SOUTHERN
UNIVERSITY COLLEGE
南方大学学院

BTIS3204
FINAL YEAR PROJECT

Meeting Minute System

Lim G Wei (B200262C)

Supervisor: Noor Fatihah Bt Mazlam

11/1/2022

Department of Computer Science
Faculty of Engineering and Information Technology

Contents

DECLARATION OF COPYRIGHT

I hereby declare that this FYR/dissertation/thesis entitled **“Meeting Minute System”** to be kept by Southern University College subject to the regulations as follows:

1. This FYR/dissertation/thesis is a property of Southern University College.

2. Southern University College or its library has the right to make and transmit copies (print or electronic) of this FYR/dissertation/thesis for academic purposes.

3. Southern University College or its library has the right to store this FYR/dissertation/thesis in a retrieval system and supply copies of this FYR/dissertation/thesis if requested by other universities and research libraries.

Signature :

Name : Lim G Wei

ID No. : B200262C

Date : 6 January 2022

Abstract

Meeting minutes are notes that are recorded during a meeting. They highlight the key issues that are discussed, ideas proposed or voted on, and activities to be undertaken. The minutes of a meeting are usually taken by a designated member of a group. Their task is to provide an accurate record of what transpired during the meeting. Once the person done writing the minutes, he will share the document with the meeting attendees. It is very insufficient when most of the work is highly similar each time. By using Minutes Meeting System, it helps the process by automation the process flow. The Minutes Meeting System helps the process with decrease the step of creating docx and add reference to the pdf file. It can save a lot of time compare to traditional minutes meeting. Besides that, this system also allows to store record so that the record won't lose easily.

Acknowledgements

I am grateful because managed to finish the project within the time given by our lecturer. The system cannot be completed without the effort and co-operation with the lectures. I also sincerely thanks to the friends that giving me review on my system on user acceptance test for future improvements.

1. Chapter One: System Introduction

1.1 Introduction

My system is about creating a minute meeting system so that secretary can reduce their job. Besides that, people could also join meeting and viewpoint created by secretary in our system too.

1.2 Background Research

From book "The Importance of Corporate Minutes" by Robert, it states that minutes of meeting are the records of the meetings. The information included agenda discussed, attendance, responses and decisions make during the meetings.

Nowadays, most of the secretary is still use docx file to create minute document for each of the meeting. The secretary must start from writing down each agenda and write key points for each agenda and paste each attachment into the word to generate as pdf file. It is very insufficient when most of the work is highly similar each time. Figure below shows that the workflow of how meeting minute is done.

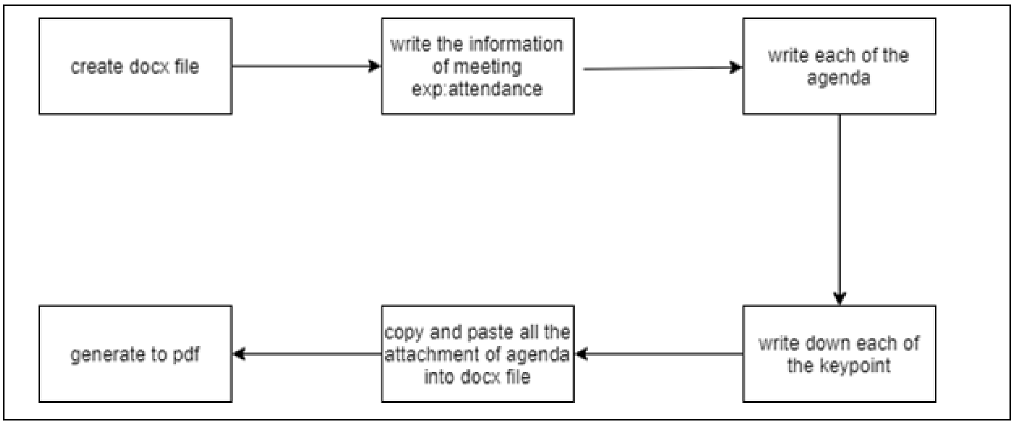

create docx file
write the information of meeting exp:attendance
write each of the agenda
write down each of the keypoint
copy and paste all the attachment of agenda into docx file
generate to pdf

Figure 1.1: Example of the flow of how minute documentation is formed

With my system, it can help them to reduce their jobs by automation the flow of some part.

1.3 User Role Involved

Role name	People involved	Description
Admin of group	Secretary	People who use the application to create the group and create meeting
Member	Normal worker which is not group member	People who create agenda for discuss in meeting and assign action to.

Table 1.1: The user role involved and their description

1.4 Objectives

The objectives of this project defined as below:

i. To study the requirements for building and enhancing the meeting minutes online system.

ii. To design and develop the proposed meeting minutes online system.

iii. To test and evaluate the proposed system.

1.5 Scope

The scopes of the project are defined as below:

- Potential Users: The potential users should be everyone who need a meeting app.
- Platform: Web app for writing key points and mark attendance and download pdf
- Framework used: Bootstrap, Laravel

1.6 Thesis organisation

This thesis consists of six chapters. Chapter One discuss the problem background of the study, project objectives, project scopes, user role involved and thesis organisation for this project.

Chapter Two discuss about the literature review. It will review the current framework use and the programming language we use to develop our system. Then, it will also discuss about what are the features of current market meeting system to learn from them and enhance my project.

Chapter Three explain on the methodology for this project. It will explain about the model we use during the whole system. Chapter Four will discuss about the design of whole system. It helps us to be more understand about how the flow works.

Chapter Five is about the testing of the system. It includes test case and make us more understand about what the testing has been done. Chapter Six is about the evaluation of the system. Chapter Seven is about the conclusion of system and future's hope.

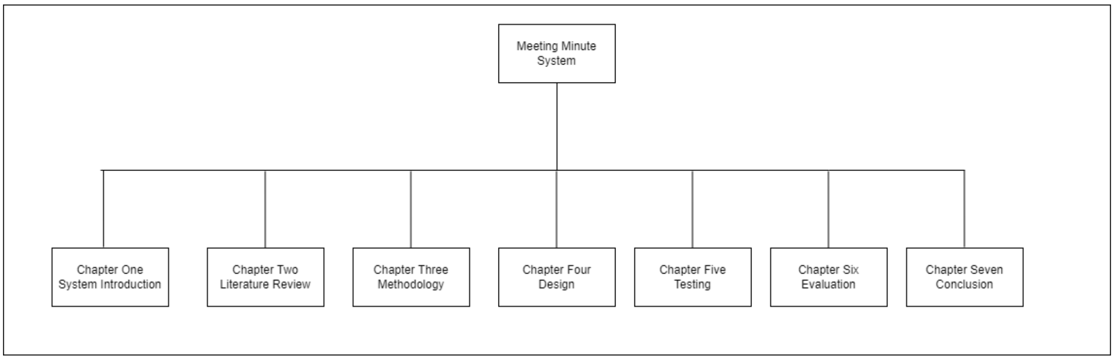


Figure 1.2: The thesis organisation of report

2. Chapter Two: Literature Review

2.1 Introduction

Current minute meetings are really inconvenience and hard to store since it is still built-in traditional way. People need to write everything down and generate it to pdf by themselves. It is really wasting time and money when we can use those time to create a better software.

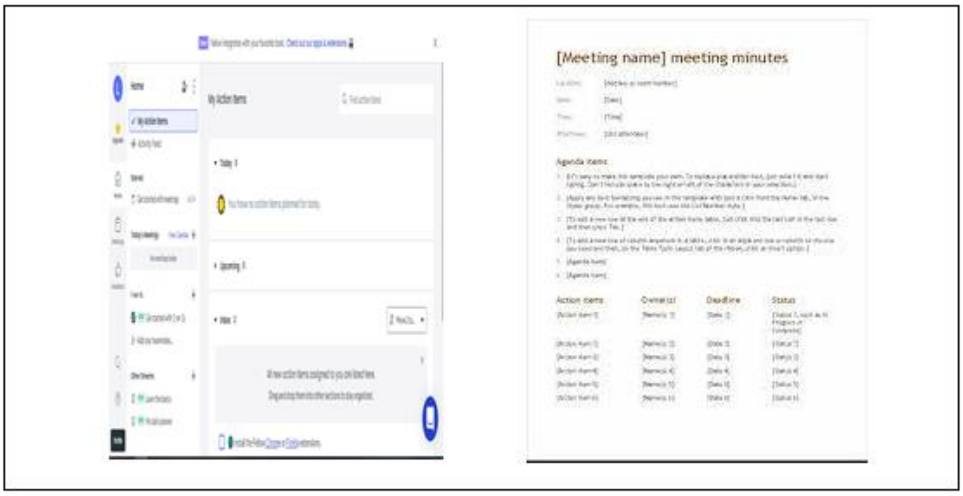
The purpose of meeting minute system is to help to reduce the current workflow of the system. It brings many advantages such as saving time and cost. It is because it saves the cost of buying new papers and do note recording every time. Then, it also decreases the space use compare to traditional minute meetings since there is no paper required. Next, it makes the whole work become tidier and neater and hard to lost. For example, in traditional minute meetings, losing of any paper would be a big problem since every file are hard to find. In minute meeting system, everything is recorded so they would not be any chance of losing files since everything is in system.

There are solutions given out by the other application current exist online which are Fellow, Meetingking and Hypercontext.

2.1.1 Fellow

Fellow is developed in 2007 by Tom Colemon. Fellow focus on more effective meetings able to show up prepared scheduled. People can also make collaboration on notes given. It can also record action items since it is the most important function of meeting minuet system to understand objective of meeting. Last, it can send notification to people in same group to avoid forgetting the meeting time.

After creates the workspace in Fellow, it allows us to add every points inside, the points can be existing in two ways, which are point form or action items. Point allows people to understand the main discussion in meeting while action items act as to do list. It can be assigned to someone and set due date for each action items. This increases the flexibility of system and the use of system. Still, they are some disadvantages. For example, this system can't generate PDF format file for future paperwork. This system more work as a note sharing system instead of meeting. Still this system got something are useful such as inviting people to join the group.



[Meeting name] meeting minutes

[Date]

[Time]

[List attendees]

Agenda items

5. [Agenda item]
6. [Agenda item]

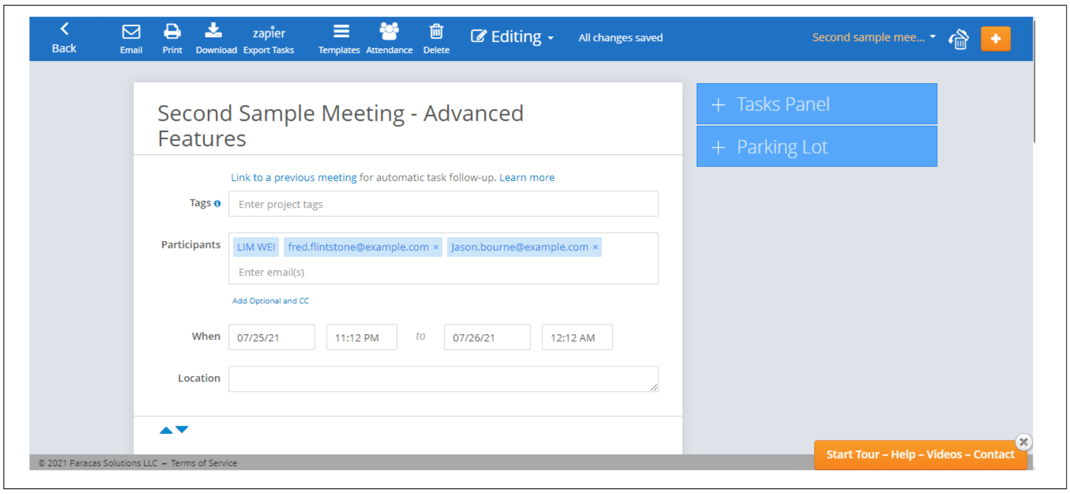
Action items	Owner(s)	Deadline	Status
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

Figure 2.1: The interface between fellow app and real paper of meeting minute

2.1.2 MeetingKing

When create meetings, it allows us to fill in some important information such as the participants for each time, the time used for each meeting and the location of meetings. After that, it will require topic for each of the agenda. Each point of agenda would be writing down inside each topic and tidy up with subtopic mode. It can also integrate with google calendar for easier manage for meeting. It has separated previous meeting and upcoming meeting which is clear for user.

This system is useful reference to create my system as many functions are fulfill for the meetings. Still, there are some weaknesses. First, the participants must be fill in for each meeting. Maybe it is flexible enough for each meeting, but it increases the work of secretary each time. This function can be abandoned since each meeting mostly have the same person. Then, it can’t assign task to anyone. It is a big problem when we try to assign task to anyone during agenda discussion. So, my system would try to improve this kind of problem.



Second Sample Meeting - Advanced Features

Link to a previous meeting for automatic task follow-up. Learn more

Tags: Enter project tags

Participants: LIM WEI, fred.flintstone@example.com ×, Jason.bourne@example.com ×

Enter email(s)

Add Optional and CC

When: 07/25/21 11:12 PM to 07/26/21 12:12 AM

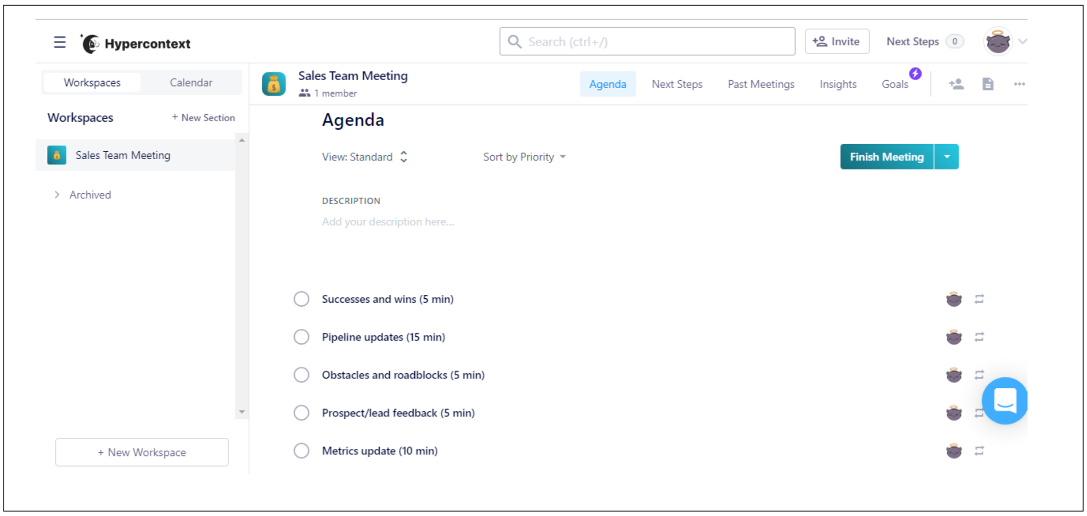
Location:

Figure 2.2: The interface of MeetingKing

2.1.3 Hypercontext

At first, it also needs to create agenda title first, then it needs to add new items inside the agenda title. Every point is act as checkbox inside Hypercontext. It allows to integrate with Google Calendar, Microsoft Teams, and Slack so that we can streamline workflow and process with ease.

Still, there are some disadvantages in the system. First, it can’t make attendance list for each of the meeting. Then, the point exist for each title can only act as action items to be done but not in point form. Next, all the action items can’t be assigned to someone else and only can show who added the action items.



Agenda

○ Successes and wins (5 min)

○ Pipeline updates (15 min)

○ Obstacles and roadblocks (5 min)

○ Prospect/lead feedback (5 min)

○ Metrics update (10 min)

Figure 2.3: The interface of Hypercontext

2.2 Framework: Bootstrap, Laravel

From TechTerms, framework means a platform for developing software applications. It provides a foundation on which software developers can build programs for a specific platform. A framework may also include code libraries, a compiler, and other programs used in the software development process.

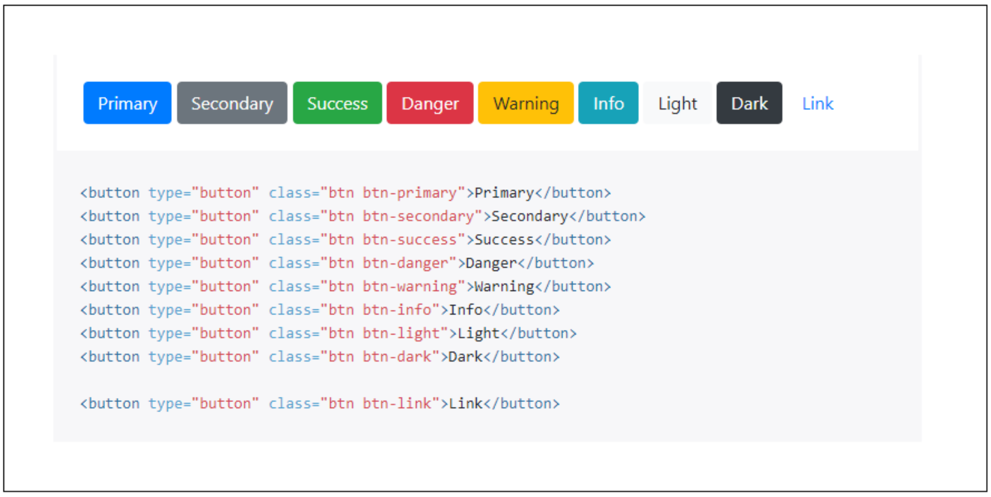
The frameworks help people to decrease developing cost and time. It is because Framework has built out a prefix function or user interface function for use. People can save time compare to construct a whole project from start. Plus, most of the functions inside are applicable in real situation. It helps people to understand the real logic in markets for people who are still persuade studying. It makes people become more professional and formal. The way of applying framework can be difference according to type of framework such as apply framework through CDN or apply framework through command code.

2.2.1 Bootstrap

From Croydon Council, Bootstrap is the most popular CSS framework for responsive and mobile-first websites, an open-source toolkit for developing with HTML, CSS, and JS. Bootstrap uses Sass variables and mixings for theming, responsive grid system for layout, pre-built components for design patterns, and JS plugins for user interaction.

With using Bootstrap, it can decrease a lot of work while trying to construct the system. It saves time to write CSS and build class for each CSS column. For example, btn primary can be apply to button to change the button to blue background white colour fonts. It makes whole system construction become clear and nice.

If we want to apply Bootstrap in our system, we need to find CDN for bootstrap and apply them in each of our pages. After applying CDN, we can use every function inside the system such as class=" btn btn-primary" inside button tags.



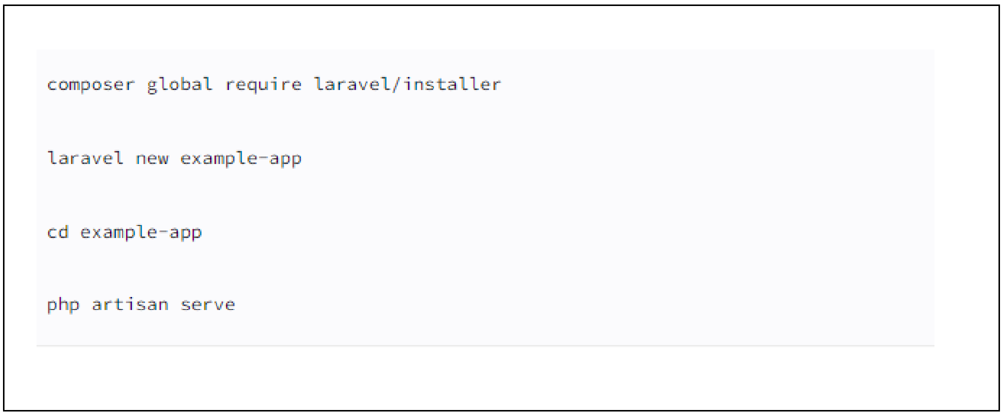
```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Figure 2.4: The way to apply Bootstrap button classes

2.2.2 Laravel

From Laravel official site, Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model-view-controller (MVC) architectural pattern and based on Symfony.

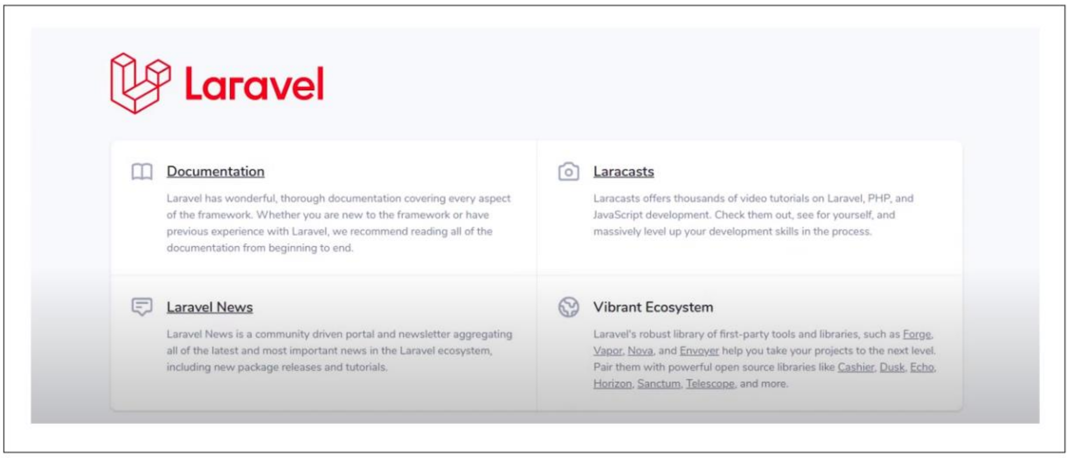
There is prerequisite condition which is you must learn the basic and logic of PHP before using Laravel. It helps to understand the function inside Laravel before trying to use it. If we want to use Laravel inside our computer (OS: Windows), first we need to install the Laravel inside our system. We need to open command and type in “composer global require laravel/installer” to install Laravel inside computer. After installed, we can create new Laravel project via “laravel new example-app” command to create a project with name “example-app”.



```
composer global require laravel/installer

laravel new example-app

cd example-app

php artisan serve
```

Figure 2.5: The process to install Laravel inside computer

After installed are done, we can find our system on server 127.0.0.1:8000



Laravel

Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as Forge, Vapor, Nova, and Envoyer help you take your projects to the next level. Pair them with powerful open source libraries like Cashier, Dusk, Echo, Horizon, Sanctum, Telescope, and more.

Figure 2.6: The user interface after Laravel’s project has been created.

2.3 Database

From oracle official website, database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

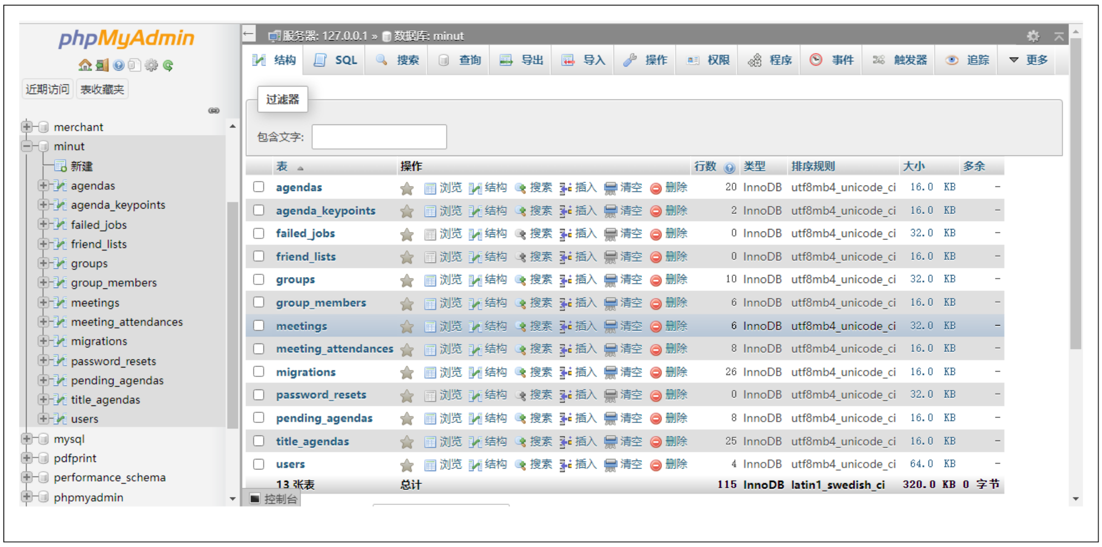
Database helps to stored data and find the way to store them. It is very important for any dynamic websites since each of them require data to run. It can also modify data so that the system can be dynamically change with coding implemented. For example, change the quantity of items in project with MySQL commands.

Before creating a database, we are required to create an ER-diagram to understand the flow works. Without creating a diagram, it is easy to be lost in nowhere since everything is unorganized. After creating ER-diagram, we can start creating our database according to which database we want to use.

2.3.1 MySQL

From MySQL official website, MySQL is a databased system that used on the web and runs on a server. It is very fast, reliable, and easy to use. MySQL is developed, distributed, and supported by Oracle Corporation. It is named after co-founder Monty Widenius's daughter: My.

MySQL can easily modify data inside database and easy to learn. The database can be modified through SQL command such as "SET quantity=2 where id=1" inside the function. It also can create database inside Xampp when Xampp is installed or downloaded in system.



包含文字:

	表	操作	行数	类型	排序规则	大小	多余
☐	agendas	☆ 浏览 结构 搜索 插入 清空 删除	20	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	agenda_keypoints	☆ 浏览 结构 搜索 插入 清空 删除	2	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	failed_jobs	☆ 浏览 结构 搜索 插入 清空 删除	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	–
☐	friend_lists	☆ 浏览 结构 搜索 插入 清空 删除	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	groups	☆ 浏览 结构 搜索 插入 清空 删除	10	InnoDB	utf8mb4_unicode_ci	32.0 KB	–
☐	group_members	☆ 浏览 结构 搜索 插入 清空 删除	6	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	meetings	☆ 浏览 结构 搜索 插入 清空 删除	6	InnoDB	utf8mb4_unicode_ci	32.0 KB	–
☐	meeting_attendances	☆ 浏览 结构 搜索 插入 清空 删除	8	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	migrations	☆ 浏览 结构 搜索 插入 清空 删除	26	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	password_resets	☆ 浏览 结构 搜索 插入 清空 删除	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	–
☐	pending_agendas	☆ 浏览 结构 搜索 插入 清空 删除	8	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	title_agendas	☆ 浏览 结构 搜索 插入 清空 删除	25	InnoDB	utf8mb4_unicode_ci	16.0 KB	–
☐	users	☆ 浏览 结构 搜索 插入 清空 删除	4	InnoDB	utf8mb4_unicode_ci	64.0 KB	–
	13 张表	总计	115	InnoDB	latin1_swedish_ci	320.0 KB	0 字节

Figure 2.7: The interface of phpMyAdmin after Xampp is installed.

2.4 Programming Language

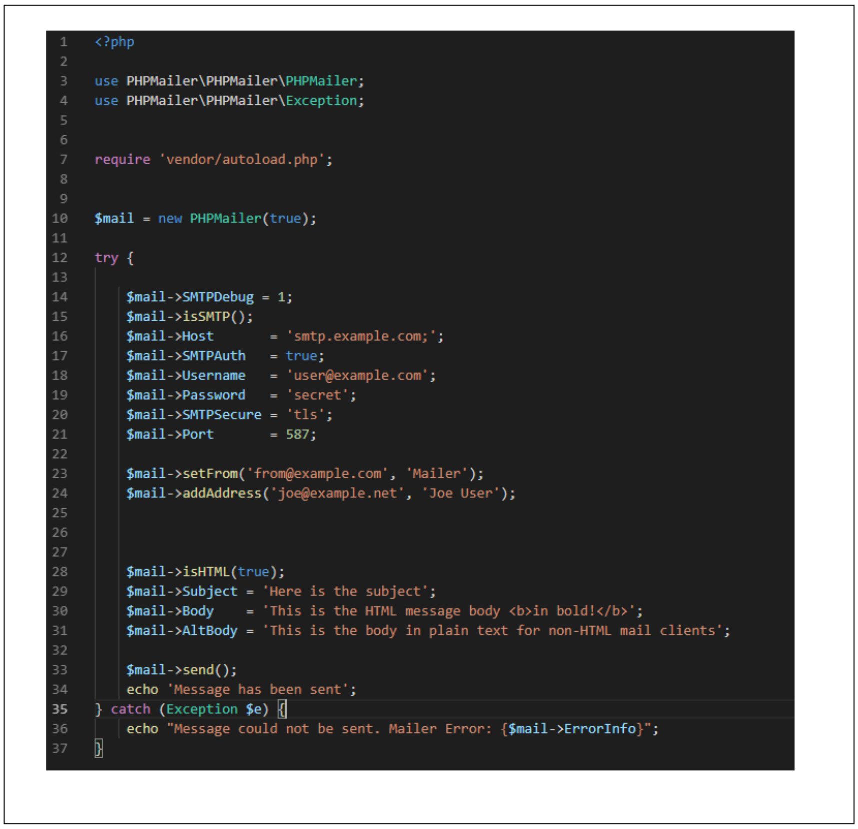
From TechTerms, it says that is a set of commands, instructions, and other syntax use to create a software program Languages that programmers use to write code are called "high-level languages." This code can be compiled into a "low-level language," which is recognized directly by the computer hardware.

Programming language is the way to makes our logic from mind become true. It makes the real changes and create what we want with coding. The way or the syntax to write it are call programming Languages. If we want to become a programmer someday, programming language are unavoidable to learn.

2.4.1 PHP

From PHP official website, it said that PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is also widely used, free, and efficient alternative to competitors such as Microsoft's ASP.

If we want to use PHP in our system, it can be easily implemented by leaving <?php inside our html column so we can use PHP. When using PHP, it helps people to understand the server-side logic since it is the first server-side languages that could embedded in HTML. It can be easily use for rookie too without any prerequisite.



```php
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;


require 'vendor/autoload.php';


$mail = new PHPMailer(true);

try {

    $mail->SMTPDebug = 1;
    $mail->isSMTP();
    $mail->Host       = 'smtp.example.com;';
    $mail->SMTPAuth   = true;
    $mail->Username   = 'user@example.com';
    $mail->Password   = 'secret';
    $mail->SMTPSecure = 'tls';
    $mail->Port       = 587;

    $mail->setFrom('from@example.com', 'Mailer');
    $mail->addAddress('joe@example.net', 'Joe User');



    $mail->isHTML(true);
    $mail->Subject = 'Here is the subject';
    $mail->Body    = 'This is the HTML message body <b>in bold!</b>';
    $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

    $mail->send();
    echo 'Message has been sent';
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```

Figure 2.8: The example of code in PHP

3. Chapter Three: Methodology

3.1 Introduction

This chapter will explain the methodology that used that I used in gathering data and analysis which are relevant to research. The methodology will include each process while developing system.

With a Software Development Life Cycle (SDLC) model, it helps people to organise the sequence to develop a system. It will help in future maintenance and develop a high-quality software since everything is planned and traceable. Without SDLC mode, it is likely to develop a less professional app compare to others. In this system, I will use waterfall model as my SDLC model to develop the system.

3.2 Waterfall model

Waterfall model is a Software Development Life Cycle (SDLC) model to helps in developing the system. It is very simple and easy to be use. It illustrates a linear sequence flow while developing the product. There are some sequences that needed to be follow according to order while developing the system which are communication, planning, design, development, and deployment.

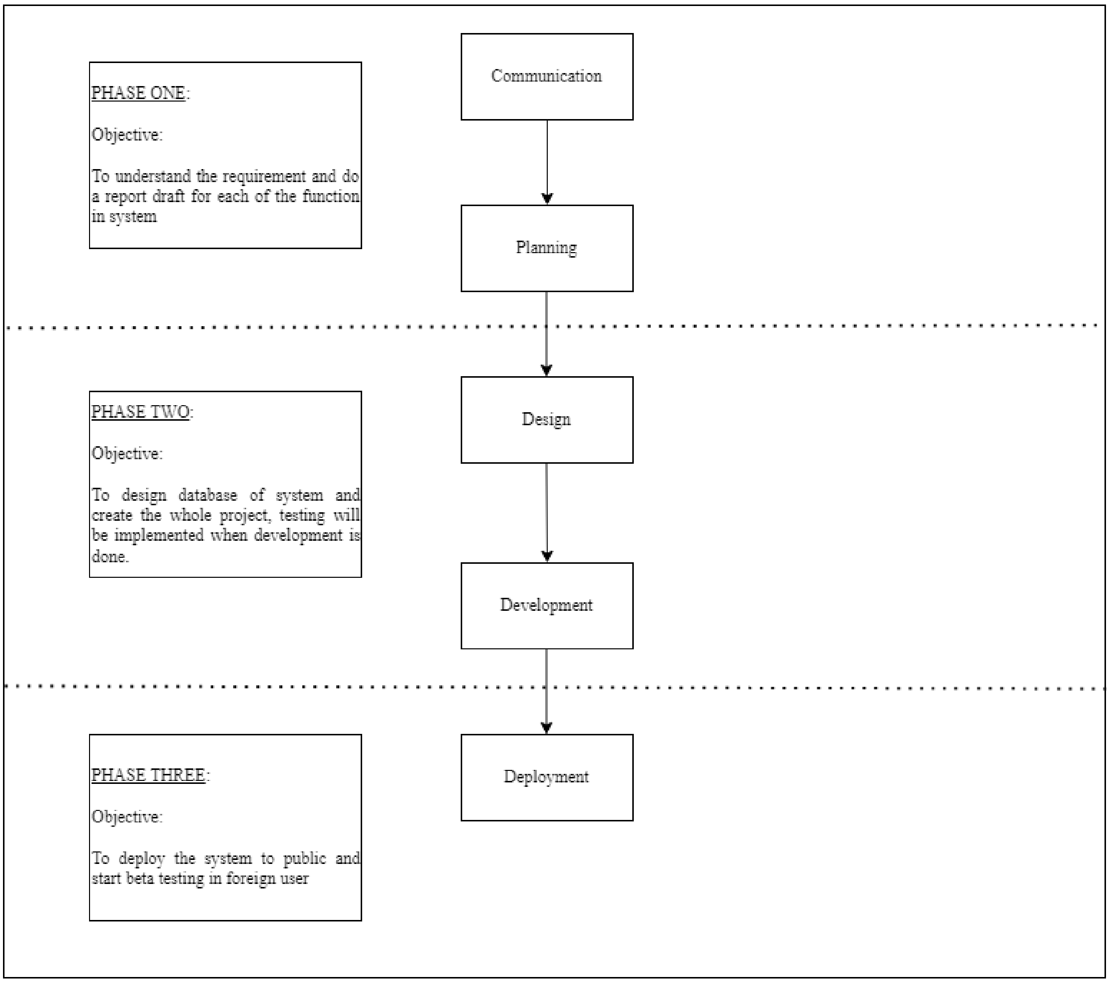

PHASE ONE:
Objective:
To understand the requirement and do a report draft for each of the function in system
Communication
Planning
PHASE TWO:
Objective:
To design database of system and create the whole project, testing will be implemented when development is done.
Design
Development
PHASE THREE:
Objective:
To deploy the system to public and start beta testing in foreign user
Deployment

Figure 3: The methodology of system development

3.3 Project framework

3.3.1 Communication

First, the requirements of users must be gathering. It helps me to understand the real use of user instead of thinking myself. The information can be gathering in many ways. For example, information can be gathering from secretary in school, whether it is club's secretary or department's secretary. It is because most of the function in this system would be use by secretary only.

There are some requirements gathered during process. First, point of each of the agenda is necessary. It is because each of the agenda require points after each of the discussion for future discuss. Then, people also require generate pdf form once the report is done. The paperwork is also needed for everyone to view after the discussion is done and able to check in next meeting. After that, they also request for Google Meet to makes the meeting become more convenient. Chat room for each of the members are also requested to replace messenger to use while meeting.

3.3.2 Planning

During planning phase, we need to have deep thought about each of the requirement's list in communication phase. Some of the requirement can be keep but some of the requirements are useless to do. For example, chat room function is not useful for the system since this requirement can be achieve once the Google Meet function achieve.

This phase helps to understand the objective of this system. The functions needed are listed down for future use in design phase. This process helps us to more understand about each of the requirements and detail about them. For example, generate pdf requirements would have some precondition before done it such as meeting form. Each of the requirements that wants to achieved need to be list out with detail precondition and flow needed.

3.3.3 Design

In Design phase, we would try to sketch out the diagram for each of the function and the overall. The diagram should include in these parts are activity diagram, entity relationship diagram, use case, class diagram and dataflow diagram. These all diagrams help a lot in understanding the flow and the development sequence which work in development phase.

First, activity diagram helps people to understand the flow of each function. For example, payment function require flow from select item to buy until payment method select. This help software engineer understands what to develop. Next, entity relationship diagram helps us to understand how the database table and each of the columns needed in table. It helps backend to construct database table. Use case write the overall function for each user inside the system to have an overview of each use functions. Class diagram created for each of the objects for each of the class. It also includes functions for each class to helps in construct backend. Dataflow diagrams helps in understand the data flow of system, so we won’t make mistake while making the system flow.

3.3.4 Development

In development phase, the code should be implemented. The code should follow the policy stated to ensure the code quality. The testing would be done after the function is done to ensure that every function is work. After the code has been implemented and pass alpha test, which is test by software developer, it will enter the last phase which is deployment phase.

3.3.5 Deployment

In deployment phase, the product will be release to customer to view feedback from customer and prepare for maintenance. The report will be generated so that the experience gain in this system can be used for other system in future. After receive beta testing feedback from customer, the system can prepare for next waterfall model to evolve the system.

4. Chapter Four: Design

4.1 Introduction

This chapter will discuss the design of this project that list in requirement analysis. The design diagram of this project includes use case diagram, sequence diagram, activity diagram, class diagram, entity relationship diagram. User interface design would also include in this section.

4.2 Requirement Analysis

4.2.1 Use case diagram

In figure 4.1, there are 22 features in this meeting minutes system. The purpose of the meeting minute system for user is to record the minute after each of the meeting is done.

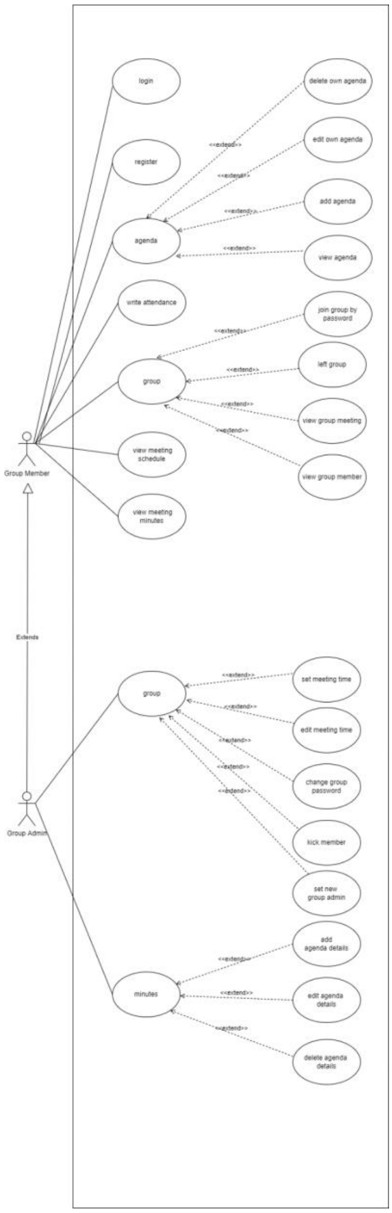


Figure 4.1: Use Case Diagram of Meeting Minute System

Table 4.1: List of use case and its description

Use Case	Description
Login	To allow user to login to use other function
Register	User can register an account to use function
Change group password	Admin can change the password of group to join
Edit meeting time	Admin can edit the meeting time set
Kick member	Admin can kick member inside group
Create meeting	Admin can create meeting inside group
Set new group admin	Admin can set new group admin for group
Add agenda key point	Admin can add key point for each agenda
Delete agenda key point	Admin can delete key point for each agenda
Edit agenda key point	Admin can edit key point for each agenda
Generate PDF	Admin can export the meeting result to an external file
Check attendances	Admin can check attendances for who attends the meeting
Add agenda	User can add agenda for each group and save them as pending agenda
Delete own agenda	User can delete their own agenda
Edit own agenda	User can edit their own agenda
View agenda	User can view the detail of agenda
Join group with password	User can join group with password set by admin
Leave group	User can leave group
View group meeting	User view group meeting list inside group
View group member	User can view group member inside group
View meeting minutes	User can view detail of meeting minutes
View meeting schedules	User can view meeting schedules once they login

4.2.2 Sequence Diagram

The purpose of sequence diagram is to illustrate the relationship between objects in an interaction. It shows how sequence of message is exchanged between object for each functionality in a system.

4.2.2.1 Admin

4.2.2.1.1 Change group password

Based on the figure below, admin can change the group password after they login and get into their group. Click update password will be able to update password function.

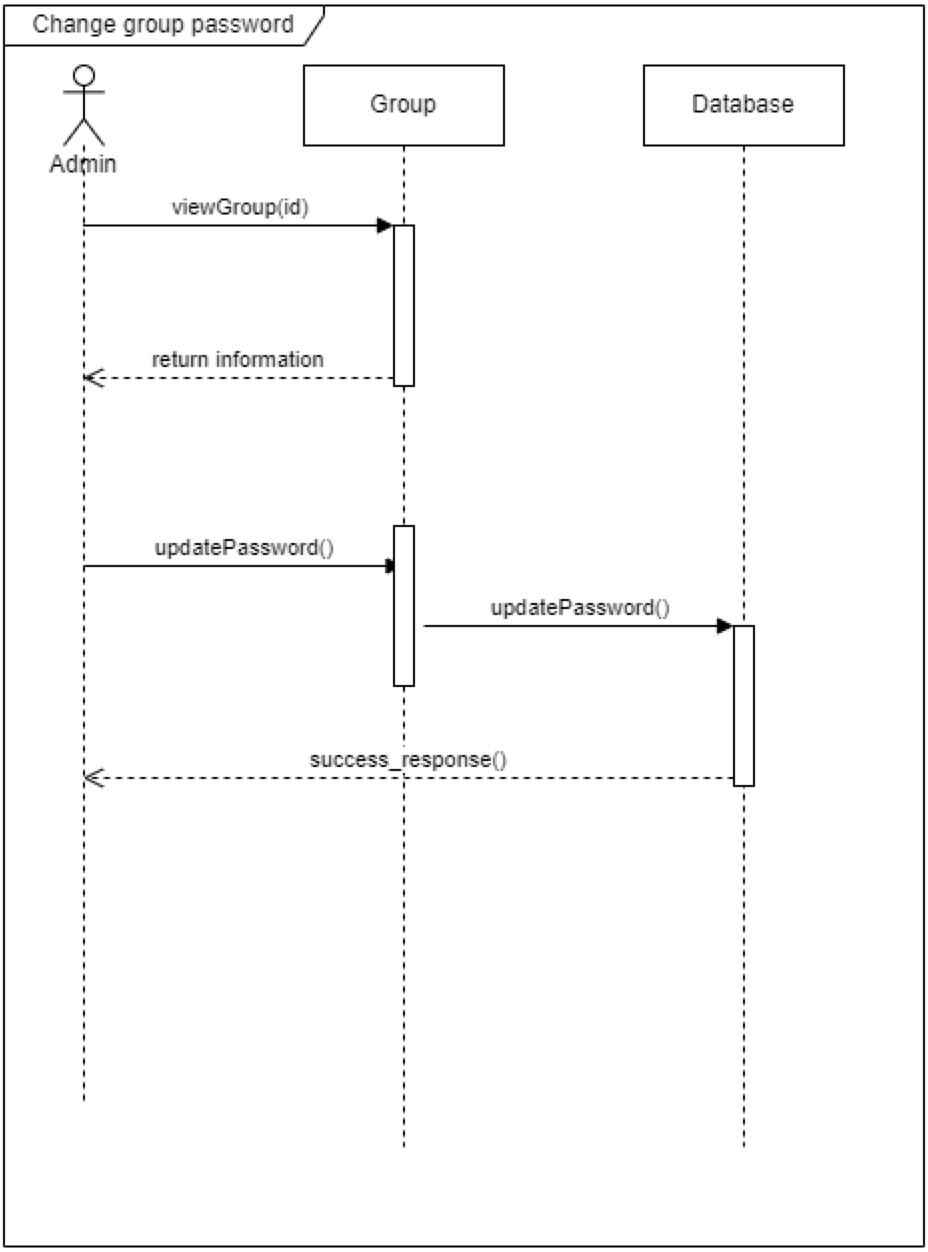


Figure 4.2: Sequence diagram for change group password

4.2.2.1.2 Edit meeting time

Based on the figure below, admin allow to edit meeting time after login. Click on the specific meeting and key in new date and time and click update will be able to update meeting time.

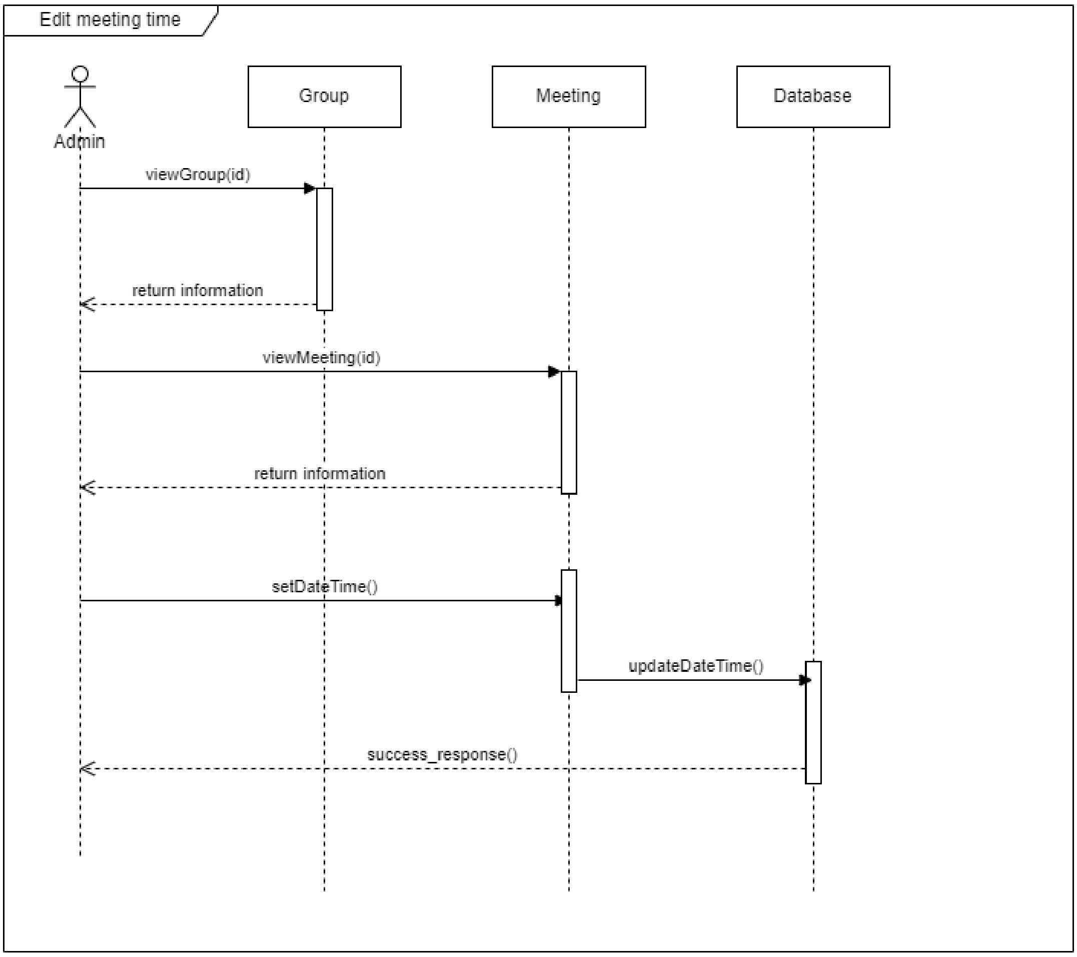


Figure 4.3: Sequence diagram for edit meeting time

4.2.2.1.3 Kick member

Based on the figure below, admin allow to kick member where belongs to their group. Select a specific group to edit and select a member and kick member would allow to kick member.

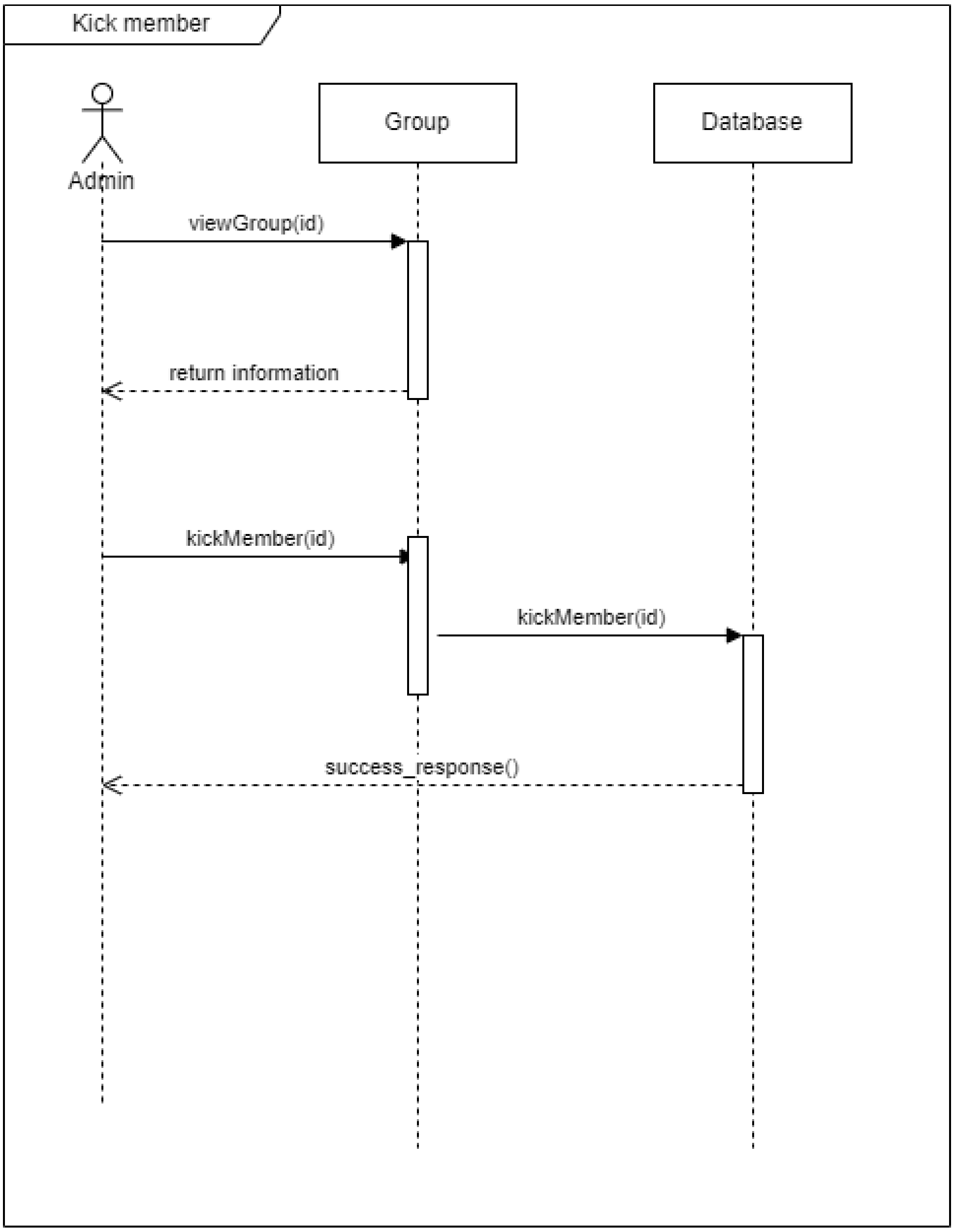


Figure 4.4: Sequence diagram for kick member

4.2.2.1.4 Create meeting

Based on the figure below, admin allow to create meeting after login. Admin will need to key in date, time, title and assign title to create meeting.

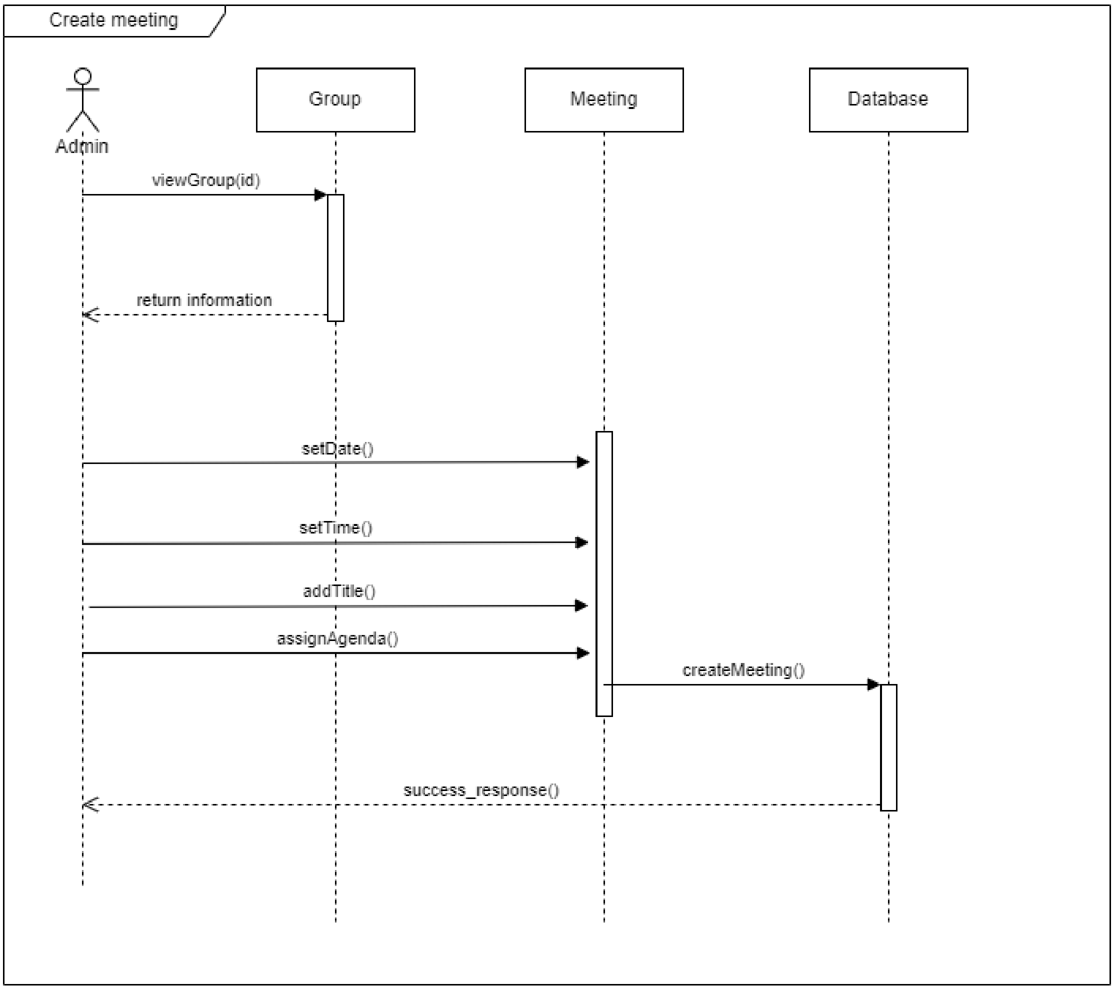


Figure 4.5: Sequence diagram of create meeting

4.2.2.1.5 Set new group admin

Based on the figure below, admin allow to set new admin for the group. Select specific group and select specific member can set them as admin.

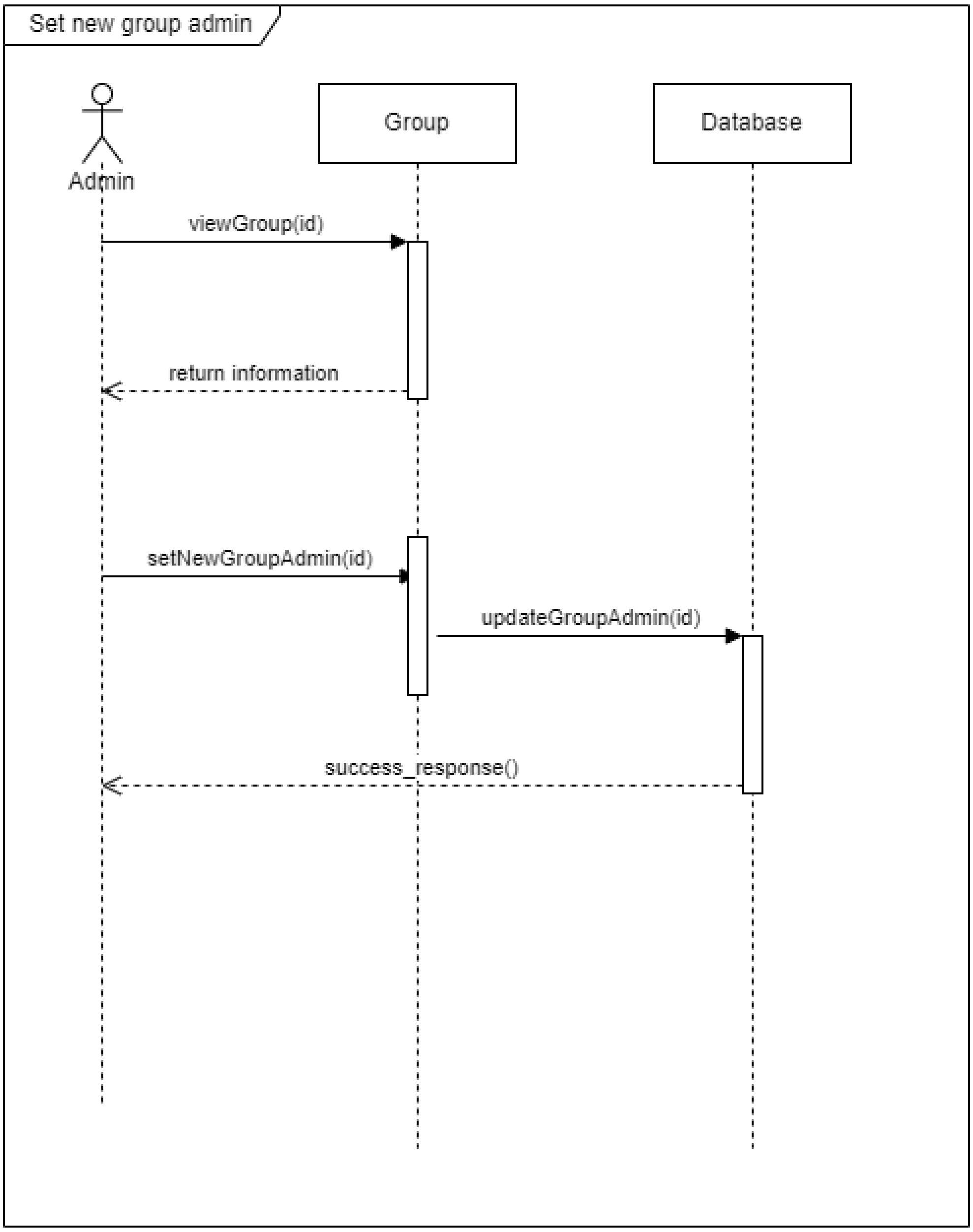


Figure 4.6: Sequence diagram of set new group admin

4.2.2.1.6 Add agenda key point

Based on the figure below, admin allow to add key point for each of the agenda after login. After meeting is created, click on the agenda will allow to create key point.

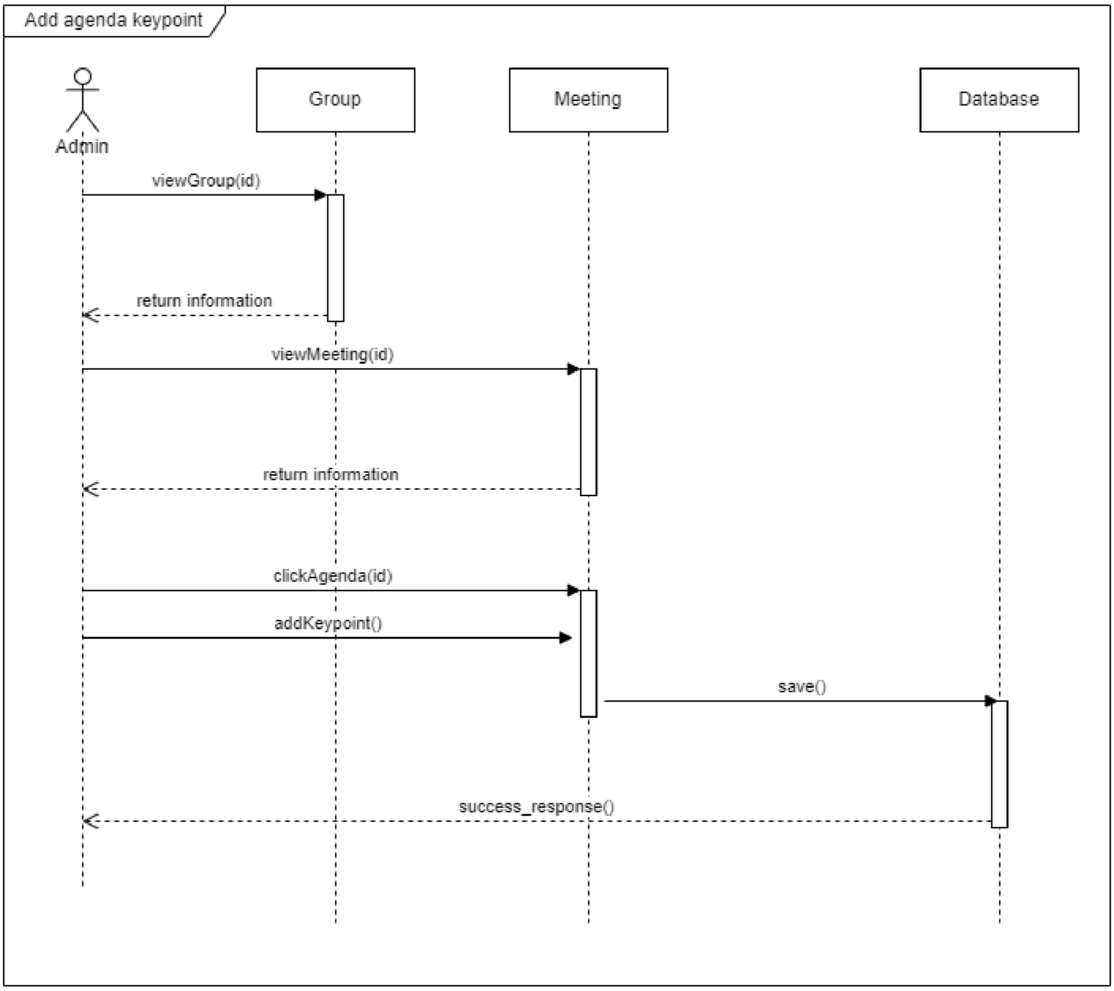


Figure 4.7: Sequence diagram of add agenda key point

4.2.2.1.7 Delete agenda key point

Based on the figure below, admin allow to delete key point for each of the agenda after login. After meeting is created, click on the agenda will allow to delete key point.

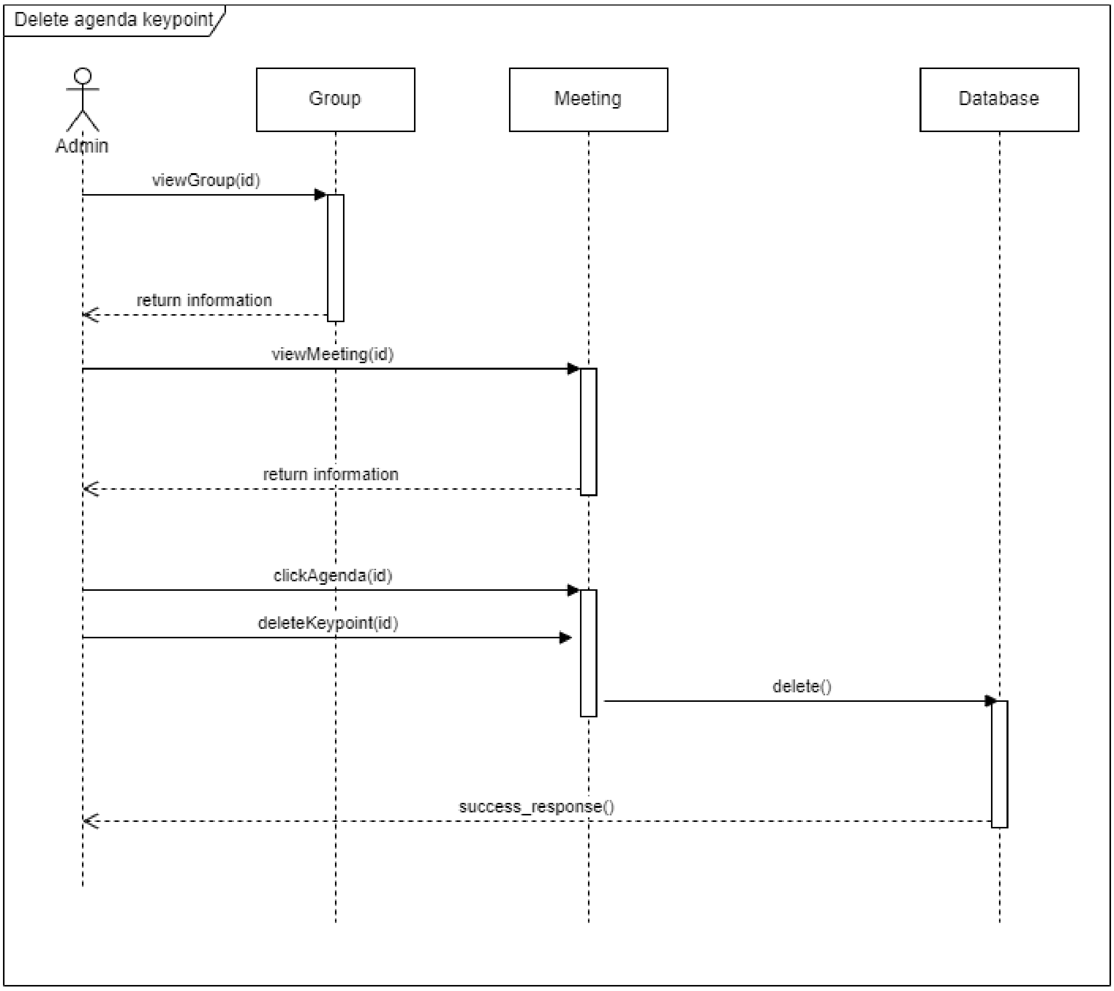


Figure 4.8: Sequence diagram of delete agenda key point

4.2.2.1.8 Edit agenda key point

Based on the figure below, admin allow to edit key point for each of the agenda after login. After meeting is created, click on the agenda will allow to edit key point exist.

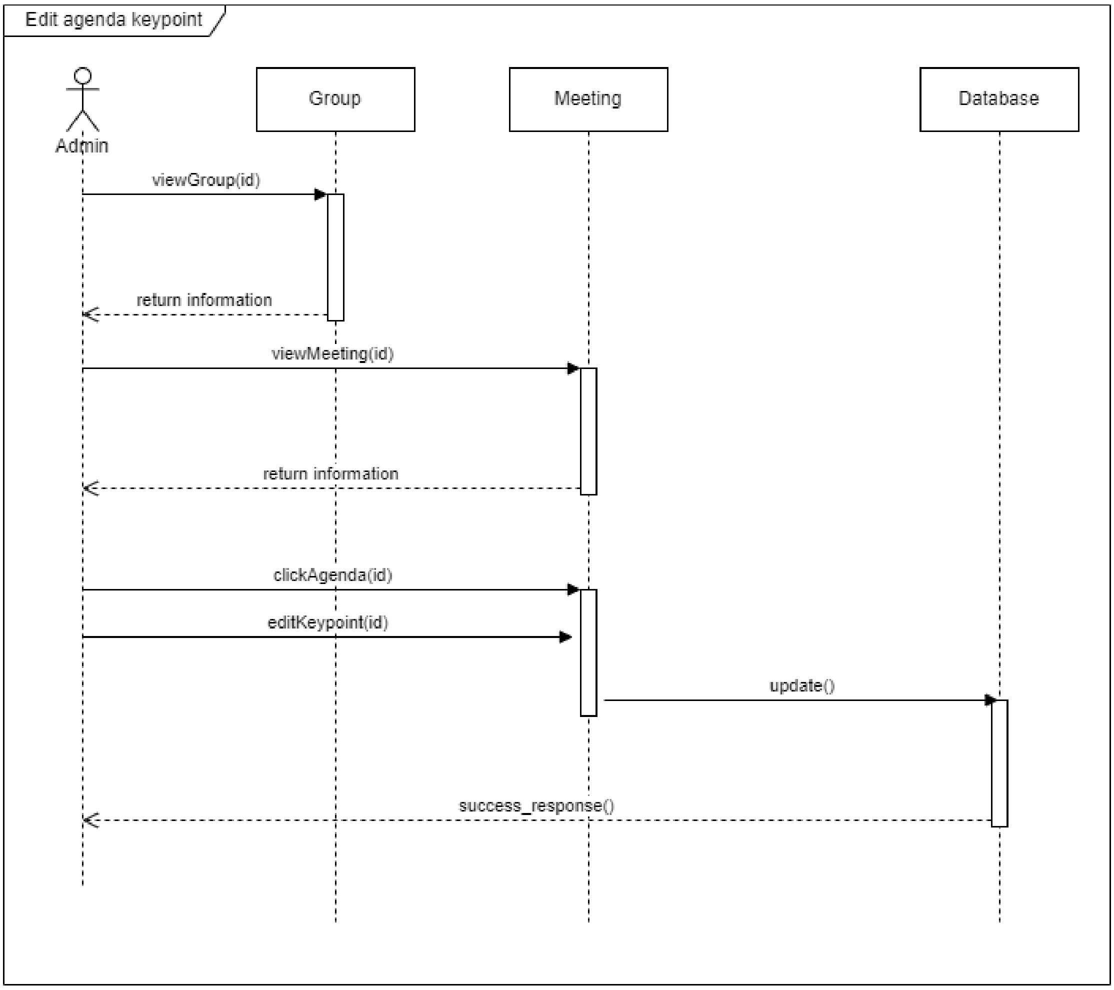


Figure 4.9: Sequence diagram for edit agenda key point

4.2.2.1.9 Generate PDF

Based on the figure below, admin allow to generate PDF after login. Click on generate PDF will display a pdf allow to view.

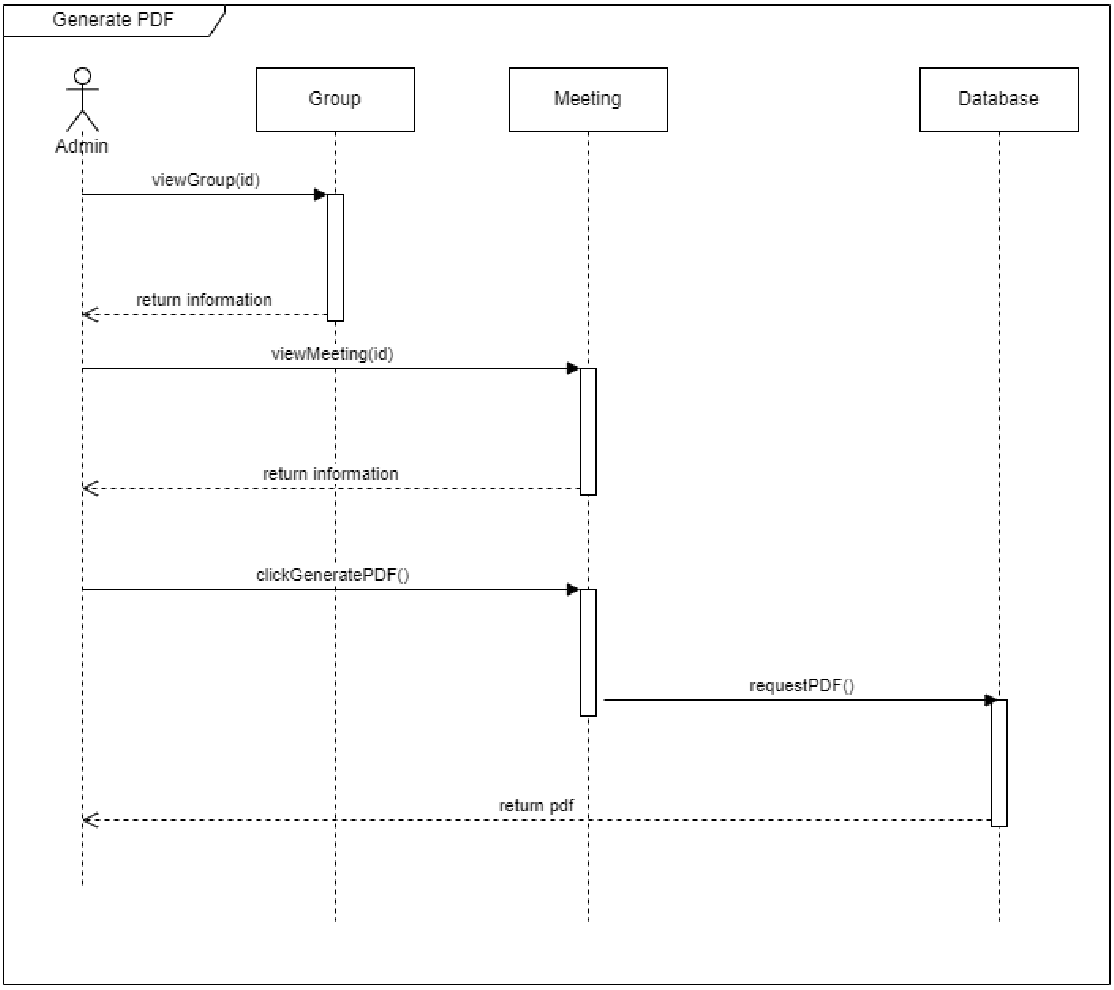


Figure 4.10: Sequence diagram for generate PDF

4.2.2.1.10 Check attendances

Based on the figure below, admin allow to check attendances after login. Select on group and specific meeting and check attendances for each of the user attends.

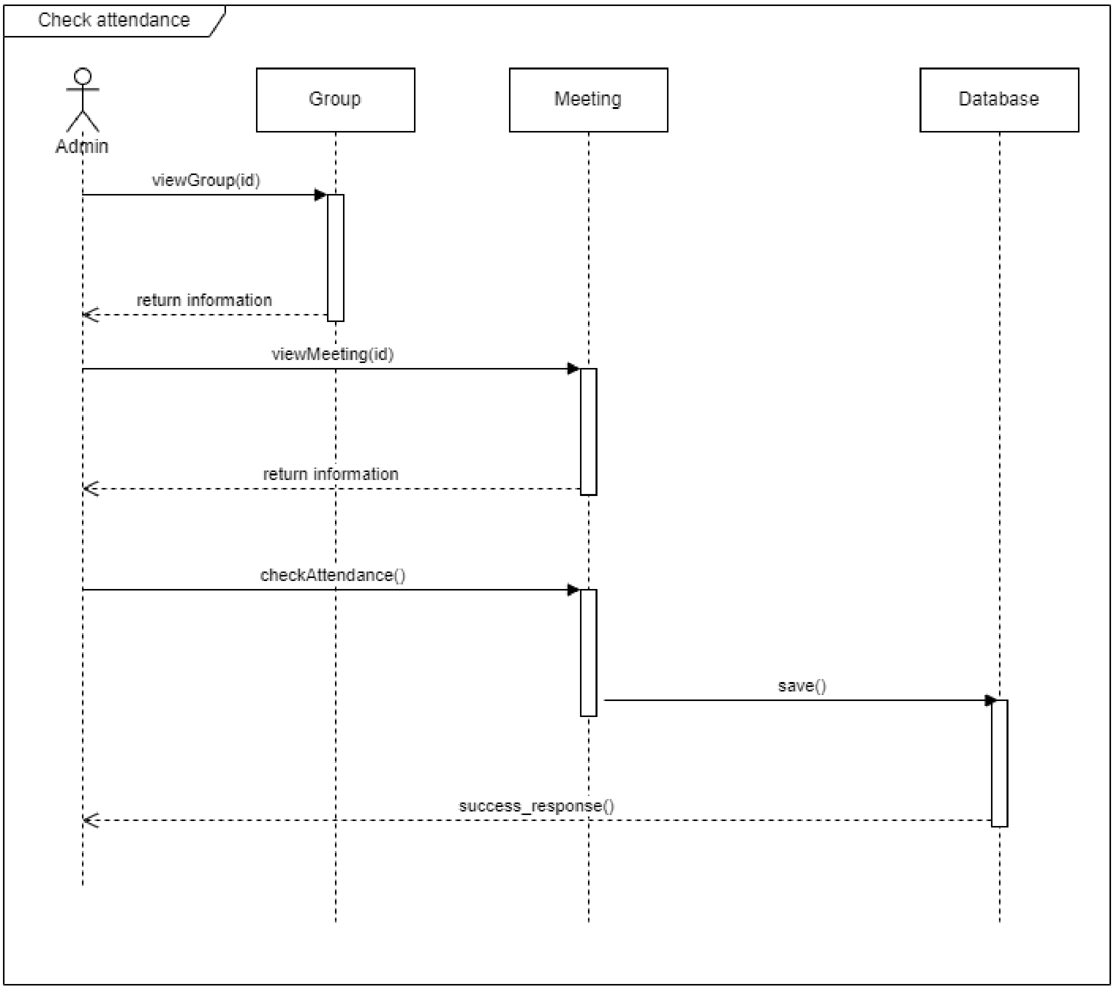


Figure 4.11: Sequence diagram for check attendances

4.2.2.1.11 Send invitations

Based on the figure below, admin allow to send invitations after login. Select on group and group members and send invitations to users.

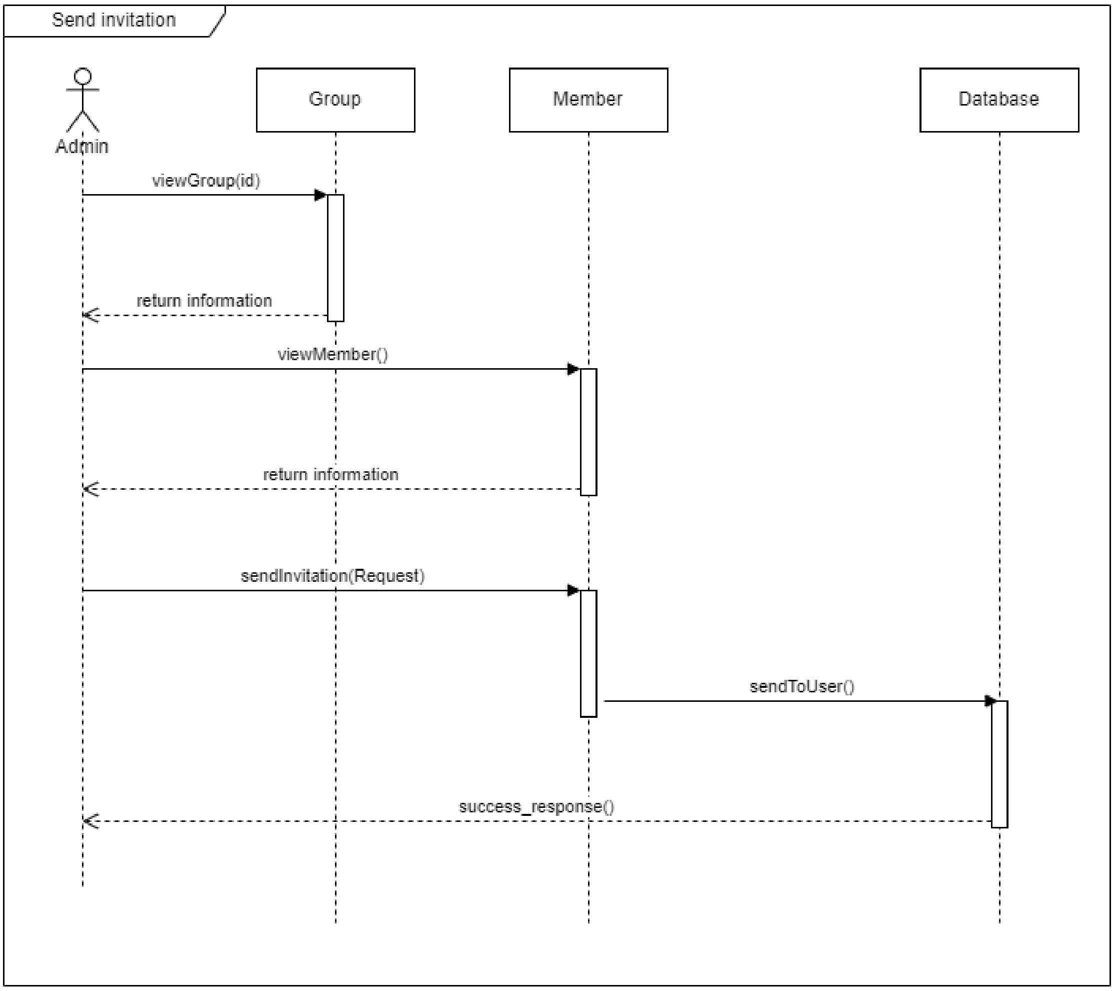


Figure 4.12: Sequence diagram for send invitations

4.2.2.2 Member

4.2.2.2.1 Add agenda

Based on the figure below, member allow to add agenda after login. Click on the specific group and allow to add agenda to database for future use after attachment file and title is filled in.

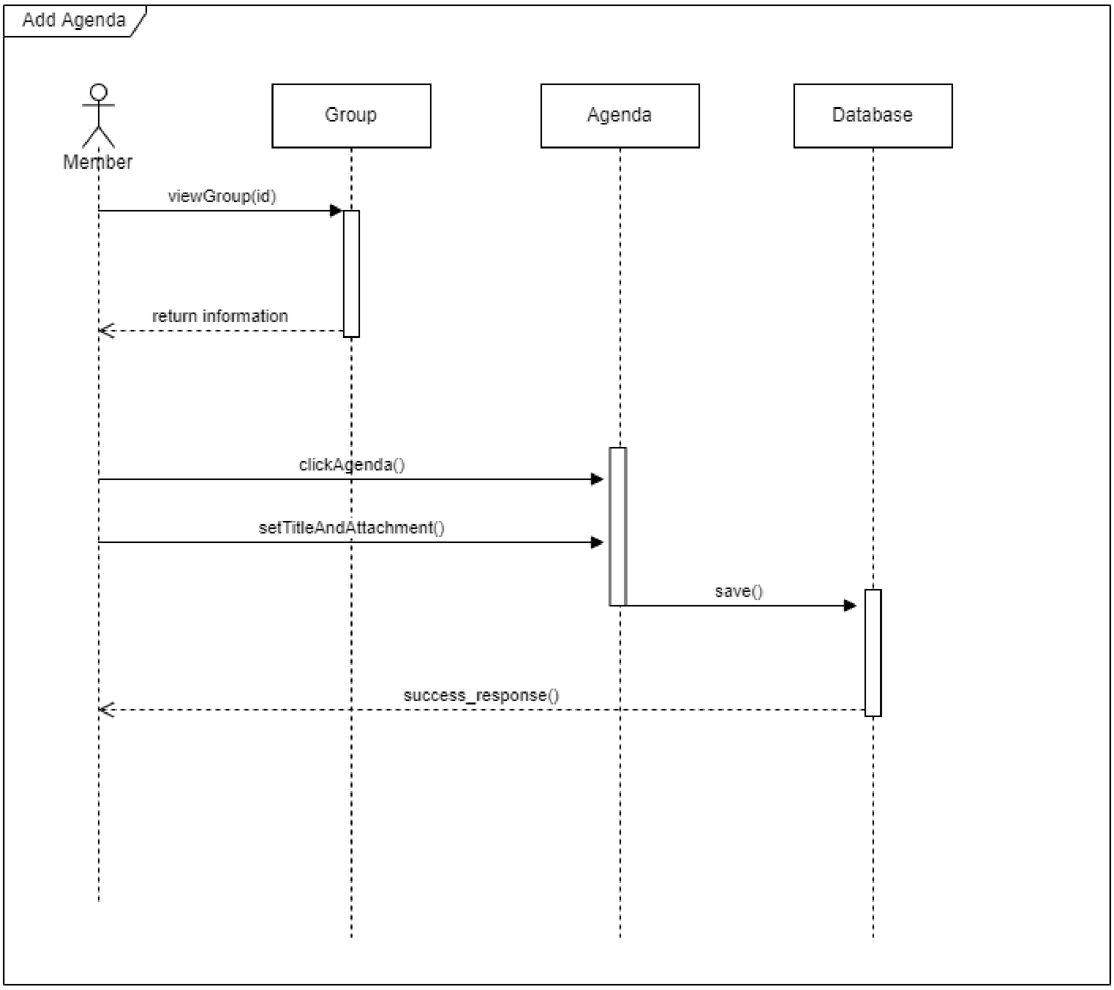


Figure 4.13: Sequence diagram for add agenda

4.2.2.2.2 Delete own agenda

Based on the figure below, member allow to delete own agenda after login. Select group and specific meeting then click on own agenda to delete.

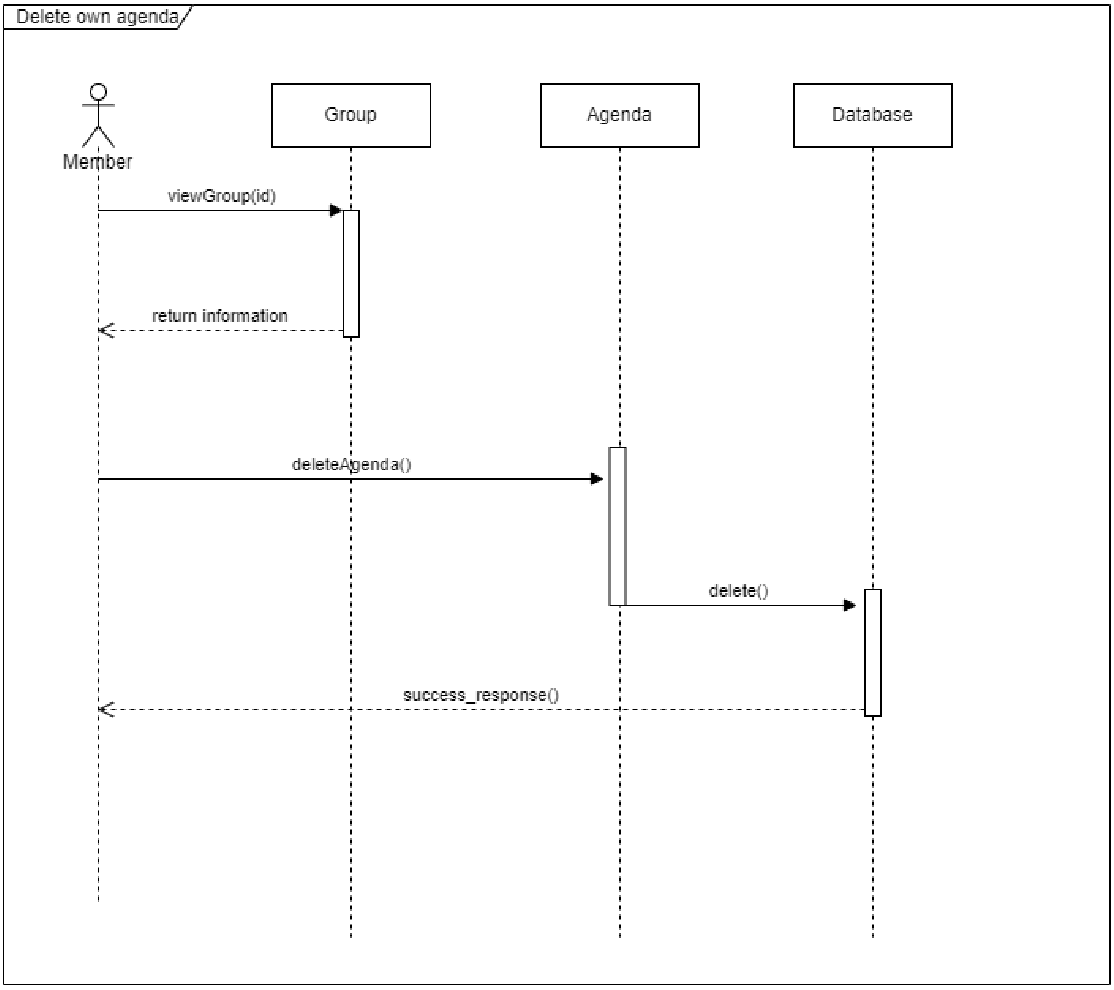


Figure 4.14: Sequence diagram for delete own agenda

4.2.2.2.3 Edit own agenda

Based on the figure below, member allow to edit own agenda after login. Select group and specific meeting then click on own agenda to edit. Fill in new title and attachment file will allow to make edit.

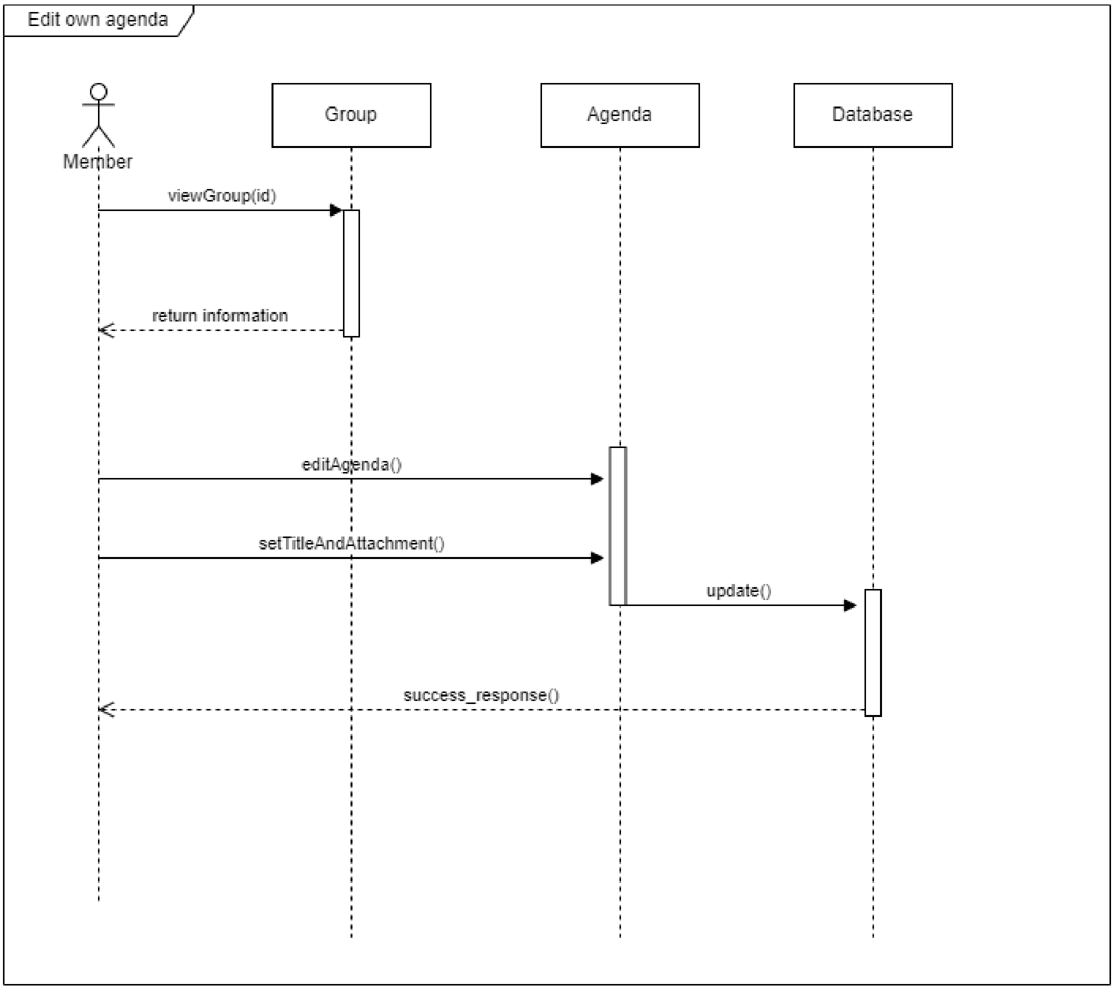


Figure 4.15: Sequence diagram for edit own agenda

4.2.2.2.4 View agenda

Based on the figure below, member allow to view agenda after login. Select group and specific meeting then click on agenda to view all agenda list.

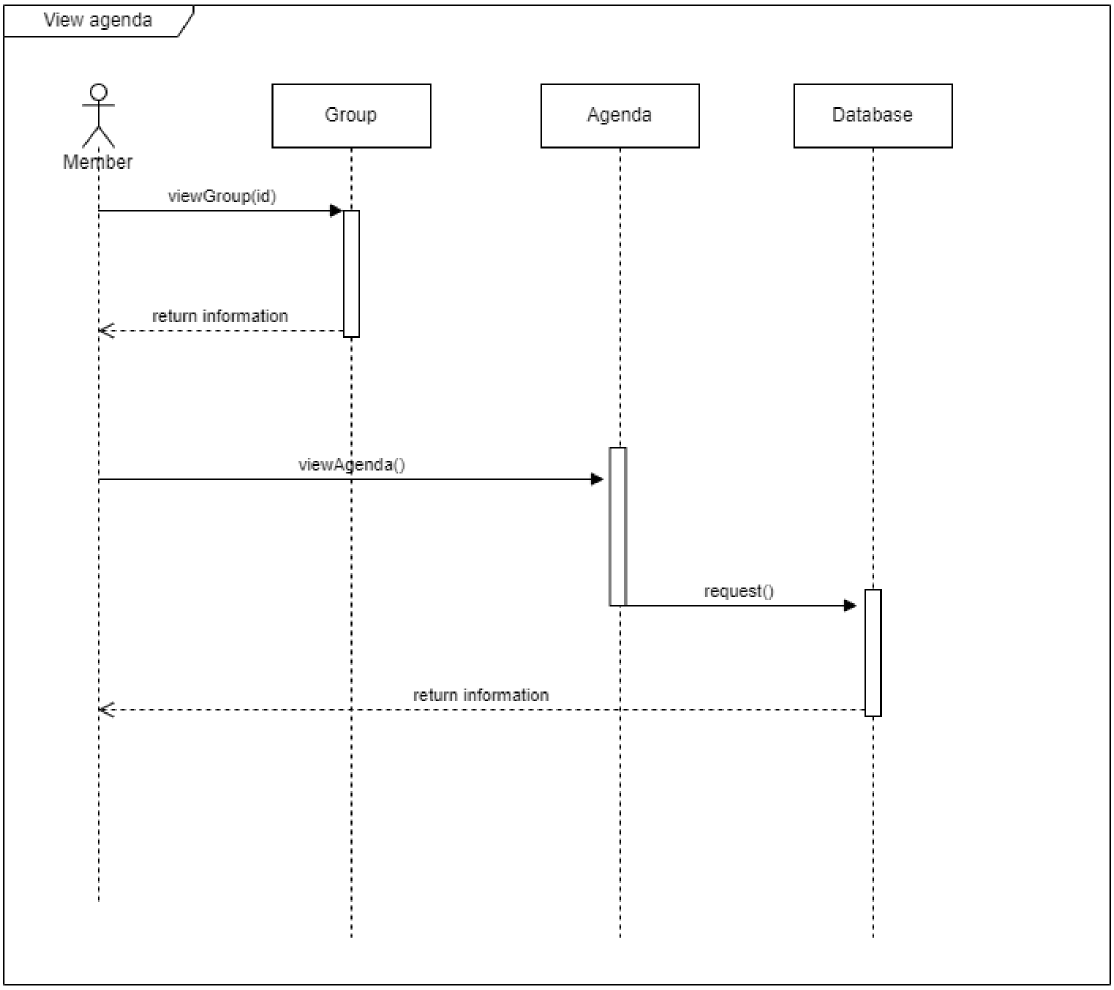


Figure 4.16: Sequence diagram for view agenda

4.2.2.2.5 Join group with password

Based on the figure below, member allow to join group with password after login. They can fill in the group password in textbox and find for fit group password. After the password is correct and member haven’t joined the group, member allow to join the group.

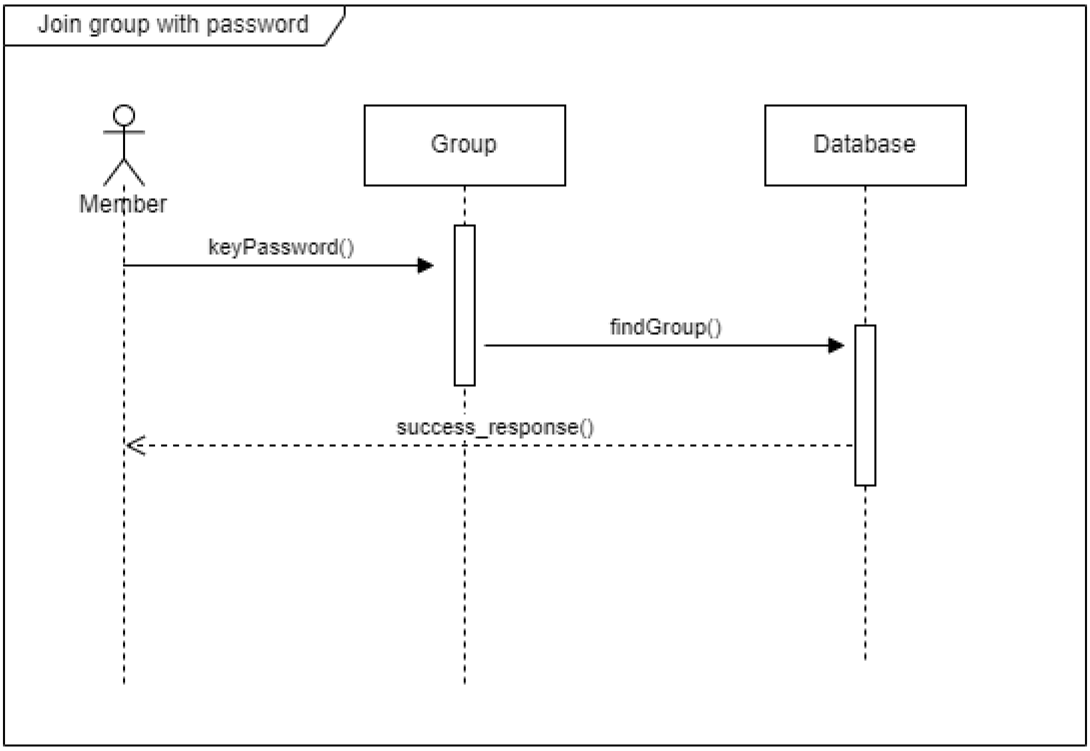


Figure 4.17: Sequence diagram for join group with password

4.2.2.2.6 Left group

Based on the figure below, member allow to left group once they join the group. Select on the left group option would allow to leave the group.

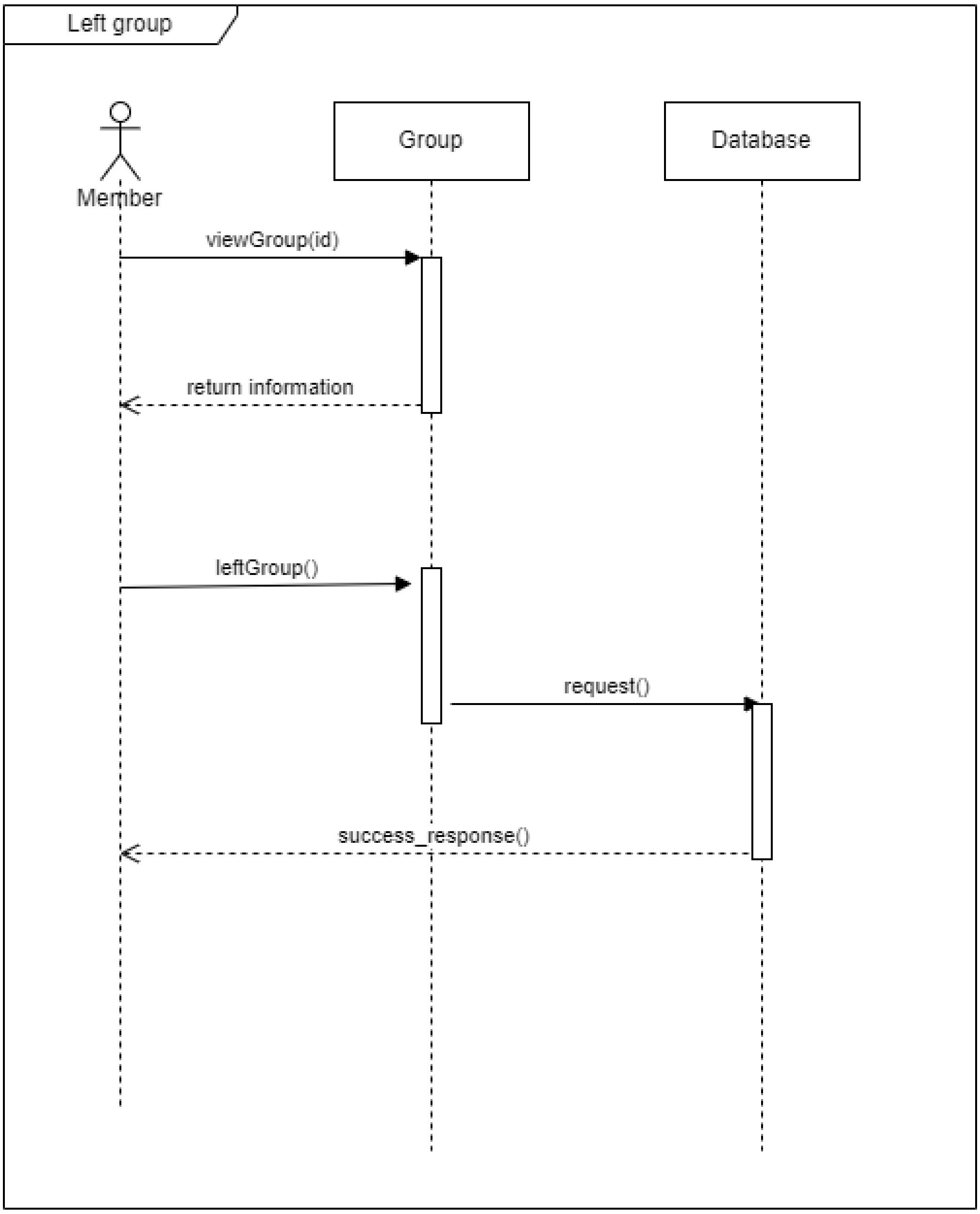


Figure 4.18: Sequence diagram for left group

4.2.2.2.7 View group meeting

Based on the figure below, member allow to view group meeting once they join the group. Click on the meeting will be able to view meeting list.

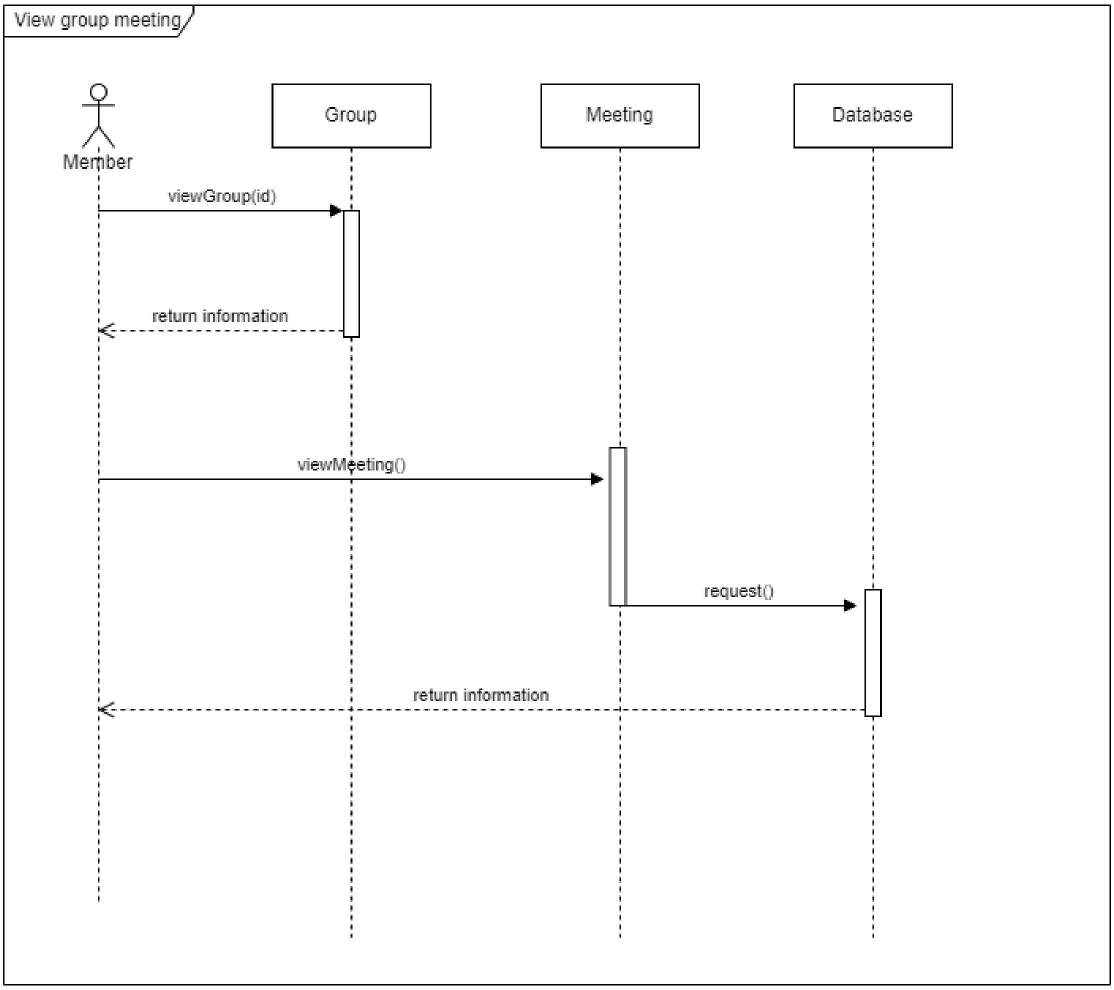


Figure 4.19: Sequence diagram for view group meeting

4.2.2.2.8 View group member

Based on the figure below, member allow to view group member after joined the group. Select group member on the selection would allow to join group.

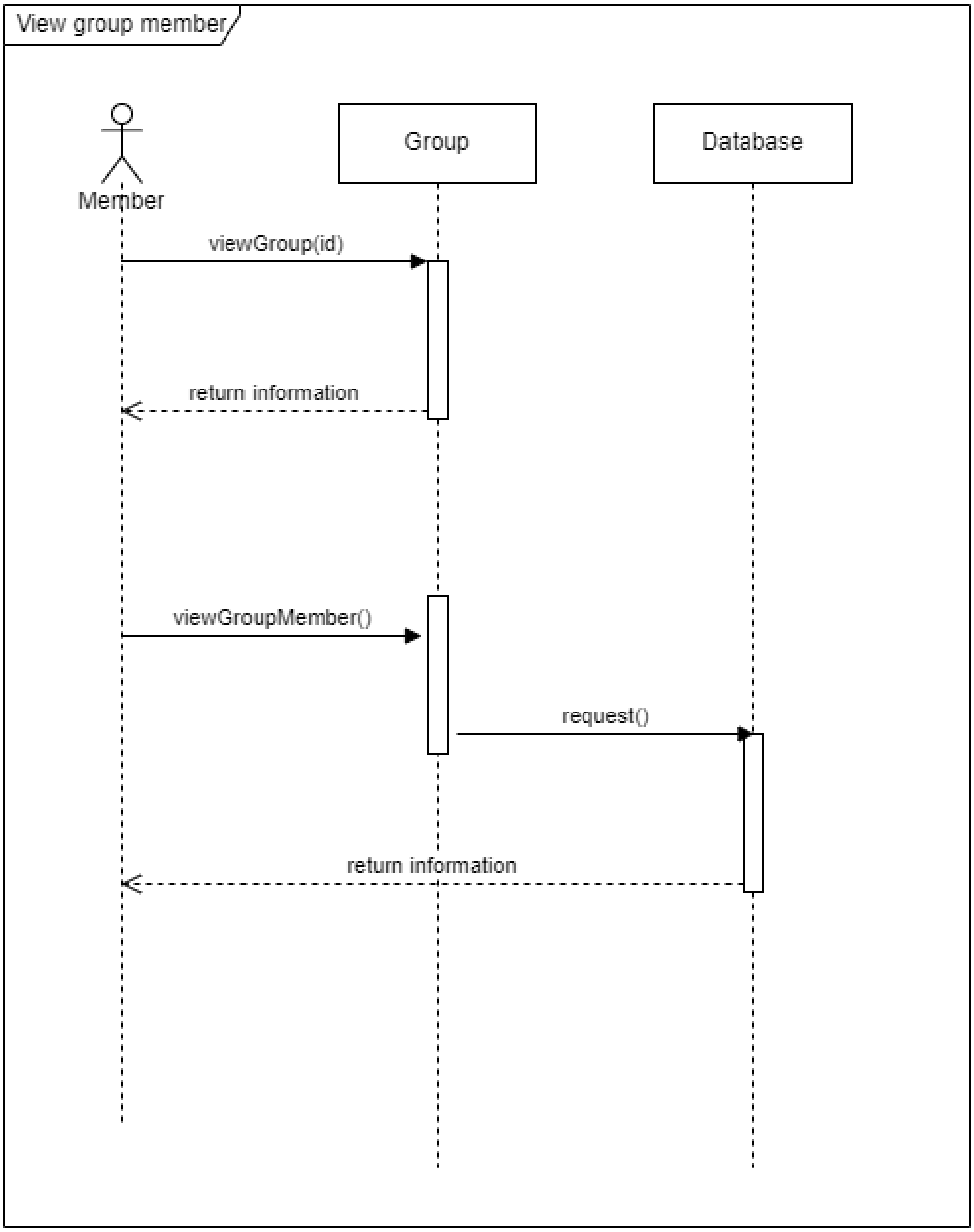


Figure 4.20: Sequence diagram for view group member

4.2.2.2.9 View meeting minutes

Based on the figure below, member allow to view detail of meeting minutes once they joined the group. Select on specific meeting minutes allow to view detail of meeting minutes.

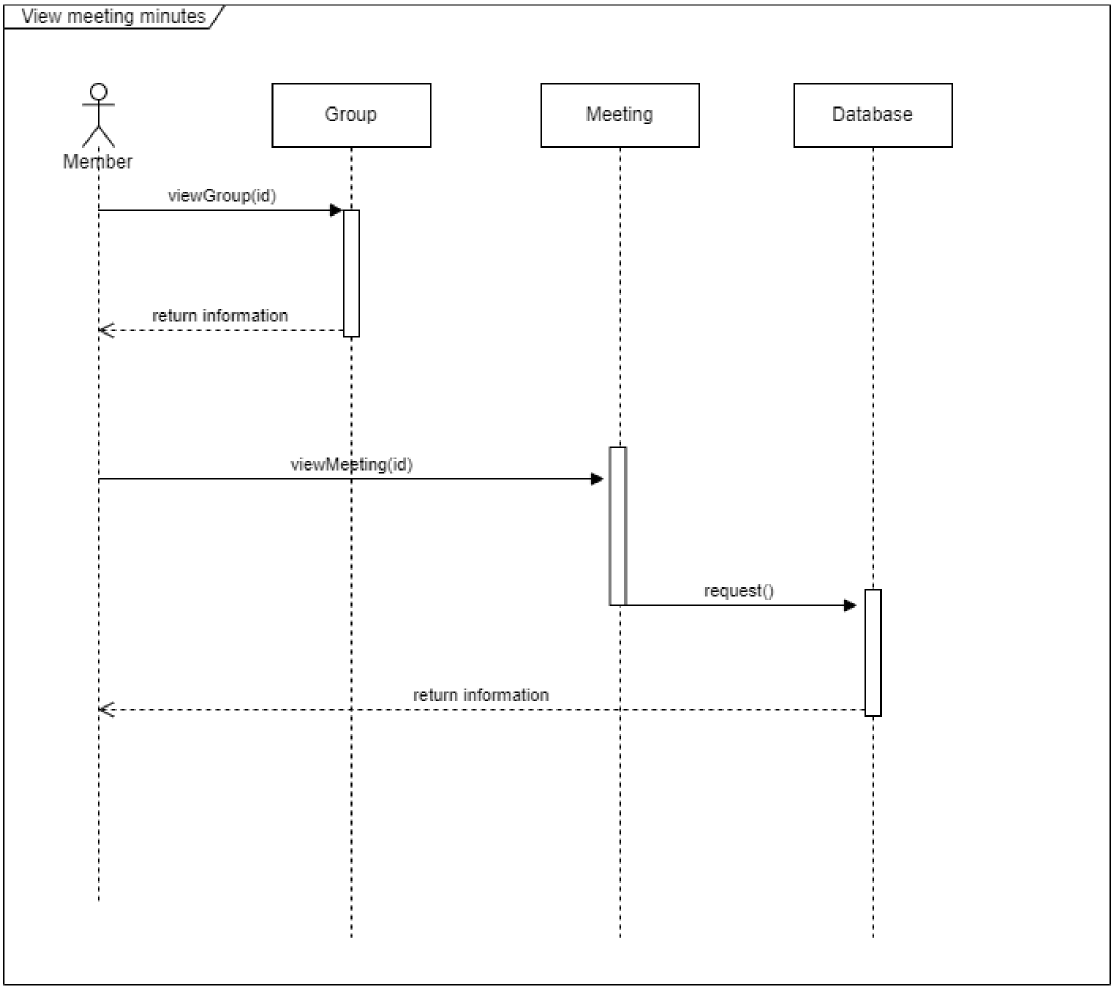


Figure 4.21: Sequence diagram of view meeting minutes

4.2.2.2.10 View meeting schedules

Based on the figure below, member allow to view meeting schedules after they login. Click on the meeting schedules allow to view meeting schedules.

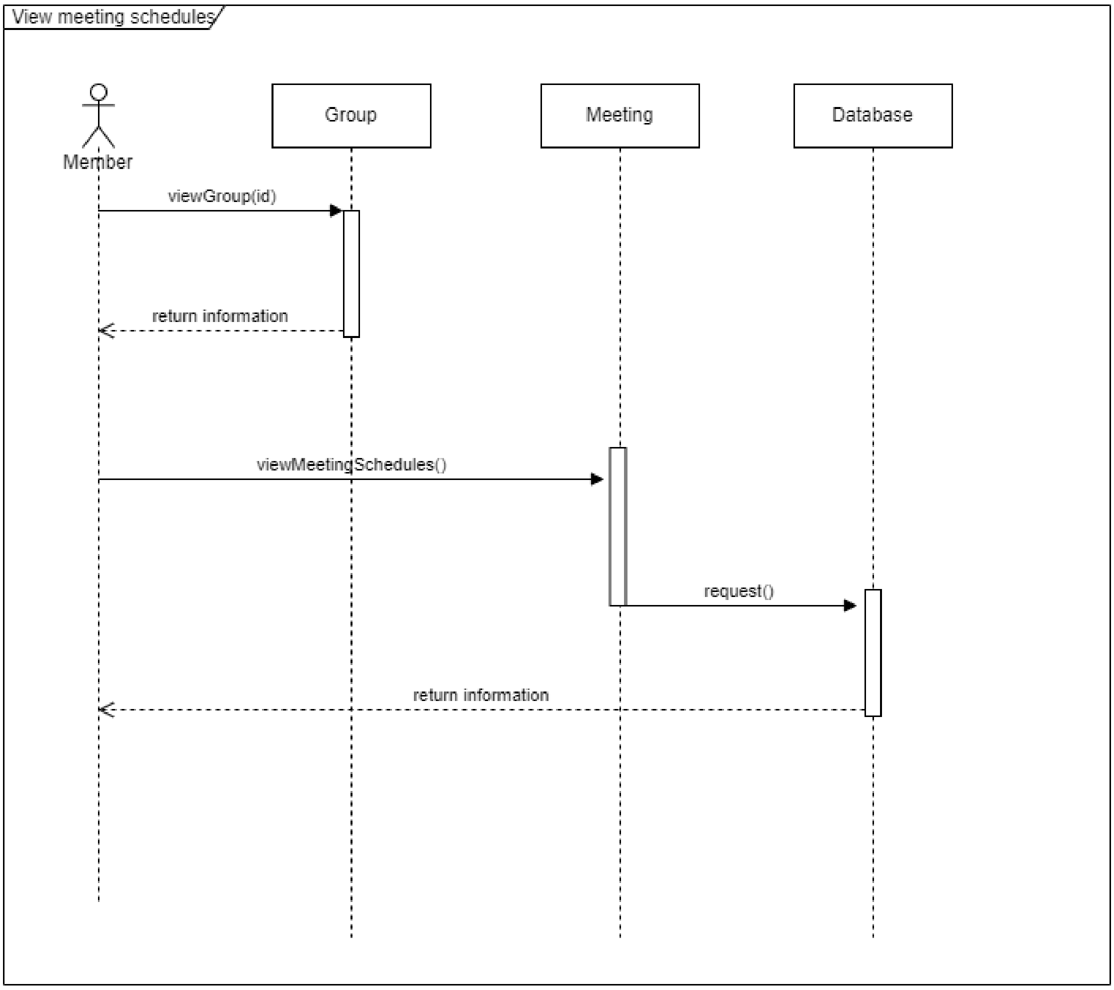


Figure 4.22: Sequence diagram for view meeting schedules

4.2.2.2.11 Login

Based on the figure below, user must provide email and password to login. This system require login to use all the function. Once there is exist account in database, member allow to login to the function.

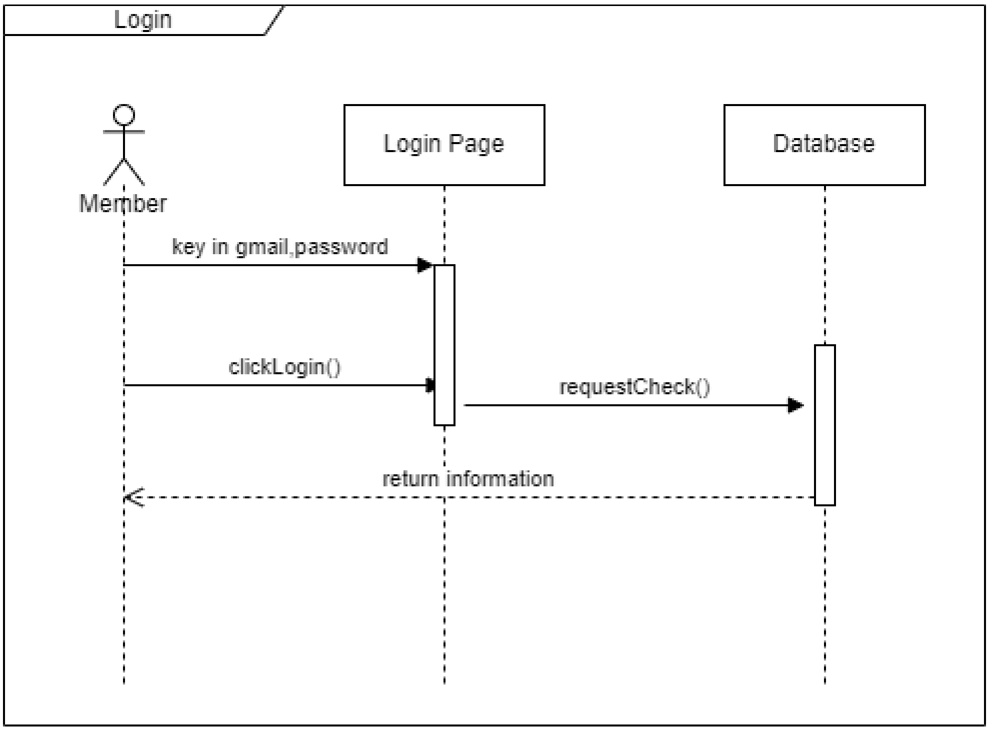


Figure 4.23: Sequence diagram of login

4.2.2.2.12 Register

Based on the figure below, user must provide email, password, phone, and password to register. Once there is no same email exist in system, user will successfully register.

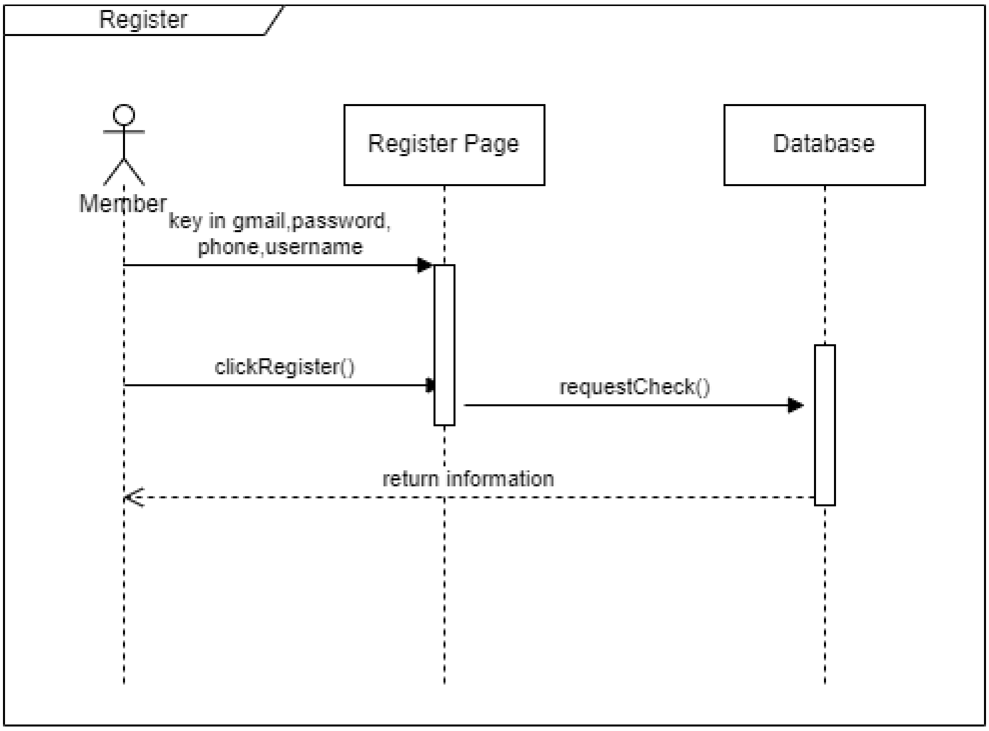


Figure 4.24: Sequence diagram for register

4.2.2.2.13 Reply invitation

Based on the figure below, user allow to reply invitation after admin invited. Click on the settings and view notification.

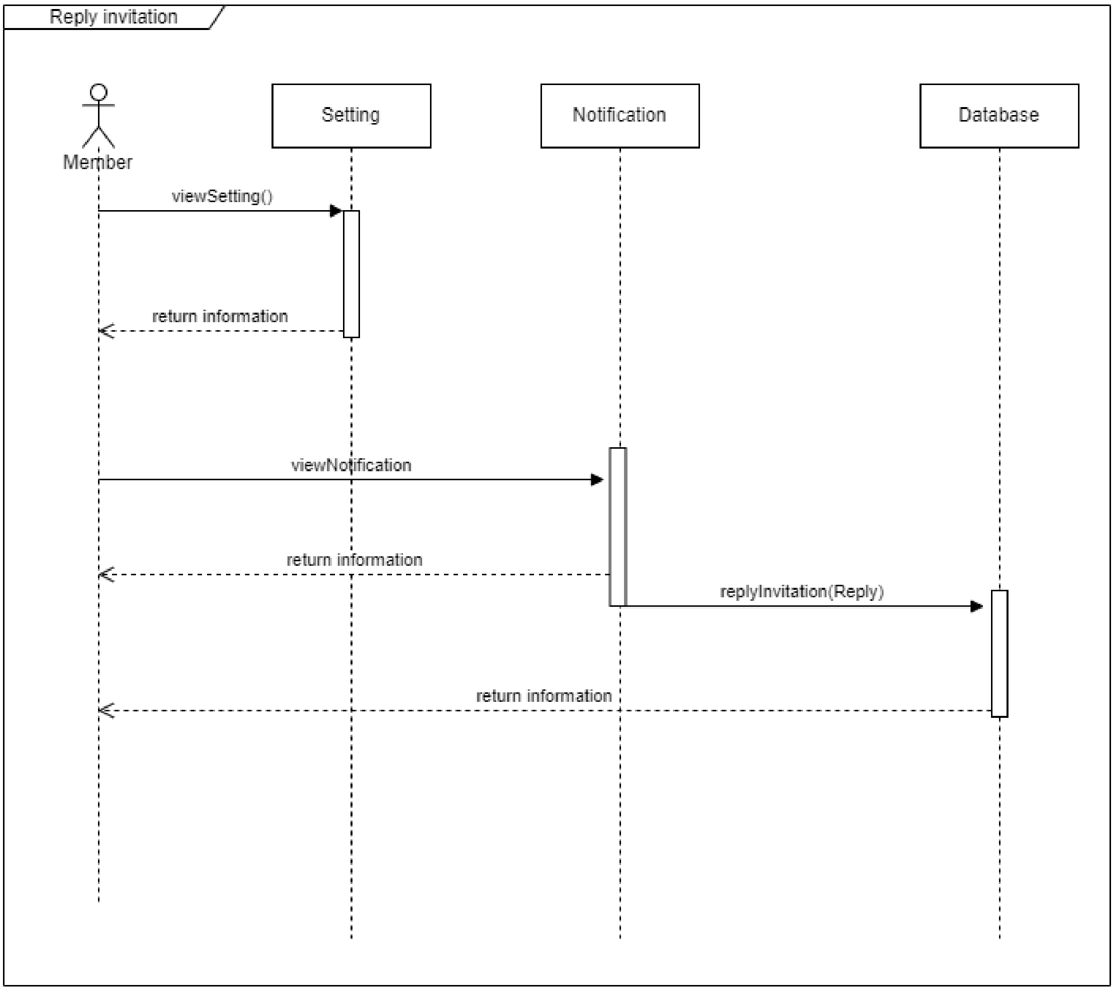


Figure 4.25: Sequence diagram for reply invitation

4.3 Activity Diagram

Activity diagram shows the behaviour of the system with flow chart. The following section will explain the activity diagram for the user in this system.

4.3.1 Admin

4.3.1.1 Change group password

Based on the figure below, admin can change the group password after they login and get into their group. Click update password will be able to update password function.

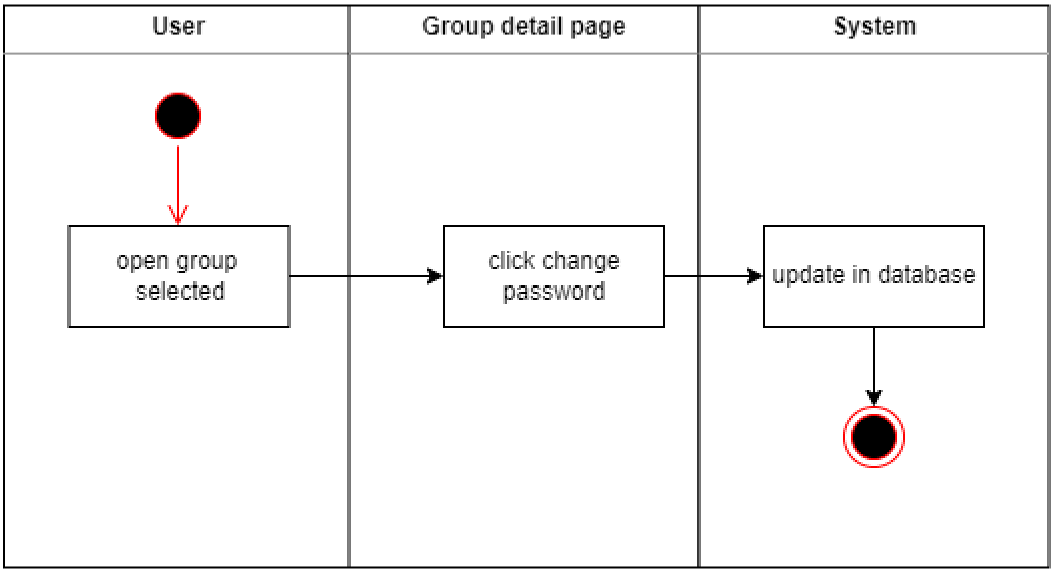


Figure 4.26: Activity diagram for change group password

4.3.1.2 Edit meeting time

Based on the figure below, admin allow to edit meeting time after login. Click on the specific meeting and key in new date and time and click update will be able to update meeting time.

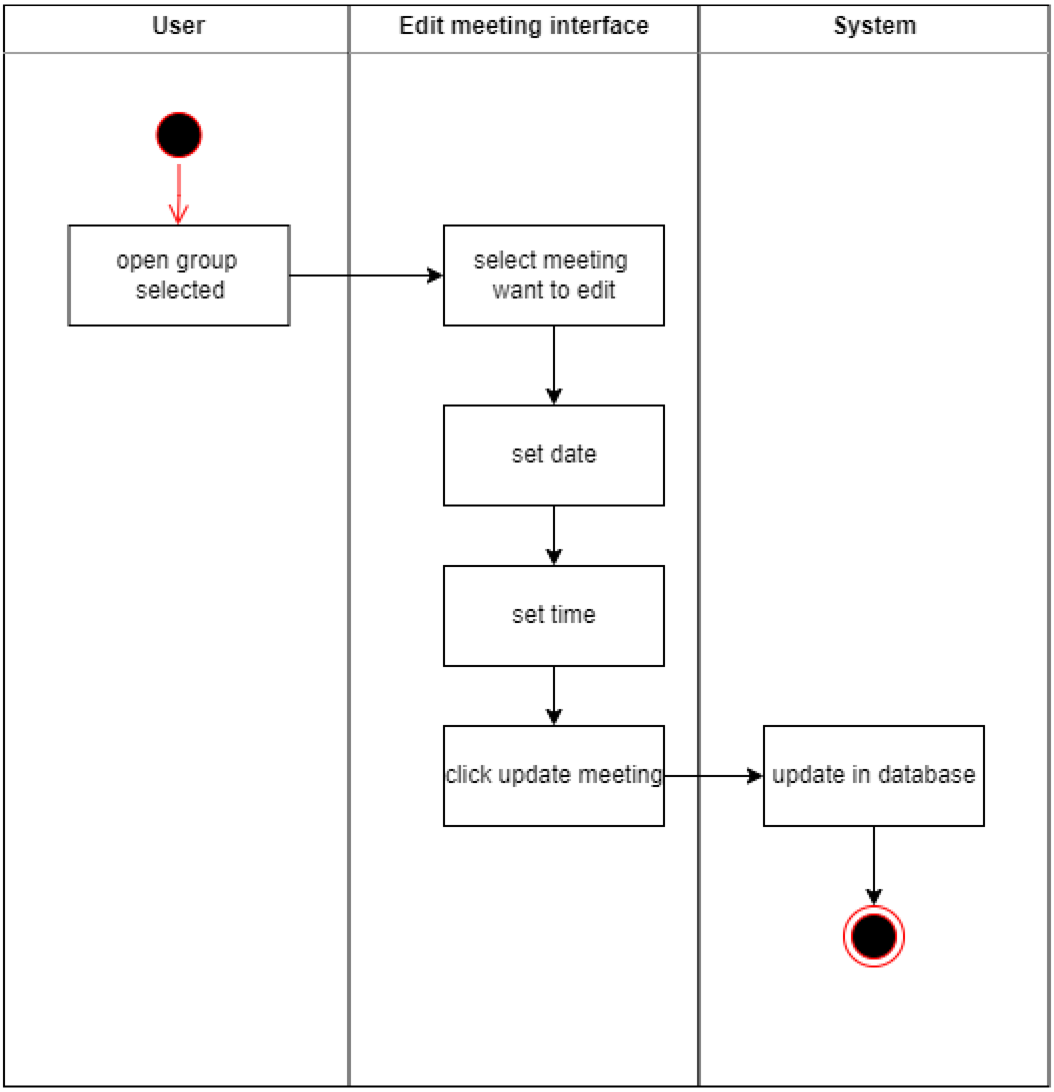


Figure 4.27: Activity diagram for edit meeting time

4.3.1.3 Kick member

Based on the figure below, admin allow to kick member where belongs to their group. Select a specific group to edit and select a member and kick member would allow to kick member.

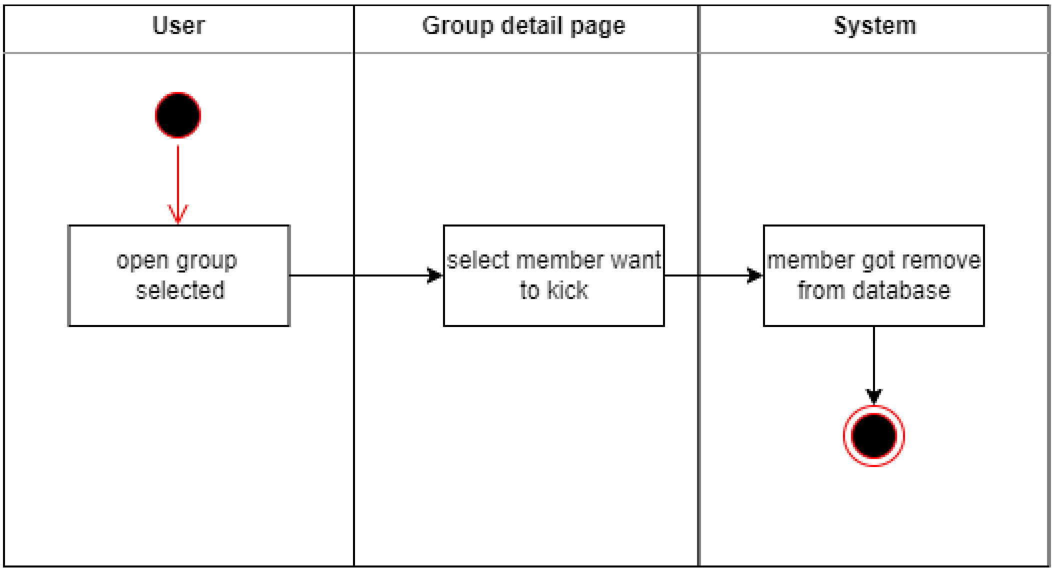


Figure 4.28: Sequence diagram for kick member

4.3.1.4 Create meeting

Based on the figure below, admin allow to create meeting after login. Admin will need to key in date, time, title and assign title to create meeting.

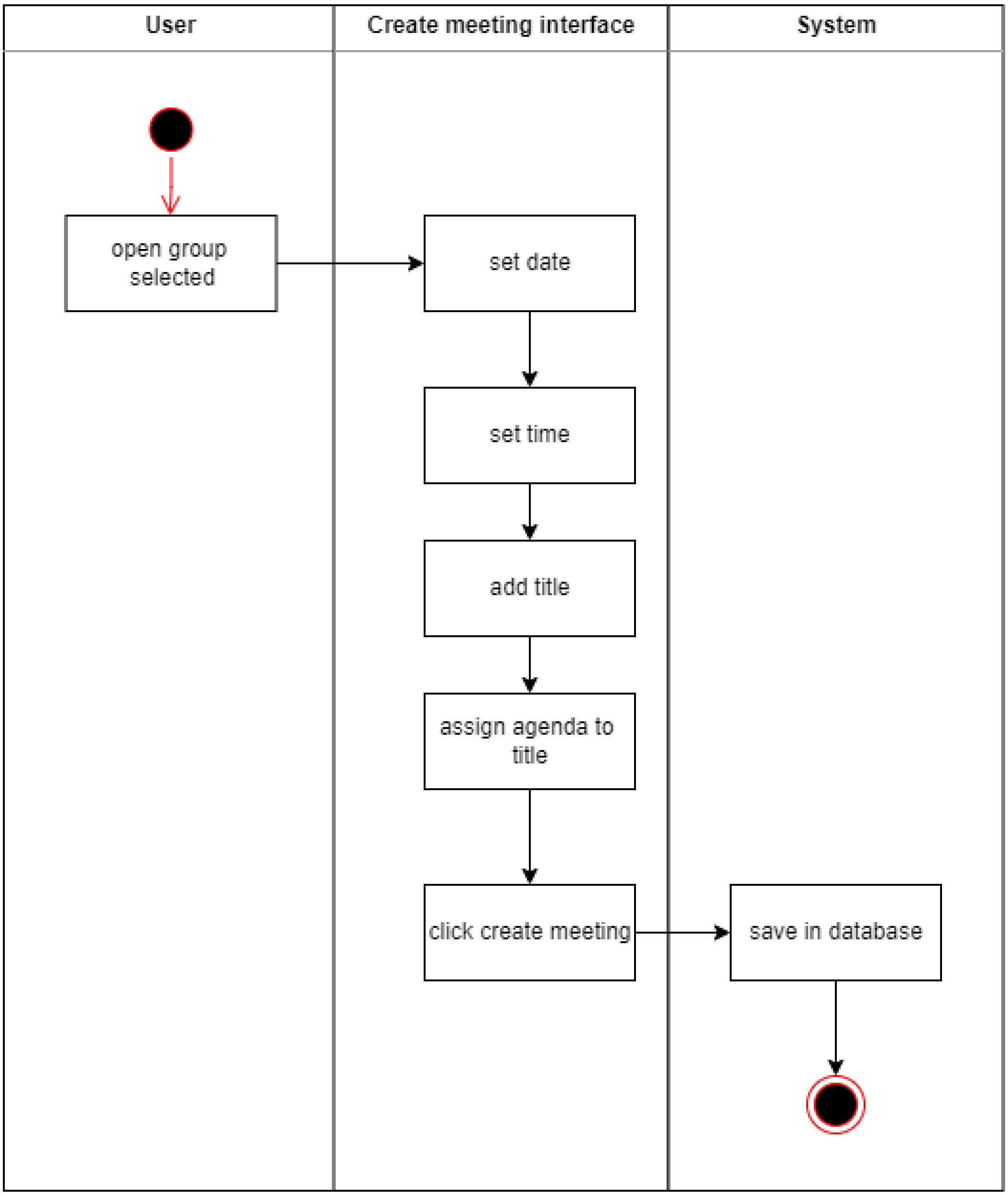


Figure 4.29: Activity diagram of create meeting

4.3.1.5 Set new group admin

Based on the figure below, admin allow to set new admin for the group. Select specific group and select specific member can set them as admin.

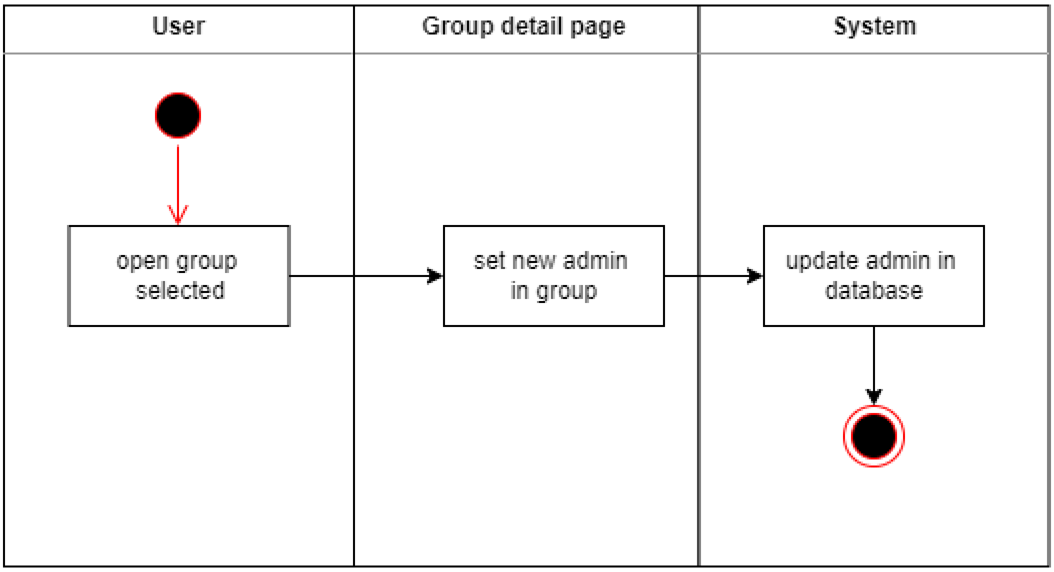


Figure 4.30: Activity diagram of set new group admin

4.3.1.6 Add agenda key point

Based on the figure below, admin allow to add key point for each of the agenda after login. After meeting is created, click on the agenda will allow to create key point.

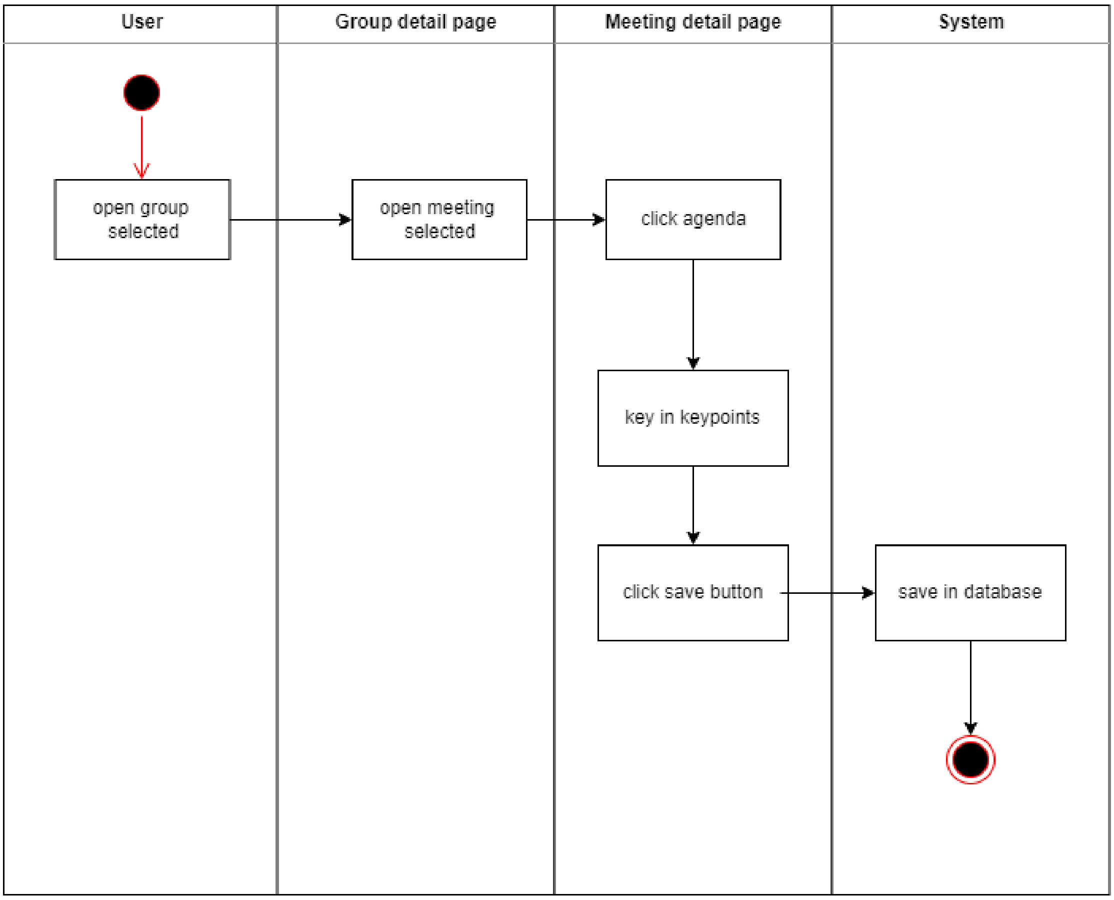


Figure 4.31: Activity diagram of add agenda key point

4.3.1.7 Delete agenda key point

Based on the figure below, admin allow to delete key point for each of the agenda after login. After meeting is created, click on the agenda will allow to delete key point.

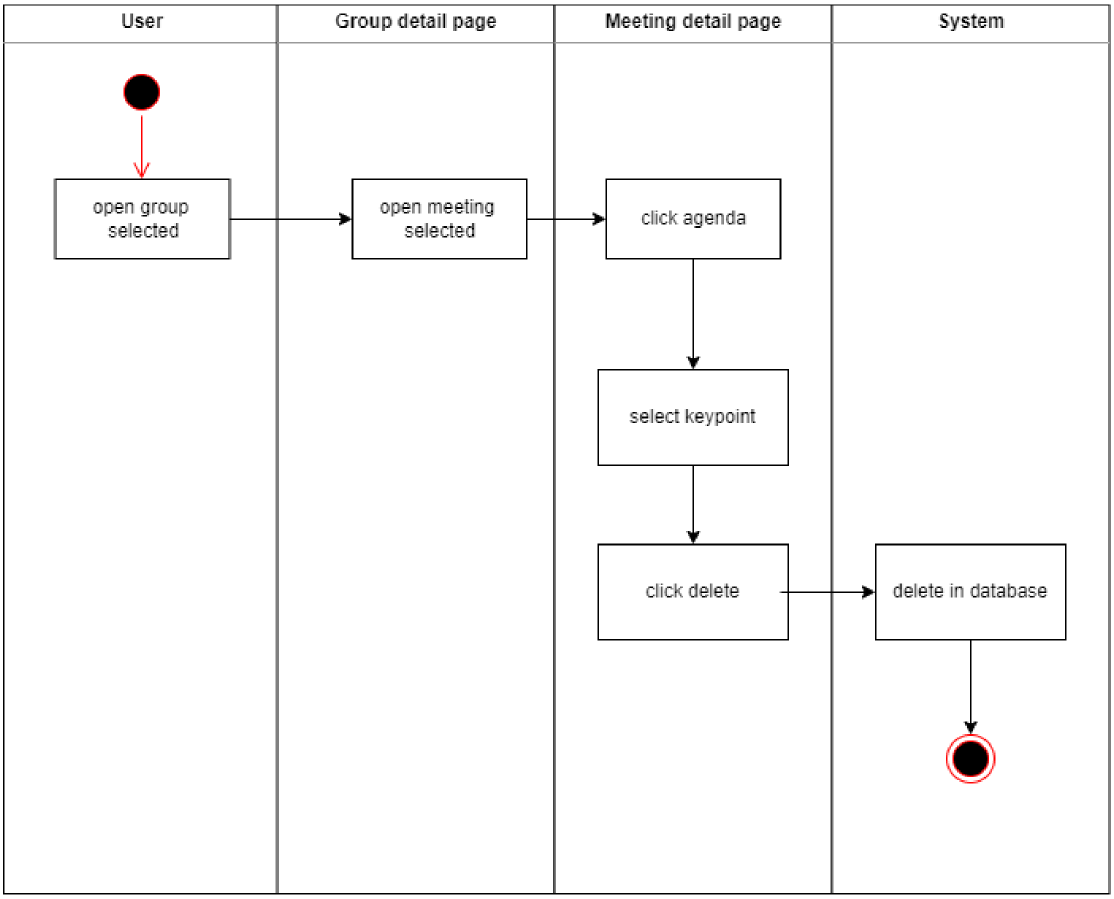


Figure 4.32: Activity diagram of delete agenda key point

4.3.1.8 Edit agenda key point

Based on the figure below, admin allow to edit key point for each of the agenda after login. After meeting is created, click on the agenda will allow to edit key point exist.

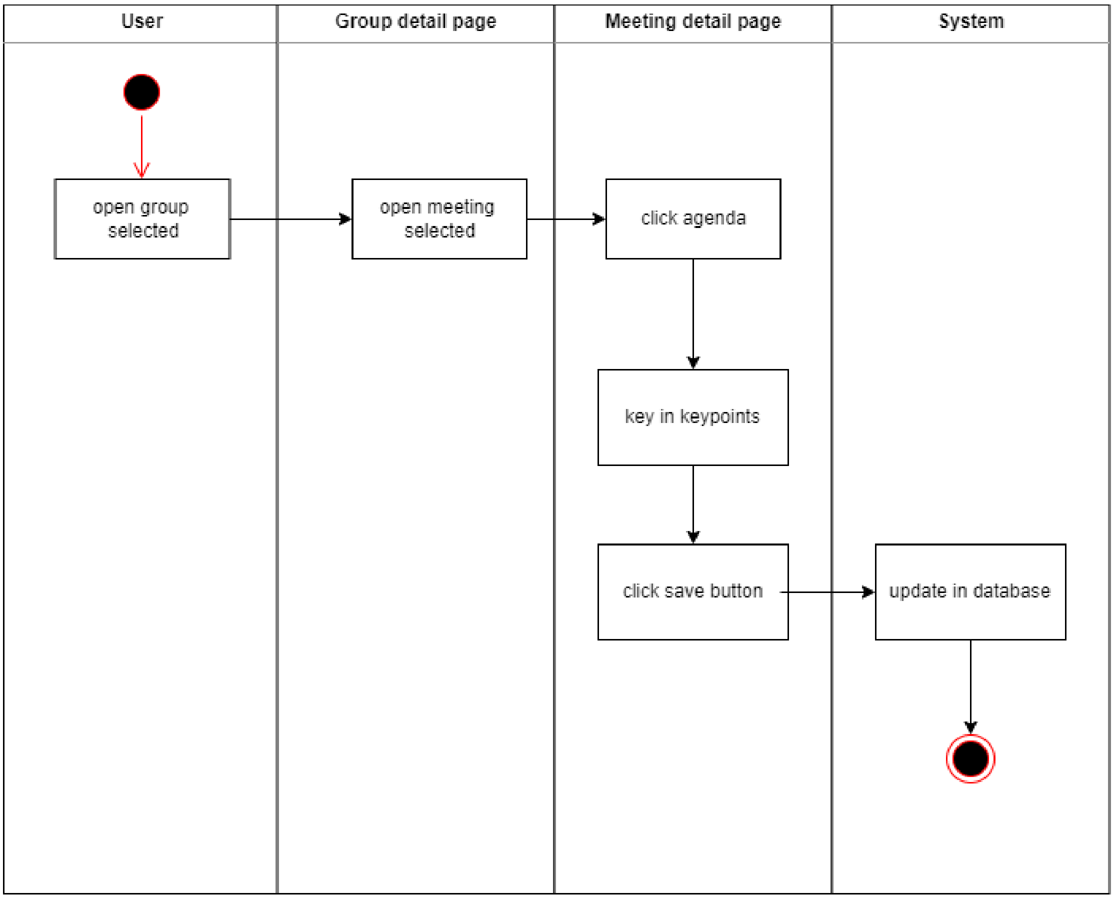


Figure 4.33: Activity diagram for edit agenda key point

4.3.1.9 Generate PDF

Based on the figure below, admin allow to generate PDF after login. Click on generate PDF will display a pdf allow to view.

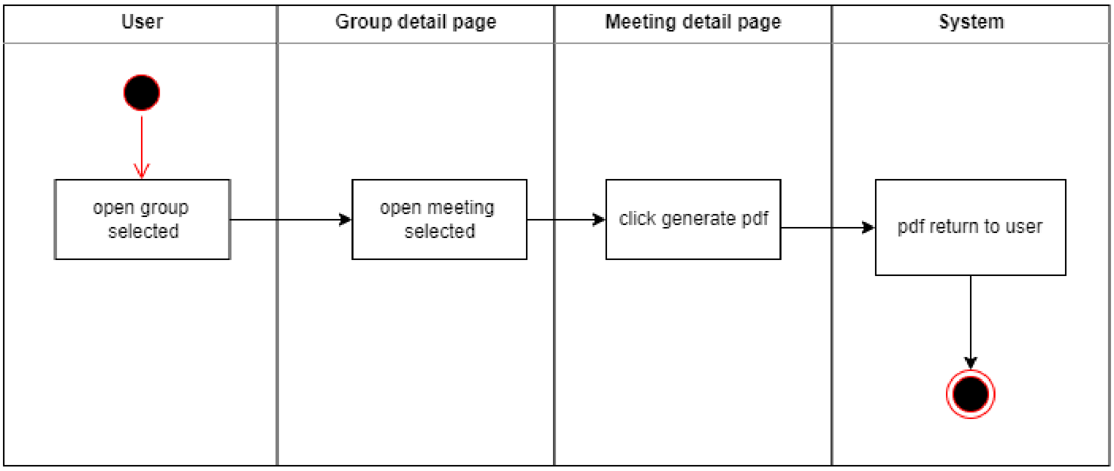


Figure 4.34: Activity diagram for generate PDF

4.3.1.10 Check attendances

Based on the figure below, admin allow to check attendances after login. Select on group and specific meeting and check attendances for each of the user attends.

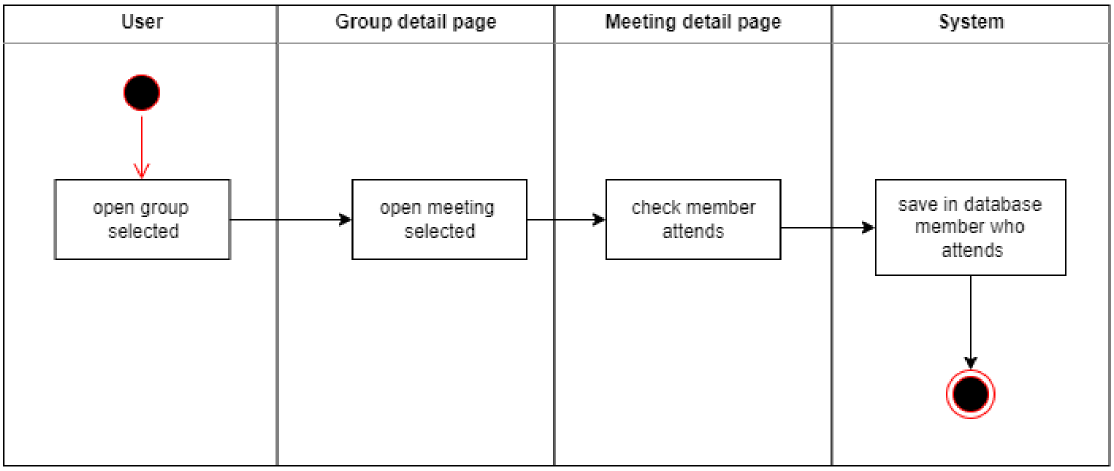


Figure 4.35: Activity diagram for check attendances

4.3.1.11 Send invitations

Based on the figure below, admin allow to send invitations after login. Select on group and group members and send invitations to users.

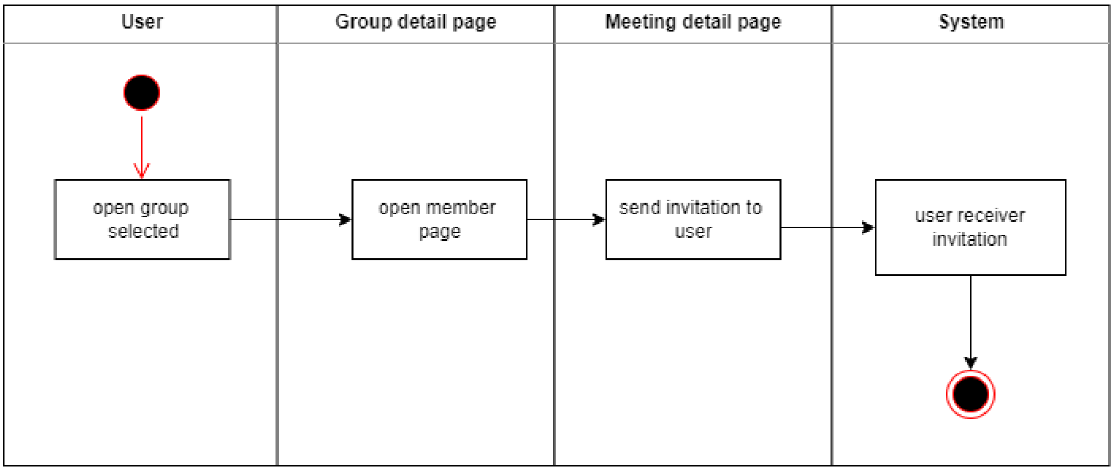


Figure 4.36: Activity diagram for send invitations

4.3.2 Member

4.3.2.1 Add agenda

Based on the figure below, member allow to add agenda after login. Click on the specific group and allow to add agenda to database for future use after attachment file and title is filled in.

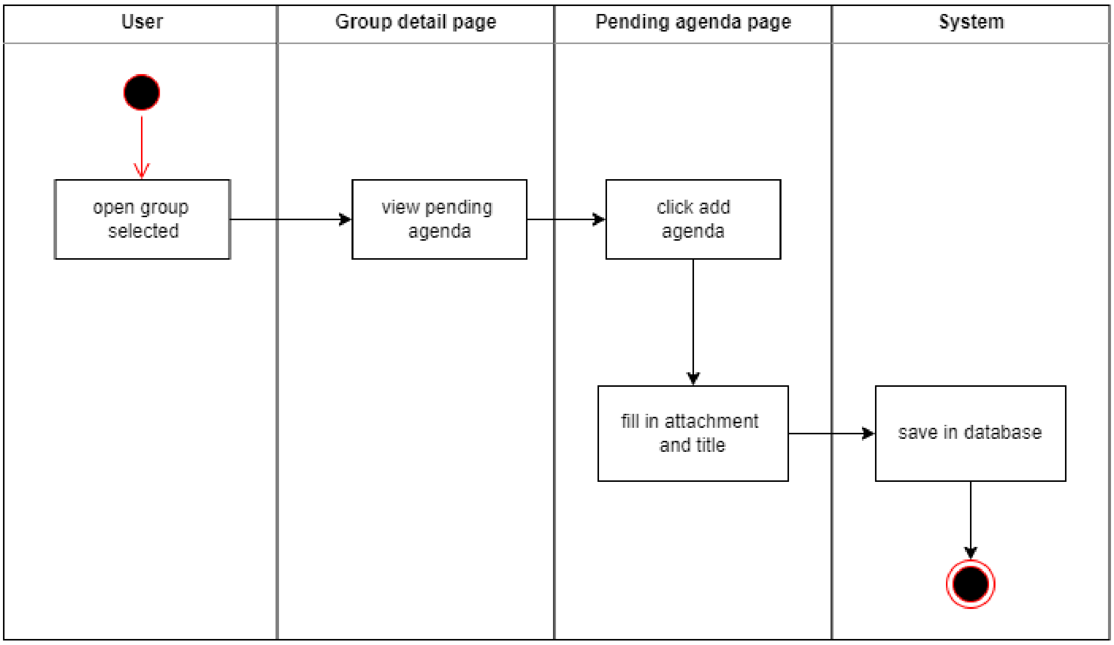


Figure 4.37: Activity diagram for add agenda

4.3.2.2 Delete own agenda

Based on the figure below, member allow to delete own agenda after login. Select group and specific meeting then click on own agenda to delete.

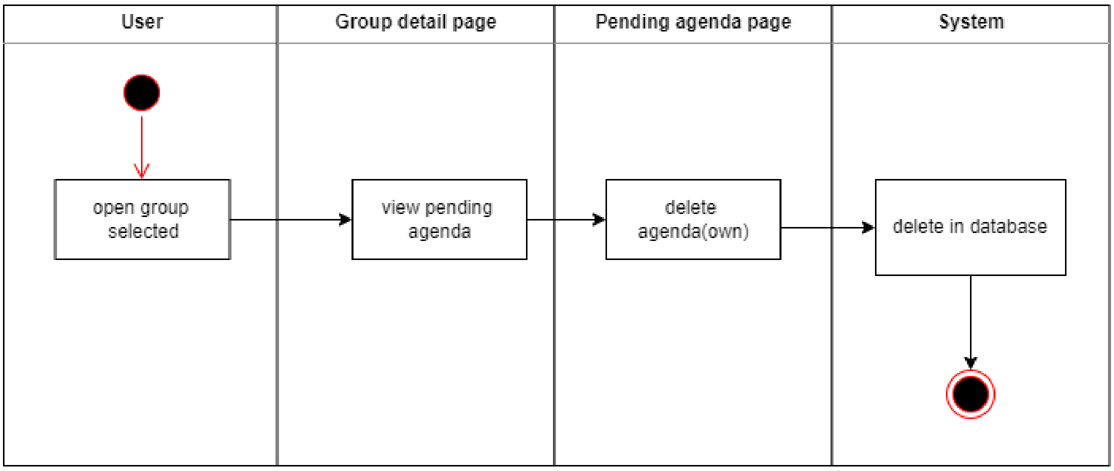


Figure 4.38: Activity diagram for delete own agenda

4.3.2.3 Edit own agenda

Based on the figure below, member allow to edit own agenda after login. Select group and specific meeting then click on own agenda to edit. Fill in new title and attachment file will allow to make edit.

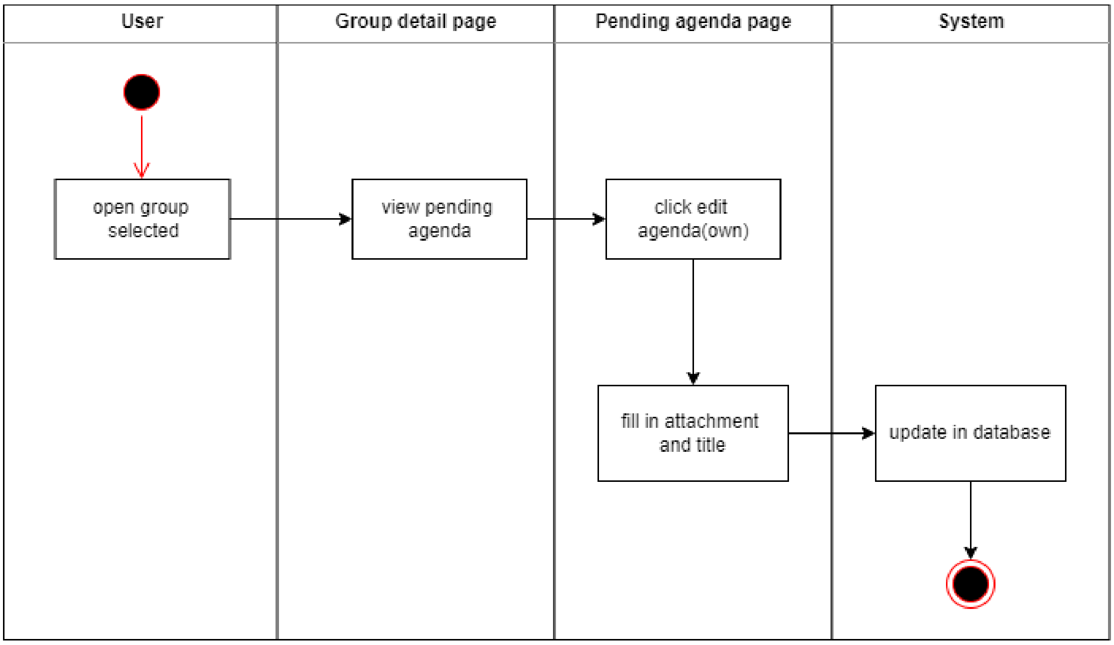


Figure 4.39: Activity diagram for edit own agenda

4.3.2.4 View agenda

Based on the figure below, member allow to view agenda after login. Select group and specific meeting then click on agenda to view all agenda list.

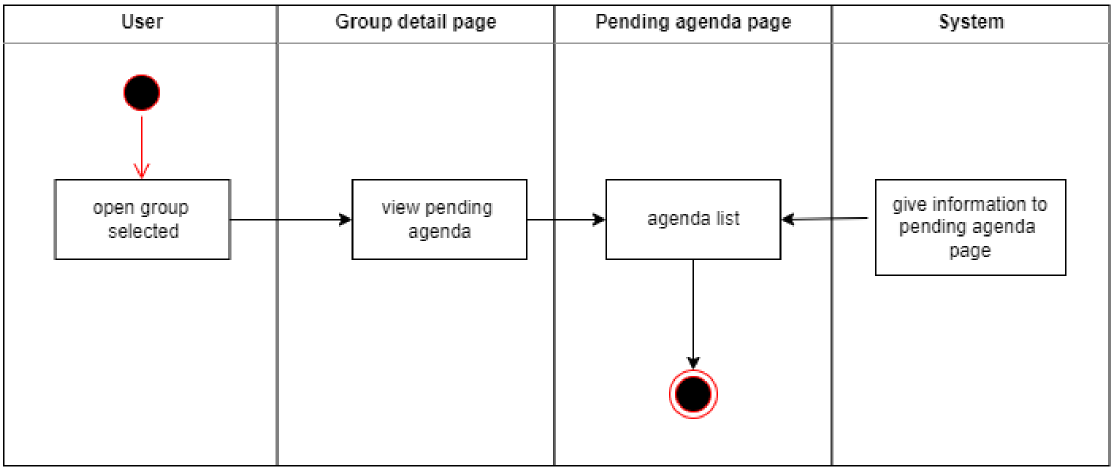


Figure 4.40: Activity diagram for view agenda

4.3.2.5 Join group with password

Based on the figure below, member allow to join group with password after login. They can fill in the group password in textbox and find for fit group password. After the password is correct and member haven’t joined the group, member allow to join the group.

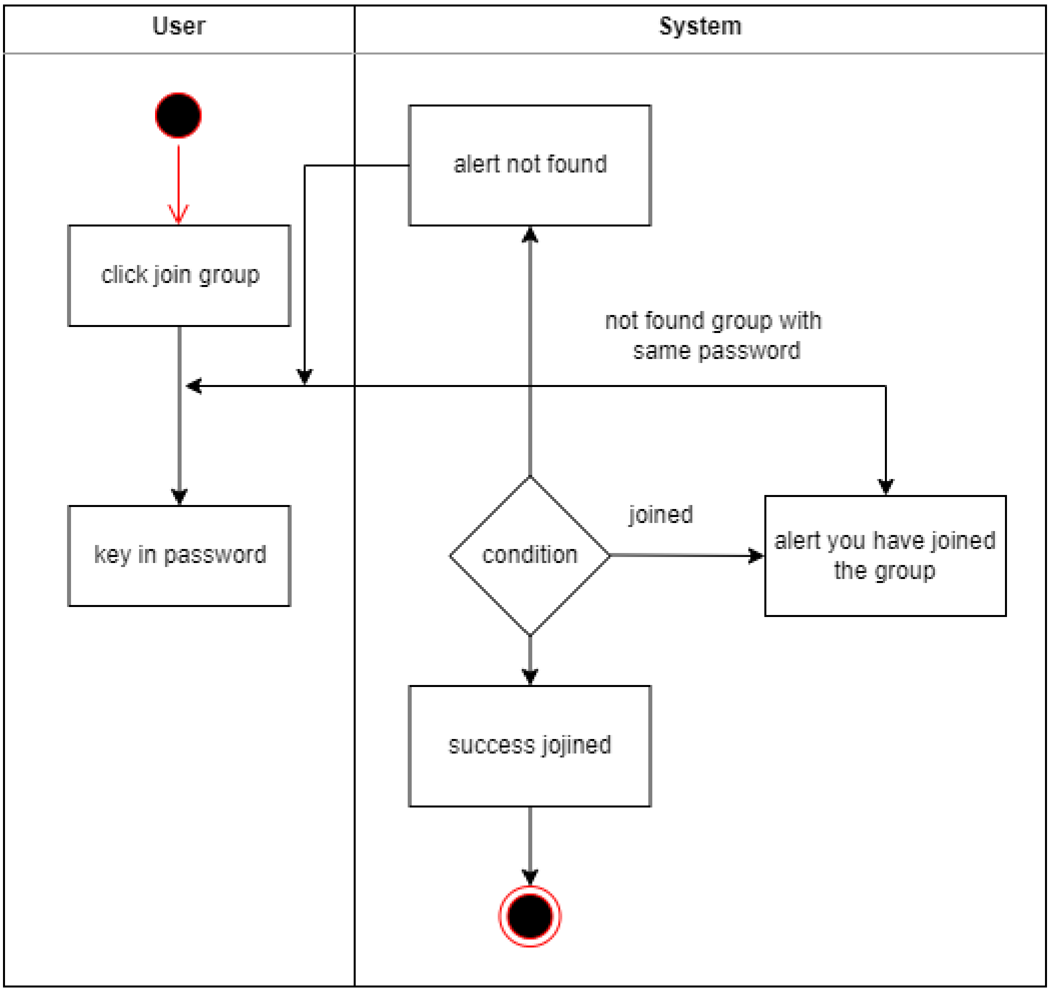


Figure 4.41: Activity diagram for join group with password

4.3.2.6 Left group

Based on the figure below, member allow to left group once they join the group. Select on the left group option would allow to leave the group.

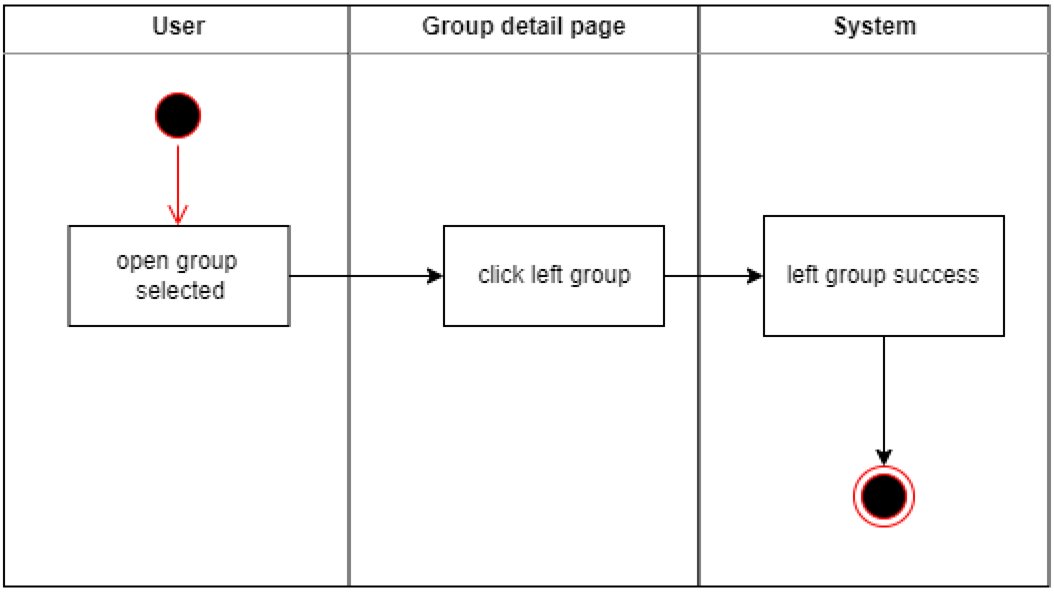


Figure 4.42: Activity diagram for left group

4.3.2.7 View group meeting

Based on the figure below, member allow to view group meeting once they join the group. Click on the meeting will be able to view meeting list.

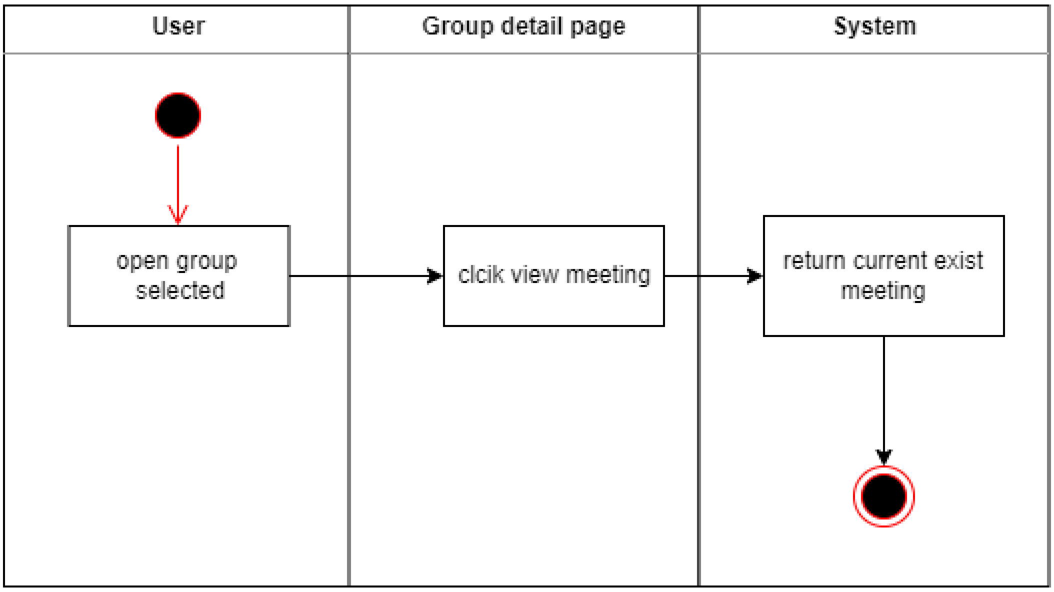


Figure 4.43: Activity diagram for view group meeting

4.3.2.8 View group member

Based on the figure below, member allow to view group member after joined the group. Select group member on the selection would allow to join group.

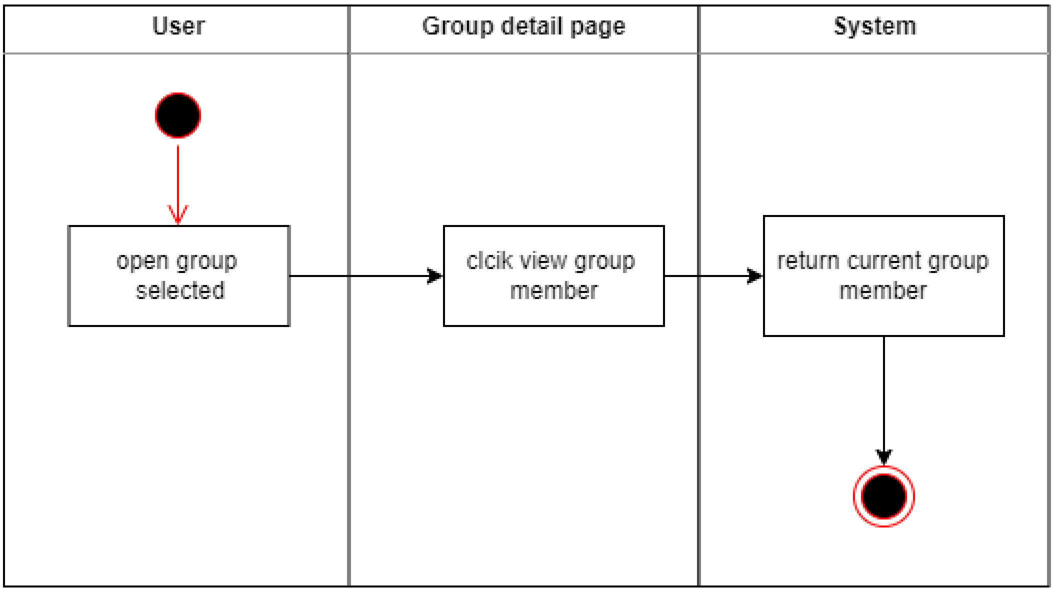


Figure 4.44: Activity diagram for view group member

4.3.2.9 View meeting minutes

Based on the figure below, member allow to view detail of meeting minutes once they joined the group. Select on specific meeting minutes allow to view detail of meeting minutes.

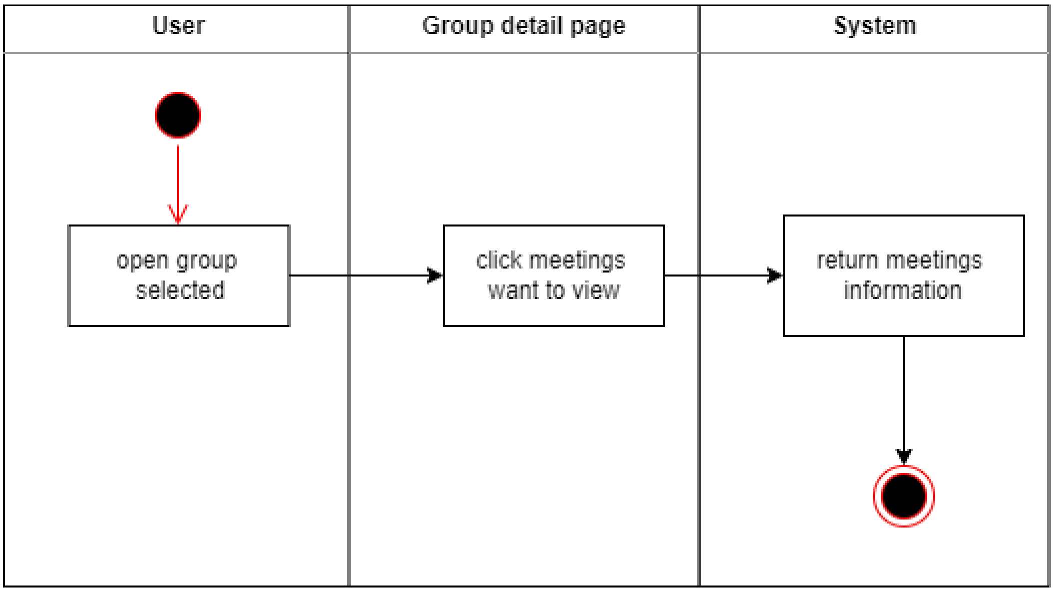


Figure 4.45: Activity diagram of view meeting minutes

4.3.2.10 View meeting schedules

Based on the figure below, member allow to view meeting schedules after they login. Click on the meeting schedules allow to view meeting schedules.

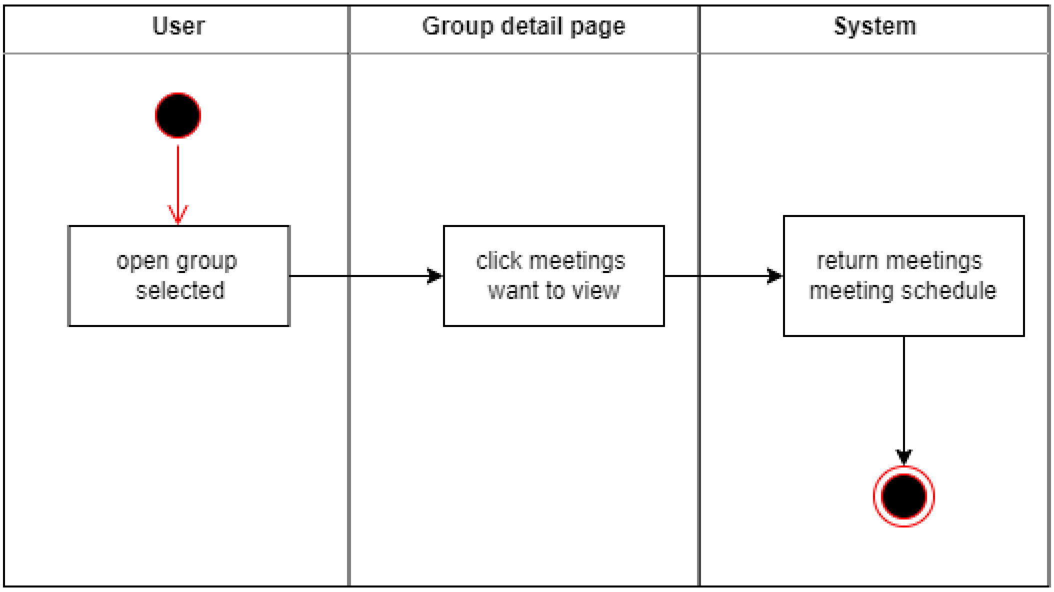


Figure 4.46: Activity diagram for view meeting schedules

4.3.2.11 Login

Based on the figure below, user must provide email and password to login. This system require login to use all the function. Once there is exist account in database, member allow to login to the function.

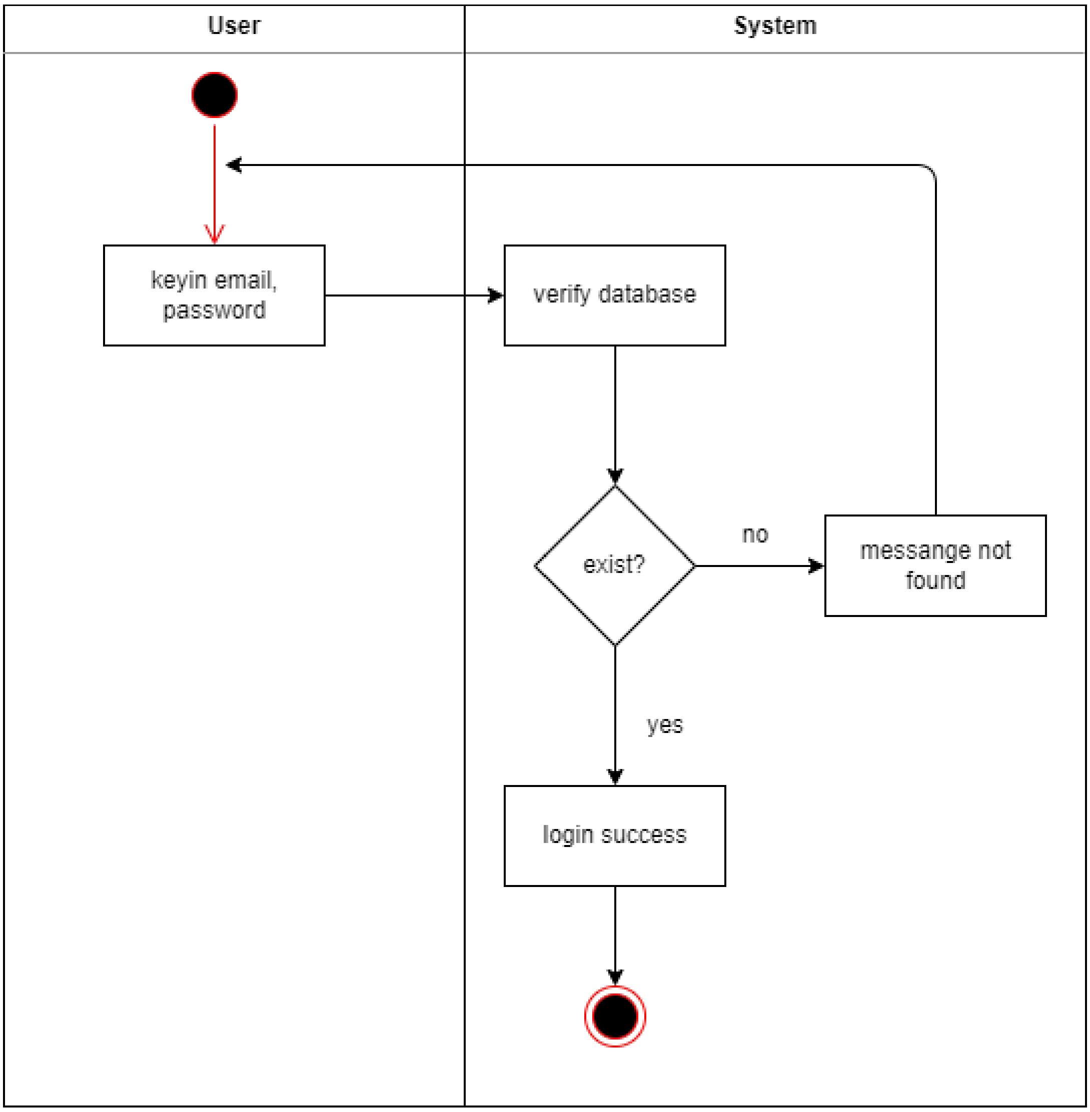


Figure 4.47: Activity diagram of login

4.3.2.12 Register

Based on the figure below, user must provide email, password, phone, and password to register. Once there is no same email exist in system, user will successfully register.

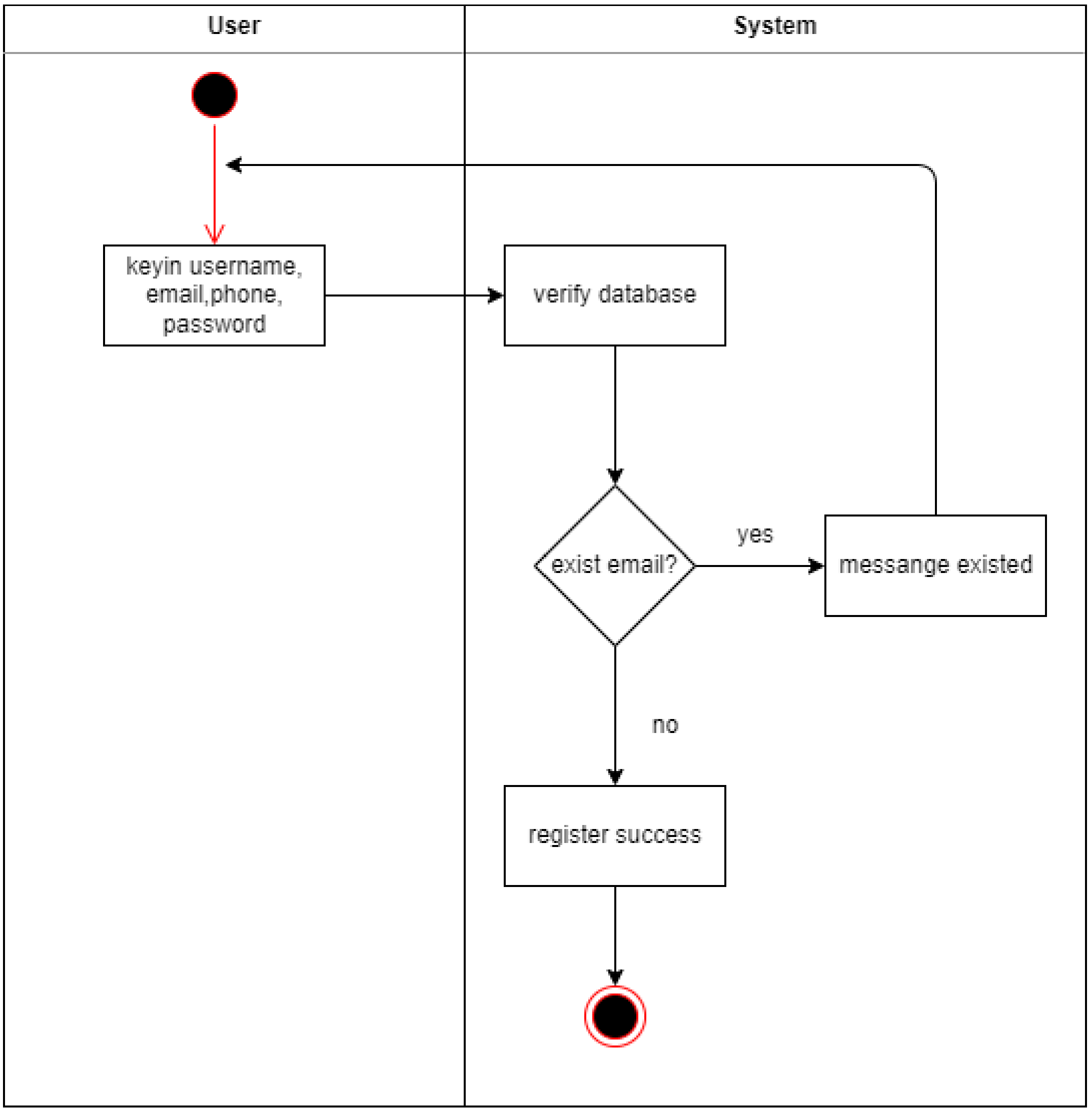


Figure 4.48: Activity diagram for register

4.3.2.13 Reply invitation

Based on the figure below, user allow to reply invitation after admin invited. Click on the settings and view notification.

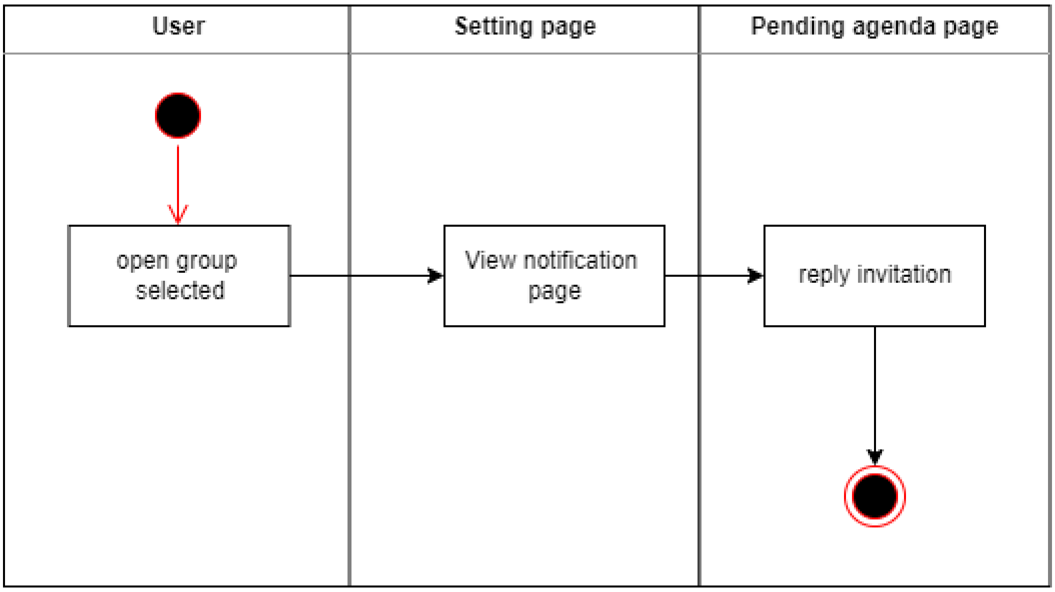


Figure 4.49: Activity diagram for reply invitation

4.4 System Architecture

The web server is the core of this system where the web application is boosted. The web application is responsible in controlling the process of meeting minutes. The web servers also provide auth function and generate pdf after the meeting record is done.

4.4.1 Class Diagram

In Unified Modelling Language (UML), a class diagram is a type of static structure diagram that illustrates a system structure based on the system classes, class attribute, methods, and class object relationship. Figure 4.46 below shows the class diagram for this system.

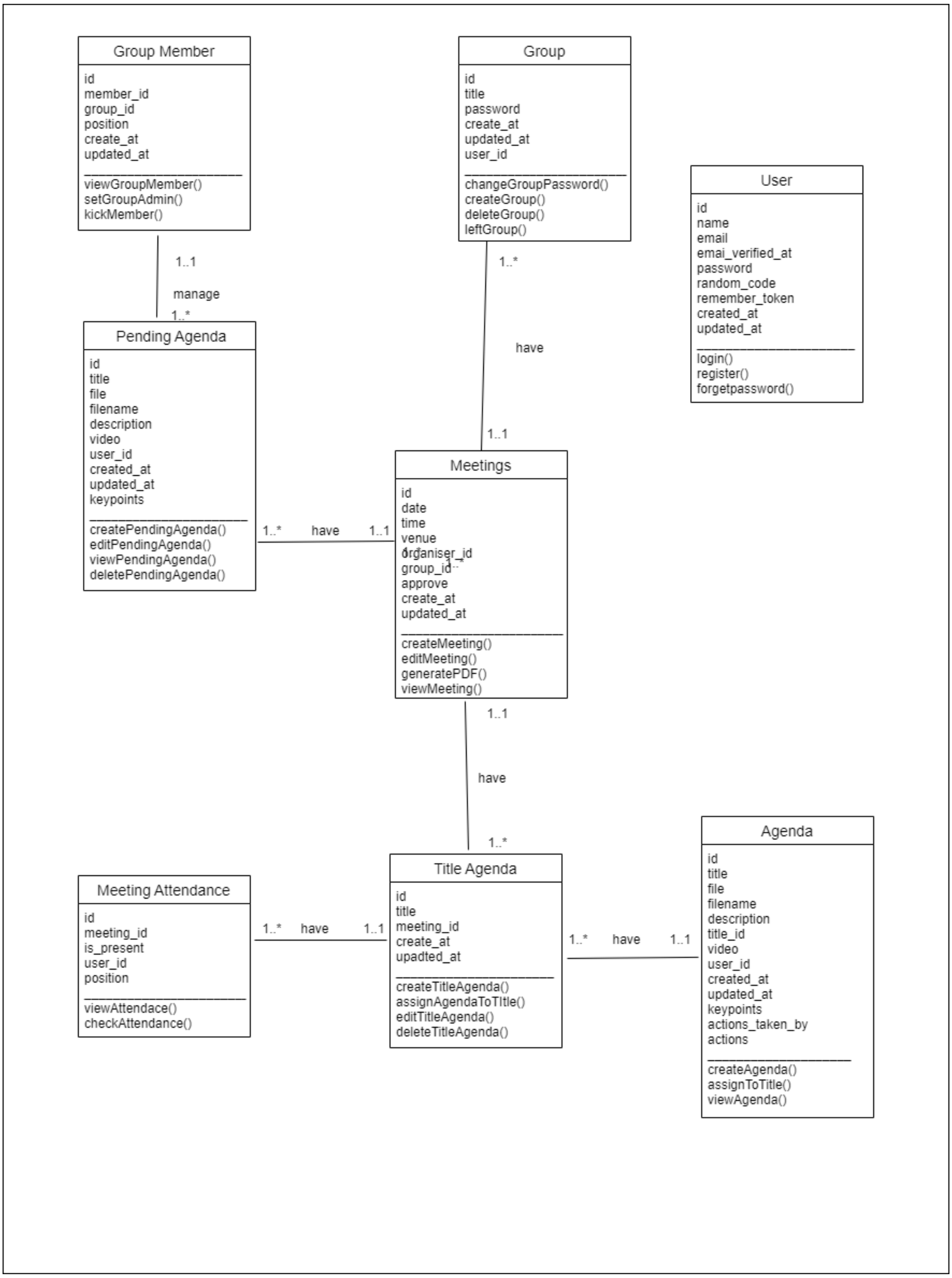


Figure 4.50: Class diagram of meeting minutes system

4.5 Database Design

Database design is a series of process comprising of planning, development, implementation, and maintenance system. A system with well-designed database can be easily maintain and achieve better data consistency.

4.5.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) illustrates how different entity work with each other in a system. ERD helps developers to visualize the starting point in a database design to determine the system requirement.

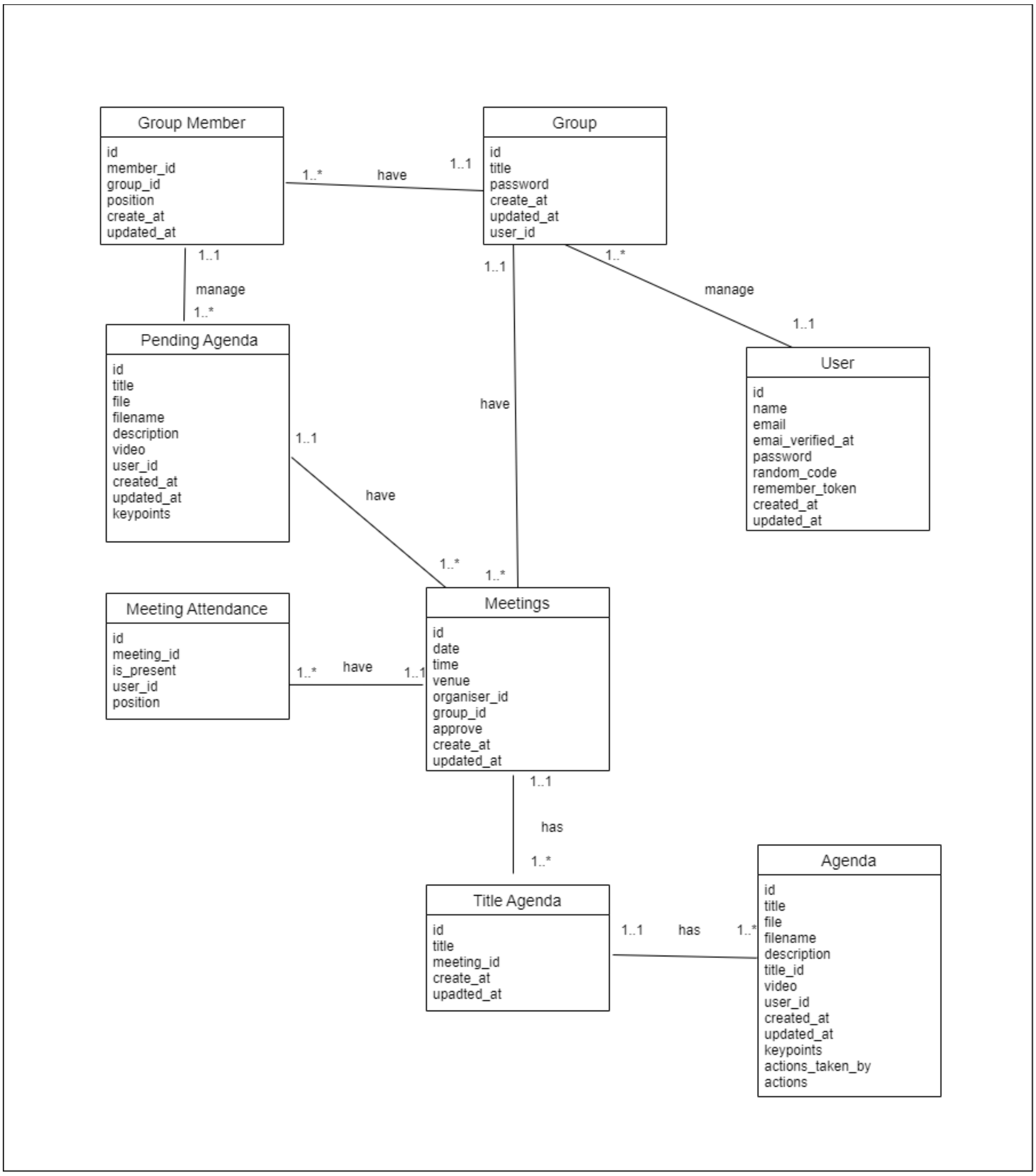
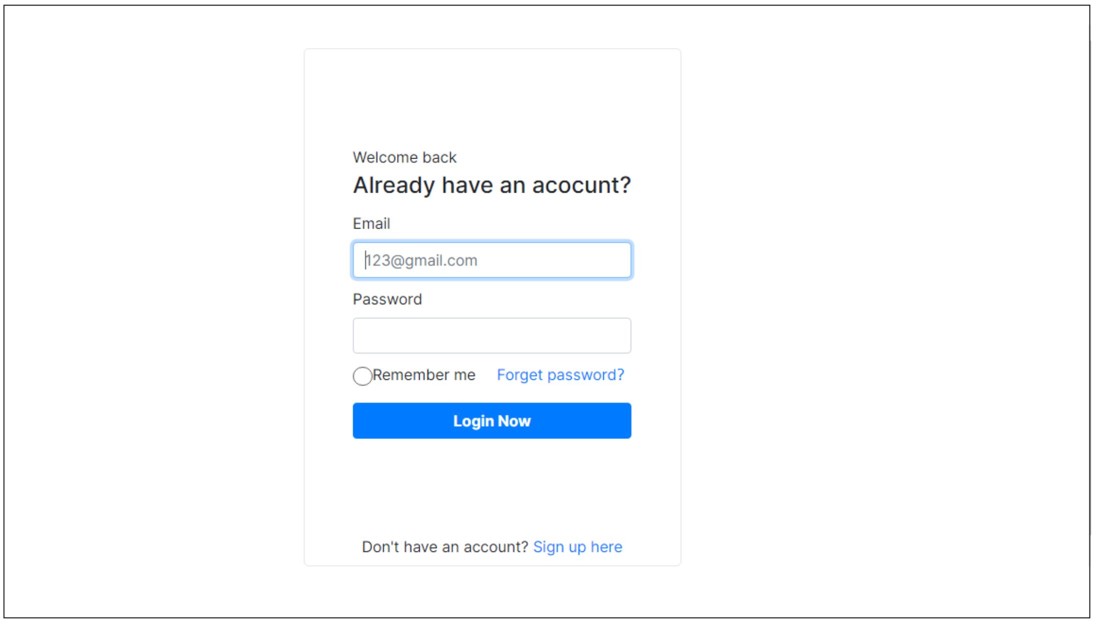


Figure 4.51: Entity Relationship Diagram (ERD) for meeting minute system

4.6 User Interface Design

User Interface (UI) is the bridge that connects the users and the system. All inputs and output are display in the Ui. Hence, a well-designed UI should allow the users to easily understand the software system and perform any desired functions at ease.



Welcome back

Already have an acocunt?

Email

123@gmail.com

Password

○Remember me Forget password?

Login Now

Don't have an account? Sign up here

Figure 4.52: The login page for all users.

Figure 4.48 shows the login page for all users. Below link able for users who haven’t registered to register.

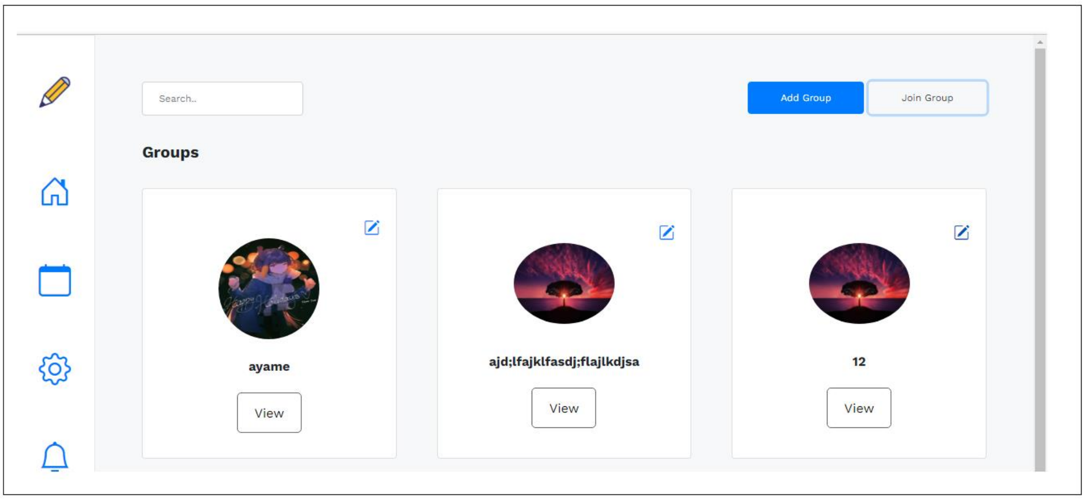


Figure 4.53: The main page of the system.

Figure 4.49 show the main page of the system. It allows to show current group joined in system. Left side able to see meeting schedules and settings.

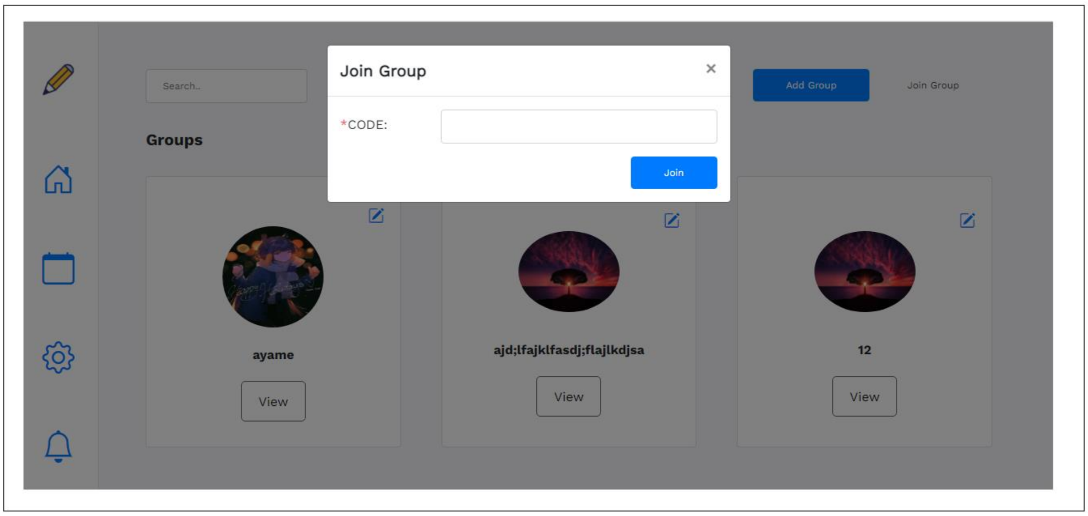
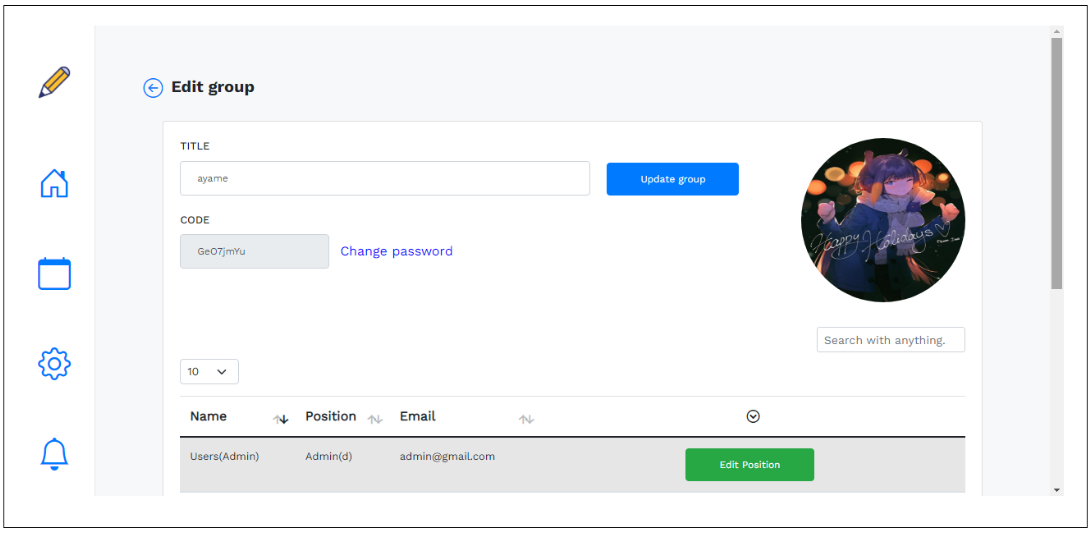


Figure 4.54: The join group page.

Figure 4.50 shows the join group page. After code is fill in and join button clicked, user able to join group if there is group exist.



Edit group

TITLE

ayame

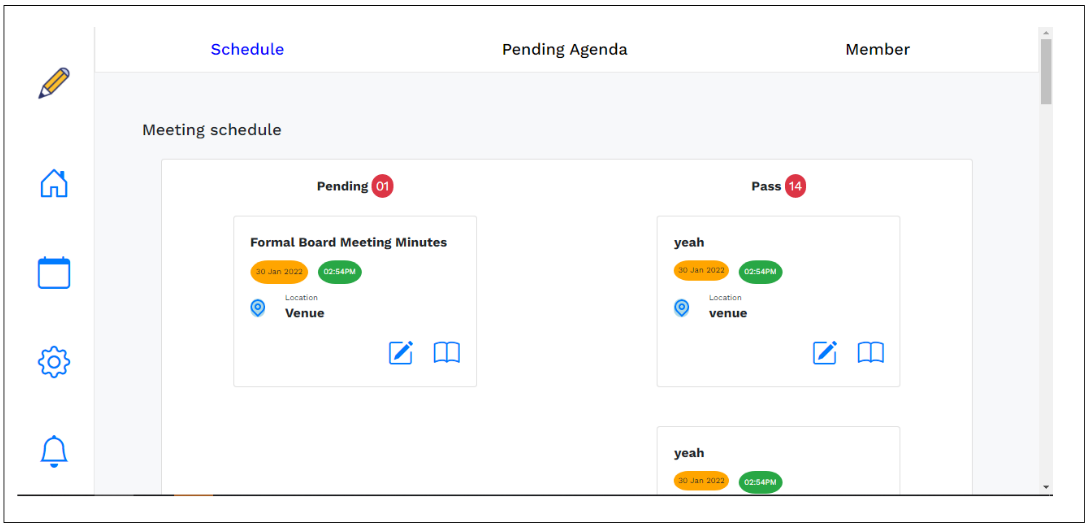
CODE

GeO7jmYu

Name	Position	Email	
Users(Admin)	Admin(d)	admin@gmail.com	Edit Position

Figure 4.55: Edit group page

Figure 4.51 show the edit group page. Admins allow to edit position of member, kick member, set admin of member and update password or title of the group in this page.



Meeting schedule

Pending 01

Formal Board Meeting Minutes
30 Jan 2022 02:54PM
Location
Venue

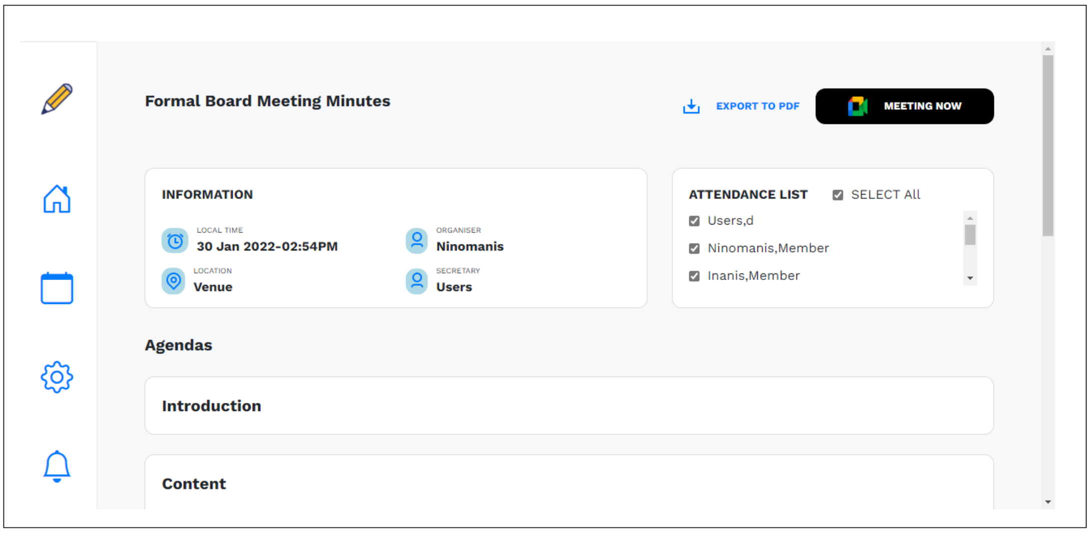
Pass 14

yeah
30 Jan 2022 02:54PM
Location
venue

yeah
30 Jan 2022 02:54PM

Figure 4.52: Group detail page

Figure 4.52 shows the group detail page. It allows to show current pending meeting and pass meeting. The edit icon is only applicable for admin in group.



Formal Board Meeting Minutes

INFORMATION

LOCAL TIME 30 Jan 2022-02:54PM

ORGANISER Ninomanis

LOCATION Venue

SECRETARY Users

ATTENDANCE LIST ☑ SELECT All

☑ Users,d
☑ Ninomanis,Member
☑ Inanis,Member

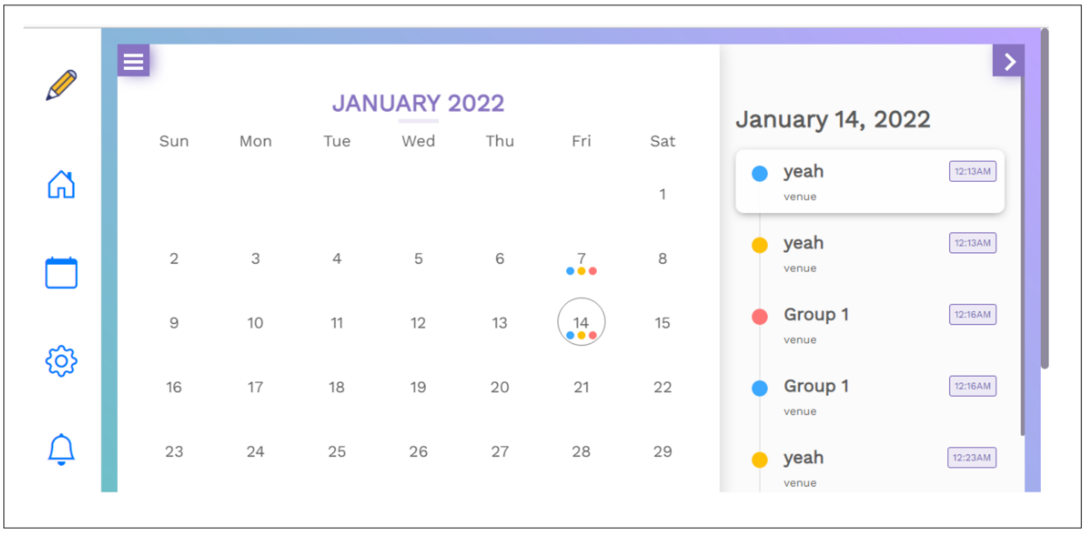
Agendas

Introduction

Content

Figure 4.56: Meeting detail page

Figure 4.53 shows the meeting detail page. Admins allow to control key points in each of the agenda and check attendances for who attends. After save is clicked, the pdf allows to generate for all members in group.



JANUARY 2022

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

January 14, 2022

●	yeah venue	12:13AM
●	yeah venue	12:13AM
●	Group 1 venue	12:16AM
●	Group 1 venue	12:16AM
●	yeah venue	12:23AM

Figure 4.57: Meeting schedule page

Figure 4.54 shows the meeting schedule page. Users allow to check their current schedule for each of the meetings in this page.

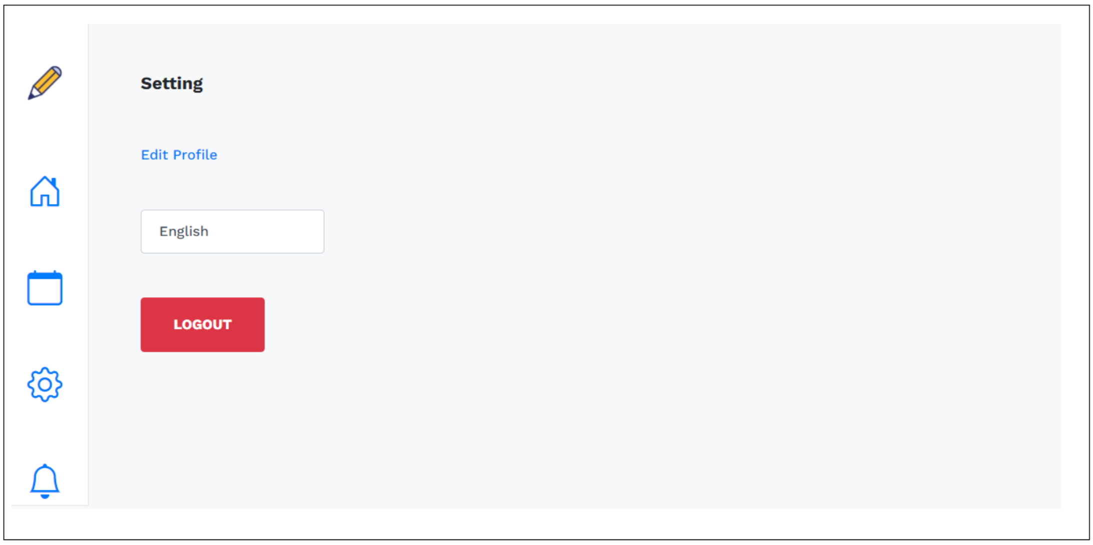


Figure 4.58: Settings page

Figure 4.55 shows settings page. Members allow to edit profile, change language, and logout during this page.

4.7 Chapter Summary

This chapter explain the components of this online minute meeting system. The first part of this system describes the system modelling such as use case and sequence diagram. The user interface design is also discussed in this chapter.

5.Chapter Five: System Implementation

5.1 Introduction

This chapter will discuss the system implementation of this project including the development environment and screenshot of coding in system.

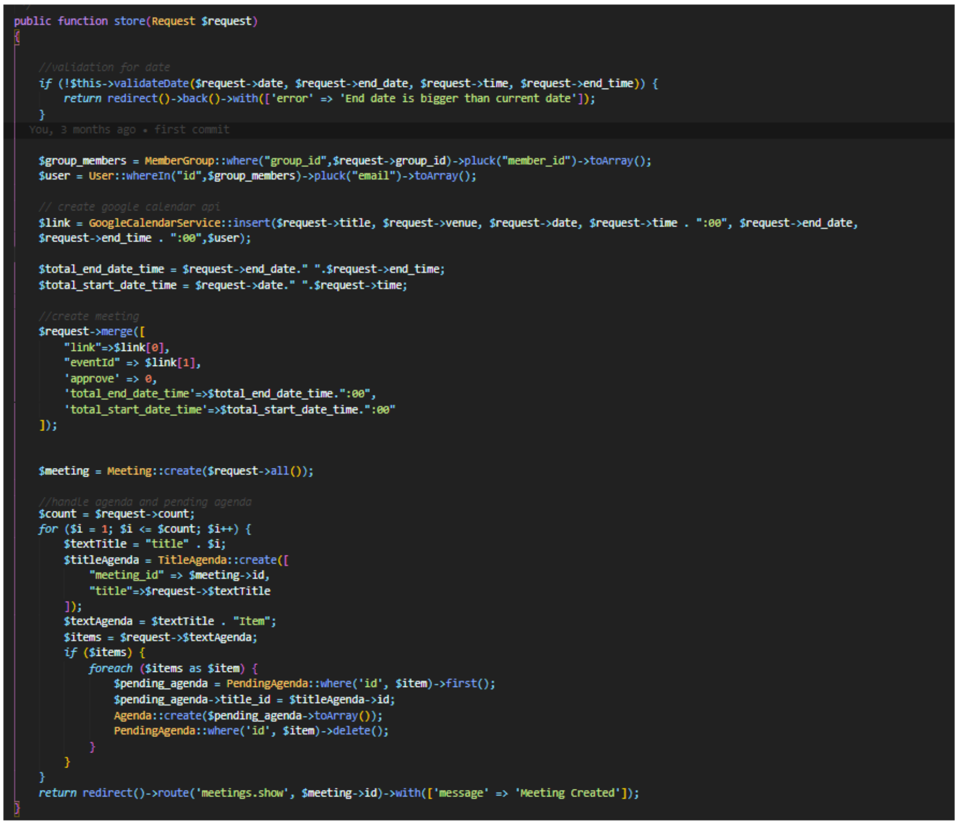
5.2 Development Environment

Visual Studio Code is used to develop this system. Some external plugins and library package is installed to ensure the system development process runs smoothly. The main programming language used in this system is PHP. On the other hand, HML, CSS and JavaScript are used to build the frontend of the system. The database used is MySQL.

5.3 System Implementation

The main phase in system development is the implementation phase. The programming code is implemented in this phase which the user can interact with the system.

5.3.1 Create meeting

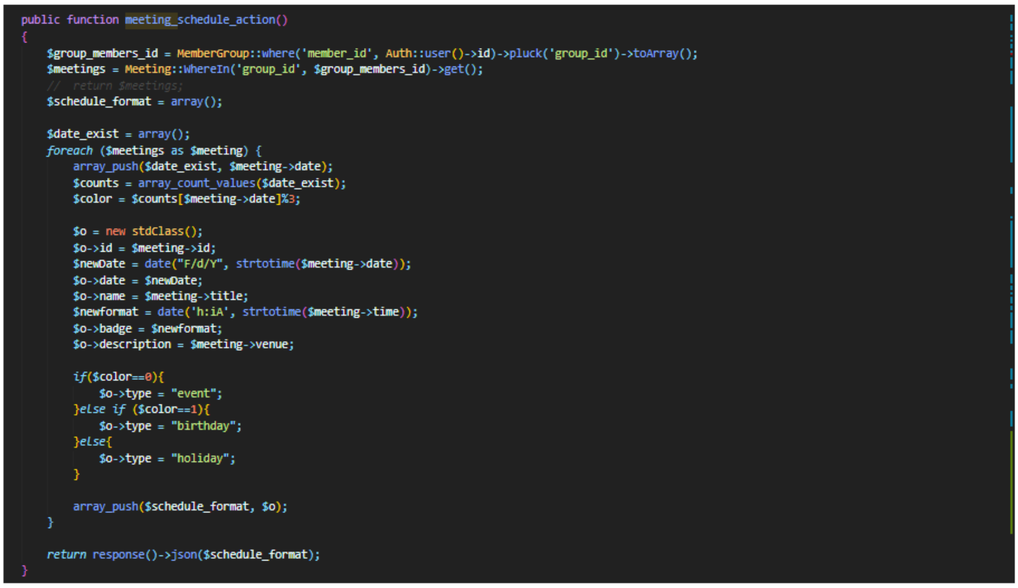


```
public function store(Request $request)
{

    //validation for date
    if (!$this->validateDate($request->date, $request->end_date, $request->time, $request->end_time)) {
        return redirect()->back()->with(['error' => 'End date is bigger than current date']);
    }
  You, 3 months ago • first commit

    $group_members = MemberGroup::where("group_id",$request->group_id)->pluck("member_id")->toArray();
    $user = User::whereIn("id",$group_members)->pluck("email")->toArray();

    // create google calendar api
    $link = GoogleCalendarService::insert($request->title, $request->venue, $request->date, $request->time . ":00", $request->end_date,
    $request->end_time . ":00",$user);

    $total_end_date_time = $request->end_date." ".$request->end_time;
    $total_start_date_time = $request->date." ".$request->time;

    //create meeting
    $request->merge([
        "link"=>$link[0],
        "eventId" => $link[1],
        'approve' => 0,
        'total_end_date_time'=>$total_end_date_time.":00",
        'total_start_date_time'=>$total_start_date_time.":00"
    ]);


    $meeting = Meeting::create($request->all());

    //handle agenda and pending agenda
    $count = $request->count;
    for ($i = 1; $i <= $count; $i++) {
        $textTitle = "title" . $i;
        $titleAgenda = TitleAgenda::create([
            "meeting_id" => $meeting->id,
            "title"=>$request->$textTitle
        ]);
        $textAgenda = $textTitle . "Item";
        $items = $request->$textAgenda;
        if ($items) {
            foreach ($items as $item) {
                $pending_agenda = PendingAgenda::where('id', $item)->first();
                $pending_agenda->title_id = $titleAgenda->id;
                Agenda::create($pending_agenda->toArray());
                PendingAgenda::where('id', $item)->delete();
            }
        }
    }
    return redirect()->route('meetings.show', $meeting->id)->with(['message' => 'Meeting Created']);
}
```

Figure 5.1: Code for create meeting

Figure 5.1 shows the code for create meeting. When admin has set the date, time, and titles for each agenda, it will be able to create meeting. First, it will call Google Calendar Service API to generate meeting link on Google meet, then it will save each of the information gained in form to create meeting. The pending agenda will be removed to agenda as it already set in meeting.

5.3.2 Meeting Schedule

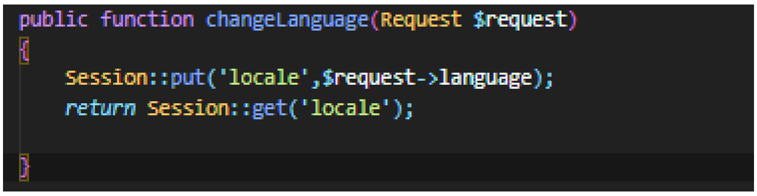


```php
public function meeting_schedule_action()
{
    $group_members_id = MemberGroup::where('member_id', Auth::user()->id)->pluck('group_id')->toArray();
    $meetings = Meeting::WhereIn('group_id', $group_members_id)->get();
    //  return $meetings;
    $schedule_format = array();

    $date_exist = array();
    foreach ($meetings as $meeting) {
        array_push($date_exist, $meeting->date);
        $counts = array_count_values($date_exist);
        $color = $counts[$meeting->date]%3;

        $o = new stdClass();
        $o->id = $meeting->id;
        $newDate = date("F/d/Y", strtotime($meeting->date));
        $o->date = $newDate;
        $o->name = $meeting->title;
        $newformat = date('h:iA', strtotime($meeting->time));
        $o->badge = $newformat;
        $o->description = $meeting->venue;

        if($color==0){
            $o->type = "event";
        }else if ($color==1){
            $o->type = "birthday";
        }else{
            $o->type = "holiday";
        }

        array_push($schedule_format, $o);
    }

    return response()->json($schedule_format);
}
```

Figure 5.2: Code for meeting schedule

Figure 5.2 shows the code for meeting schedule. Since meeting schedule use the plugins, it should follow the way that plugins want to appear message. The color stated is used to differentiate different event on the same date.

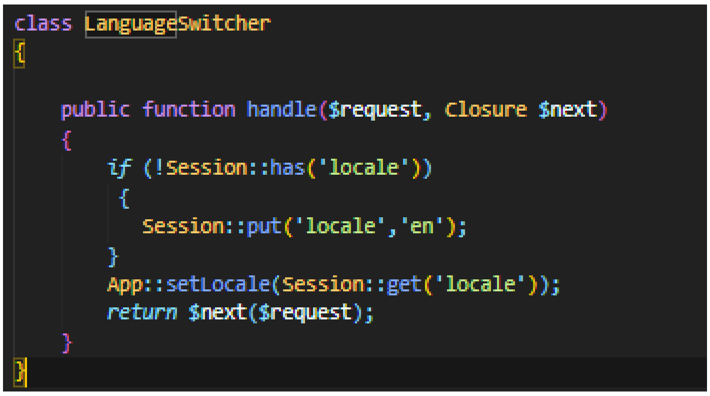
5.3.3 Change Language



```
public function changeLanguage(Request $request)
{
    Session::put('locale',$request->language);
    return Session::get('locale');

}
```

Figure 5.3: Code for change language

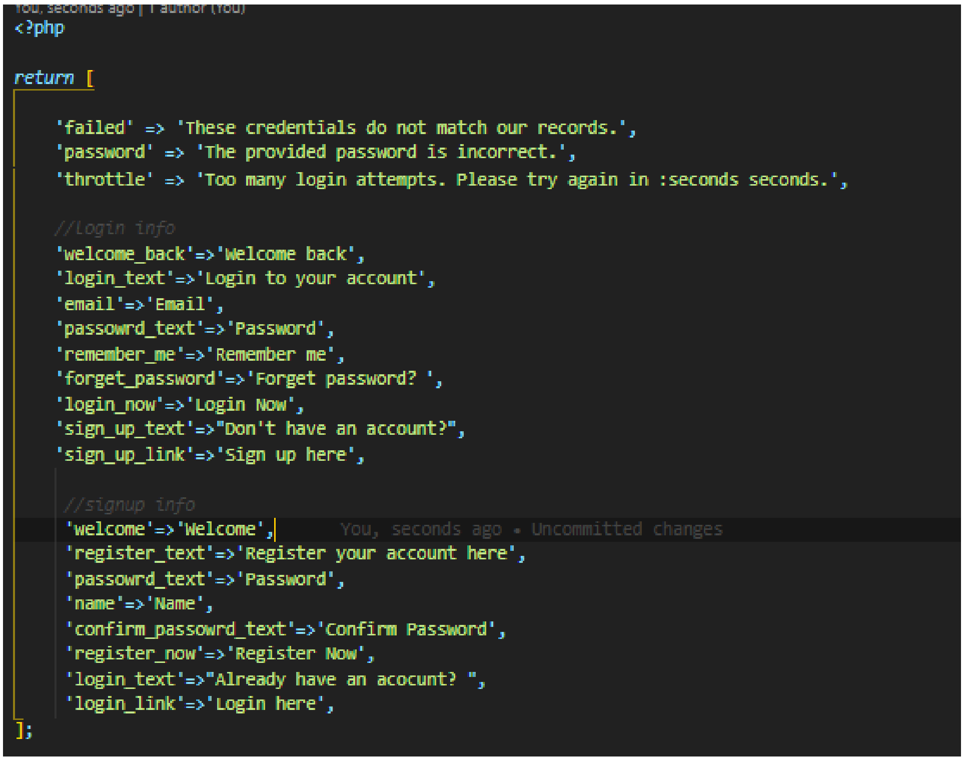
Figure 5.3 shows the code for change language display in system. After language has been changed, it will pass through middleware to check current session language as figure 5.4.



```
class LanguageSwitcher
{

    public function handle($request, Closure $next)
    {
        if (!Session::has('locale'))
         {
           Session::put('locale','en');
        }
        App::setLocale(Session::get('locale'));
        return $next($request);
    }
}
```

Figure 5.4: Code for middleware change language

Figure 5.4 shows the middleware to change language. It will set language according to current locale.

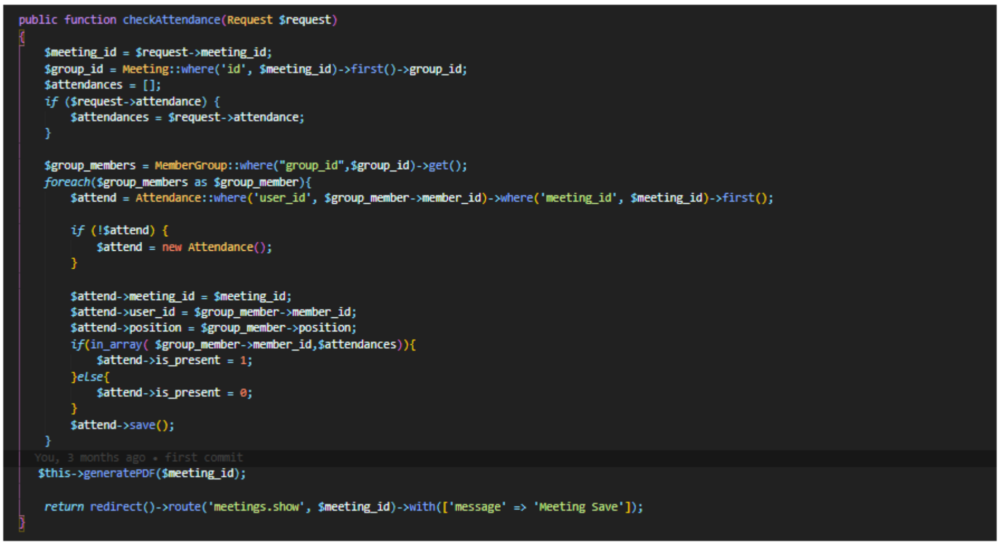


```
You, seconds ago | 1 author (You)
<?php

return [

    'failed' => 'These credentials do not match our records.',
    'password' => 'The provided password is incorrect.',
    'throttle' => 'Too many login attempts. Please try again in :seconds seconds.',

    //login info
    'welcome_back'=>'Welcome back',
    'login_text'=>'Login to your account',
    'email'=>'Email',
    'passowrd_text'=>'Password',
    'remember_me'=>'Remember me',
    'forget_password'=>'Forget password? ',
    'login_now'=>'Login Now',
    'sign_up_text'=>"Don't have an account?",
    'sign_up_link'=>'Sign up here',

     //signup info
     'welcome'=>'Welcome',        You, seconds ago • Uncommitted changes
     'register_text'=>'Register your account here',
     'passowrd_text'=>'Password',
     'name'=>'Name',
     'confirm_passowrd_text'=>'Confirm Password',
     'register_now'=>'Register Now',
     'login_text'=>"Already have an acocunt? ",
     'login_link'=>'Login here',
];
```

Figure 5.5: Code for language change

Figure 5.5 shows the language would be implemented if our locale is English.

5.3.4 Check attendances



```
public function checkAttendance(Request $request)
{
    $meeting_id = $request->meeting_id;
    $group_id = Meeting::where('id', $meeting_id)->first()->group_id;
    $attendances = [];
    if ($request->attendance) {
        $attendances = $request->attendance;
    }

    $group_members = MemberGroup::where("group_id",$group_id)->get();
    foreach($group_members as $group_member){
        $attend = Attendance::where('user_id', $group_member->member_id)->where('meeting_id', $meeting_id)->first();

        if (!$attend) {
            $attend = new Attendance();
        }

        $attend->meeting_id = $meeting_id;
        $attend->user_id = $group_member->member_id;
        $attend->position = $group_member->position;
        if(in_array( $group_member->member_id,$attendances)){
            $attend->is_present = 1;
        }else{
            $attend->is_present = 0;
        }
        $attend->save();
    }
     You, 3 months ago • first commit
   $this->generatePDF($meeting_id);

    return redirect()->route('meetings.show', $meeting_id)->with(['message' => 'Meeting Save']);
}
```

Figure 5.7: Code for check attendances

Figure 5.7 shows the code for check attendances. After attendance is checked, it will generate a pdf and allow everyone to download the pdf file from function generate PDF.

5.4 Chapter Summary

This chapter explain the implementation of code in minute meeting system. The first part of this system describes the development environment in this system and the second part of this system describe about some core function inside the system.

6. Chapter Six: Testing and Evaluation

6.1 Introduction

This chapter will discuss the testing done in this project including the user acceptance toward the system.

6.2 Functional Testing

6.2.1 Admin

6.2.1.1 [TC001] Change group password

Table 6.1: Test case of change group password and its testing flow

[ID] Name	[TC001] Change group password
Summary	Change the password of group when is clicked
Items to be tested	Module X: Join group with password
Users	Admin
Pre-conditions	Admin has login and enter page to edit group
Basic Course of Events	1. Open group selected 2. Click change password 3. Updated in database
Input	N/A
Expected output	The password has been changed

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Click change password	N/A	The password is updated for group	Message password updated	Success

6.2.1.2 [TC002] Edit meeting time

Table 6.2: Test case of edit meeting time and its testing flow

[ID] Name	[TC002] Edit meeting time
Summary	Edit the meeting time created
Items to be tested	Module X: View group meeting
Users	Admin
Pre-conditions	Admin has login and the meeting has been created
Basic Course of Events	1. Open group selected 2. Select meeting time want to edit 3. Set date 4. Set time 5. Updated in database
Input	Date, time
Expected output	The meeting has been updated

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Select meeting time want to edit	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success
3	Input date	12-25-2021	N/A	N/A	Success

4	Input time	07:59	N/A	N/A	Success
5	Click update	N/A	Date and time updated	Redirected to meeting detail page	

6.2.1.3 [TC003] Kick member

Table 6.3: Test case of kick member and its testing flow

[ID] Name	[TC003] Kick member
Summary	Kick the member inside the group
Items to be tested	Module X: View group member
Users	Admin
Pre-conditions	Admin has login
Basic Course of Events	1. Open group selected 2. Select member want to kick 3. Member got remove from database
Input	Date, time
Expected output	The member removes from group

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Select member want to kick	N/A	Member got remove from database	Alert member removed	Success

6.2.1.4 [TC004] Create meeting

Table 6.4: Test case of create meeting and its testing flow

[ID] Name	[TC004] Create meeting
Summary	Create new meeting in group
Items to be tested	Module X: View group meeting
Users	Admin
Pre-conditions	Admin has login
Basic Course of Events	1. Open group selected 2. Set date 3. Set time 4. Add title 5. Assign agenda to title 6. Save in database
Input	Date, time, title, agenda, venue
Expected output	The meeting has been created

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Input date	12-25-2021	N/A	N/A	Success
3	Input time	07:59	N/A	N/A	Success
4	Add title	Introduction	N/A	N/A	Success

5	Assign agenda to title	First agenda	N/A	N/A	Success
6	Click save	N/A	The meeting has been created	Alert meeting created	Success

6.2.1.5 [TC005] Set new group admin

Table 6.5: Test case of new group admin and its testing flow

[ID] Name	[TC005] Set new group admin
Summary	Set a new admin for group
Items to be tested	Module X: View group member
Users	Admin
Pre-conditions	Admin has login
Basic Course of Events	1. Open group selected 2. Pick admin from group member list 3. Update in database
Input	

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Pick admin from group member list	N/A	Update admin in database	Alert admin changed	Success

6.2.1.6 [TC006] Add agenda key points

Table 6.6: Test case of add agenda key points and its testing flow

[ID] Name	[TC006] Add agenda details
Summary	Add point and details for each of the agenda
Items to be tested	Module X: View group meeting
Users	Admin
Pre-conditions	Admin has login, meeting has created
Basic Course of Events	1. Open group selected 2. Open meeting selected 3. Click agenda 4. Type in key points 5. Save in database
Input	Details of agenda

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Open meeting selected	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success
3	Click agenda	N/A	Open agenda dropdown	Open agenda dropdown	Success

4	Type in key points	Use of meetings	N/A	N/A	Success
5	Click save	N/A	Save in database	Alert key points saved	Success

6.2.1.7 [TC007] Delete agenda key points

Table 6.7: Test case of delete agenda key points and its testing flow

[ID] Name	[TC007] Delete agenda details
Summary	Delete the agenda details
Items to be tested	Module X: View group meeting
Users	Admin
Pre-conditions	Admin has login, meeting has created
Basic Course of Events	1. Open group selected 2. Open meeting selected 3. Click agenda key points 4. Confirm delete 5. Delete in database
Input	N/A

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Open meeting selected	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success
3	Click agenda	N/A	Open agenda dropdown	Open agenda dropdown	Success

4	Confirm delete	N/A	Key points delete in database	Key points removed	Success

6.2.1.8 [TC008] Edit agenda key points

Table 6.8: Test case of edit agenda key points and its testing flow

[ID] Name	[TC008] Edit agenda details
Summary	Edit agenda details in meeting
Items to be tested	Module X: View group meeting
Users	Admin
Pre-conditions	Admin has login, meeting has created
Basic Course of Events	1. Open group selected 2. Open meeting selected 3. Click agenda selected 4. Type in key points 5. Update in database
Input	Agenda details

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Open meeting selected	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success
3	Click agenda	N/A	Open agenda dropdown	Open agenda dropdown	Success

4	Type in key points	Use of meetings	N/A	N/A	Success
5	Click save	N/A	Update in database	Alert key points saved	Success

6.2.1.9 [TC009] Generate PDF

Table 6.9: Test case of generate PDF and its testing flow

[ID] Name		[TC009] Generate PDF			
Summary		Generate pdf that done in meeting			
Items to be tested		Module X: View group meeting			
Users		Admin			
Pre-conditions		Admin has login, meeting has created, attendance checked			
Basic Course of Events		1. Open group selected 2. Open meeting selected 3. Click generate pdf 4. Pdf shown and able to download			
Input					
No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Open meeting selected	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success
3	Click generate pdf	N/A	Pdf shown and able to download	Pdf shown and able to download	Success

6.2.1.10 [TC010] Check attendances

Table 6.10: Test case of check attendances and its testing flow

[ID] Name		[TC010] Check attendances			
Summary		Check attendances for who come to meetings			
Items to be tested		Module X: View group meeting			
Users		Admin			
Pre-conditions		Admin has login, meeting has created			
Basic Course of Events		1. Open group selected 2. Open meeting selected 3. Check member attends 4. Save in database for member who attends and not attends			
Input		N/A			
No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Open meeting selected	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success
3	Check member attends	N/A	N/A	N/A	Success

4	Click save	N/A	Attendances save	Alert attendances save	Success

6.2.1.11 [TC011] Send invitation

Table 6.11: Test case of generate PDF and its testing flow

[ID] Name		[TC011] Generate PDF			
Summary		Generate pdf that done in meeting			
Items to be tested		Module X: View group meeting			
Users		Admin			
Pre-conditions		Admin has login			
Basic Course of Events		1. Open group selected 2. Open view member 3. Type user email 4. Click send invitation			
Input					
No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Open view member	N/A	Redirected to group member page	Redirected to group member page	Success
3	Type user email	test@gmail.com	N/A	N/A	Success

4	Click send invitation	N/A	Send invitation to user	Show invitation sent	Success

6.2.2 Member

6.2.2.1 [TC012] Add agenda

Table 6.12: Test case of add agenda and its testing flow

[ID] Name	[TC012] Add Agenda
Summary	Add agenda for future meetings
Items to be tested	Module X: View agenda
Users	Member
Pre-conditions	Member has login
Basic Course of Events	1. Click add agenda 2. Fill in attachment file and title 3. Saved in database
Input	Attachment file, title

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Click add agenda	N/A	Redirected to add agenda page	Redirected to add agenda page	Success
2	Fill in attachment file and title	Agenda.pdf, Cutting cost of software update	N/A	N/A	Success
3	Click save	N/A	Agenda saves as pending agenda	Alert agenda saved	Success

6.2.2.2 [TC013] Delete own agenda

Table 6.13: Test case of delete own agenda and its testing flow

[ID] Name	[TC013] Delete own agenda
Summary	Delete agenda that created by user itself
Items to be tested	Module X: View agenda
Users	Member
Pre-conditions	Member has login
Basic Course of Events	1. Click agenda wanted to delete (own) 2. Confirm again to delete agenda 3. Deleted in database
Input	

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Click agenda delete button (own)	N/A	Pops up delete confirmation	Pops up delete confirmation	Success
2	Click confirm	N/A	Agenda deleted from database	Alert agenda deleted	Success

6.2.2.3 [TC014] Edit own agenda

Table 6.14: Test case of edit own agenda and its testing flow

<table>
<tr><td colspan="2">[ID] Name</td><td colspan="4">[TC014] Edit own agenda</td></tr>
<tr><td colspan="2">Summary</td><td colspan="4">Edit agenda that created by user itself</td></tr>
<tr><td colspan="2">Items to be tested</td><td colspan="4">Module X: View agenda</td></tr>
<tr><td colspan="2">Users</td><td colspan="4">Member</td></tr>
<tr><td colspan="2">Pre-conditions</td><td colspan="4">Member has login</td></tr>
<tr><td colspan="2">Basic Course of Events</td><td colspan="4">1. Click agenda wanted to edit (own)
2. Fill in attachment and title
3. Updated in database</td></tr>
<tr><td colspan="2">Input</td><td colspan="4">Attachment file, title</td></tr>
<tr><td><b>No.</b></td><td><b>Test</b></td><td><b>Test Data</b></td><td><b>Expected Outcome</b></td><td><b>Actual Outcome</b></td><td><b>Status</b></td></tr>
<tr><td>1</td><td>Click agenda edit button (own)</td><td>N/A</td><td>Redirect to agenda detail page</td><td>Redirect to agenda detail page</td><td>Success</td></tr>
<tr><td>2</td><td>Fill in attachment file and title</td><td>Agenda.pdf, Cutting cost of software update</td><td>N/A</td><td>N/A</td><td>Success</td></tr>
<tr><td>3</td><td>Click save</td><td>N/A</td><td>Agenda updated</td><td>Alert agenda updated</td><td>Success</td></tr>
</table>

6.2.2.4 [TC015] View agenda

Table 6.15: Test case of view agenda and its testing flow

[ID] Name	[TC015] View agenda
Summary	View agenda that created in group
Items to be tested	Module X: View agenda
Users	Member
Pre-conditions	Member has login
Basic Course of Events	1. Click group current agenda 2. View agenda and its attachment file
Input	

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Click group current agenda	N/A	Redirect to current agenda page	Redirect to current agenda page	Success
2	View agenda	N/A	Redirect to agenda detailed page	Redirect to agenda detailed page	Success

6.2.2.5 [TC016] Join group with password

Table 6.16: Test case of join group with password and its testing flow

[ID] Name	[TC016] Join group with password
Summary	Join the group with password
Items to be tested	Module X: View group
Users	Member
Pre-conditions	Member has login
Basic Course of Events	1. Click home 2. Key in password 3. Join group
Input	Password
Alternative Path	1. In step 2, when password is wrong, it will jump out alert not found password with the group 2. In step 2, when password written have already joined the group, it will alert already joined in group

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Click home	N/A	Redirect to home page	Redirect to home page	Success
2	Key in password	12345678	N/A	N/A	Success
3	Click join group	N/A	Join group success	Alerts join group success	Success

6.2.2.6 [TC017] Left group

Table 6.17: Test case of left group and its testing flow

[ID] Name	[TC017] Left group
Summary	Left the group
Items to be tested	Module X: View group
Users	Member
Pre-conditions	Member has login
Basic Course of Events	1. Click group selected 2. Click left group 3. Group left
Input	

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Click left group	N/A	Left group	Alerts left group	Success

6.2.2.7 [TC018] View group meeting

Table 6.18: Test case of view group meeting and its testing flow

[ID] Name	[TC018] View group meeting
Summary	View meeting inside group
Items to be tested	Module X: View group meeting
Users	Member
Pre-conditions	Member has login, member is the member of group
Basic Course of Events	1. Click group selected 2. View meeting list
Input	N/A

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	View meeting list	N/A	Meeting list shown	Meeting list shown	Success

6.2.2.8 [TC019] View group member

Table 6.19: Test case of view group member and its testing flow

[ID] Name		[TC019] View group member			
Summary		View member inside group			
Items to be tested		Module X: View group meeting			
Users		Member			
Pre-conditions		Member has login, member is the member of group			
Basic Course of Events		1. Click group selected 2. View group member			
Input		N/A			
No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	View group member	N/A	Redirected to group member page	Redirected to group member page	Success

6.2.2.9 [TC020] View meeting minutes

Table 6.20: Test case of view meeting minutes and its testing flow

[ID] Name		[TC020] View meeting minutes			
Summary		View meeting minutes inside group			
Items to be tested		Module X: View meeting minutes			
Users		Member			
Pre-conditions		Member has login, member is the member of group, the meeting created			
Basic Course of Events		1. Click group selected 2. Click meetings wanted to view 3. View meeting minutes data, date, times, venue, agenda			
Input		N/A			
No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Open group selected	N/A	Redirected to group details page	Redirected to group details page	Success
2	Select meeting time want to view	N/A	Redirected to meeting detail page	Redirected to meeting detail page	Success

6.2.2.10 [TC021] View meeting schedules

Table 6.21: Test case of view meeting schedules and its testing flow

[ID] Name		[TC021] View meeting schedules			
Summary		View meeting schedules			
Items to be tested		Module X: View meeting schedule			
Users		Member			
Pre-conditions		Member has login			
Basic Course of Events		1. Click meeting schedules 2. View upcoming, pass meeting schedule in order			
Input		N/A			
No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Click meeting schedules	N/A	Redirected to meeting schedule page	Redirected to meeting schedule page	Success

6.2.2.11 [TC022] Login

Table 6.22: Test case of login and its testing flow

[ID] Name	[TC022] Login
Summary	Login to the system
Items to be tested	Module X: View group
Users	Member
Pre-conditions	Member has registered
Basic Course of Events	1. Key in Gmail, password 2. Login to the system 3. Login success
Input	Gmail, password
Alternative Path	1. In step 2, when there is no match in database, it will alert error in system

No.	**Test**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
1	Key in Gmail, password	admin@gmai.com, 123456	N/A	N/A	Success
2	Click login	N/A	Login successfully to system	Redirected to home page	Success

6.2.2.12 [TC023] Register

Table 6.23: Test case of register and its testing flow

[ID] Name	[TC023] Register
Summary	Register account in the system
Items to be tested	Module X: Register function
Users	Anonymous user
Pre-conditions	
Basic Course of Events	1. Key-in Gmail, username, phone, password in system 2. Register account 3. Save in database
Input	Gmail, password, phone, username
Alternative Path	1. In step 2, if there is Gmail same as the text key-in, it will alert already registered

No.	Test	Test Data	Expected Outcome	Actual Outcome	Status
1	Key in Gmail	admin@gmai.com	N/A	N/A	Success
2	Key in password	123456	N/A	N/A	Success
3	Key in phone	0123456789	N/A	N/A	Success
4	Key in username	Lim G Wei	N/A	N/A	Success

5	Click register	N/A	Successfully registered	Redirect to home page	Success

6.2.2.13 [TC024] Reply invitation

Table 6.24: Test case of view group member and its testing flow

<table>
<tr><td colspan="2">[ID] Name</td><td colspan="4">[TC024] View group member</td></tr>
<tr><td colspan="2">Summary</td><td colspan="4">View member inside group</td></tr>
<tr><td colspan="2">Items to be tested</td><td colspan="4">Module X: View group meeting</td></tr>
<tr><td colspan="2">Users</td><td colspan="4">Member</td></tr>
<tr><td colspan="2">Pre-conditions</td><td colspan="4">Member has login, member is the member of group</td></tr>
<tr><td colspan="2">Basic Course of Events</td><td colspan="4">1. Click settings
2. View notification
3. Click accept or reject</td></tr>
<tr><td colspan="2">Input</td><td colspan="4">N/A</td></tr>
<tr><td><b>No.</b></td><td><b>Test</b></td><td><b>Test Data</b></td><td><b>Expected Outcome</b></td><td><b>Actual Outcome</b></td><td><b>Status</b></td></tr>
<tr><td>1</td><td>Click settings</td><td>N/A</td><td>Redirected to settings page</td><td>Redirected to settings page</td><td>Success</td></tr>
<tr><td>2</td><td>View notification</td><td>N/A</td><td>Redirected to notification page</td><td>Redirected to notification page</td><td>Success</td></tr>
<tr><td>3</td><td>Click accept or reject</td><td></td><td>Save response in database</td><td>Redirected to group if accept, reject no change</td><td>Success</td></tr>
</table>

6.3 User Acceptance Testing

People who have actively participate club in school are invited to review my system. This is because this system can be used for member who are involved in meeting a lot. They will use this system rather than others and understand what is more important.

There were 17 participants selected to be reviewers for the user acceptance testing activity. These 17 participants were chosen based on their experiences. All of them have joined in club before and join meeting and view meeting minute before. It is totally qualified to review the system. The system was evaluated based on the questions distributed prior to the testing activity.

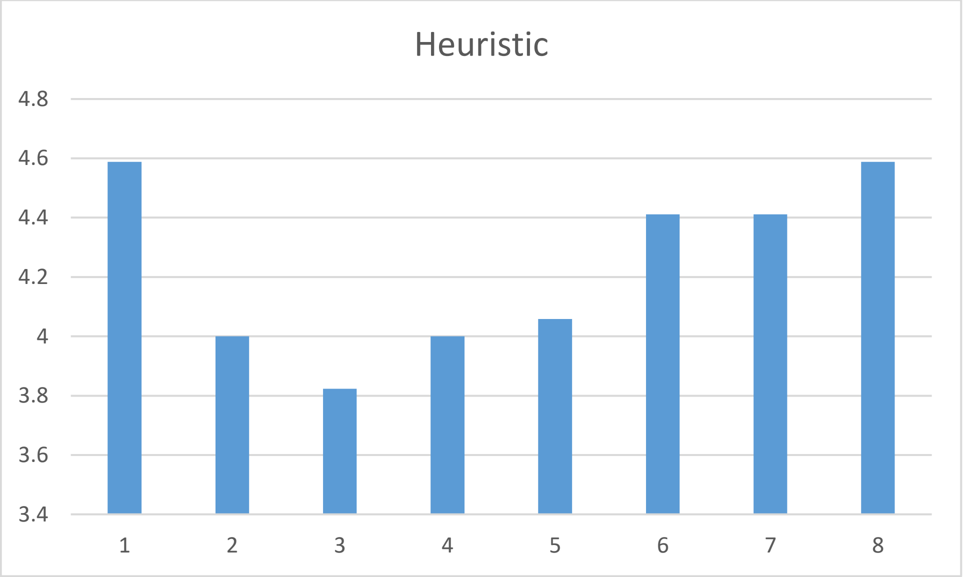


Figure 6.1: Average review of user acceptance testing of each section in heuristic

The number in x-axis represent:

1. Primary goal of the system/purpose is clear
2. Clean, simple design
3. Pleasing colour scheme
4. Appropriate use of space
5. Consistent design
6. Text and colour are consistent
7. Icons are universally understood
8. Images are meaningful and serve a purpose
   The average result gain for Heuristic is 4.25.

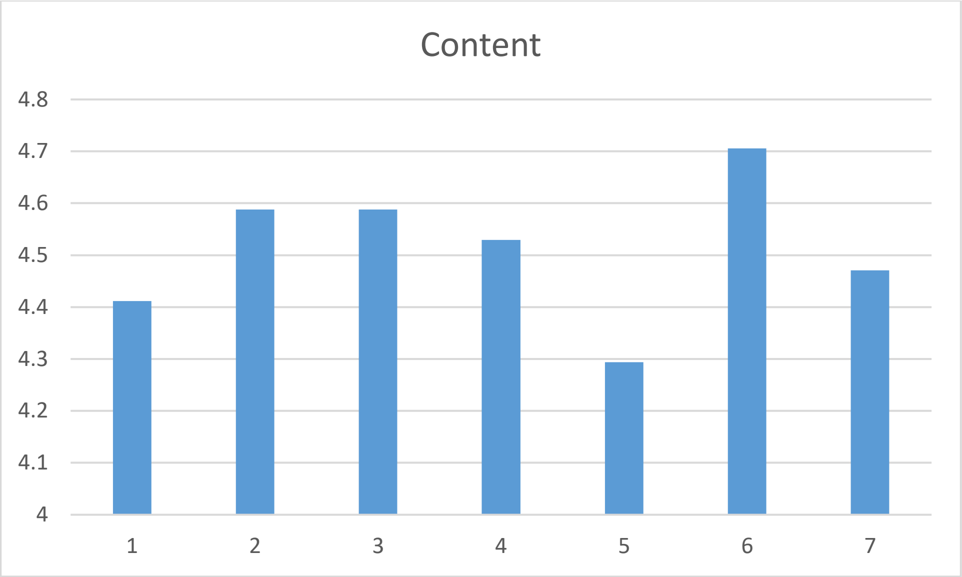


Figure 6.2: Average review of user acceptance testing of each section in content

The number in x-axis represent:

1. Major headings are easy to understand
2. Minimal text/information presented
3. Clear terminology, no jargon
4. Links are clear and follow conventions
5. Help is available on every page
6. Important content is above the fold
7. Search box is easy to identify and easy to use.
   The average result gain for Content is 4.51.

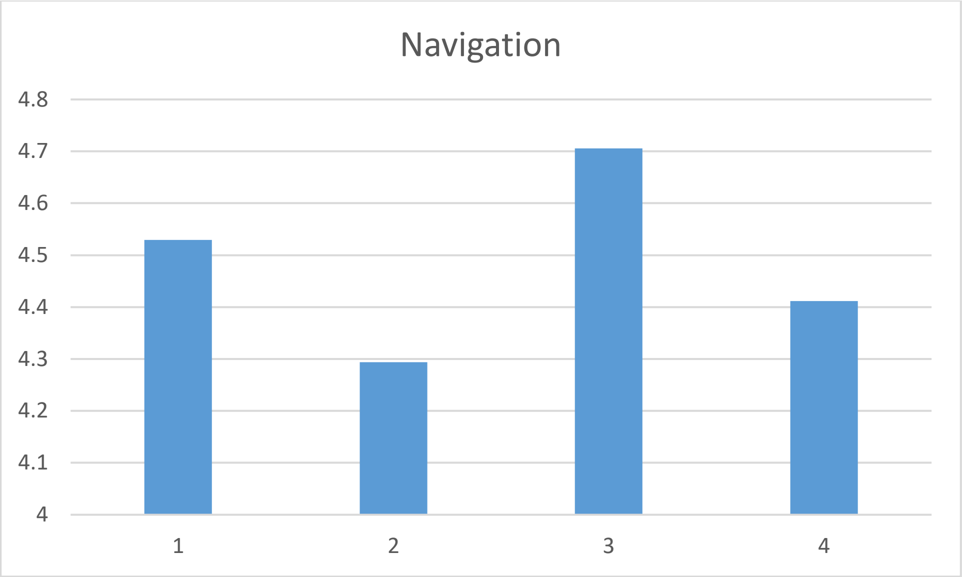


Figure 6.3: Average review of user acceptance testing of each section in navigation

The number in x-axis represent:

1. Consistent navigation
2. Easy to identify your location on the site
3. Consistent way to return to ‘main page/system’
4. Organization of information makes sense

   The average result gain for Navigation is 4.49.

There are some recommendation or message provided by reviewers:

1. Здорово
2. Good and User-Friendly System
3. Need to implement more CSS.
4. The simple user interface design is easy for users to understand and get started, and the operation is very convenient.
5. Present of header / topic whichever you call it can help user to know where or what page they are current located, or maybe what certain group of items represent. E.g., the item in first page makes me thought that it is user, people or member kind of that, but it turns out to be group. Voice input and description of system in the video will help us in better understanding the system. Anyway, overall is good.
6. 整理一下就 OK

6.4 Summary

This chapter explain the testing done in the system. The first part discusses about functional testing done by developer while second part of this chapter discuss about user acceptance towards the system.

7. Chapter Seven: Recommendations

7.1 Introduction

This chapter will discuss about the weakness of the system compare to others and the suggestion for future implementation.

7.2 System weaknesses

There are few shortcomings identified for this system during development phase:

i. The overall design of the system is not very suit for user experience.
ii. The record functions are not done for this system.

7.3 Suggestion for future implementation

The following improvements are suggested to increase the system functionality and its effectiveness:

i. Enhance the design of the system.
ii. Add record functions in the system.

8.Chpater Eight: Conclusion

8.1 Introduction

This chapter will discuss about the concluding remarks of the system.

8.2 Concluding Remarks

The minute meeting system allows users to record their meetings records and generate them into PDF. User only requires signing up their account in the system.

9 References

“One-on-one meeting AGENDA SOFTWARE,” *Fellow.app*. [Online]. Available: https://fellow.app/lp/meeting-minutes-software. [Accessed: 23-Jul-2021].

L. Ingram and A. G. User, “Meeting minutes software – the fastest and easiest way to write your minutes,” *Meeting Agenda & Meeting Minutes Software*. [Online]. Available: https://meetingking.com/meeting-minutes-software/. [Accessed: 23-Jul-2021].

“The app built for managers: Collaborative meeting agendas,” *Hypercontext*, 25-Jul-2021. [Online]. Available: https://hypercontext.com/. [Accessed: 27-Jul-2021].

Bootstrap framework. (n.d.). Croydon Council. https://www.croydon.gov.uk/design-and-content-guidelines/resources/front-end-framework/bootstrap-framework#:~:text=Bootstrap%20is%20the%20most%20popular,JS%20plugins%20for%20user%20interaction

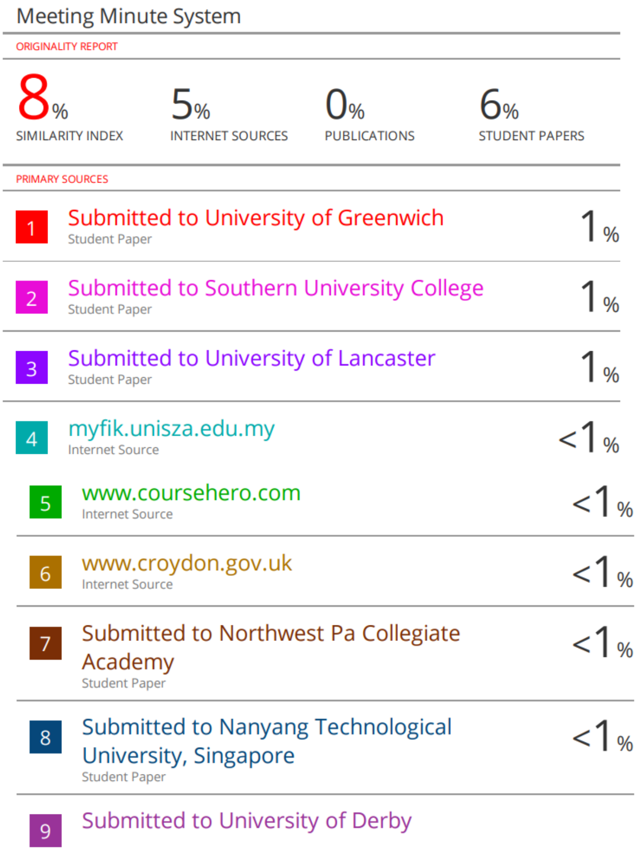
Framework definition. (n.d.). The Tech Terms Computer Dictionary. https://techterms.com/definition/framework#:~:text=A%20framework%2C%20or%20software%20framework,programs%20for%20a%20specific%20platform.&text=A%20framework%20may%20also%20include,in%20the%20software%20development%20process

(n.d.). Laravel - The PHP Framework For Web Artisans. https://laravel.com/

MySQL. (n.d.). https://www.mysql.com/

What is a database? (n.d.). https://www.oracle.com/database/what-is-database/

Appendix



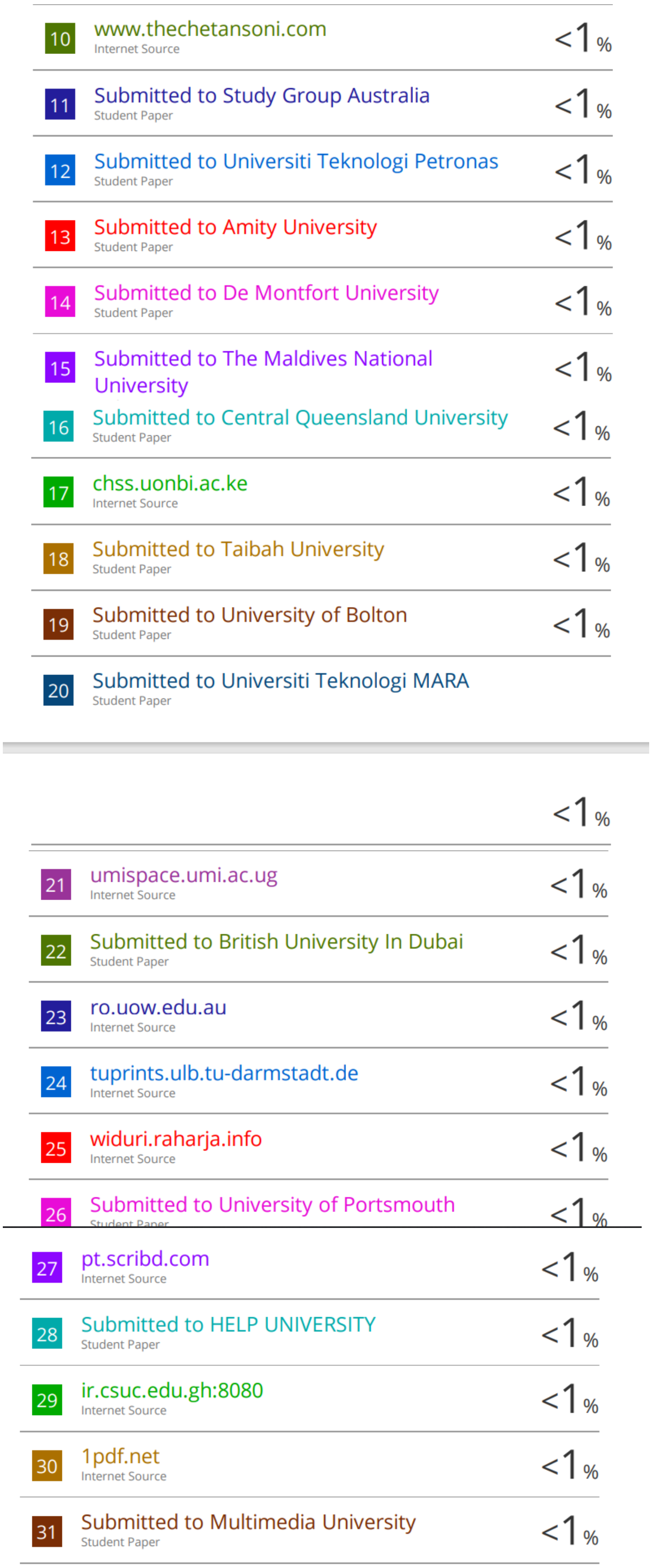
Meeting Minute System

ORIGINALITY REPORT

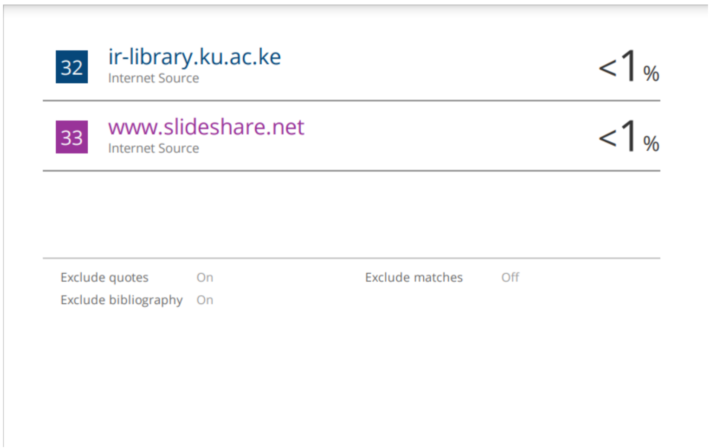
8%	5%	0%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Greenwich Student Paper	1 %
2	Submitted to Southern University College Student Paper	1 %
3	Submitted to University of Lancaster Student Paper	1 %
4	myfik.unisza.edu.my Internet Source	<1 %
5	www.coursehero.com Internet Source	<1 %
6	www.croydon.gov.uk Internet Source	<1 %
7	Submitted to Northwest Pa Collegiate Academy Student Paper	<1 %
8	Submitted to Nanyang Technological University, Singapore Student Paper	<1 %
9	Submitted to University of Derby	



10	www.thechetansoni.com Internet Source	<1 %
11	Submitted to Study Group Australia Student Paper	<1 %
12	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
13	Submitted to Amity University Student Paper	<1 %
14	Submitted to De Montfort University Student Paper	<1 %
15	Submitted to The Maldives National University	<1 %
16	Submitted to Central Queensland University Student Paper	<1 %
17	chss.uonbi.ac.ke Internet Source	<1 %
18	Submitted to Taibah University Student Paper	<1 %
19	Submitted to University of Bolton Student Paper	<1 %
20	Submitted to Universiti Teknologi MARA Student Paper	<1 %
21	umispace.umi.ac.ug Internet Source	<1 %
22	Submitted to British University In Dubai Student Paper	<1 %
23	ro.uow.edu.au Internet Source	<1 %
24	tuprints.ulb.tu-darmstadt.de Internet Source	<1 %
25	widuri.raharja.info Internet Source	<1 %
26	Submitted to University of Portsmouth Student Paper	<1 %
27	pt.scribd.com Internet Source	<1 %
28	Submitted to HELP UNIVERSITY Student Paper	<1 %
29	ir.csuc.edu.gh:8080 Internet Source	<1 %
30	1pdf.net Internet Source	<1 %
31	Submitted to Multimedia University Student Paper	<1 %



32	ir-library.ku.ac.ke Internet Source	<1 %
33	www.slideshare.net Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		