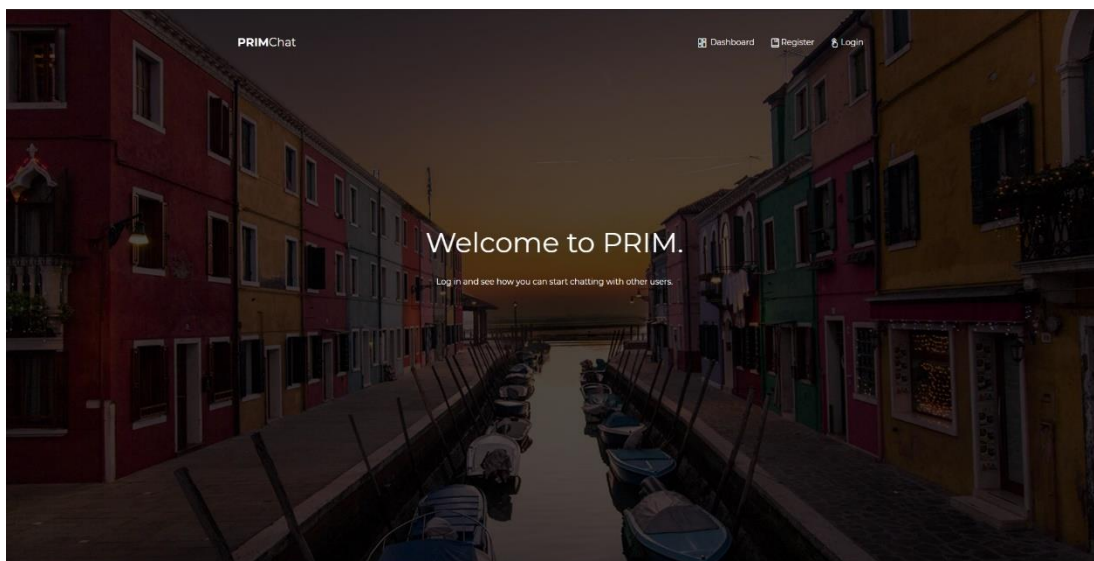


Figure 4.5: Landing Page



## ii) Register

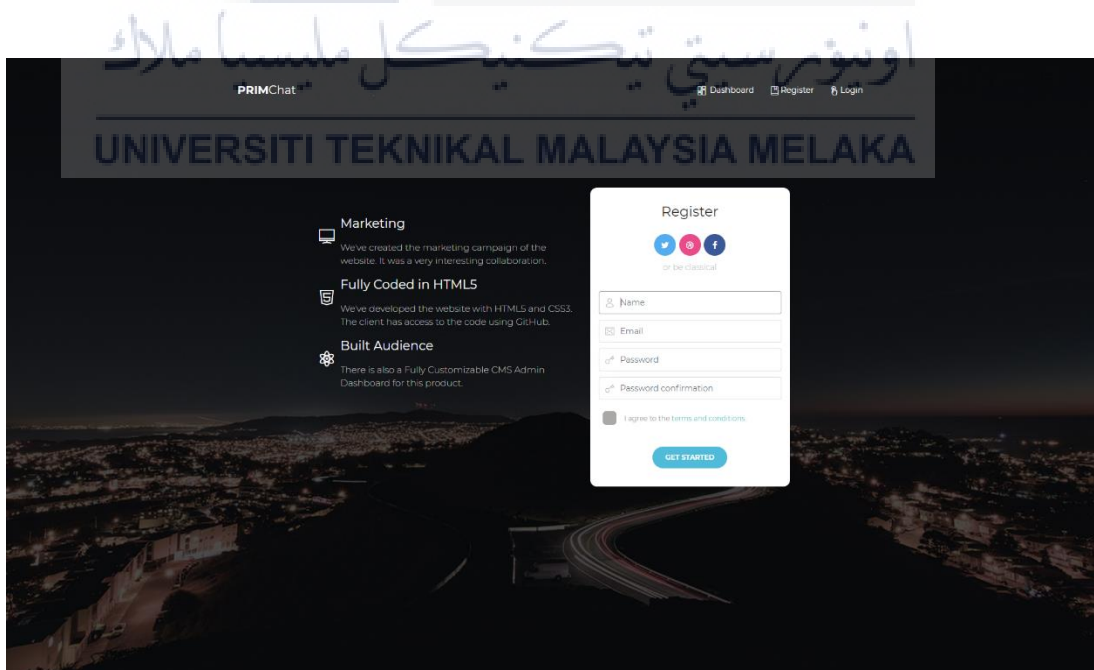


Figure 4.6: Register Page

iii) Login

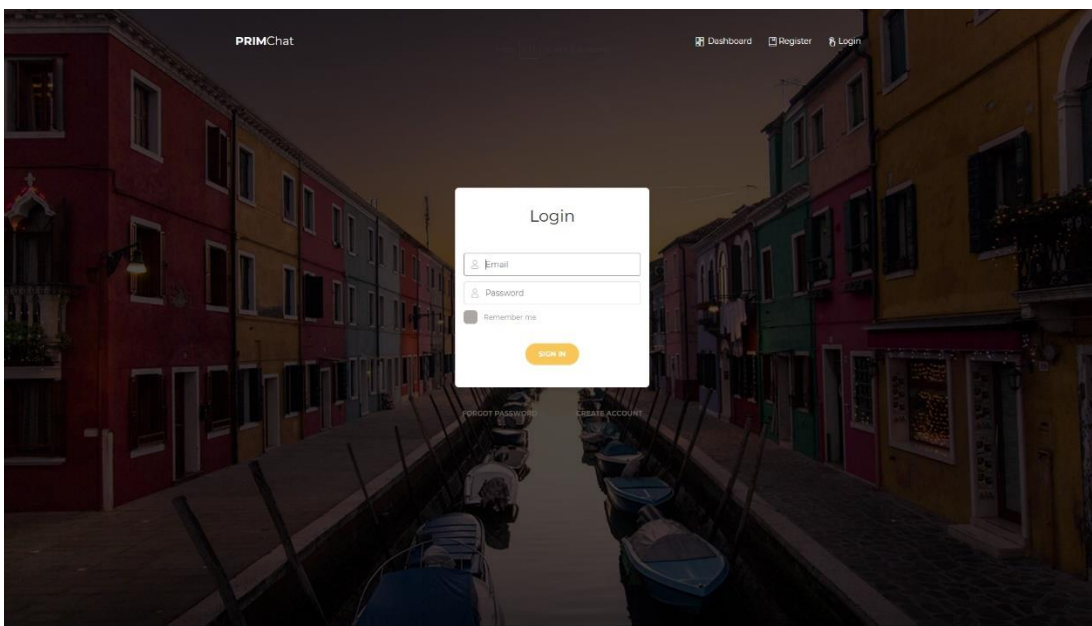
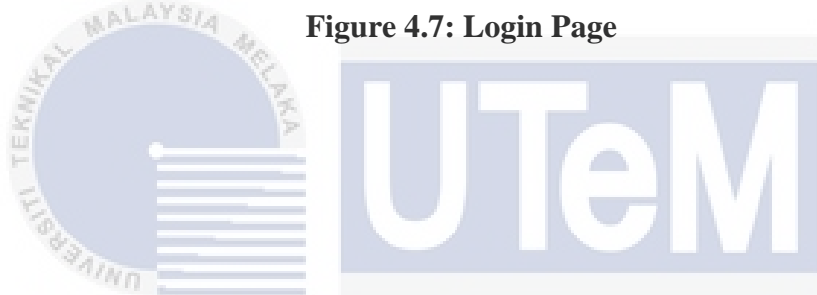


Figure 4.7: Login Page



iv) Dashboard (Logged-in)

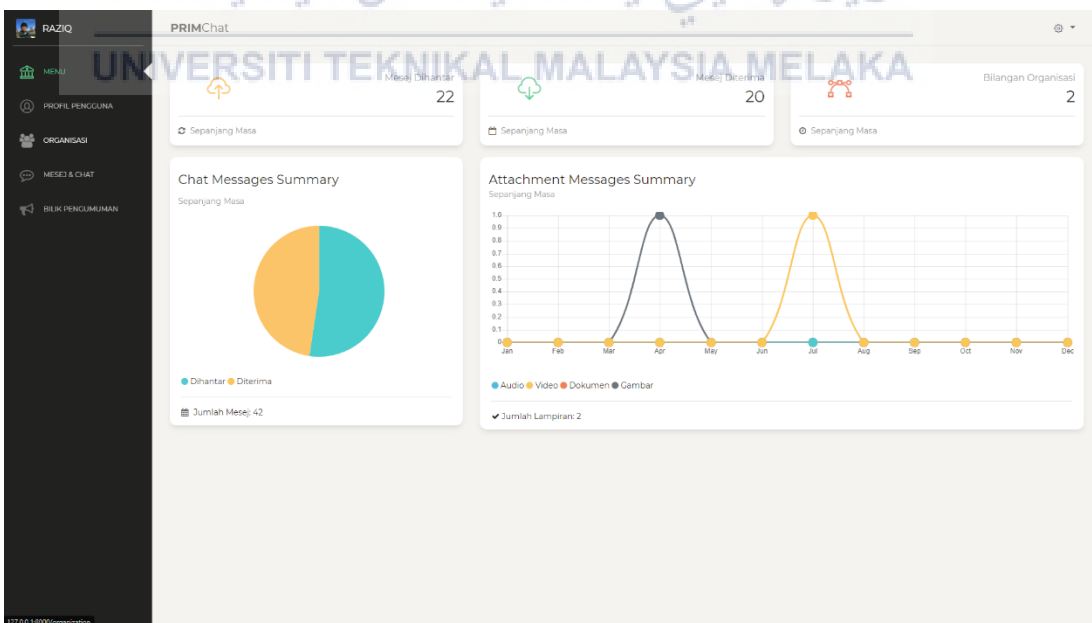


Figure 4.8: Dashboard (System Usage Reporting)

## v) User Profile

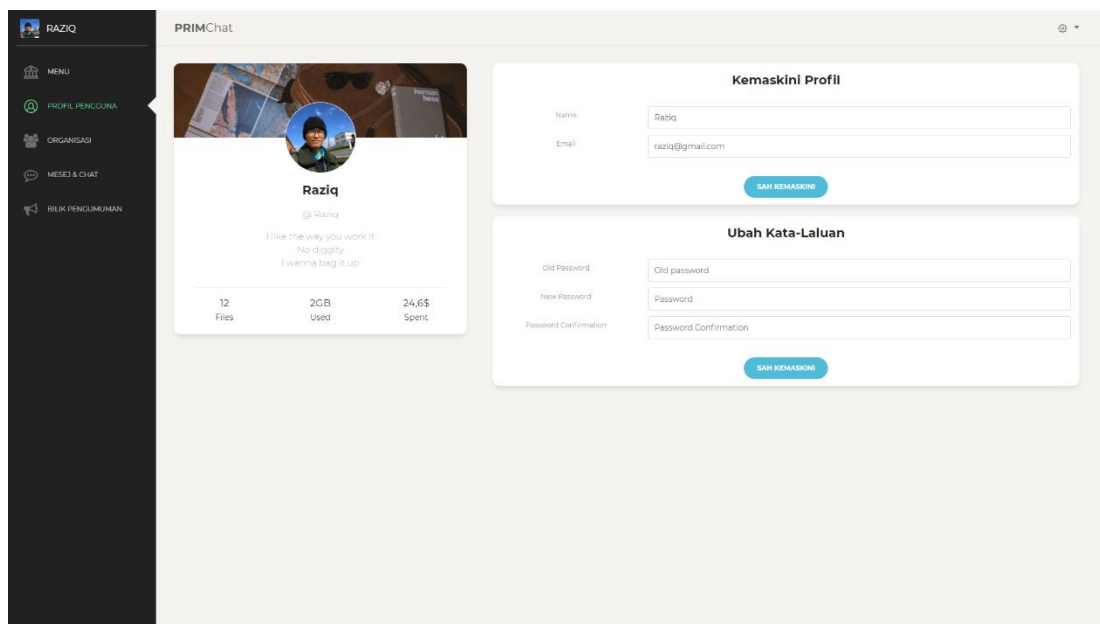


Figure 4.9: Profile Page

## vi) Organization Page

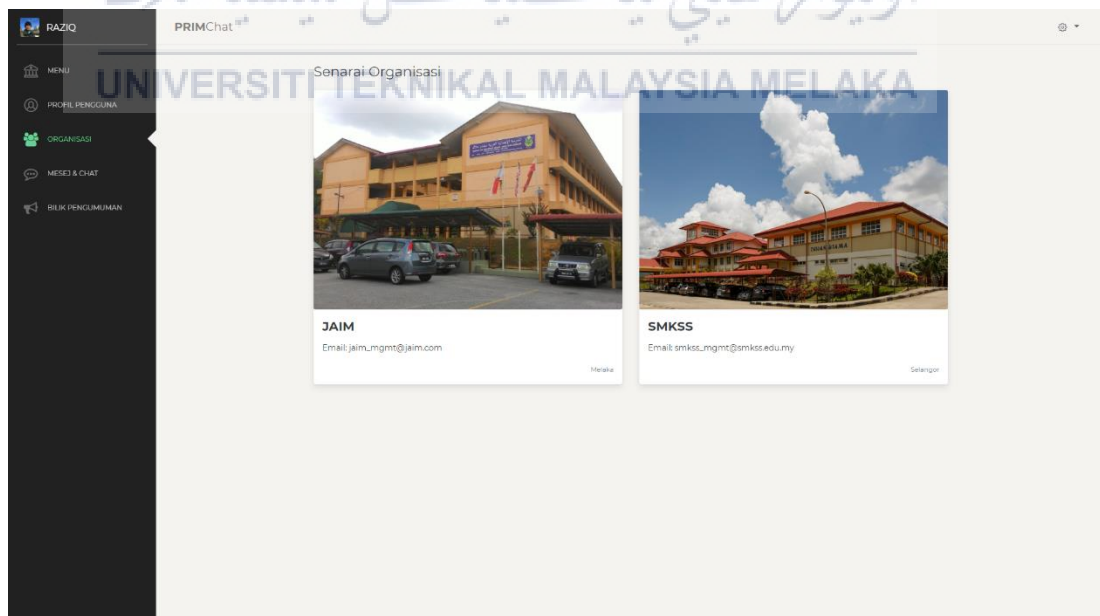
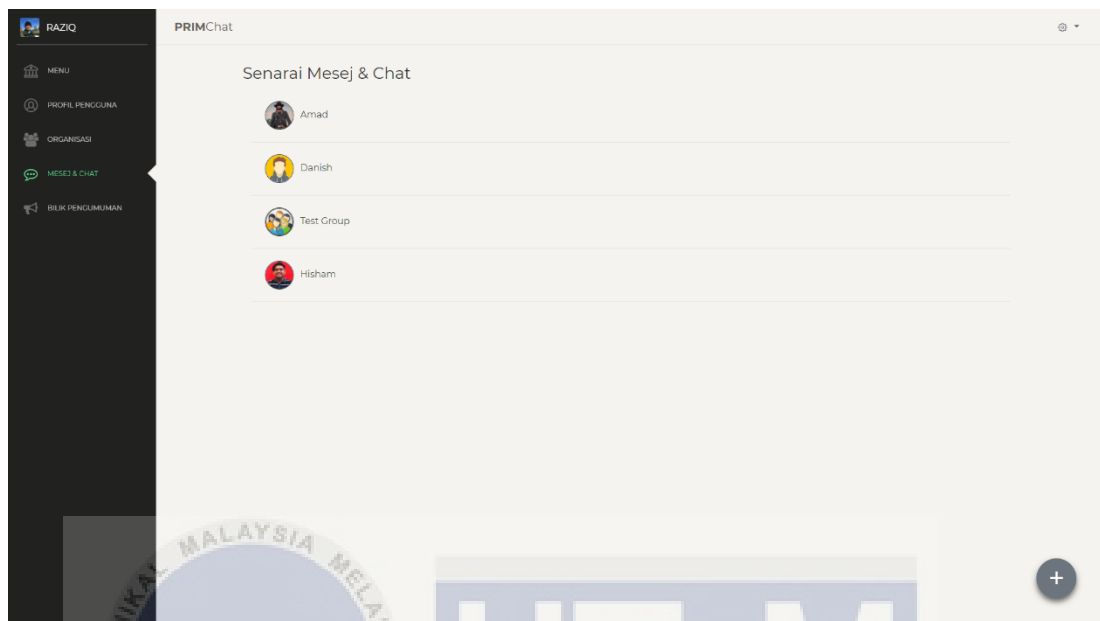


Figure 4.10: Organization Page

## vii) Chat Messaging



**Figure 4.12: Main Messaging Page**



**Figure 4.13: Open New Group chat**

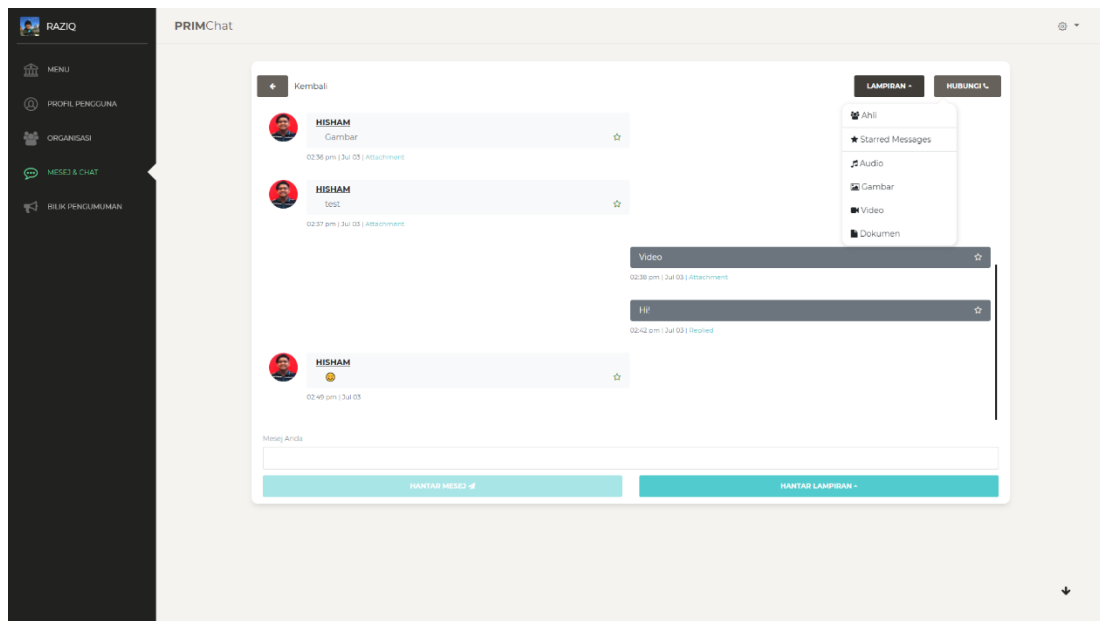


Figure 4.15: Chatting page

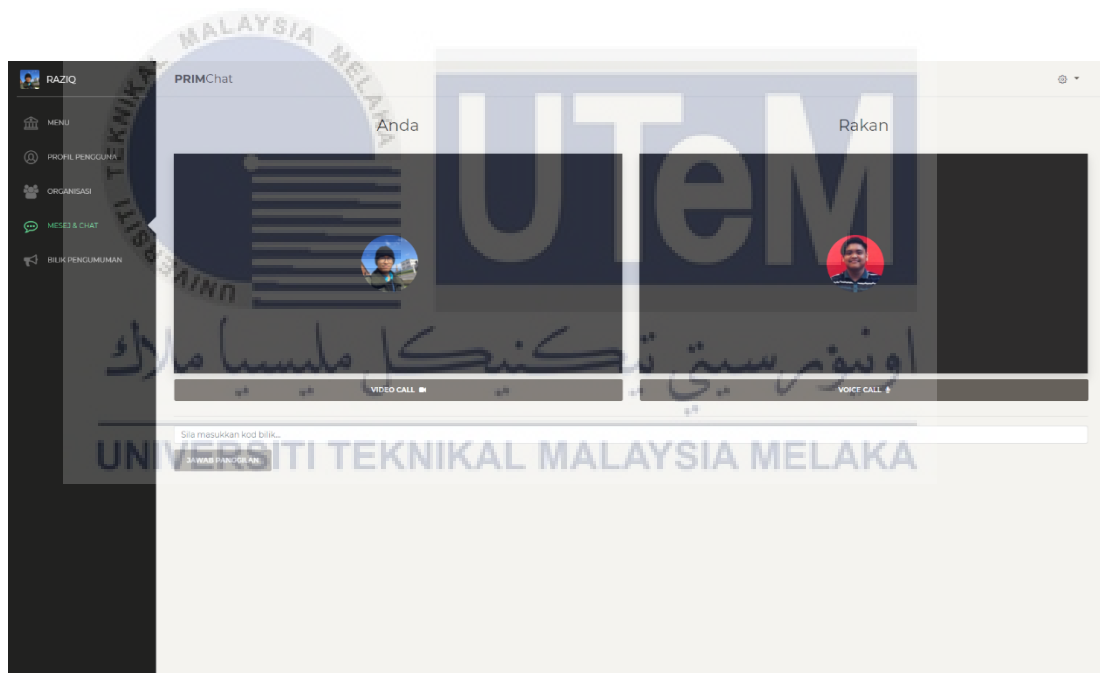


Figure 4.14: Voice/Video Calling Page

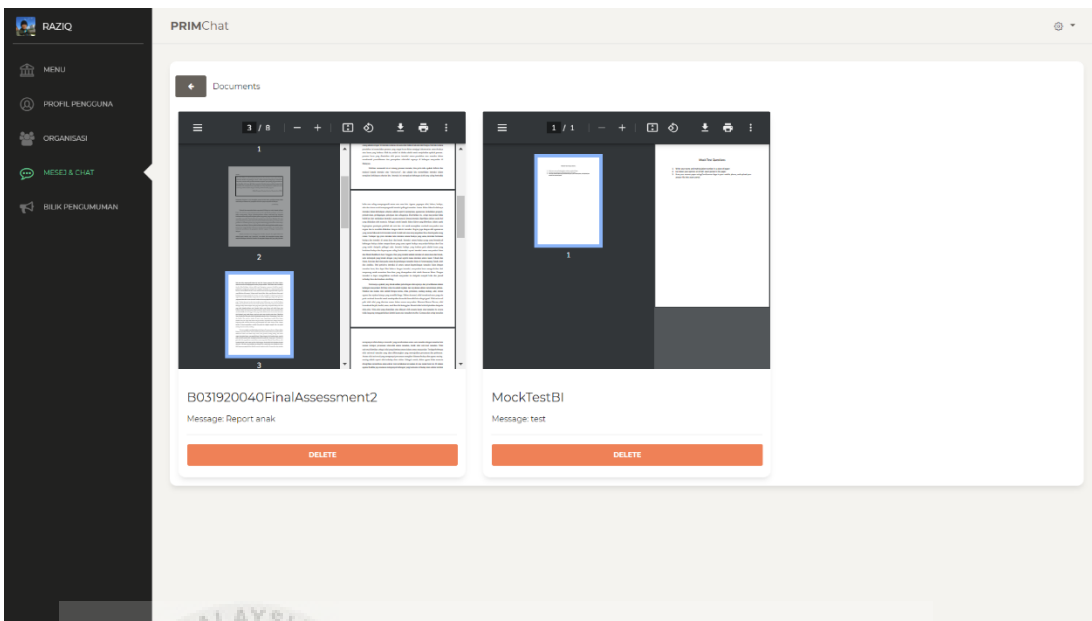
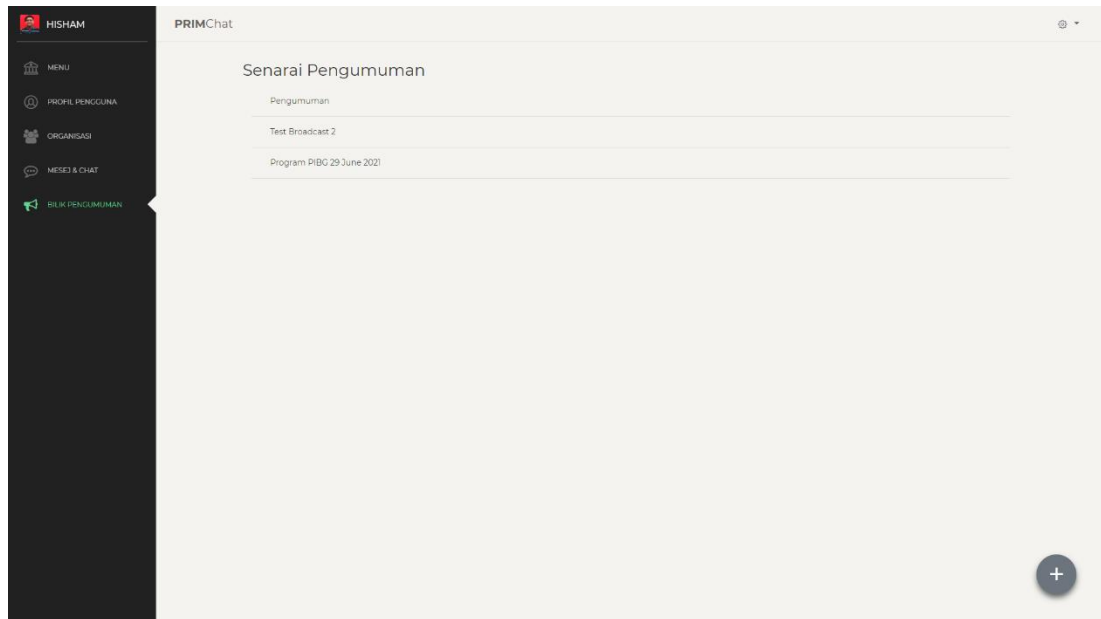


Figure 4.16: Attachment Page (e.g. document)

viii) Announcement Page



Figure 4.17: Main Announcement Page (Display Broadcast Room)

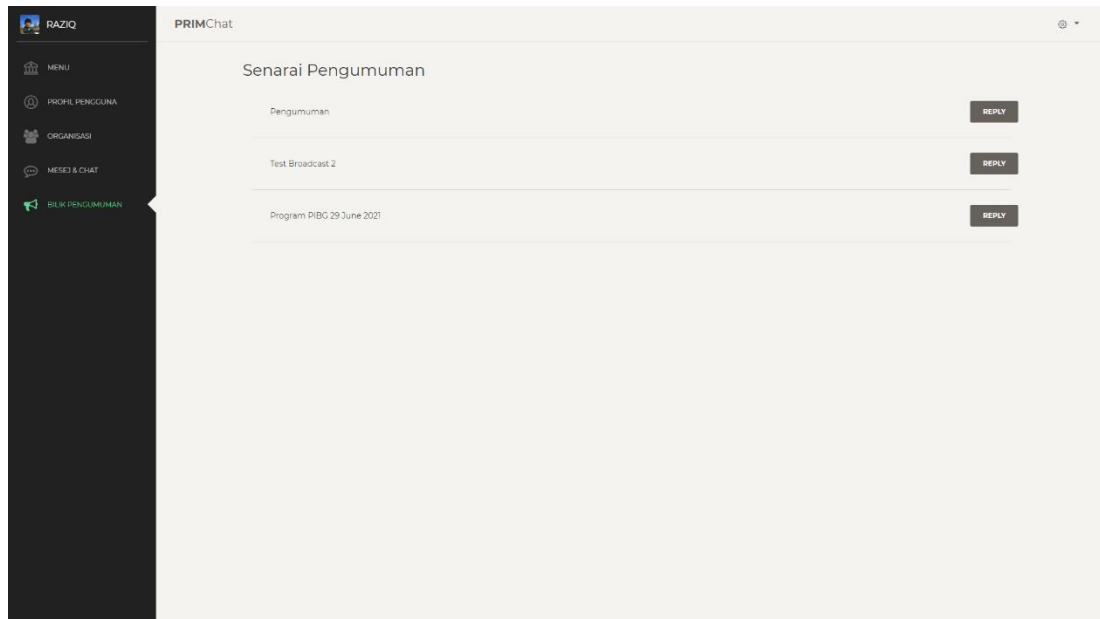


**Figure 4.19: Broadcast Message for Class**



**Figure 4.18: Create new broadcast message (Teacher)**





**Figure 4.20: Broadcast message (for Parent)**

#### 4.4 Conclusion

Finally, this chapter has covered project design, which comprises the selection of DBMS for the Physical Design, Business Rules for the Conceptual Design, Entity Relationship Diagram, Data Dictionary and Normalization for the Logical Design, and Graphical User Interface (GUI) Design.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

## CHAPTER 5: IMPLEMENTATION

### 5.1 Introduction

The major goal of this chapter is to successfully implement the previously specified database. The installation and configuration of the database will be demonstrated. In this phase, MySQL is installed on Windows 10, and Data Definition Language (DDL) and Data Manipulation Language (DML) are also implemented.

### 5.2 Software Development Environment Setup

Before constructing the PRIMChat System, the software development environment must be set up. The Web Server Application with Laragon and Nginx, with HeidiSQL as DBMS and laravel as the backend framework. The frontend is developed using standard web tool HTML, CSS and JS. These are the important tool to set up in order to develop PRIMChat System.

#### 5.2.1 Step of Installation

Step 1: Go to <https://laragon.org/download/> and choose preferred installation mode (Full & Portable).

## Edition

### Download Laragon - Full (147 MB)

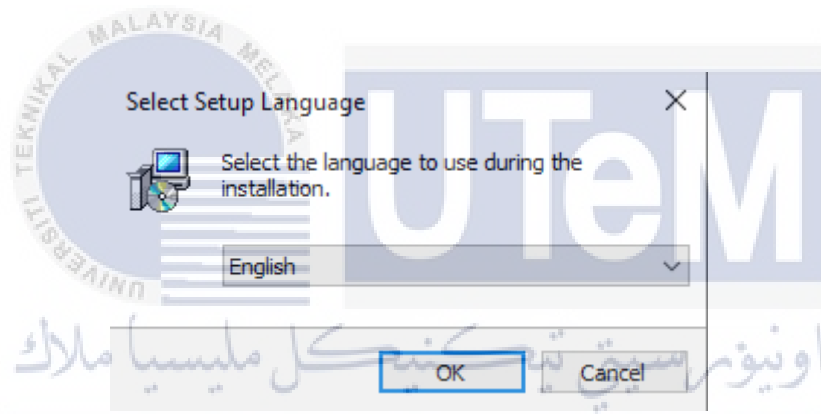
- **Laragon Full (64-bit):** Apache 2.4, Nginx, MySQL 5.7, PHP 7.4, Redis, Memcached, Node.js 14, npm, git, bitmana...

### Download Laragon - Portable (38 MB)

- **Laragon Portable:** PHP 5.4, MySQL 5.1, bitmana - Good for getting started with PHP, then you can add newer versions of PHP/MySQL easily later using "Tools > Quick add"

**Figure 5.1: Download Page**

Step 2: Double click on the downloaded .exe installation file. Choose preferred language and click OK.



**Figure 5.2: First Installation page**

Step 3: Let the default option ticked and click Next.

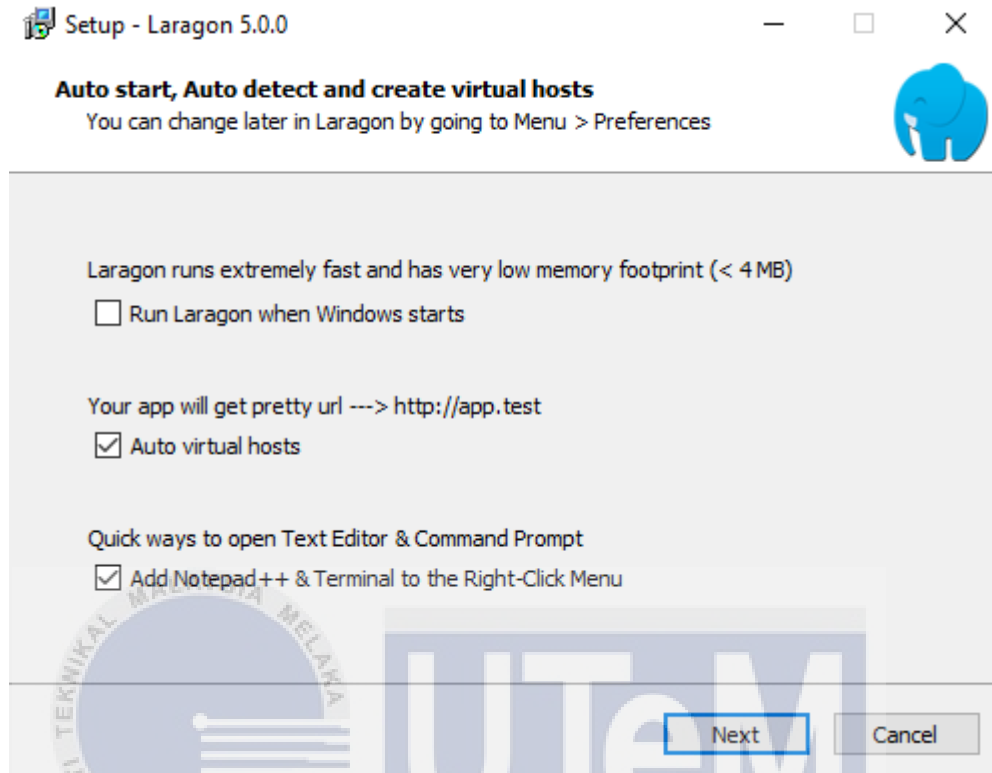
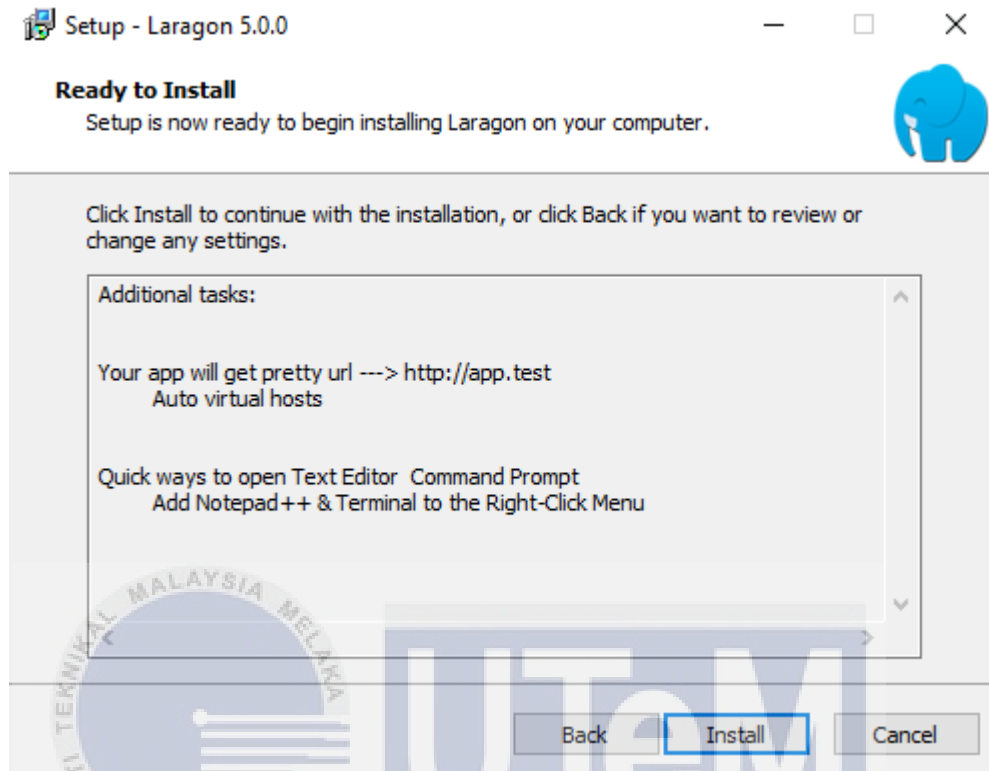


Figure 5.3: Installation Option page

Step 4: Revise the installation option summary and click Install.



**Figure 5.4: Installation Setup Summary**

Step 5: After finished installing Laragon, open the application.



Figure 5.5: Laragon main page

Step 6: You can click setting icon at top right corner to open Preference menu and setup preferred web service and DBMS.

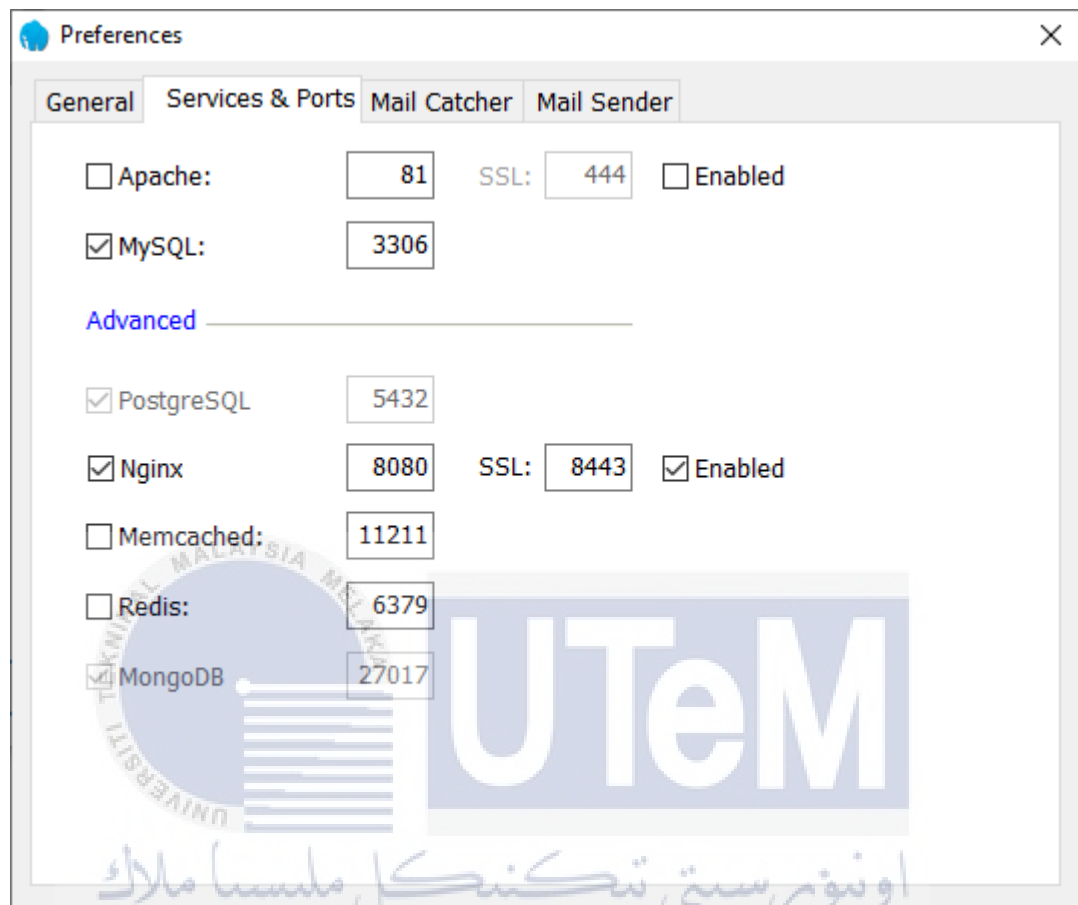
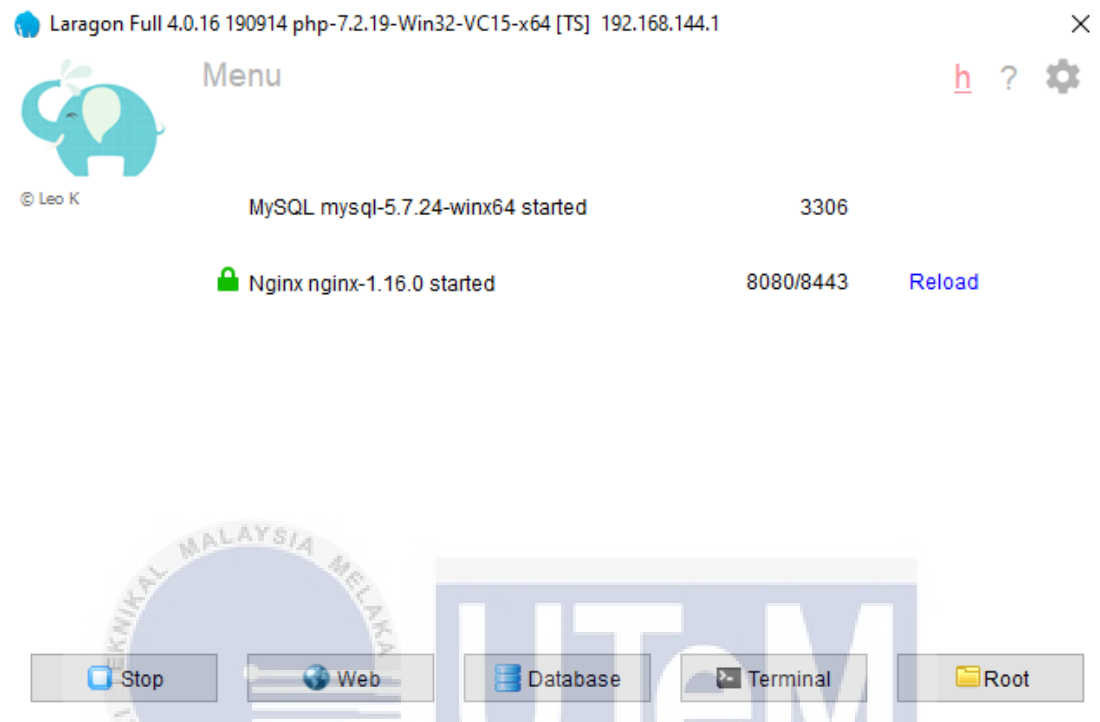


Figure 5.6: Preferences page

Step 7: Click Start All to start all the checked service, in this case it will start Nginx and MySQL. And there it goes, everything set up successfully.



**Figure 5.7: Laragon services running**

### 5.3 Database Implementation

During the database implementation phase, the database is used to conduct database queries in the system, which include simple searches, complicated queries, aggregate functions, stored procedures, and triggers. The queries' aim is to insert, retrieve, validate, and verify data in a database.

#### 5.3.1 Data Definition Language (DDL)

The SQL command that builds and manipulates tables in a relational database is referred to as Data Definition Language (DDL). DDL statements can be used to build, modify, and delete database objects such as tables, procedures, functions, and triggers for PRIMChat System.



```

1 CREATE TABLE `users` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `name` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `email` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
5   `email_verified_at` TIMESTAMP NULL DEFAULT NULL,
6   `password` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
7   `fcm_token` LONGTEXT NULL COLLATE 'utf8mb4_unicode_ci',
8   `remember_token` VARCHAR(100) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
9   `attachment_id` BIGINT(20) UNSIGNED NULL DEFAULT NULL,
10  `ppfilepath` VARCHAR(255) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
11  `role` TINYINT(3) UNSIGNED NOT NULL DEFAULT '2' COMMENT '1: Admin, 2: User',
12  `created_at` TIMESTAMP NULL DEFAULT NULL,
13  `updated_at` TIMESTAMP NULL DEFAULT NULL,
14  PRIMARY KEY (`id`),
15  UNIQUE INDEX `users_email_unique` (`email`)
16 )
17 COLLATE='utf8mb4_unicode_ci'
18 ENGINE=InnoDB
19 AUTO_INCREMENT=8
20 ;
21

```

Figure 5.8: User Table DDL

```

1 CREATE TABLE `organizations` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `name` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `email` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
5   `phoneno` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
6   `address` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
7   `city` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
8   `postcode` INT(11) NOT NULL,
9   `state` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
10  `filepath` VARCHAR(255) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
11  `created_at` TIMESTAMP NULL DEFAULT NULL,
12  `updated_at` TIMESTAMP NULL DEFAULT NULL,
13  PRIMARY KEY (`id`)
14 )
15 COLLATE='utf8mb4_unicode_ci'
16 ENGINE=InnoDB
17 AUTO_INCREMENT=3
18 ;
19

```

Figure 5.9: Organization Table DDL

```

1 CREATE TABLE `organization_roles` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `title` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `created_at` TIMESTAMP NULL DEFAULT NULL,
5   `updated_at` TIMESTAMP NULL DEFAULT NULL,
6   PRIMARY KEY (`id`)
7 )
8 COLLATE='utf8mb4_unicode_ci'
9 ENGINE=InnoDB
10 AUTO_INCREMENT=4
11 ;
12

```

Figure 5.10: Organization Role Table DDL

```

1 CREATE TABLE `organization_users` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `user_id` BIGINT(20) UNSIGNED NOT NULL,
4   `organization_id` BIGINT(20) UNSIGNED NOT NULL,
5   `role_id` BIGINT(20) UNSIGNED NOT NULL,
6   `start_date` TIMESTAMP NOT NULL,
7   `end_date` TIMESTAMP NOT NULL,
8   `status` TINYINT(1) NOT NULL,
9   `created_at` TIMESTAMP NULL DEFAULT NULL,
10  `updated_at` TIMESTAMP NULL DEFAULT NULL,
11  PRIMARY KEY (`id`)
12 )
13 COLLATE='utf8mb4_unicode_ci'
14 ENGINE=InnoDB
15 AUTO_INCREMENT=6
16 ;
17

```

Figure 5.12: Organization User Table DDL

```

1 CREATE TABLE `organization_classes` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `name` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `organization_id` BIGINT(20) UNSIGNED NOT NULL,
5   `teacher_id` BIGINT(20) UNSIGNED NOT NULL,
6   `created_at` TIMESTAMP NULL DEFAULT NULL,
7   `updated_at` TIMESTAMP NULL DEFAULT NULL,
8   PRIMARY KEY (`id`)
9 )
10 COLLATE='utf8mb4_unicode_ci'
11 ENGINE=InnoDB
12 AUTO_INCREMENT=3
13 ;
14

```

Figure 5.11: Organization Class Table DDL

```

1 CREATE TABLE `students` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `name` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `icno` BIGINT(20) UNSIGNED NOT NULL,
5   `filepath` VARCHAR(255) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
6   `created_at` TIMESTAMP NULL DEFAULT NULL,
7   `updated_at` TIMESTAMP NULL DEFAULT NULL,
8   PRIMARY KEY (`id`)
9 )
10 COLLATE='utf8mb4_unicode_ci'
11 ENGINE=InnoDB
12 AUTO_INCREMENT=6
13 ;
14

```

Figure 5.13: Student Table DDL

```

1 CREATE TABLE `parent_students` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `parent_id` BIGINT(20) UNSIGNED NOT NULL,
4   `student_id` BIGINT(20) UNSIGNED NOT NULL,
5   `role` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
6   `created_at` TIMESTAMP NULL DEFAULT NULL,
7   `updated_at` TIMESTAMP NULL DEFAULT NULL,
8   PRIMARY KEY (`id`)
9 )
10 COLLATE='utf8mb4_unicode_ci'
11 ENGINE=InnoDB
12 AUTO_INCREMENT=6
13 ;
14

```

Figure 5.14: Parent Student Table DDL

```

1 CREATE TABLE `chats` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `isGroup` TINYINT(1) NOT NULL DEFAULT '0',
4   `groupname` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
5   `created_at` TIMESTAMP NULL DEFAULT NULL,
6   `updated_at` TIMESTAMP NULL DEFAULT NULL,
7   PRIMARY KEY (`id`)
8 )
9 COLLATE='utf8mb4_unicode_ci'
10 ENGINE=InnoDB
11 AUTO_INCREMENT=15
12 ;
13

```

Figure 5.15: Chat Table DDL

```

1 CREATE TABLE `grouprooms` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `chat_id` BIGINT(20) UNSIGNED NOT NULL,
4   `user_id` BIGINT(20) UNSIGNED NOT NULL,
5   `created_at` TIMESTAMP NULL DEFAULT NULL,
6   `updated_at` TIMESTAMP NULL DEFAULT NULL,
7   PRIMARY KEY (`id`)
8 )
9 COLLATE='utf8mb4_unicode_ci'
10 ENGINE=InnoDB
11 AUTO_INCREMENT=19
12 ;
13

```

Figure 5.16: Grouproom Table DDL

```

1 CREATE TABLE `attachments` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `filepath` VARCHAR(255) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
4   `extension` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
5   `filename` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
6   `fileblob` LONGBLOB NULL,
7   `type` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
8   `created_at` TIMESTAMP NULL DEFAULT NULL,
9   `updated_at` TIMESTAMP NULL DEFAULT NULL,
10  PRIMARY KEY (`id`)
11 )
12 COLLATE='utf8mb4_unicode_ci'
13 ENGINE=InnoDB
14 AUTO_INCREMENT=43
15 ;
16

```

Figure 5.18: Attachment Table DDL

```

1 CREATE TABLE `broadcastrooms` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `groupname` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `class_id` BIGINT(20) UNSIGNED NOT NULL,
5   `created_at` TIMESTAMP NULL DEFAULT NULL,
6   `updated_at` TIMESTAMP NULL DEFAULT NULL,
7   PRIMARY KEY (`id`)
8 )
9 COLLATE='utf8mb4_unicode_ci'
10 ENGINE=InnoDB
11 AUTO_INCREMENT=2
12 ;
13

```

Figure 5.17: Broadcast Room Table DDL

```

1 CREATE TABLE `broadcastmessages` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `message` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `status` TINYINT(1) NOT NULL DEFAULT '1',
5   `broadcastroom_id` BIGINT(20) UNSIGNED NOT NULL DEFAULT '0',
6   `created_at` TIMESTAMP NULL DEFAULT NULL,
7   `updated_at` TIMESTAMP NULL DEFAULT NULL,
8   PRIMARY KEY (`id`)
9 )
10 COLLATE='utf8mb4_unicode_ci'
11 ENGINE=InnoDB
12 AUTO_INCREMENT=6
13 ;
14

```

Figure 5.19: Broadcast Message Table DDL

```

1 CREATE TABLE `messages` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `message` VARCHAR(255) NOT NULL COLLATE 'utf8mb4_unicode_ci',
4   `sender_id` BIGINT(20) UNSIGNED NOT NULL,
5   `isFile` TINYINT(1) NOT NULL DEFAULT '0',
6   `chat_id` BIGINT(20) UNSIGNED NOT NULL,
7   `attachment_id` BIGINT(20) UNSIGNED NULL DEFAULT NULL,
8   `bcm_id` BIGINT(20) UNSIGNED NULL DEFAULT NULL,
9   `type` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_unicode_ci',
10  `isStarred` TINYINT(1) NOT NULL DEFAULT '0',
11  `created_at` TIMESTAMP NULL DEFAULT NULL,
12  `updated_at` TIMESTAMP NULL DEFAULT NULL,
13  PRIMARY KEY (`id`)
14 )
15 COLLATE='utf8mb4_unicode_ci'
16 ENGINE=InnoDB
17 AUTO_INCREMENT=100
18 ;
19

```

Figure 5.21: Message Table DDL

```

1 CREATE TABLE `class_students` (
2   `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
3   `class_id` BIGINT(20) UNSIGNED NOT NULL,
4   `student_id` BIGINT(20) UNSIGNED NOT NULL,
5   `year` YEAR NULL DEFAULT NULL,
6   `created_at` TIMESTAMP NULL DEFAULT NULL,
7   `updated_at` TIMESTAMP NULL DEFAULT NULL,
8   PRIMARY KEY (`id`)
9 )
10 COLLATE='utf8mb4_unicode_ci'
11 ENGINE=InnoDB
12 AUTO_INCREMENT=6
13 ;
14

```

Figure 5.20: Class Student Table DDL

### 5.3.2 Stored Procedures

```

1 CREATE PROCEDURE `CreateChatroom` (
2   IN `groupname` VARCHAR(50),
3   IN `groupstatus` BIT
4 )
5 LANGUAGE SQL
6 NOT DETERMINISTIC
7 CONTAINS SQL
8 SQL SECURITY DEFINER
9 COMMENT 'Insert new chat and create group room.'
10 BEGIN
11
12   insert into chats(isGroup, grouprooms) values (groupstatus, groupname);
13
14 END

```

Figure 5.22: Stored procedure for CreateChatroom

```

1 CREATE PROCEDURE `CreateUser` (
2   IN `pName` VARCHAR(100),
3   IN `pEmail` VARCHAR(50),
4   IN `pPassword` VARCHAR(100)
5 )
6 LANGUAGE SQL
7 NOT DETERMINISTIC
8 CONTAINS SQL
9 SQL SECURITY DEFINER
10 COMMENT 'Insert new user'
11 BEGIN
12
13   insert into users(name, email, password) values (pName, pEmail, pPassword);
14
15 END

```

Figure 5.23: Stored procedure for CreateUser

### 5.3.3 Triggers

```

1 CREATE TRIGGER `users_before_insert` BEFORE INSERT ON `users` FOR EACH ROW BEGIN
2   set new.password = md5(new.password);
3   set new.created_at = SYSDATE();
4   set new.updated_at = sysdate();
5 END

```

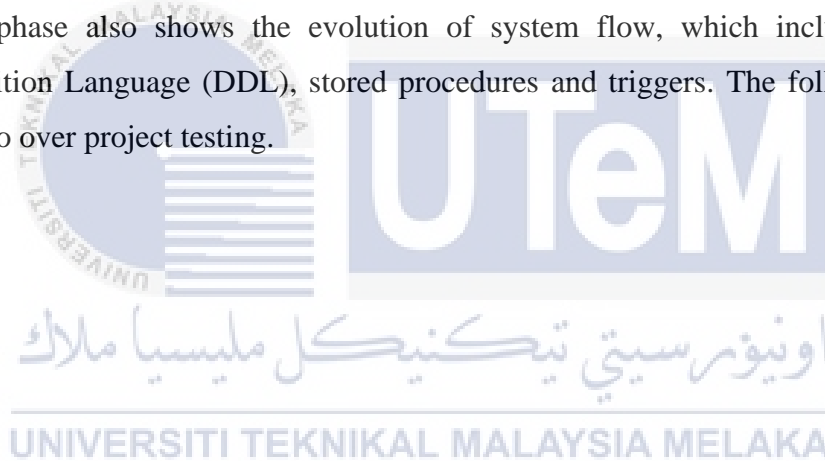
Figure 5.24: Trigger before insert users

```
1 CREATE TRIGGER `users_after_update` AFTER UPDATE ON `users` FOR EACH ROW BEGIN
2
3   set new.updated_at = sysdate();
4
5 END
```

**Figure 5.25: Trigger after update users**

## 5.4 Conclusion

Details concerning the development of the PRIMChat system have been discussed in this chapter. The setup for the software development environment, which includes methods for installing Laragon on 64-bit Windows, has been demonstrated. This phase also shows the evolution of system flow, which includes the Data Definition Language (DDL), stored procedures and triggers. The following chapter will go over project testing.



## CHAPTER 6: TESTING

### 6.1 Introduction

This chapter will go over the testing step of the approach stated in chapter two. This testing documentation includes the test plan, which is the initial study of testing on this system, the test organization, the test environment, the test schedule, the test strategy, the test design, which includes two types of testing (test description and test data), as well as the test result and analysis. The test environment is made up of hardware and software that is used to put this system through its paces. Furthermore, test strategy is crucial during this testing phase because it directs the testing methodologies to be employed.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

### 6.2 Test Plan

A test plan is a document that depicts the approach, scope, resources, and timeline of planned test exercises. It distinguishes, among other things, the test items, the highlights to be tested, the testing assignments, who will do each task, the level of analyst autonomy, the test condition, the test structure systems and section and leave criteria to be used, and the method of reasoning for their choice, as well as any dangers requiring possibility planning. It is a report on the test planning procedure.



### 6.2.1 Test Organization

The test organization in this project consists of a developer, tester and system analyst. The scopes that will be assessed include functional and non-functional requirements. Both the tester and the developer collaborate to uncover defects and problems in the application; every failure and error in the system's output is logged and rectified.

**Table 6.1: Roles and Responsibilities**

Roles	Responsibilities
Analyst	<ul style="list-style-type: none"> <li>- Identify and acquire requirements from clients.</li> <li>- Plan the optimum time to develop and deliver the project outcome.</li> <li>- Prepare necessary document and understand current problem to provide ideas to solution.</li> </ul>
Developer	<ul style="list-style-type: none"> <li>- Understand the requirement from client and system analyst.</li> <li>- Continuous developing and debugging the system.</li> <li>- Make sure all modules are working.</li> </ul>
Tester	<ul style="list-style-type: none"> <li>- Detect bugs and errors in the system.</li> <li>- Validate data input as a new-comer user.</li> </ul>

### 6.2.2 Test Environment

The test environment is made up of components that aid in test execution such as software, hardware, and network configuration that can connect more than two components set up by the developer. The test environment design must be a carbon copy of the creation condition in order to identify any condition or arrangement flaws. The hardware components used in this development are shown in Table 6.2. Table 6.3, on the other hand, displays the programmes and software computers that configure the application system and database configuration.

**Table 6.2: Test Environment for Hardware Component**

Hardware Specification	Description
Model	Desktop PC
Central Processing Unit (CPU)	Ryzen 5 2600
Graphic Processing Unit (GPU)	Nvidia GTX 1080
Display	Asus TUF VG279QM 27"
RAM	16GB
Storage	128 SSD & 1TB Hard Disk

**Table 6.3: Test Environment for Software**

Software Specification	Description
Database	HeidiSQL
Web Server	Nginx
Operating System	Windows 10
Integrated Development Environment (IDE)	Visual Studio Code 2019
Documentation	MS Word 2016

### 6.2.3 Test Schedule

A test schedule is a plan for software testing that includes the testing phases or tasks, the target start and finish dates, and responsibilities. It should also show how the test will be examined, followed, and confirmed. Table 6.4 shows the activity, testing description, start and end dates, as well as the duration of the testing.

**Table 6.4: Test Schedule in PRIMChat System**

Activities	Description	Start Date	End Date	Duration
Security Testing	Security testing is a type of programming testing that aims to expose framework vulnerabilities and ensure that the framework's information and assets are safe from prospective intruders or unauthorised users.	7/8/2021	13/8/2021	6 Days
Functionality Testing	Functionality testing is defined as a type of testing that ensures that each capacity of the product application operates in accordance with the required specification.	14/8/2021	21/8/2021	7 Days

### 6.3 Test Strategy

There are four forms of testing in Test Strategy: black box testing, white box testing, bottom-up testing, and top-down testing. Black box testing, often known as Behavioural Testing, is a software testing approach in which the tester is unaware of the internal structure, design, or implementation of the items being evaluated. This testing is mostly used to compare functional requirements against non-functional needs. White box testing, also known as Code-Based Testing or Structural Testing, is a software testing approach chosen by the tester to practise paths through the programming language and establish the right outputs from the input resources. Furthermore, bottom-up testing checks each segment at the lower progressive system separately, followed by the segments that rely on these segments. At the same time, top-down testing is a coordination testing system that is used to simulate the behaviour of lower-level modules that have not yet been integrated. However, this project solely supports black box testing techniques like as equivalence testing and boundary value analysis.

### 6.3.1 Classes of Tests

For this testing process in the PRIMChat System, two sorts of test classes are used.

#### i) Functionality Testing

This testing will ensure that each capacity of the product application operates in accordance with the specification.

#### ii) Security Testing

This testing aims to expose application vulnerabilities and ensure that the application's information and assets are protected from potential unauthorised users.

## 6.4 Test Design

Test design describes how to construct and write test suites for software testing. The goal of the test design is to ensure that the requirements specified are met in accordance with what the client requires and desires. The test description and test data are the two components of test design.

### 6.4.1 Test Description

Every module test case includes a test description that explains the test case identification, kind of testing, pre-conditions, test requirements, step procedure in each test case identification, and expected output result.

Tables 6.5 to 6.9 indicate the modules that are examined in order to acquire the best possible result.

**Table 6.5: Test Description for Registration**

Test ID	A001		
Module Name	Register User		
Description	Register new user account in the system.		
Test Case ID	Test Case	Procedure	Expected Output
A001_01	Get Started button clicked before filling Account Holder Name.	Click Get Started button without fill Name field.	Validation message “Please fill out this field”.
A001_02	Get Started button clicked before filling Email.	Click Get Started button without fill Email field.	Validation message “Please fill out this field”.
A001_03	Get Started button clicked with incorrect email format.	Click Get Started button without fill the correct format Email field.	Validation message “The @ is missing” or “The @ is incomplete”.
A001_04	Get Started button clicked before fill password.	Click Get Started button without fill password field.	Validation message “Please fill out this field”.
A001_05	Get Started button clicked before fill confirm password.	Click Get Started button without fill confirm password field.	Validation message “Please fill out this field”.
A001_06	Get Started button clicked without fill at least 8 long characters password.	Click Get Started button without fill at least 8 long characters password field.	Validation message “The password must be at least 8 characters”.
A001_07	Get Started button clicked with existing email.	Click Get Started button with existing Email.	Validation message “The email is already taken”.

A001_08	Get Started button clicked without click agree to T&C.	Click Get Started button without accept T&C agreement.	Validation message “The agree terms and conditions field is required”.
A001_09	Get Started button clicked with mismatch confirm password to password.	Click Get Started button with incorrect confirm password.	Validation message “The password confirmation does not match”.
A001_10	Get Started button clicked with perfect input.	Click Get Started button with valid data inputted into the system.	User will be automatically logged in and redirected to the main dashboard.

**Table 6.6: Test Description for Login**

<b>Test ID</b>	A002		
<b>Module Name</b>	Login User Account		
<b>Description</b>	Sign in to the system with the new created account.		
<b>Test Case ID</b>	<b>Test Case</b>	<b>Procedure</b>	<b>Expected Output</b>
A002_01	Sign In button clicked before filling email.	Click Sign In button without fill email field.	Validation message “Please fill out this field”.
A002_02	Sign In button clicked before filling Password.	Click Sign In button without fill Password field.	Validation message “Please fill out this field”.
A002_03	Sign In button clicked with incorrect email.	Click Sign In button without fill existing Email.	Validation message “These credentials do not match our records”.

A002_04	Sign In button with incorrect password.	Click Sign In button with incorrect password.	Validation message “These credentials do not match our records”.
A002_05	Sign In button clicked with correct credential.	Click Get Started button after filling valid credential.	User will be redirected to main dashboard.

**Table 6.7: Test Description for Update Profile**

<b>Test ID</b>	A003		
<b>Module Name</b>	Update User		
<b>Description</b>	Update user information/detail.		
<b>Test Case ID</b>	<b>Test Case</b>	<b>Procedure</b>	<b>Expected Output</b>
A003_01	Sah Kemaskini button clicked before filling Account Holder Name.	Click Sah Kemaskini button without fill Name field.	Validation message “Please fill out this field”.
A003_02	Sah Kemaskini button clicked before filling Email.	Click Sah Kemaskini button without fill Email field.	Validation message “Please fill out this field”.
A003_03	Sah Kemaskini button clicked with incorrect email format.	Click Sah Kemaskini button without fill the correct format Email field.	Validation message “The @ is missing” or “The @ is incomplete”.
A003_04	Sah Kemaskini button clicked before fill old password.	Click Sah Kemaskini button without fill old password field.	Validation message “Please fill out this field”.

A003_05	Sah Kemaskini button clicked before fill new password.	Click Sah Kemaskini button without fill new password field.	Validation message “Please fill out this field”.
A003_06	Sah Kemaskini button clicked before fill confirm password.	Click Sah Kemaskini button without fill confirm password field.	Validation message “Please fill out this field”.
A003_07	Sah Kemaskini button clicked with incorrect old password.	Click Sah Kemaskini button with incorrect old password.	Validation message “Current password field does not match with your password”.
A003_08	Sah Kemaskini button clicked with new password less than 8 characters.	Click Sah Kemaskini button new password less than 8 characters.	Validation message “The password must be at least 8 characters”.
A003_09	Sah Kemaskini button clicked with confirm password less than 8 characters.	Click Sah Kemaskini button with confirm password less than 8 characters.	Validation message “The password confirmation must be at least 8 characters”.
A003_10	Sah Kemaskini button clicked with mismatch confirm password.	Click Sah Kemaskini button with mismatch confirm password.	Validation message “The password confirmation does not match”.
A003_11	Save button clicked	Click Save button in edit profile	Validation message “Error!



	without provide profile picture.	picture without provide image file.	Sila Pilih Gambar Untuk Diupload”.
A003_12	Sah Kemaskini button clicked with valid email and name.	Click Sah Kemaskini button in Kemaskini Profil.	Message “Profile Updated”.
A003_13	Sah Kemaskini button clicked with all valid password.	Click Sah Kemaskini button in Ubah Kata-Laluan.	Message “Password Update”.
A003_14	Save button clicked with correct image file.	Click Save button under Profile Picture section.	Message “Berjaya!”.

**Table 6.8: Test Description for Chatting**

<b>Test ID</b>	B001		
<b>Module Name</b>	Chatting Module		
<b>Description</b>	All procedures involving chatting starting from create chat room until voice/video calling.		
<b>Test Case ID</b>	<b>Test Case</b>	<b>Procedure</b>	<b>Expected Output</b>
B001_01	Click “+” fab button to invite user to chat.	Click “+” in Mesej & Chat page.	Modal of list of users will be displayed.
B001_02	Teruskan button clicked before inviting at least 1 user.	Click Teruskan button tick the invite checkbox.	Validation message “Error! Sila pilih pengguna”.
B001_03	Search box to search for user.	Search for email or person name.	Data table will display relevant users based on search.

B001_04	Teruskan button clicked with invitation for existing chat room.	Click Teruskan button while choosing a user who already had created chat room with us.	Validation message “Error! Chat room sudah tersedia Bersama user”.
B001_05	Create chat room with teacher/parent based on student.	Click on student profile in organization classes page.	A message “Konfirmasi? Adakah anda ingin mesej Bersama guru/ibu bapa?”.
B001_06	OK button clicked with existing chatroom with the teacher/parent.	Click OK button with existing chat room already between 2 users.	Validation message “Bilik sudah tersedia”. User will be redirected to chat page.
B001_07	OK button clicked with to create new chat room for the teacher/parent.	Click OK button without existing chat room.	Successful message “Room has been created”.
B001_08	Send a text message that includes emoji, special character etc.	Fill the message field to send to the other users that consist any kind of character or emoji and click send.	Message will be appeared on the chat screen and saved in database.
B001_09	Send message with an attachment.	Hantar Lampiran button is clicked with a caption in chatting page.	Message will appear in chat screen with a link to open the sent attachment.

B001_10	Open attachment page based on category to display all attachment sent within the chat room.	Click Lampiran at top of chat screen to display dropdown of attachment category.	User will be redirected to desired attachment type.
B001_11	Star message to favorite a message.	Click star button at the right of a message.	The star icon will change color means it is successful.
B001_12	Voice call with user.	Click Hubungi button in chat screen. A notification will appear, asking permission to use mic and webcam. Then choose voice call.	User will be redirected to voice call page and will generate a voice room invitation link to give to another user.
B001_13	Video call with user.	Click Hubungi button in chat screen. A notification will appear, asking permission to use mic and webcam. Then choose video call.	User will be redirected to video call page and will generate a video room invitation link to give to another user.

**Table 6.9: Test Description for Broadcast Message**

Test ID	C001		
Module Name	Broadcast Message		
Description	Create broadcast room and broadcast message to parents.		
Test Case ID	Test Case	Procedure	Expected Output
C001_01	Create new broadcast room for a class that already had 1.	In organization class, click “+” fab button to create broadcast room.	Validation message “Bilik broadcast sudah tersedia”. User will be redirected to the page.
C001_02	Create new broadcast room for a class.	In organization class, click “+” fab button to create broadcast room.	A success message will popup “The room created successfully”.
C001_03	Broadcast a new message to the class but without content.	Go to Bilik Pengumuman and choose broadcast room. Then, click “+” fab button to broadcast new message. Click Teruskan without fill the message.	Validation message “Error!”.
C001_04	Broadcast a new message to the class.	Go to Bilik Pengumuman and choose broadcast room. Then, click “+” fab button to broadcast new message. Click Teruskan.	Broadcast message successfully created.

C001_05	Reply to broadcast message.	Click on a reply button beside the broadcast message to reply.	User will be redirected to the chat page with a popup modal to reply towards the broadcast message.
---------	-----------------------------	--	---

#### 6.4.2 Test Data

To assure the task of test description in Table 6.5 until Table 6.9, test data must be studied for each predicted result shown in Table 6.10. Tables 6.10 through 6.14 will display test data for different test descriptions.

**Table 6.10: Test Data for Registration**

<b>Test ID</b>	A001
<b>Module Name</b>	Register User
<b>Test Case ID</b>	A001_01
<b>Test Case</b>	Get Started button clicked before filling Account Holder Name.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty Name field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_02
<b>Test Case</b>	Get Started button clicked before filling Email.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty Email field.
<b>Role</b>	Parent and Teacher (User)

<b>Test Case ID</b>	A001_03
<b>Test Case</b>	Get Started button clicked with incorrect email format.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Incorrect email format e.g. asdasd@
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_04
<b>Test Case</b>	Get Started button clicked before fill password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty password field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_05
<b>Test Case</b>	Get Started button clicked before fill confirm password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty password confirmation field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_06
<b>Test Case</b>	Get Started button clicked without fill at least 8 long characters password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Password less than 8 characters.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_07
<b>Test Case</b>	Get Started button clicked with existing email.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Enter email that already exist in the system.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_08
<b>Test Case</b>	Get Started button clicked without click agree to T&C.

<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Unchecked T&C agreement checkbox.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A001_09
<b>Test Case</b>	Get Started button clicked with mismatch confirm password to password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Mismatch password confirmation to password field.
<b>Role</b>	Parent and Teacher (User)

**Table 6.11: Test Data for Login**

<b>Test ID</b>	A002
<b>Module Name</b>	Login User Account
<b>Test Case ID</b>	A002_01
<b>Test Case</b>	Sign In button clicked before filling email.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty Email field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A002_02
<b>Test Case</b>	Sign In button clicked before filling Password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty Password field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A002_03
<b>Test Case</b>	Sign In button clicked with incorrect email.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Enter email that has not been registered yet.

<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A002_04
<b>Test Case</b>	Sign In button with incorrect password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Incorrect password to access account.
<b>Role</b>	Parent and Teacher (User)

**Table 6.12: Test Data for Update Profile**

<b>Test ID</b>	A003
<b>Module Name</b>	Update User
<b>Test Case ID</b>	A003_01
<b>Test Case</b>	Sah Kemaskini button clicked before filling Account Holder Name.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty Name field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_02
<b>Test Case</b>	Sah Kemaskini button clicked before filling Email.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty Email field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_03
<b>Test Case</b>	Sah Kemaskini button clicked with incorrect email format.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Incorrect email format e.g. asdasd@
<b>Role</b>	Parent and Teacher (User)



<b>Test Case ID</b>	A003_04
<b>Test Case</b>	Sah Kemaskini button clicked before fill old password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty old password field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_05
<b>Test Case</b>	Sah Kemaskini button clicked before fill new password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty new password confirmation field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_06
<b>Test Case</b>	Sah Kemaskini button clicked before fill confirm password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty confirm password field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_07
<b>Test Case</b>	Sah Kemaskini button clicked with incorrect old password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Incorrect old password field to current account password.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_08
<b>Test Case</b>	Sah Kemaskini button clicked with new password less than 8 characters.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	New password less than 8 characters.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_09

<b>Test Case</b>	Sah Kemaskini button clicked with confirm password less than 8 characters.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Confirm password less than 8 characters.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_10
<b>Test Case</b>	Sah Kemaskini button clicked with mismatch confirm password.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Mismatch password confirmation to password field.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	A003_11
<b>Test Case</b>	Save button clicked without provide profile picture.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Empty image file during save changes.
<b>Role</b>	Parent and Teacher (User)

Table 6.13: Test Data for Chatting

<b>Test ID</b>	B001
<b>Module Name</b>	Chatting Module
<b>Test Case ID</b>	B001_01
<b>Test Case</b>	Click “+” fab button to invite user to chat.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	In Mesej & Chat page upon clicking + button.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_02
<b>Test Case</b>	Teruskan button clicked before inviting at least 1 user.
<b>Input Field</b>	<b>Test Data</b>

<b>Situation</b>	Did not tick at least 1 user in the invite checkbox to create room.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_03
<b>Test Case</b>	Search box to search for user.
<b>Input Field</b>	<b>Test Data</b>
<b>Search Field</b>	“ahmadraziqdanish@gmail.com” or “Hisham”
<b>Test Case ID</b>	B001_04
<b>Test Case</b>	Teruskan button clicked with invitation for existing chat room.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Create a new chat room while already exist with invited user.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_05
<b>Test Case</b>	Create chat room with teacher/parent based on student.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Teacher or parent trying to find each other through students.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_06
<b>Test Case</b>	OK button clicked with existing chatroom with the teacher/parent.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	A chat room between the teacher and parent already existed.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_07
<b>Test Case</b>	OK button clicked with to create new chat room for the teacher/parent.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Create chat room from student page.
<b>Role</b>	Parent and Teacher (User)

<b>Test Case ID</b>	B001_08
<b>Test Case</b>	Send a text message that includes emoji, special character etc.
<b>Input Field</b>	<b>Test Data</b>
Message	Hi! 😊
<b>Test Case ID</b>	B001_09
<b>Test Case</b>	Send message with an attachment.
<b>Input Field</b>	<b>Test Data</b>
Message	Picture
File	Image.png
<b>Test Case ID</b>	B001_10
<b>Test Case</b>	Open attachment page based on category to display all attachment sent within the chat room.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Click lampiran button at the top to display a dropdown of attachment type to view.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_11
<b>Test Case</b>	Star message to favourite a message.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Save or favourite a message.
<b>Role</b>	Parent and Teacher (User)
<b>Test Case ID</b>	B001_12
<b>Test Case</b>	Voice call with user.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Create voice call room will result a link being generated and can distribute it towards another user.
<b>Role</b>	Parent and Teacher (User)

<b>Test Case ID</b>	B001_13
<b>Test Case</b>	Video call with user.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Create video call room will result a link being generated and can distribute it towards another user.
<b>Role</b>	Parent and Teacher (User)

**Table 6.14: Test Data for Broadcast Message**

<b>Test ID</b>	C001
<b>Module Name</b>	Broadcast Message
<b>Test Case ID</b>	C001_01
<b>Test Case</b>	Create new broadcast room for a class that already had 1.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	When teacher try to create broadcast room for class that already have 1.
<b>Role</b>	Teacher (User)
<b>Test Case ID</b>	C001_02
<b>Test Case</b>	Create new broadcast room for a class.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Create new broadcast room for class.
<b>Role</b>	Teacher (User)
<b>Test Case ID</b>	C001_03
<b>Test Case</b>	Broadcast a new message to the class but without content.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	No message content to broadcast upon submit form.
<b>Role</b>	Teacher (User)
<b>Test Case ID</b>	C001_04

<b>Test Case</b>	Broadcast a new message to the class.
<b>Input Field</b>	<b>Test Data</b>
Message	We have PIBG meeting mid-year.
<b>Test Case ID</b>	C001_05
<b>Test Case</b>	Reply to broadcast message.
<b>Input Field</b>	<b>Test Data</b>
<b>Situation</b>	Parent who has inquiry for the broadcast may reply to the teacher who broadcasted the message.
<b>Role</b>	Parent (User)

## 6.5 Test Result and Analysis

The test result is crucial to every testing since it shows the actual result and compares it to the expected result that is tested in the test description to determine the progress of this testing. Tables 6.15 to 6.19 provide the identification test case and real outcome, as well as the status of passing or failing during the implementation of this testing. The testing for Table 6.15 demonstrates that all feasible scenarios for registering a new user are covered.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**Table 6.15: Test Result for Registration**

<b>Test ID</b>	A001	
<b>Module Name</b>	Register User	
<b>Test Case ID</b>	<b>Result</b>	<b>Status (Pass/Fail)</b>
A001_01	“Please fill this field” message	Pass
A001_02	“Please fill this field” message	Pass
A001_03	“The @ is missing” or “The @ is incomplete” message	Pass
A001_04	“Please fill this field” message	Pass
A001_05	“Please fill this field” message	Pass

A001_06	“The password must be at least 8 characters” message	Pass
A001_07	“The email already taken” message	Pass
A001_08	“The agree terms and conditions field is required” message	Pass
A001_09	“The password confirmation does not match” message	Pass
A001_10	Redirect to main dashboard.	Pass

**Table 6.16: Test Result for Login**

<b>Test ID</b>	A002	
<b>Module Name</b>	Login User Account	
<b>Test Case ID</b>	<b>Result</b>	<b>Status (Pass/Fail)</b>
A002_01	“Please fill this field” message	Pass
A002_02	“Please fill this field” message	Pass
A002_03	“These credentials do not match our records” message	Pass
A002_04	“These credentials do not match our records” message	Pass
A002_05	Redirected to main dashboard	Pass

**Table 6.17: Test Result for Update Profile**

<b>Test ID</b>	A003	
<b>Module Name</b>	Update Profile	
<b>Test Case ID</b>	<b>Result</b>	<b>Status (Pass/Fail)</b>
A003_01	“Please fill this field” message	Pass
A003_02	“Please fill this field” message	Pass
A003_03	“The @ is missing” or “The @ is incomplete” message	Pass
A003_04	“Please fill this field” message	Pass

A003_05	“Please fill this field” message	Pass
A003_06	“Please fill out this field” message	Pass
A003_07	“Current password field does not match with your password” message	Pass
A003_08	“The password must be at least 8 characters” message	Pass
A003_09	“The password confirmation must be at least 8 characters” message	Pass
A003_10	“The password confirmation does not match” message	Pass
A003_11	“Error! Sila Pilih Gambar Untuk Diupload” message	Pass
A003_12	“Profile Updated” message	Pass
A003_13	“Password Updated” message	Pass
A003_14	“Berjaya!” message	Pass

**Table 6.18: Test Result for Chatting**

<b>Test ID</b>	B001	
<b>Module Name</b>	Chatting Module	
<b>Test Case ID</b>	<b>Result</b>	<b>Status (Pass/Fail)</b>
B001_01	List of users in a modal form.	Pass
B001_02	“Error! Sila pilih pengguna” message	Pass
B001_03	Display all users based on filtered search (Email/Name).	Pass
B001_04	“Error! Chat room sudah tersedia Bersama user” message	Pass
B001_05	“Konfirmasi? Adakah anda ingin mesej Bersama guru/ibu bapa?” message	Pass
B001_06	“Bilik sudah tersedia” message	Pass
B001_07	“Room has been created” message	Pass



B001_08	Message stored in database	Pass
B001_09	A message with attachment link appeared	Pass
B001_10	Redirect to attachment page	Pass
B001_11	Star icon change color based on star status	Pass
B001_12	Users can hear each other voices in the voice call	Pass
B001_13	Users can see each other live streaming via web cam panel	Pass

**Table 6.19: Test Result for Broadcast Message**

Test ID	C001	Module Name	Broadcast Message Module
Test Case ID	Result	Status (Pass/Fail)	
C001_01	“Bilik broadcast sudah tersedia” message	Pass	
C001_02	“The room created successfully” message	Pass	
C001_03	“Error!” message	Pass	
C001_04	“Broadcast message posted” message	Pass	
C001_05	Redirected to chatting page with popup modal to reply the broadcast message.	Pass	

## 6.6 Conclusion

The purpose of this chapter is to reveal the strategies used to confirm and approve the system and its functionality. The test plan includes tests that examine modifications made to the system. The test plan is used to inspect work done in the system to avoid developing a system that does not fulfil the needs and desires of the consumers. The test organisation will describe who is in charge of this project, including the developer. The test environment is the platform on which the programme is run, and it includes the hardware and software that are utilised to run the system throughout the testing period. The test schedule is a timetable that is created to coordinate each individual testing period. The test strategy is the approach used to test the system by employing multiple methodologies to address two major areas of the system, in this example, the functionality and security of the system. Meanwhile, the test description is the projected possible outcomes of the above-mentioned testing technique. Simultaneously, the test data is the data that is entered into the system for testing purposes. Finally, the test findings and analysis are the gathering of data from the tests performed as well as the analysis of the obtained data in relation to the predicted outcome and reaction.

Finally, the last following chapter will go through the project's completion and the lessons learnt.

## **CHAPTER 7: PROJECT CONCLUSION**

### **7.1 Introduction**

This chapter will analyse the overall performance of the PRIMChat System, including an analysis of the system's strengths and limitations as well as recommendations for improvements based on the analysis. Furthermore, the project's contribution will be defined in this chapter.

### **7.2 Observation on Strengths and Weaknesses**

Every system has its own set of strengths and flaws. This section demonstrates the advantages and disadvantages of the PRIMChat System.

#### **7.2.1 Strengths**

**i) Medium of Communication**

Communication between 2 users (Parent and Teacher) became easier over the internet.

**ii) Easy to use**

Navigations inside the system are not complicated and easy to understand.

**iii) Important for user satisfactory**

With chatting module, current PRIM system will be able to connect with their users or between users to perform transaction better using the system.

**iv) Cross-platform**

User will be able to use the system using any devices as long as it has internet connection and web browser.

### 7.2.2 Weaknesses

**i) Not Integrated with Main System**

The system is not integrated with the main system yet as the status still in development

**ii) Limitation with the web browser version**

There are several bugs encountered when using older version of web browser or others that do not support certain services like adobe flash or webcam/microphone permission. Thus, users must always use latest and relevant version of web browser.

### 7.3 Propositions for Improvement

The PRIMChat System has a lot of room for improvement. To begin, in the future, improving programming technique by adding comments may help new programmers to read and learn the code easier and simplify the coding may help reduce disk storage for the system file. Following that, a mobile application also might be a good idea to propose as most of chatting application use mobile environment. Finally, the user interface can be improved to make it more interesting.

## 7.4 Project Contribution

The web application PRIMChat System is meant to include additional functions of text messaging and voice/video calling for existing system called PRIM. First, the project is important to develop a communication medium for user to connect with each other over the internet within the system. Secondly, it is also intended to make things easier for teacher or parent to monitor student in school. Finally, this project should be useful for current PRIM users as it will elevate their satisfaction for the current system.

## 7.5 Conclusion

To conclude, the purpose and the scope set out in Chapter 1. PRIMChat System has achieved all the specified goals, which are to provide communication for users within the system and make things easier for schoolteacher and parent to use the system. The System Development Life Cycle (SDLC) has been chosen as development approach that encompasses planning, analysis, design, execution, maintenance. The system's faults and defects were also detected during the maintenance period. Finally, the PRIMChat System meets the non-functional and functional requirements successfully. But the technology remains faulty and needs to be adapted for usage in the future. At the conclusion, this system was effectively constructed and the requirement of the Bachelor of Computer Science (Database Management) has been met.

## REFERENCES

Alessia Scott (2020). SMS Vs. Text: What Is the Difference? Taken from

<https://blog.textedly.com/sms-vs-text>

Berkeley Library, University of California (2020). How to Write a Good

Documentation. Taken from <https://guides.lib.berkeley.edu/how-to-write-good-documentation>

Firestore Admin SDK for PHP Documentation (2021). Taken from [https://firebase-](https://firebase-php.readthedocs.io/en/5.x/)

[php.readthedocs.io/en/5.x/](https://firebase-php.readthedocs.io/en/5.x/)

Google Firebase Documentation (2021). Taken from

<https://firebase.google.com/docs/>

Heidi Hemmer (2009). Impact of Text Messaging on Communication. Mankato:

Minnesota State University.

Laravel 7.x Official Documentation (2021). Taken from

<https://laravel.com/docs/7.x/readme>

WhatsApp Business API Documentation (2021). Taken from

<https://developers.facebook.com/docs/whatsapp>