FTMK NAVIGATOR

KOH KOK SHENG

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FTMK Navigator

KOH KOK SHENG

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

[FTMK Navigator]

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

STUDENT : _____ Date : 09/09/2021
(KOH KOK SHENG)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR : _____ Date : 9/9/2021
(ASSOC. PROF. DR. SABRINA BINTI AHMAD)

ACKNOWLEDGEMENTS

First and foremost, I would like to express my special thanks of gratitude to my supervisor (Assoc. Professor Dr. Sabrina binti Ahmad) for her encouragement, continuous supervision and knowledge about the project and support for it in the completion of the project. Her advice is very beneficial to the project completion.

Besides my supervisor, I would like to thank my parents for giving birth to me at the first place and supporting me spiritually throughout my life. Other than that, I would like to thank all my friends that help me throughout the project.

Last but not least, I would like to thank Universiti Teknikal Malaysia Melaka (UTeM), the Faculty of Information and Communications Technology, for providing me the opportunity to incorporate my expertise in the project.

ABSTRACT

This project is developed based on the idea of navigation in an indoor environment. Sometimes it is difficult to find a venue in a building like a shopping mall or faculty because of its complex structure or familiarity of user. So it is a good idea to have a mobile application that can guide user inside a building. FTMK in UTeM is one of the good example of complex building. It is hard to search a venue or route in FTMK especially for new students, hence that's the reason why FTMK Navigator is built in this project. FTMK Navigator is an android mobile application, it shows the floor plans of FTMK, user can use it to search a venue or route between two venues. It calculate the estimated distance when a route is being searched and gives some more information to user for them to search their destination more easily. FTMK Navigator is useful to students and guests that visit FTMK since they can find destination on their fingertip.

ABSTRAK

Projek ini dibangunkan berdasarkan idea pelayaran di persekitaran dalaman. Kadang-kadang sukar untuk mencari tempat di bangunan seperti pusat membeli-belah atau fakulti kerana strukturnya yang kompleks atau keakraban pengguna. Oleh itu, adalah idea yang baik untuk mempunyai aplikasi mudah alih yang dapat membimbing pengguna di dalam bangunan. FTMK di UTeM adalah salah satu contoh bangunan kompleks yang baik. Sukar untuk mencari tempat atau laluan di FTMK terutamanya untuk pelajar baru, oleh itu itulah sebabnya FTMK Navigator dibina dalam projek ini. FTMK Navigator adalah aplikasi mudah alih android, ia menunjukkan denah lantai FTMK, pengguna dapat menggunakannya untuk mencari tempat atau laluan antara dua tempat. Ia mengira anggaran jarak ketika laluan sedang dicari dan memberikan lebih banyak maklumat kepada pengguna agar mereka dapat mencari destinasi dengan lebih mudah. FTMK Navigator berguna untuk pelajar dan tetamu yang mengunjungi FTMK kerana mereka dapat mencari destinasi di hujung jari mereka.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FTMK	-	**Fakulti Teknologi Maklumat & Komunikasi**
UTeM	-	**Universiti Teknikal Malaysia Melaka**
Wi-Fi	-	**Wireless Fidelity**
RSS	-	**Receive Signal Strength**
AP	-	**Access Point**
IMU	-	**Inertial Measurement Unit**
SDLC	-	**System Development Life Cycle**
DBLC	-	**Database Life Cycle**
ERD	-	**Entity Relationship Diagram**
SQL	-	**Structured Query Language**
RDBMS	-	**Relational Database Management System**
IDE	-	**Integrated Development Environment**
UML	-	**Unified Modelling Language**

CHAPTER 1: INTRODUCTION

1.1 Introduction

FTMK Navigator is an android mobile application that helps students mainly in searching venues and directing students to their destination in FTMK. FTMK is large and complex in structure, students always can't find their destination and get lost in FTMK. Hence, FTMK navigator is designed to solve the problems faced by students. This project is built using the Navigine, it is a global provider of integrated mobile indoor positioning technologies that enable advanced indoor and outdoor navigation. At the end of this project, FTMK Navigine will be useful to the students and guests in the FTMK.

1.2 Problem Statement

- The structure of FTMK is complex that students always get lost in there.
- Students always can't find destination in FTMK.
- Students always wasting time on searching destination in FTMK.

1.3 Objective

This project embarks on the following objectives:

- To design a solution to facilitate students, guests in indoor navigation.

- To develop an android-based mobile application as a platform for students and guests to navigate them to their destination in FTMK.

- To test the system for efficiency and user satisfaction in terms of usability by the students and guests.

1.4 Scope

a) Target User

This project is target for students in FTMK and guests that visit FTMK.

b) Operating System

Android is the operating system used for FTMK Navigator in this project. Android is a software stack for mobile devices that includes an operating system, middleware and key application (Kharisma and Nurhasan, no date). It is used because it provides a rich application framework that allows developer to build innovative application.

c) Modules to be developed

- Display Floor Plan

FTMK Navigator will display floor plan for a specific floor when user click on up and down button to switch floor number. The ground floor's floor plan will be displayed as default after user enter FTMK Navigator.

- Search Venue by Venue's Name

FTMK Navigator will search a specific venue and draw a pinpoint on the venue after user enter correct venue's name.

- Draw Route from Source to Destination

  FTMK Navigator will draw a route on the floor plan once it receives two valid venues name. The route drawn by FTMK Navigator will be the shortest one.

- Search Venues by Filter

  It is a feature that allow user to search multiple venues by selecting one of the filters provided by FTMK Navigator. Examples of filters are lecture room, lab, rest room and so on.

- Estimate Distance From Source to Destination

  The distance from source to destination will be estimated once user enter two valid venues name.

d) Deliverable

The deliverables of this project are the literature review, project methodology, analysis, design, implementation, testing and finally the complete FTMK Navigator application.

1.5 Project Significance

This project is built especially for students in FTMK and guests that visit FTMK, they are the one who will get benefits from this project. By using navigation technique, this will definitely decrease the time consume on searching their destination. As a result, it reduce the probability of students and guests getting late to lecture or lab session and important meeting respectively.

1.6 Expected Output

There are some of the expected output in this project. Firstly, FTMK Navigator is built as expected at the end of this project, all of the modules stated above are built accurately. Secondly, the application is widely accepted by students and guests, it is

easy to use and learn to use. Thirdly, the application helps students and guests in term of reducing time consume in searching the destination. Lastly, the developer of this application grasp all the knowledge about indoor navigation.

1.7 Conclusion

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

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter aims to discuss on the related works regarding indoor navigation. The literature will describe in details on the types of indoor navigation. It will act as the summary of the related topic of published researched. This will give the clear explanation on what has just been done, what is commonly acknowledged, what is emerging and what is the present state of thinking on the topic.

2.2 Facts and findings

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

2.2.1 Domain

Indoor navigation is a function that used to lead people to reach their destination in an indoor environment. There are many different types of technology that can be used to build indoor navigation system. One of the most popular technology used to build indoor navigation is the Wi-Fi technology. Indoor navigation based on Wi-Fi are used in many projects, it is because a wide range of different types of existing Wi-Fi hotspots can be used for this and it is easy to implement. This technique uses

already existing infrastructure and Wi-Fi access points (APs) to calculate where a device is located. The device needs to be able to listen for the Wi-Fi AP but does not need to connect to it. By analyzing the signal strength of multiple Wi-Fi signals and knowing the location of those APs, the approximate location of the device can be determined. There are still many other technologies used to build indoor navigation system, which will be discuss later.

2.2.2 Existing System

Till date, there are many applications that provide indoor navigation functionality. One of the example is the mobile application offered by Sunway Pyramid shopping mall (*Sunway Pyramid launches real-time in-mall navigation mobile app*, no date). The Sunway Pyramid mobile application was created in partnership with Google Maps and uses indoor map technology to provide customers an easier way to navigate the mall. Wi-Fi fingerprinting is the technique used by Sunway Pyramid application to achieve indoor positioning and navigation (Zulkiflie, Kamaruddin and Wahab, 2020). This technique collects the signal strength of the Wi-Fi access point (AP) in the local surrounding at various points in a covered area. With this technique, Sunway Pyramid application locate customer's position accurately and navigate customer throughout the whole shopping mall. Figure 2.1 shows the user interface of Sunway Pyramid for user to enter source and destination while figure 2.2 shows the route drawn from source to destination.

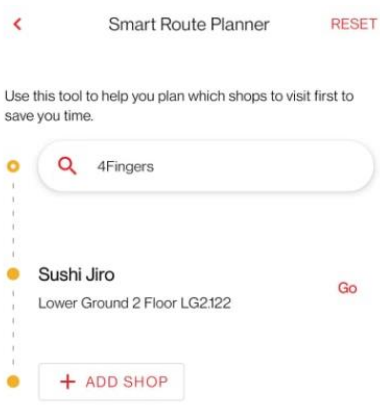

Figure 2.1 Smart Route Planner User Interface

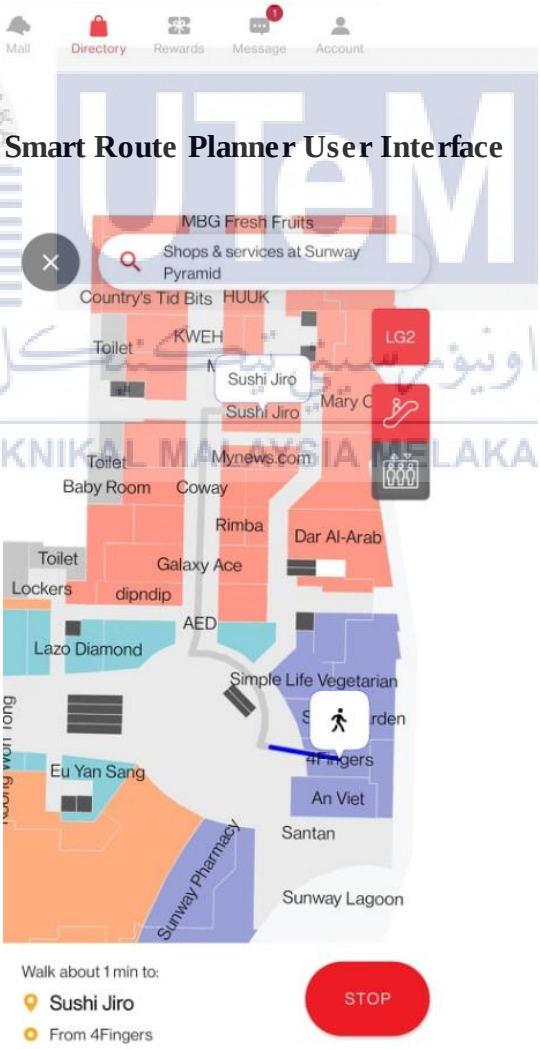

Figure 2.2 Route Drawn on Sunway Pyramid Floor Plan

2.2.3 Technique

Other than using Wi-Fi fingerprinting that used by Sunway Pyramid, there are many other techniques can be used to achieve indoor navigation which is describe in section below.

- IMU (Inertial Measurement Unit)

  Inertial Systems inform about the relative movement of the tag with the integration of several sensors such as accelerometer, magnetometer, and gyroscope, in a tiny module. These sensors are useful in determining the direction and orientation of movement. Combined, they can provide an estimate of the relative motion with regards to the previous position.

- Infrared Light

  This type of system can be used as a very reliable room detector. Since light cannot traverse walls, it is not possible for a tag to detect light from an anchor without being in the same room. For precise localization, they require installing many anchors and can struggle due to the low quality of the signal strength measurements required to compute the position from multiple anchors.

- Ultrasound System

  Ultrasound systems use sound instead of light. It does not interfere with electromagnetic waves and does not require line of sight. The system requires a set of anchors and a tag. It uses Time of Flight, that is, the time required by sound to travel from an anchor to a tag or vice versa, to estimate the distance between them.

2.3 Project Methodology

Project methodology including system development life cycle (SDLC) and database life cycle (DBLC) is important because they help to transform the idea of a

project into a functional and completely operational structure and defines the stages involved for implementing a database respectively.

2.3.1 System Development Life Cycle (SDLC)

In this project, object-oriented analysis and design (OOAD) is used as the system development life cycle approach. OOAD is a software engineering approach that models a system as a group of interacting objects (*Object-oriented analysis and design (OOAD) | Guide books*, no date). OOAD is used in this project because it is iterative and incremental, it allows development team to develop prioritized requirements first, it is more flexible, it is easier to test and debug during small iteration and so on. There are five phases in OOAD, which is requirement gathering and analysis, system design, object design, implementation and testing. Figure 2.3 visualize the life cycle of OOAD.

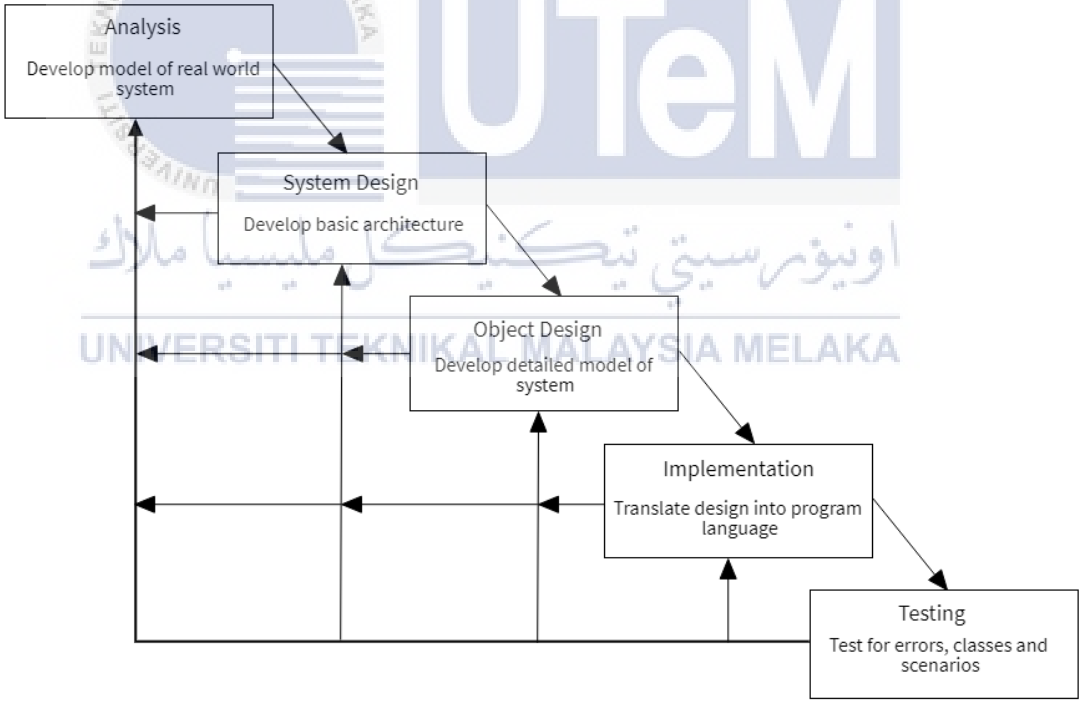

Figure 2.3: Object-oriented Analysis and Design Model

2.3.2 Database Life Cycle (DBLC)

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

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

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

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

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

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

2.4 Project Requirements

Project requirements such as software requirement and hardware requirement are important to get the FTMK Navigator done. Table 2.1 and table 2.2 list out the software and hardware requirement for this project respectively.

2.4.1 Software Requirement

Table 2.1: List of Software

Android Studio	An official Integrated Development Environment (IDE) for Android application development, based on IntelliJ IDEA.
Star UML	An open source software modelling tool that supports the Unified Modelling Language (UML) framework for system and software modelling.
Microsoft Word	A word processor published by Microsoft. It is used to type and save document for this project.
Navigine	A platform provides development tools for mobile developers and system integrators for creating indoor navigation.

2.4.2 Hardware Requirement

Table 2.2: List of Hardware

Laptop/Desktop	Laptop or desktop is required to build the FTMK Navigator.
Android Mobile Phone	Android mobile phone is used to run and test FTMK Navigator.
USB Cable	USB cable is used to connect android smartphone to development machine.

2.4.3 Other Requirements

N/A

2.5 Project Schedule and Milestones

A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes, although they are particularly useful for simplifying complex projects. Table 2.3 shows the Gantt chart for this project.

Table 2.3: Gantt Chart

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Proposal Preparation																
Proposal Correction/Improvement & Approval																
Proposal Submission																
Define Problem Statement, Objective & Project Scope																
Research and Discovery																
System Analysis																
System Design																
System Implementation																

System Testing																
Final Presentation																
Submission																

2.6 Conclusion

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

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter aims to discuss on the analysis of current system and the proposed FTMK Navigator. Some of the techniques and diagram such as activity diagram, use case diagram will be used in this chapter to illustrate the analysis and requirements of the system. This will give the clear explanation on the shortage of current system and what will be done by the proposed system.

3.2 Problem Analysis

In chapter 1, the problem statements are listed for this project. As stated in the problem statements, the structure of FTMK is too complex that students and guests always get lost in there and thus wasting time on searching their desired destination. And sometime they can't even find the destination after wasting a lot of time. Hence, it is better to let students and guests to search venues and routes of FTMK on their fingertip.

By looking on the current system which is FTMK Navigation developed by Zakwan Shaj, it does not really let navigate user to their destination. It just display the sketch floor plan with highlighted destination after user enter the name of the destination. Thus, there is still room for improvement which will be built in the proposed system. Figure 3.1 shows the basic flow of current system.

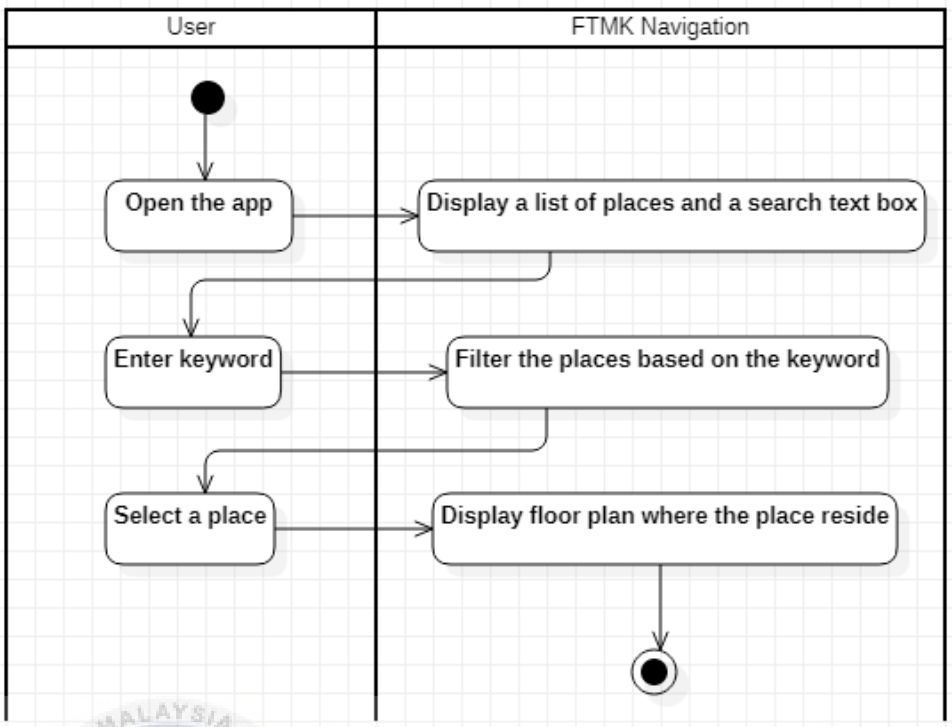

Figure 3.1 Basic Flow of Current System

3.3 Requirement Analysis

Requirement analysis is significant and essential activity after elicitation. It analyze, refine, and scrutinize the gathered requirements to make consistent and unambiguous requirements. This activity reviews all requirements and provide a graphical view of the entire system.

3.3.1 Functional Requirement

Functional requirements are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations (*CS 410/510 - Software Engineering class notes*, no date). Functional requirements of this project are stated below.

- The system shall be able to display correspond floor plan's image after user press on up and down arrow button to switch floor.

- The system shall be able to provide label which indicate the current floor number.

- The system shall be able to enlarge or reduce size of the image of floor plan when user press on plus or minus button or pinch on the image.

- The system shall be able to display correspond floor plan with pinpoint on venue after user search a venue by entering the venue's name.

- The system shall be able to draw multiple pinpoints on floor plan image after user select the filter for venues.

- The system shall be able to draw route from source to destination after user enter the name of source and destination.

- The system shall be able to estimate the distance from source to destination after user enter the name of source and destination.

3.3.2 Non-functional Requirement

Non-functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standard (*CS 410/510 - Software Engineering class notes*, no date). Table 3.1 describes the general non-functional requirements for this project.

Table 3.1 Non-functional Requirement

No	Requirement	Description
1.	Constant	The route drawn by the system should be the same and has a shortest distance every single time the user search a route.
2.	Portability	FTMK Navigator shall be able to operate on any Android devices with API level 21 and higher.

3.	Usability	The user interfaces of FTMK Navigator shall be easy to understand and easy to learn to use.
4.	Availability	FTMK Navigator shall be available at any time.

3.3.3 Other Requirement

N/A

3.4 Use Case Diagram

A use case diagram is visual tool that provides a way for developers to come to a common understanding with system's end users and domain experts (Sengupta and Bhattacharya, no date). Figure 3.2 shows the use case diagram for the FTMK Navigator.

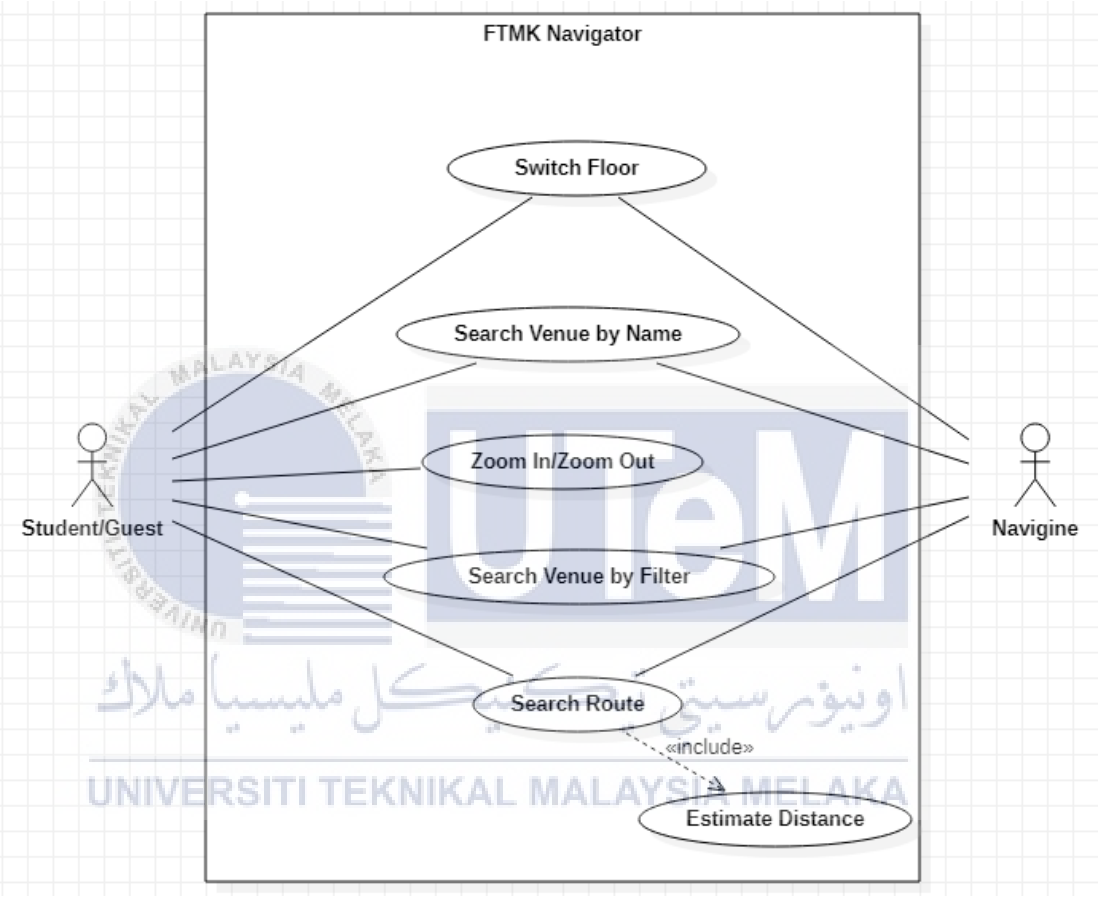

Figure 3.2 FTMK Navigator Use Case Diagram

Based on the figure 3.2, there are two primary actors in this project. Primary actor is the actor who triggers the use case. There are five main use cases and one included use case. Navigine is the secondary actor which provide a specific result or information to a use case. With the help of Navigine, most of the use cases like switch floor, search venue by name or filter and search route can be done easily.

3.5 Sequence Diagram

A sequence diagram is used to model the interactions between the actors and the objects in a system and the interactions between the objects themselves. The sequence diagram is drawn based on the FTMK Navigator use case diagram. Section below shows the sequence diagrams of this project.

3.5.1 Switch Floor

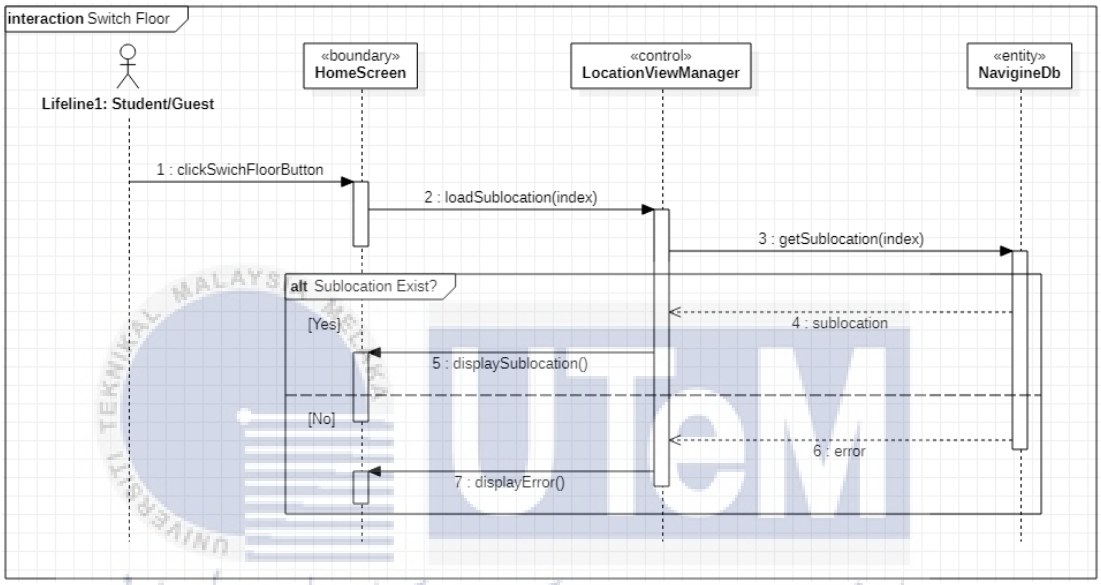

Figure 3.3 Switch Floor Sequence Diagram

Figure 3.3 is the sequence diagram for switch floor use case. It starts when user click on the switch floor button. The switch floor request will be handle by LocationViewManager. LocationView is a container that is used to display the floor plan image. Hence, every operation on the LocationView will be manage by LocationViewManager. The particular floor plan image will be retrieved from database if the floor plan image is exist, or else error message will be displayed.

3.5.2 Search Venue by Name

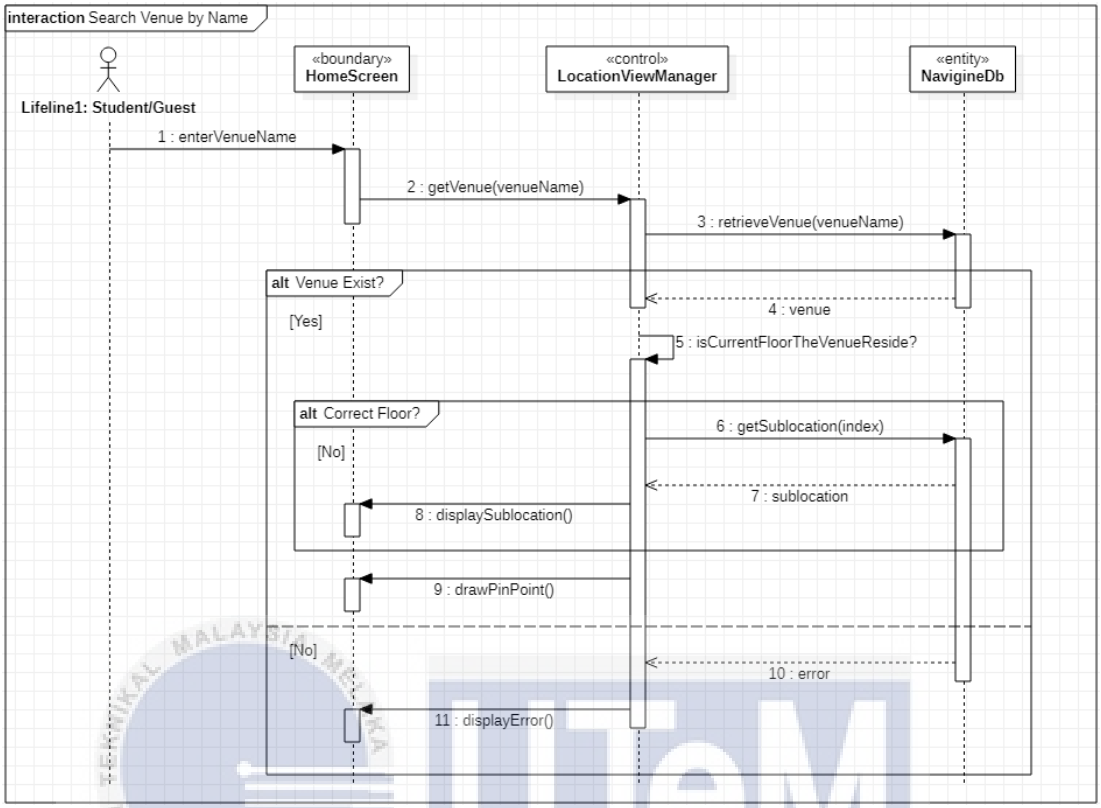

Figure 3.4 Search Venue by Name Sequence Diagram

Figure 3.4 is the sequence diagram for search venue by name use case. It starts when user enter the venue's name. The search venue request will be handle by LocationViewManager. The particular venue's data will retrieved from database and display on the right floor plan, or else an error message will be displayed.

3.5.3 Zoom In/Zoom Out

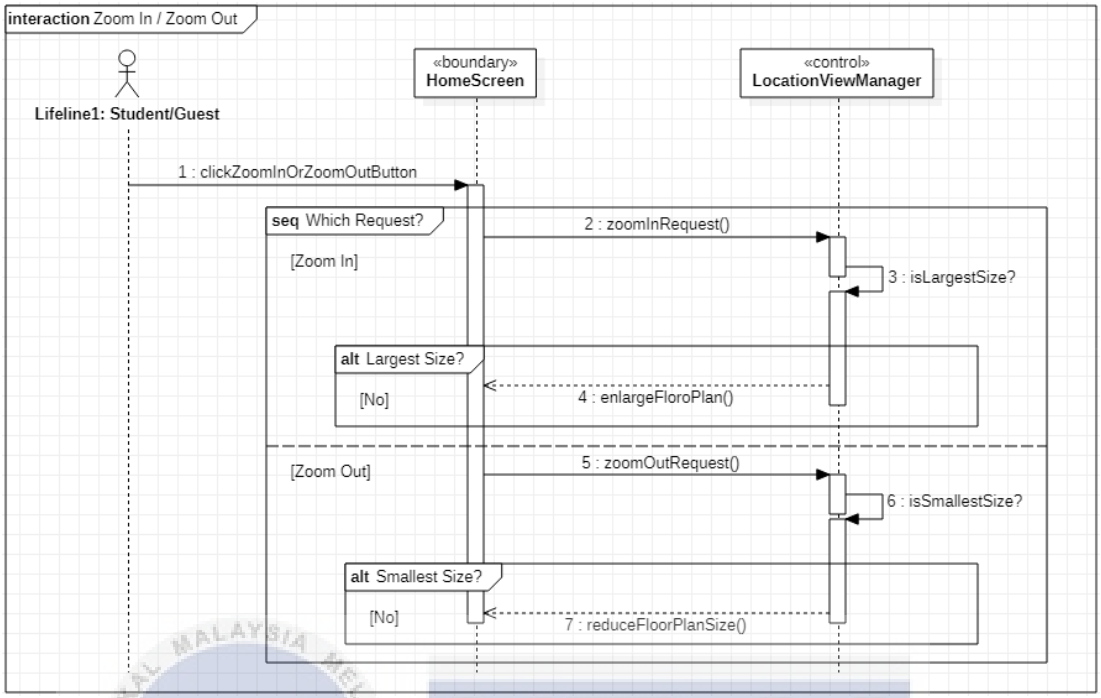

Figure 3.5 Zoom In/Zoom Out Sequence Diagram

Figure 3.5 is the sequence diagram for zoom in or zoom out use case. It starts when user click on zoom in or zoom out button. LocationViewManager will check if the floor plan is already the largest or smallest size. If not, the floor plan size will enlarge or reduce based on the request.

3.5.4 Search Venues by Filter

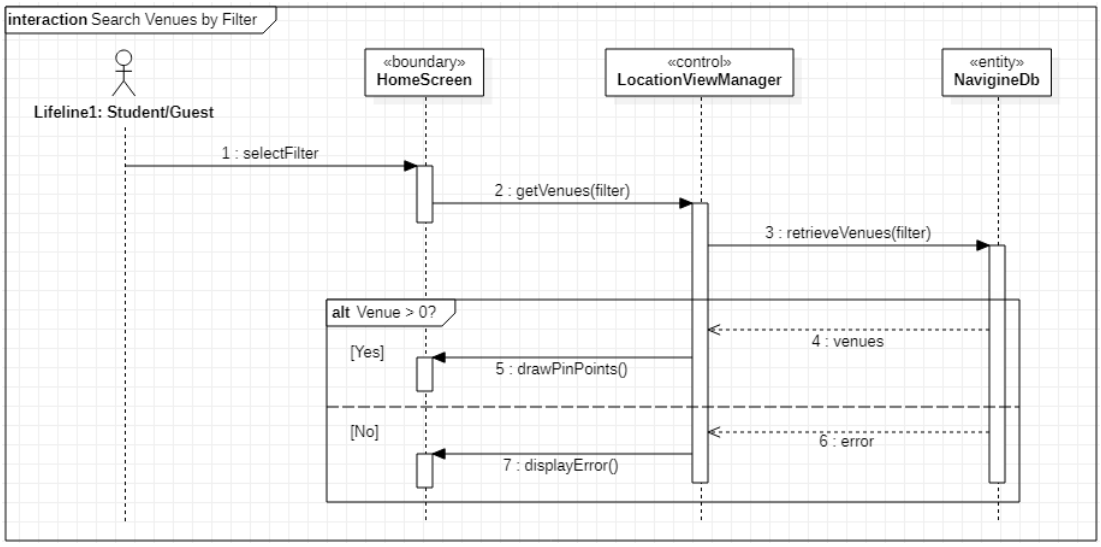

Figure 3.6 Search Venues by Filter Sequence Diagram

Figure 3.6 is the sequence diagram for search venues by filter use case. There are several filter option for user to choose on the home scree. Once a filter is chosen, it will be handled by LocationViewManager. All the venues related to the filter will be retrieved from database. Multiple pinpoints will be drawn on the floor plan if there is venue retrieved from database, or else an error message will be displayed.

3.5.5 Search Route

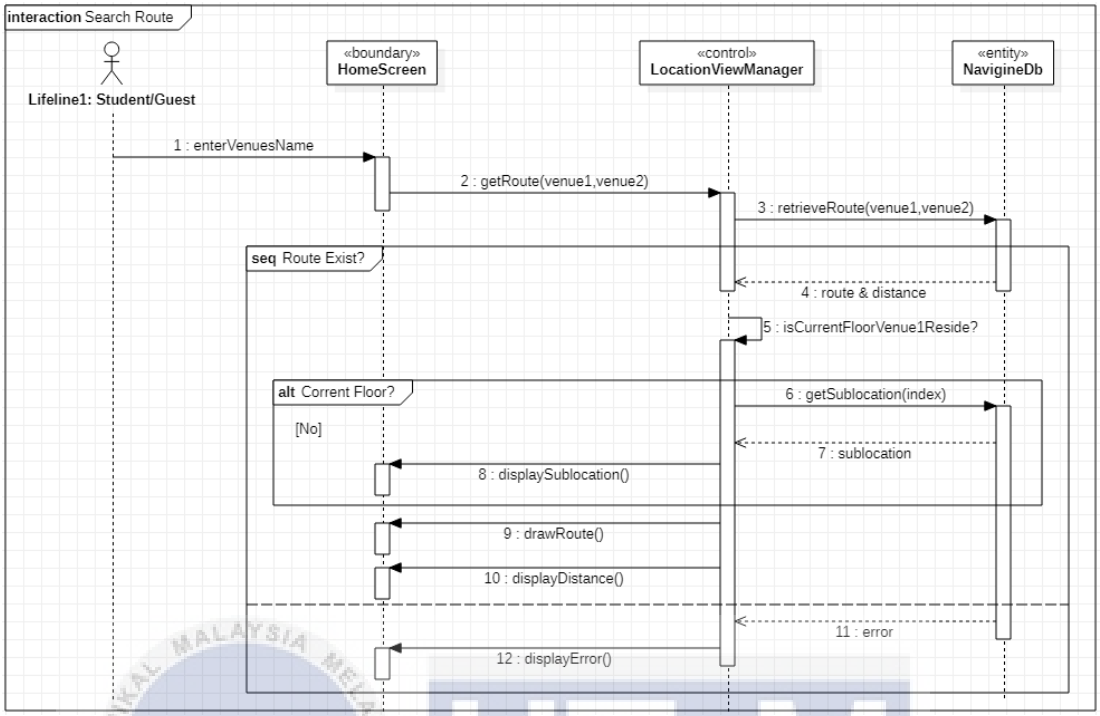

Figure 3.7 Search Route Sequence Diagram

Figure 3.7 is the sequence diagram for search route use case. It starts when user enter two venues' name. The search route request will be handled by LocationViewManager, the route and the length of route will be retrieved from database if the route is exist, else an error message will be displayed.

3.6 Conclusion

In conclusion, this chapter have analyzed the problems of current system and collected the requirements for the proposed system. In next chapter, it will visualize the requirements by using models and diagrams.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter aims to define the results of the analysis of the preliminary design and the result of the detailed design and the result of the detailed design. It is one of the most crucial part in software development, the logical design produced during previous chapters is turned into a physical design. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail in this chapter.

4.2 High-Level Design

High level design refers to the overall system design. It describes the overall architecture of the FTMK Navigator. It includes the description of system architecture, database design, and user interface design of the FTMK Navigator.

4.2.1 System Architecture

A system architecture is comprises a collection of software and system components, and constraints, a collection of system stakeholders' need statements and a rationale which demonstrates that the components, connections and constraints define a system that, if implemented, would satisfy the collection of system stakeholders' need statements (Abd-Allah *et al.*, 1997). Figure 4.1 visualize the system architecture of the FTMK Navigator.

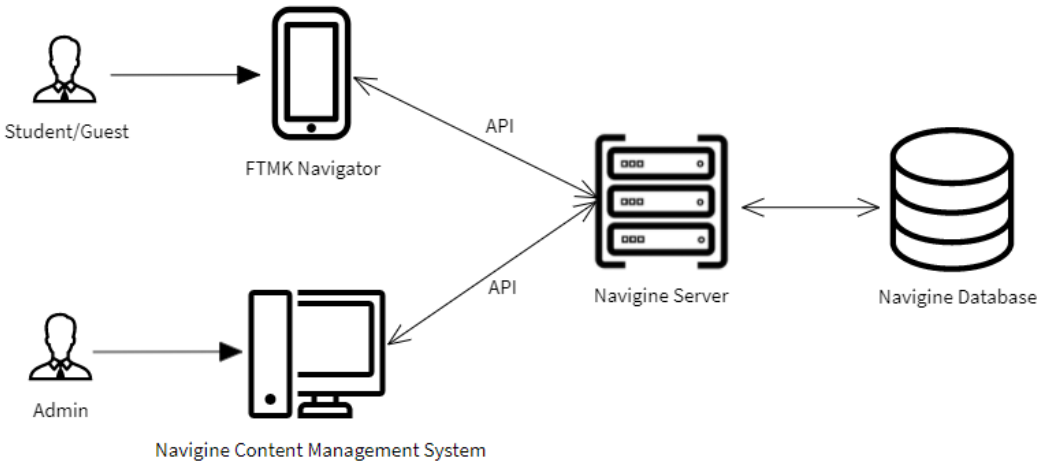

Figure 4.1 FTMK Navigator Architecture Design

Figure 4.1 shows the design of system architecture of FTMK Navigator. Student/Guest and admin are client side user. Admin is the one who use Navigine content management system (CMS) to create and manage location, floors, venues and routes. Student or guest will use the FTMK Navigator to search venues, routes and so on. Each of the request that perform by client user will be handle by Navigine server via application programming interface (API). Then the data will be insert or retrieve from database if any and will be send back to client user if any.

4.2.2 User Interface Design

User interface design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style (*What is User Interface Design? | Interaction Design Foundation (IxDF)*, no date). Section below shows all the user interfaces in FTMK Navigator.

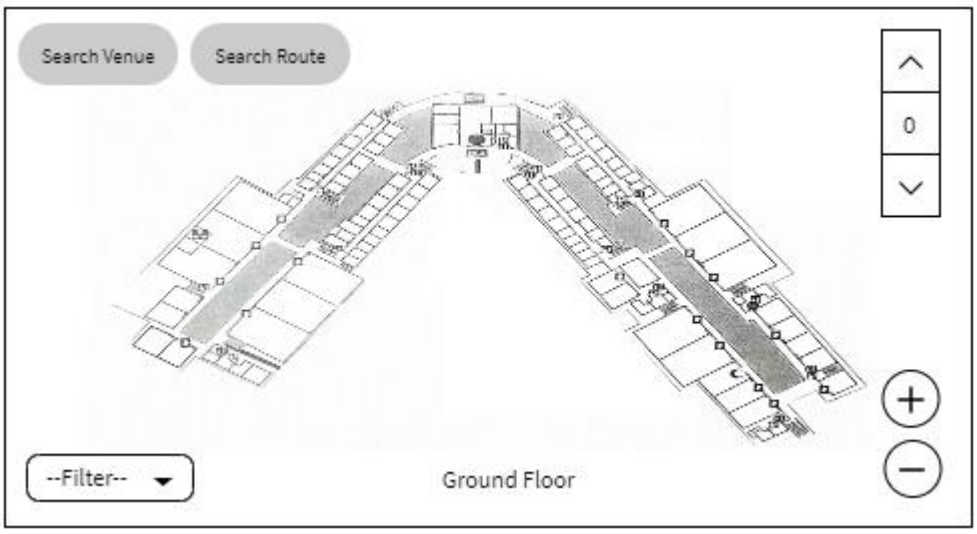

Figure 4.2 Home Screen

Figure 4.2 shows the home screen of FTMK Navigator. There are two buttons on the top left corner which allow user to search venue and search route. The up and down button on top right corner allowed user to switch floor while the buttons on bottom right corner is used to enlarge or reduce the size of floor plan image. Lastly, the dropdown menu on bottom left corner is used to search venues by a certain filter like lecture room, lab and so on.

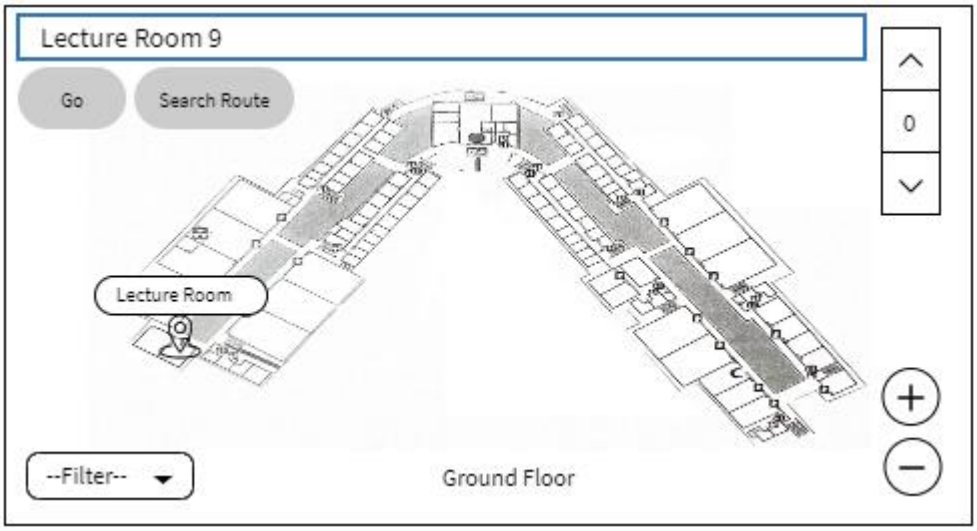

Figure 4.3 Search Venue by Name Screen

Figure 4.3 shows the search venue screen. The input field on top of the screen is for user to enter the venue's name, once user click on the "Go" button, a pinpoint will be drawn on the floor plan if the venue is found.

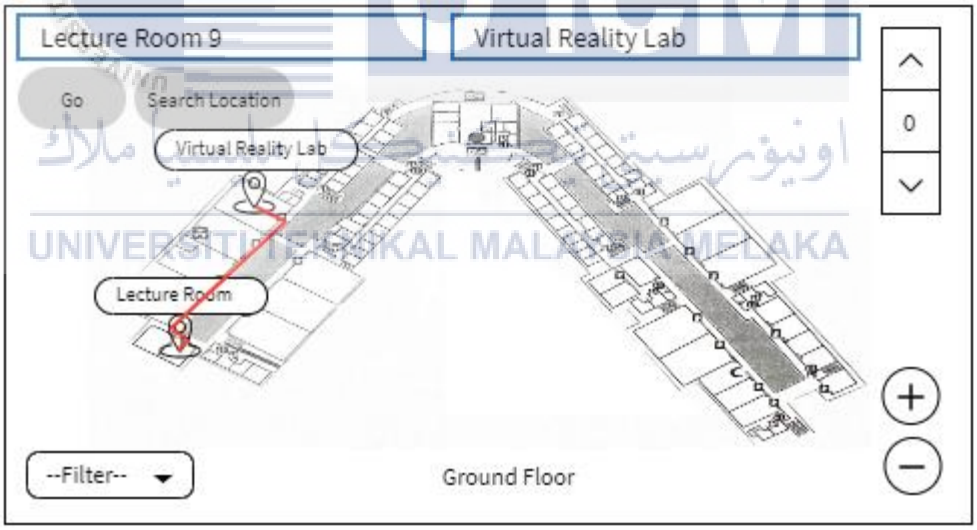

Figure 4.4 Search Route Screen

Figure 4.4 shows the search route screen. There are two input fields on top of the screen for user to enter the source and destination. Once the "Go" button is clicked, the route and pinpoints will be drawn on the screen if the route is found.

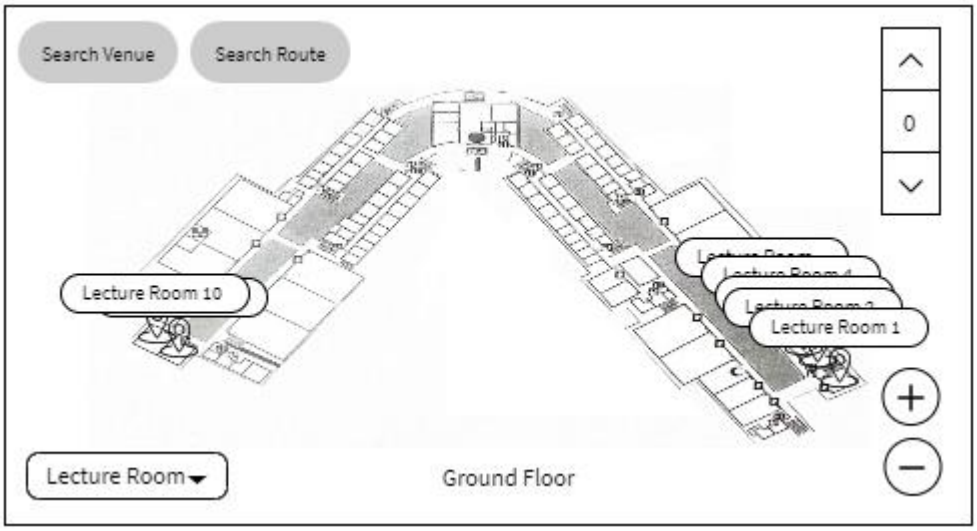

Figure 4.5 Search Venues by Filter Screen

Figure 4.5 shows the search venues by filter screen. Based on figure 4.5, the venues are found based on the "Lecture Room" filter option, so all of the venue with "Lecture Room" type will be drawn on the screen.

4.2.3 Database Design

Database design is a collection of steps that helps with designing, creating, implementing, and maintaining a business's data management system (*Database Design - Overview, Importance, and Techniques | Astera*, no date). Database design consists of three parts, the conceptual design, the logical design and the physical design which all of these three parts will be discussed in section below.

4.2.3.1 Conceptual Database Design

Conceptual database design is the first stage in the database design process. The goal at this stage is to design a database that is independent of database software and physical details. Normally conceptual database design only includes the important entities and the relationships among them. Figure 4.2 shows the conceptual database design of FTMK Navigator.

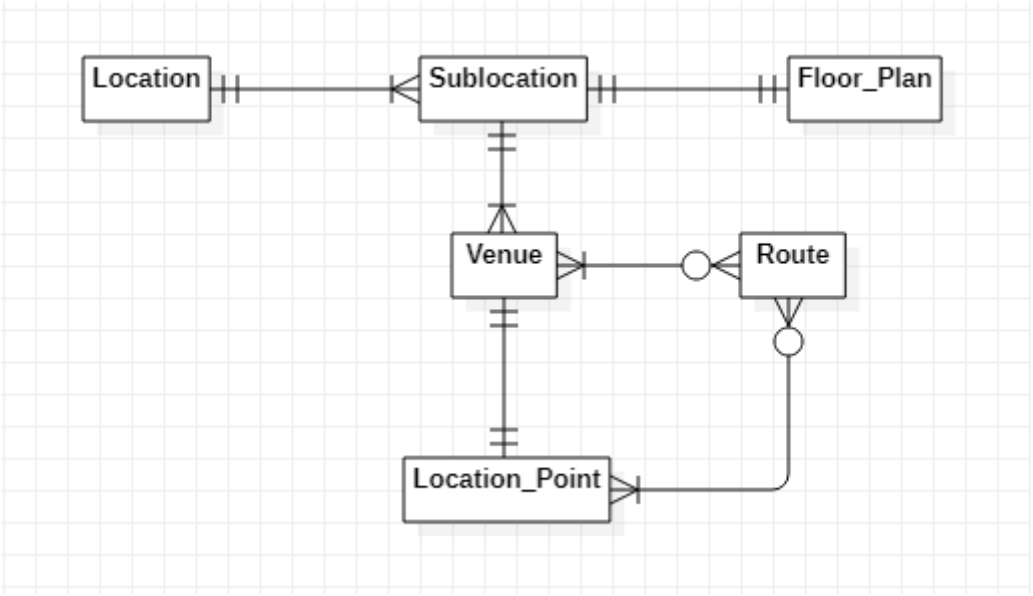

Figure 4.6 FTMK Navigator Conceptual Database Design

4.2.3.2 Logical Database Design

Logical database design is the second stage in the database design process. The goal of logical database design formalizes the objects, or entities, and the relationship between them. Logical database design add some more details to the conceptual database design such as attributes, primary key, and foreign key. Figure 4.3 shows the logical database design of FTMK Navigator.

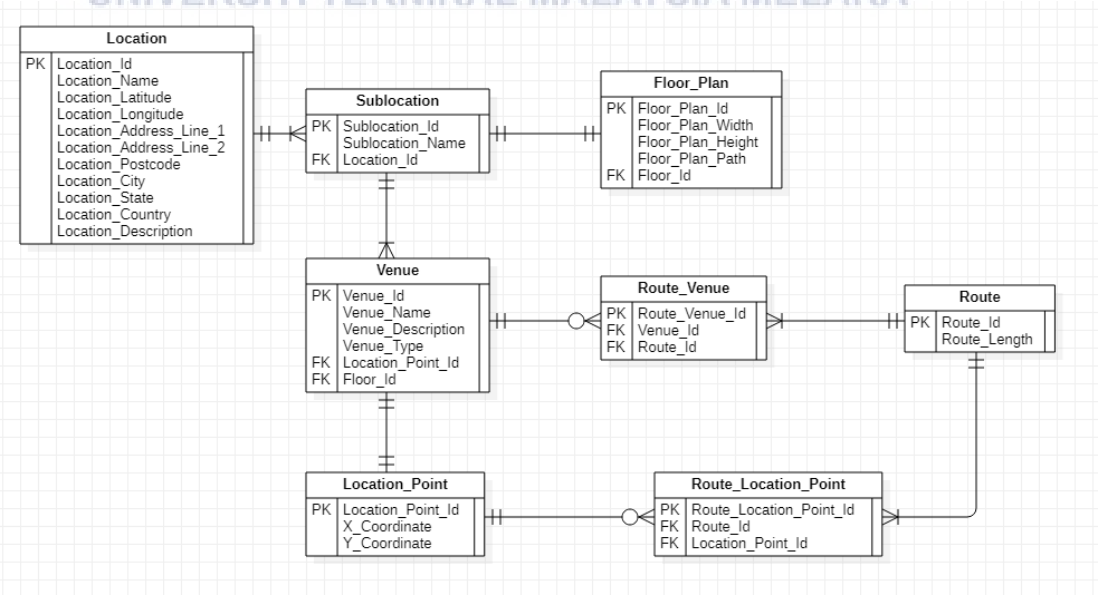

Figure 4.7 FTMK Navigator Logical Database Design

4.3 Detailed Design

Detailed design is the process of refining and expanding the preliminary design phase of a system or component to the extent that the design is sufficiently complete to be implemented (*CHAPTER 5: PRINCIPLES OF DETAILED DESIGN 1 Software Engineering Design: Theory and Practice*, no date). Section below discusses the class diagram, physical database design, and data dictionary for the FTMK Navigator.

4.3.1 Class Diagram

Class diagram is used to describe the static view of an application (Mcgill *et al.*, 2002). There are different type of design patterns for class diagram, which is the Model-View-Controller (MVC), Model-View-Presenter (MVP), and Model-View-ViewModel (MVVM). In this project, MVC design pattern is used to present the class diagram. Figure 4.8 shows the class diagram of FTMK Navigator.

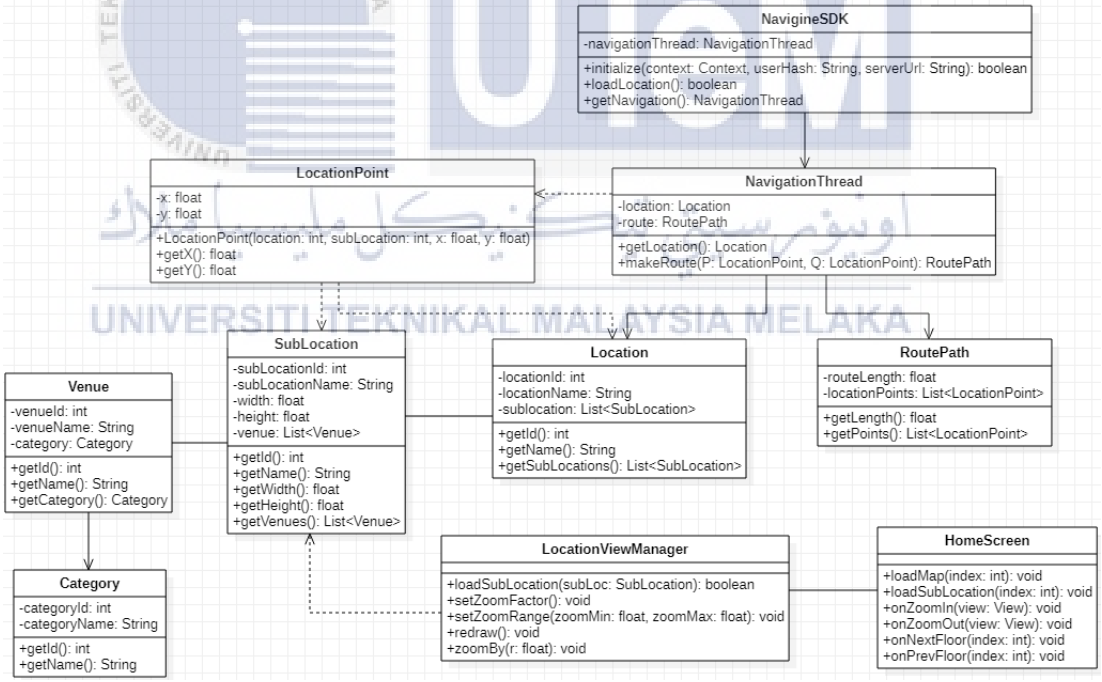

Figure 4.8 FTMK Navigator Class Diagram

4.3.2 Physical Database Design

Physical database design is the third stage of database design process. The goal of physical database design it to optimize performance while ensuring data integrity

by avoiding unnecessary data redundancies. Physical database design shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationship between tables. Figure 4.4 shows the physical database design of FTMK Navigator.

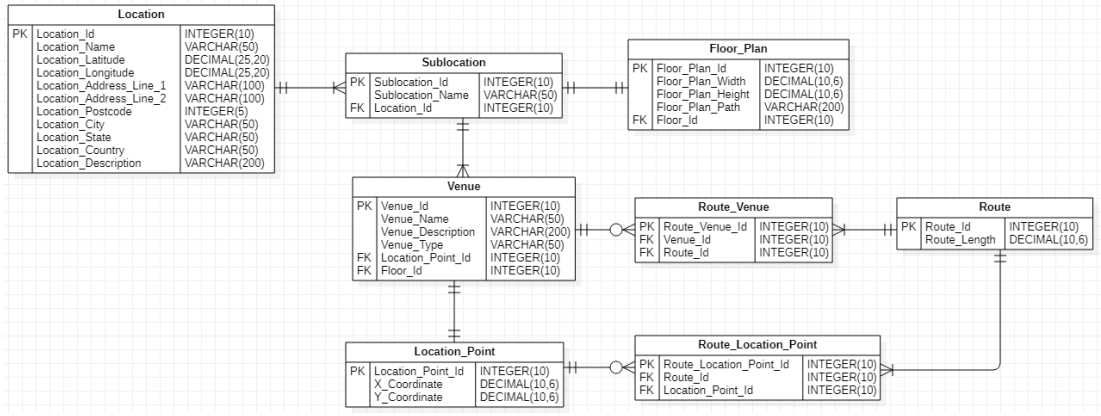

Figure 4.9 FTMK Navigator Physical Database Design

4.3.3 Data Dictionary

A data dictionary is a centralized repository of information about data descriptions such as meaning, relationships to other data, responsibility, origin, usage, and format (Uhrowczik, no date). It describes the meanings and purposes of data elements within the context of a project, and provides guidance on interpretation, accepted meanings and representation. Section below shows the data dictionary for this project.

Table 4.1 Location Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Location_Id	Integer	10	Primary Key	-	An unique id for a location
Location_Name	Varchar	50	Not Null	-	Name of the location

Location_Latitude	Decimal	25,20	Not Null	-	Latitude of the location
Location_Longitude	Decimal	25,20	Not Null	-	Longitude of the location
Location_Address	Varchar	100	-	-	Address of the location
Location_Description	Varchar	200	-	-	Description for the location

Table 3.1 shows the data requirements for location. Location is the building where the indoor navigation will be built on, which in this project, the location is the FTMK.

Table 4.2 Sublocation Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Sublocation_Id	Integer	10	Primary Key	-	An unique id for floor of a location
Sublocation_Name	Varchar	50	Not Null	-	Name for a specific floor of a location
Location_Id	Integer	10	Foreign Key	Location	The unique id of a location

					created by user

Table 3.2 shows the data requirements for sublocation. Sublocation, in simple, is the floor of a location. Each location will have at least one or many sublocation, hence the location id will be used to identify which location the floor is on.

Table 4.3 Floor_Plan Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Floor_Plan_Id	Integer	10	Not Null	-	An unique id for each floor plan
Floor_Plan_Width	Decimal	10,6	Not Null	-	Width of the floor plan
Floor_Plan_Height	Decimal	10,6	Not Null	-	Height of the floor plan
Floor_Plan_Path	Varchar	200	Not Null	-	A path to get the floor plan image
Floor_Id	Integer	10	Foreign Key	Floor	An unique id for a floor of a location

Table 3.3 shows the data requirements for floor plan. Floor plan is a scaled diagram of a room or building viewed from above. In FTMK, there are four floors/sublocations which result in four floor plans. And each floor plan is correspond to one of the floor.

Table 4.4 Location_Point Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Location_Point_Id	Integer	10	Primary Key	-	A unique id for a specific position within a sublocation
X_Coordinate	Decimal	10,6	Not Null	-	X-coordinate of the location point within a sublocation
Y_Coordinate	Decimal	10,6	Not Null	-	Y-coordinate of the location point within a sublocaiton

Table 3.4 shows the data requirements for location point. Location point is used for representing certain positions (points) within a sublocation.

Table 4.5 Venue Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Venue_Id	Integer	10	Primary Key	-	An unique id for a venue
Venue_Name	Varchar	50	Not Null	-	Name of the venue
Venue_Description	Varchar	200	-	-	Description for a venue
Venue_Type	Varchar	50	-	-	Type of the venue (e.g. classroom, lab)
Location_Point_Id	Integer	10	Foreign Key	Location Point	A unique id for a specific position within a sublocation
Floor_Id	Integer	10	Foreign Key	Floor	An unique id for floor of a location

Table 3.5 shows the data requirements for venue. Venue is a specific room or space within a sublocation. A venue is correspond to one of the floor and location point. There are many types for venues such as classroom, lab, and restroom and so on.

Table 4.6 Route Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Route_Id	Integer	10	Primary Key	-	An unique id for a route
Route_Length	Decimal	10,6	-	-	Distance between two points

Table 3.6 shows the data requirements for route. Route is a way for travel or movement, the path from a starting point to a destination. Each route is identified with a route id and has a route length which is the distance between two points of the route.

Table 4.7 Route_Venue Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Route_Venue_Id	Integer	10	Primary Key	-	An unique id for a venue in a route
Venue_Id	Integer	10	Foreign Key	Venue	An unique id for a venue
Route Id	Integer	10	Foreign Key	Route	An unique id for a route

Table 3.7 shows the data requirements for route_venue. It is a bridge between route and venue. With the data in route_venue, the venues that involve in a route can be get easily.

Table 4.8 Route_Location_Point Table

Column Name	Data Type	Field Length	Constraint	FK Reference Table	Description
Route_Location_ Point_Id	Integer	10	Primary Key	-	An unique id for location point in a route
Route_Id	Integer	10	Foreign Key	Route	An unique id for a route
Location_Point_ Id	Integer	10	Foreign Key	Locaiton_P oint	An unique id for a location point within a sublocation

Table 3.8 shows the data requirements for route_location_point. It is a bridge between route and location point. With the data in route_location_point, all location points in a route can be get easily and a route can be drawn by using the location points.

4.4 Conclusion

In conclusion, this chapter discussed all types of software design such as architectural design, high-level design and detailed design. Software design provide sufficient detailed data and information about the FTMK Navigator and it will be useful for next chapter which is the implementation phase of software development.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

After the requirements and design activities are completed in previous chapters, the implementation of FTMK Navigator will be conducted in this chapter. Details like software development environment setup, software configuration management, configuration environment setup, version control procedure, and implementation status are the expected output at the end of this chapter.

5.2 Software Development Environment Setup

Software development environment is the collection of hardware and software tools a system developer uses to build software systems (*What are environments in software development? A guide to the development, beta, and production environments.* | *Codebots*, no date). Table 5.1 descripts the software development environment for this project.

Table 5.1 Software Development Environment Setup

Development Environment	Description
Desktop	• With Windows 10
Android Studio	• Official Integrated Development Environment (IDE) for Android application development.

Navigine CMS	• A global provider of integrated mobile indoor positioning technologies that enable advanced indoor and outdoor navigation.
Fiber-Optic Internet	• With over 100 Mbps speed

5.3 Software Configuration Management

Software configuration management is the discipline that enable developers to keep evolving software products under control, and thus contributes to satisfying quality and delay constraints (Estublier, no date). Sections below descript the software configuration management for this project.

5.3.1 Configuration Environment Setup

The purpose of configure environment in this project is to ensure software runs correctly and execute as expected. Table 5.2 states the environment configuration of software used in this project.

Table 5.2 Configuration Environment Setup

Android Studio	Navigine CMS
• Minimum SDK Version 26 • Compile SDK Version 30 • Target SDK Version 30 • Build Tools Version 30.0.3	• Enable Venues • Enable Routes • Enable Elevation

• Enable Internet Access Permission • Enable Network State Access	

5.3.2 Version Control Procedure

N/A

5.4 Implementation Status

There are several modules to be built in this project, the implementation status for each module is jotted down in order to record the progress of implementation for each module. Table 5.3 shows the implementation progress of the specified module at the time of writing.

Table 5.3 Implementation Status

Module	Progress (%)
Display Floor's Plan	100
Search Venue by Venue's Name	100
Search Venues by Filter	100
Draw Route	100
Estimate Distance	100

5.5 Conclusion

In conclusion, the implementation is the action to turn system requirements and design into actual code. In this chapter, some of the important details such as software development environment setup, software configuration management and so on are discussed. In next chapter, the testing on FTMK Navigator will be carry out in order to ensure FTMK Navigator is performing well and bug free.

CHAPTER 6: TESTING

6.1 Introduction

Software testing is a process of executing a program or application with the intent of finding the software bugs. Testing is important to ensure the quality of the product and is required for an effective performance of software application or product (Arvind *et al.*, no date). In this chapter, testing details such as test plan, test strategy, test design, test results and analysis are discussed.

6.2 Test Plan

A test plan refers to a detailed document that catalogs the test strategy, objectives, schedule, estimations, deadlines and the resources required for completing that particular project (*Test Planning: A Detailed Guide | BrowserStack*, no date). In this project, three tests will be performed on FTMK Navigator which is the unit testing, system testing and usability testing. Hence, a test plan will be created for the tests to outline the test organization and environment, define the roles and responsibilities of testers and so on. Section below shows the details of test plan for this project.

6.2.1 Test Objective

This test plan embarks on the following objectives:

- To ensure the FTMK Navigator conforms to requirements stated previously.
- To ensure the FTMK Navigator meets the quality specifications defined previously.

- To identified and fixed bugs/issues.

6.2.2 Test Organization

Test organization describes the personnel that will perform the testing. In order to maximize the effect of testing, FTMK Navigator will be tested by developer itself and at least 35 students and guests that been visit FTMK. Table 6.1 shows the details of test organization.

Table 6.1 Test Organization of FTMK Navigator

Test Type	Personnel Involve	Description
Unit Testing	Developer of FTMK Navigator	Unit testing is selected in this project to test the smallest unit of code before integrate them into a larger unit. Unit test is carried by developer of FTMK Navigator. Unit testing is fall under white box testing because it is tested by developer itself.
System Testing	Developer of FTMK Navigator	System testing is selected in this project to test the complete and fully integrated FTMK Navigator. Test cases and test data will be created for each module of FTMK Navigator and the status (Success/Fail) will be recorded for further improvement. System testing in this project is fall under white box testing because it is tested by developer itself.
Usability Testing	Developer of FTMK Navigator and at	Usability testing is selected in this project to test the FTMK Navigator from user's point of view. Usability testing will be carried out after system testing. At least 35

	least 35 FTMK students and guests	students and guests will be asked to try on FTMK Navigator and a questionnaire will be answered by them. Analysis will be performed on the questionnaire to evaluate the satisfaction of user on FTMK Navigator.

6.2.3 Test Environment

Test environment is a setup of software and hardware to execute test cases (*Test Environment for Software Testing*, no date). Table 6.2 shows the test environment for this project.

Table 6.2 Test Environment of FTMK Navigator

Component	Requirement
Hardware	• Desktop/Laptop with Windows 10 • Android Mobile Phone with Android 8 or above
Software	• Android Studio

6.2.4 Test Schedule

Test schedule describe the duration and test cycle of system testing and usability testing in this project. Table 6.3 shows the details of test schedule.

Table 6.3 Test Schedule of FTMK Navigator

Test Type	Module	Duration	Test Cycle
System Testing	Display Floor Plan	5 minutes	1
	Search Venue by Venue's Name	5 minutes	1
	Search Venues by Filter	5 minutes	1
	Draw Route from Source to Destination	5 minutes	1
Usability Testing	-	3 days	1

6.2.5 Test Completeness

Test completeness define the criteria that the testing is completed. There are a few criteria in this project as shown below.

- 100% test coverage
- All manual test cases executed
- All bugs/errors are fixed

6.3 Test Strategy

In this project, two testing techniques were used to test the FTMK Navigator. The first testing technique used is the white box testing. White box testing is a software

testing technique in which the tester has knowledge about the internal structure of the software. White box testing is used in this project because it is efficient in finding hidden errors and problems as the tester is the developer itself.

The second testing technique used in this project is the black box testing. Black box testing is a software testing technique in which the tester test the application without any knowledge of the application's internal structure. The reason of using black box testing is to perform the test from user's point of view.

6.3.1 Classes of Tests

This section describes the type of testing that will be used in this project to test the FTMK Navigator.

6.3.1.1 Unit Testing

The first type of testing used in this project is the unit testing. Unit testing is a type of software testing where individual units of a software are tested. Unit tests are carried out by the developer of the FTMK Navigator during the process of system development. The purpose of unit testing in this project is to test the smallest unit of code and to validate if all the functions work as expected. Some of the methods of the FTMK Navigator may include numbers as variables or parameters, hence two techniques are used to carry out the unit testing. The two techniques are boundary value analysis and equivalence partitioning. There are more techniques used in this project such as the statement coverage, branch coverage and so no. These techniques are used to prevent exhaustive testing and try to cover all the possible data range in this project.

6.3.1.2 System Testing

The second type of testing used in this project is the system testing. System testing is defined as testing of a complete and fully integrated software product (*What is System Testing? Definition of System Testing, System Testing Meaning - The Economic Times*, no date). The purpose of system testing is to evaluate the end-to-end system specifications and check if the FTMK Navigator fulfill the requirements stated

previously. Test cases and test data to be used by the system testing will be produced in next few sections.

6.3.1.3 Usability Testing

The first type of testing used in this project is the usability testing. Usability testing is a testing to measure of how easily the system can be used by end users (*Usability Testing*, no date). Usability testing is used in this project is to evaluate the user friendliness of FTMK Navigator. To achieve the goal of usability testing, FTMK Navigator will be distributed and used by FTMK students and guests, after that a questionnaire will be answered by FTMK students and guests. From the questionnaire, evaluation will be performed to check the satisfaction of user on FTMK Navigator.

6.4 Test Design

Test design is the activity of deriving and specifying test cases from test conditions (*What Is Test Design Actually? - DZone Performance*, no date). The goal of test design is to create test cases and test data. Section below shows the test cases and test data for the system testing.

6.4.1 Test Description

System test will be carried out for each module of FTMK Navigator. Hence, the test case identification, test cases and expected result for each module are designed and documented. Test cases for system testing can be referenced in Appendix A.

Usability testing will be carried out by finding at least 35 students and guests to try on FTMK Navigator and answered a questionnaire after that. There are nine mandatory statements and one optional field for user to comment and suggest for improvement. For each statement, there is a linear scale from 1 to 5 which higher number means respondents are more agree with the statement. Evaluation on questionnaire will be performed to check the satisfaction of user on FTMK Navigator.

6.5 Test Results and Analysis

After system testing and usability testing have been carried out, the test result and analysis are recorded. Tables below show the test result and analysis for the FTMK Navigator.

6.5.1 Test Results and Test Analysis for System Testing

Test results for the system testing can be referenced in Appendix B.

After system testing has been carried out, all test cases have the actual result same as the expected result. It seems like the FTMK Navigator is error free so far. In next section, usability testing will be carried out which FTMK Navigator will be tested by at least 35 FTMK students and guests and a questionnaire will be answered by them.

6.5.2 Test Result and Analysis for Usability Testing

Usability testing is carried out for 3 days, from 1st September 2021 until 3rd September 2021. Total of 36 students and guests are found to test on FTMK Navigator and answer a questionnaire. The responses of users are analyzed to evaluate the satisfaction of user on FTMK Navigator.

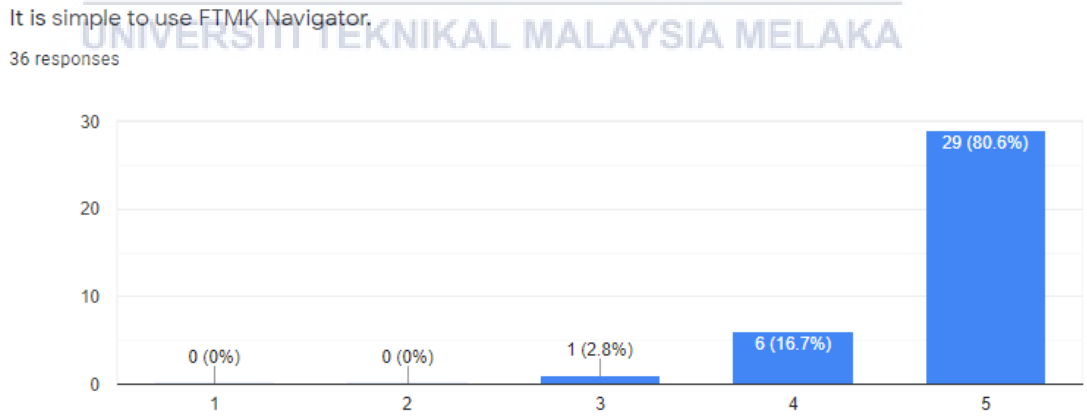

Figure 6.1 Responses of "It is simple to use FTMK Navigator"

From figure 6.1, out of 36 respondents, 29 respondents are strongly agree with that the FTMK Navigator is simple to use, 6 respondents are agree with that, and only

1 respondent is partially agree with that. It can be said that nearly all of the respondents are satisfied with the simplicity of FTMK Navigator.

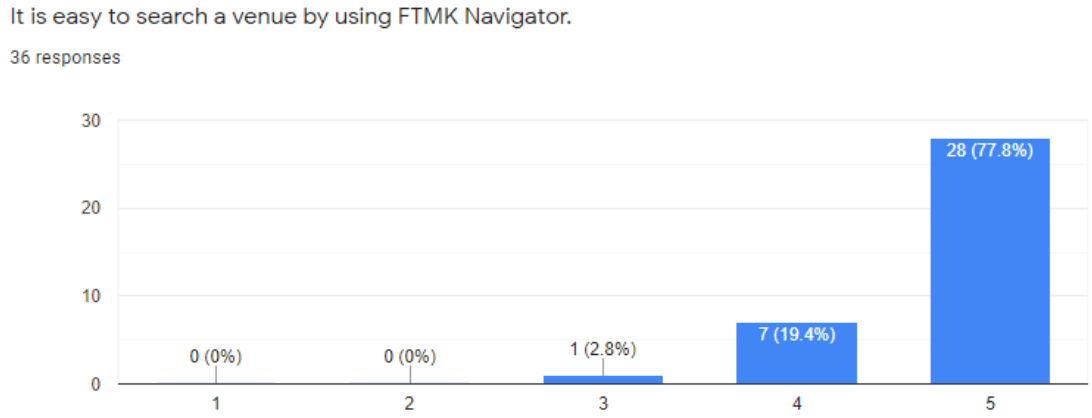

Figure 6.2 Responses of "It is easy to search a venue by using FTMK Navigator"

Figure 6.2 shows that 28 respondents think that it is very easy to search a venue by using FTMK Navigator. 7 of them are agree and only one of them is partially agree with that. As a result, it can be said that most of the respondents have no problems with the feature of searching venue.

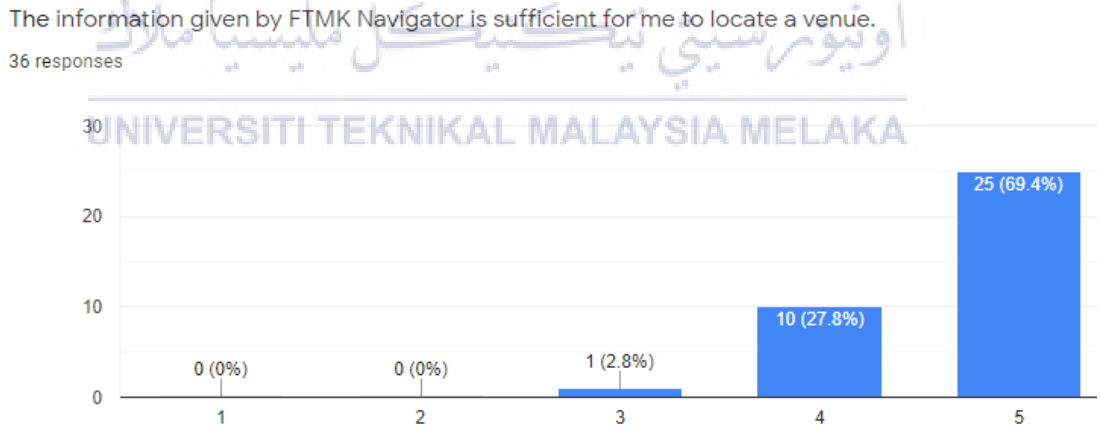

Figure 6.3 Responses of "The information given by FTMK Navigator is sufficient for me to locate a venue"

Figure 6.3 shows that 25 of respondents are strongly agree with that the information given by FTMK Navigator is sufficient for them to locate a venue. Only one of them are partially agree with that. Overall, most of the respondents can locate a venue very easily by using the information provided by FTMK Navigator.

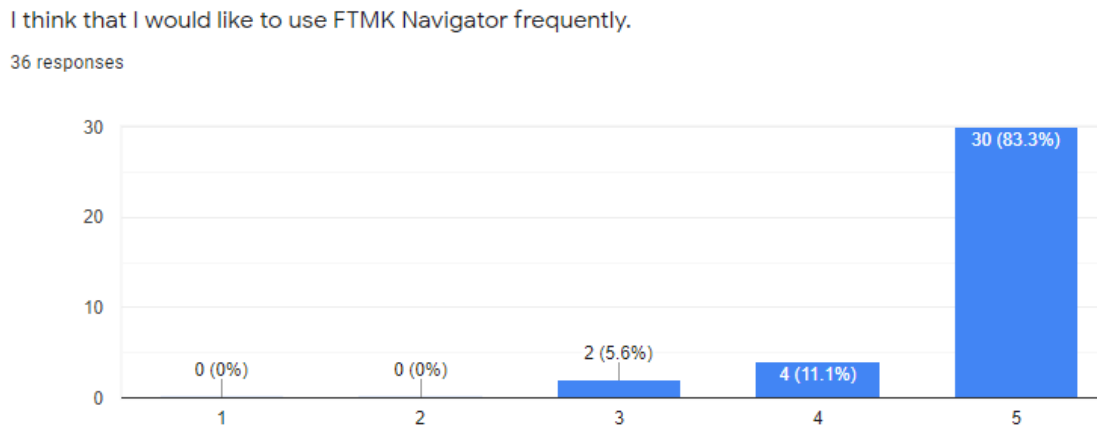

Figure 6.4 Responses of "I think that I would like to use FTMK Navigator frequently"

In figure 6.4, it shows that more than 80% of respondents would like to use FTMK Navigator frequently. From figure 6.4, it can be said that FTMK Navigator is very useful and thoughtful from the perspective of user and as a result it makes user want to use it frequently.

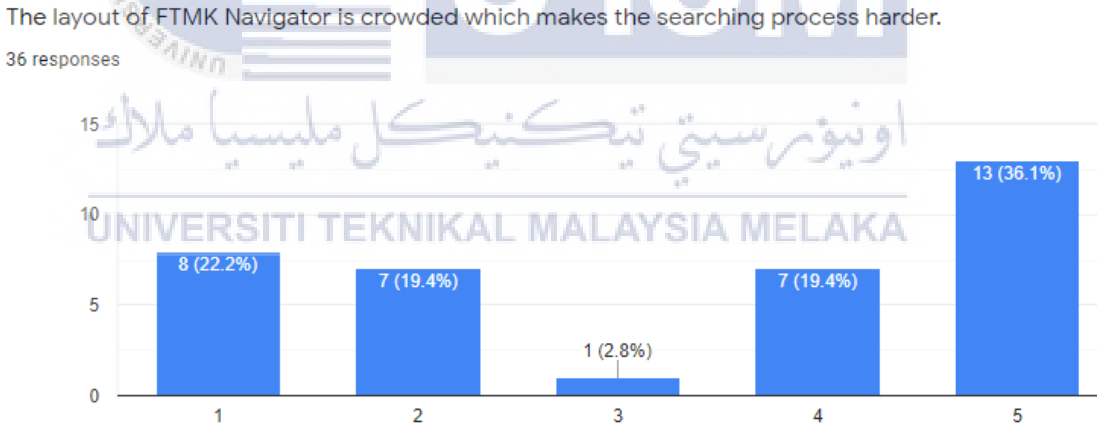

Figure 6.5 Responses of "The layout of FTMK Navigator is crowded which makes the searching process harder"

In figure 6.5, it shows that only 13 respondents think that the layout of FTMK Navigator does not burden the searching process while more than quarter of respondents think that the layout of FTMK Navigator does burden the searching process. The layout is one of the weakness of FTMK Navigator because the layout of FTMK Navigator looks crowded on small screen devices. Hence, FTMK Navigator is

not suitable for users with small screen device because it makes the searching process harder.

In short, from the responses of questionnaire, we can conclude that there is not much disadvantage of FTMK Navigator and it has many advantages to user. User think that it is simple to use, it is easy to search venue and user would like to use it frequently because of those advantages.

6.6 Conclusion

In conclusion, system testing and usability testing are carried out for the FTMK Navigator. In system testing, no failure or error are found on FTMK Navigator but error free does not mean that all user will fully satisfied with FTMK Navigator as we seen in the usability testing. Hence, testing is very important as it will reveals errors and disadvantage of FTMK Navigator.

CHAPTER 7: PROJECT CONCLUSION

7.1 Observation on Weaknesses and Strengths

From the development and testing of this project, it is observed that this project has multiple weakness to be improved. However, there are also strengths that bring benefits to users.

7.1.1 Weakness

One of the weakness of FTMK Navigator is that it is only available on android devices. Users with IOS mobile device is not able to use FTMK Navigator. Other than that, as FTMK Navigator is a mobile application, for users who has device with small screen, it might be difficult for them to use the FTMK Navigator as the layout of user interface will be crowded.

7.1.2 Strength

One of the strength of FTMK Navigator is that when user is entering text on the text fields, some suggestion will be given to user for them to select. It might be helpful sometime when the user forgot the venue's name. In addition, the information given by FTMK Navigator after user search venue or route is very thoughtful. FTMK Navigator will states the venues beside the venues search by user, it is to help user from finding or locating the venues they search.

7.2 Propositions for Improvement

One of the improvement of FTMK Navigator is that the route drawn on the floor plan should be animated. Which mean the route drawn on floor plan should be a

repeated growing line between the source and destination. With animated route on the floor plan, user can distinguish the source and destination easily with just a glance.

7.3 Project Contribution

The FTMK Navigator is specially built for FTMK students and guests that visit FTMK. FTMK Navigator provide the floor plan of each floor in FTMK for them to understand the structure of FTMK. It makes the process of searching a venue in FTMK easier and reduces the time consuming on searching a venue in FTMK.

7.4 Conclusion

In conclusion, the final version of FTMK Navigator basically meets the requirements stated in previous chapters. It is a good experience to build an application by oneself as you will understand more on the process of building an application and makes you a better developer.

REFERENCES

10 benefits of fact-finding | ThinkAdvisor (no date). Available at: https://www.thinkadvisor.com/2014/01/29/10-benefits-of-fact-finding/ (Accessed: 20 April 2021).

Abd-Allah, A. *et al.* (1997) *On the Definition of Software System Architecture SEE PROFILE On the Definition of Software System Architecture*. Available at: https://www.researchgate.net/publication/2670553 (Accessed: 23 June 2021).

Arvind, M. *et al.* (no date) 'Software Testing'. Available at: www.himpub.com (Accessed: 26 August 2021).

CHAPTER 5: PRINCIPLES OF DETAILED DESIGN 1 Software Engineering Design: Theory and Practice (no date).

CS 410/510 - Software Engineering class notes (no date). Available at: https://cs.ccsu.edu/~stan/classes/CS410/notes16/04-Requirements.html (Accessed: 15 May 2021).

Database Design - Overview, Importance, and Techniques | Astera (no date). Available at: https://www.astera.com/type/blog/all-you-need-to-know-about-database-design/ (Accessed: 16 June 2021).

Definition Of Fact Finding Techniques Information Technology Essay (no date). Available at: https://www.ukessays.com/essays/information-technology/definition-of-fact-finding-techniques-information-technology-essay.php (Accessed: 20 April 2021).

Estublier, J. (no date) 'Software Configuration Management A Road Map'.

Kharisma, A. and Nurhasan, A. S. (no date) *What is Android? Related papers IMPLEMENTAT ION OF SQL SERVER BASED ON SQLIT E ENGINE ON ANDROID PLAT FORM Edit or IJRET*. Available at: http://retrodev.com/android/ (Accessed: 23 June 2021).

Mcgill, M. J. *et al.* (2002) *UML class diagram syntax: an empirical study of comprehension*. Available at: https://www.researchgate.net/publication/2528936 (Accessed: 23 June 2021).

Object-oriented analysis and design (OOAD) | Guide books (no date). Available at: https://dl.acm.org/doi/book/10.5555/2031615 (Accessed: 18 June 2021).

Sengupta, S. and Bhattacharya, S. (no date) *Formalization of UML Use Case Diagram-A Z Notation Based Approach*.

Sunway Pyramid launches real-time in-mall navigation mobile app (no date). Available at: https://www.nst.com.my/lifestyle/bots/2019/04/477461/sunway-pyramid-launches-real-time-mall-navigation-mobile-app (Accessed: 20 April 2021).

Test Environment for Software Testing (no date). Available at: https://www.guru99.com/test-environment-software-testing.html (Accessed: 26 August 2021).

Test Planning: A Detailed Guide | BrowserStack (no date). Available at: https://www.browserstack.com/guide/test-planning (Accessed: 26 August 2021).

Uhrowczik, P. P. (no date) *Data Dictionary/Directories*.

Usability Testing (no date). Available at: https://www.tutorialspoint.com/software_testing_dictionary/usability_testing.htm (Accessed: 26 August 2021).

What are environments in software development? A guide to the development, beta, and production environments. | Codebots (no date). Available at: https://codebots.com/app-development/what-are-environments-in-software-development-a-guide-to-the-development-beta-and-production-environments (Accessed: 9 August 2021).

What is System Testing? Definition of System Testing, System Testing Meaning - The Economic Times (no date). Available at: https://economictimes.indiatimes.com/definition/system-testing (Accessed: 26 August 2021).

What Is Test Design Actually? - DZone Performance (no date). Available at: https://dzone.com/articles/what-is-test-design-actually (Accessed: 26 August 2021).

What is User Interface Design? | Interaction Design Foundation (IxDF) (no date). Available at: https://www.interaction-design.org/literature/topics/ui-design (Accessed: 16 June 2021).

Zulkiflie, S. A., Kamaruddin, N. and Wahab, A. (2020) 'Dynamic navigation indoor map using Wi-Fi fingerprinting mobile technology', *Bulletin of Electrical Engineering and Informatics*, 9(2), pp. 739–746. doi: 10.11591/eei.v9i2.2066.


UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX A

Test Case for Display Floor Plan Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result
DisplayFloorPlan_01	Verify display floor plan function for first time user.	Open the FTMK Navigator with internet connection.	1. Install the FTMK Navigator. 2. Open FTMK Navigator with internet connection.	-	The ground floor's floor plan is displayed as the first floor plan.
DisplayFloorPlan_02	Verify display floor plan function for first time user.	Open the FTMK Navigator without internet connection.	1. Install the FTMK Navigator 2. Open FTMK Navigator without internet connection.	-	A message is shown to inform user to check their internet connection.
DisplayFloorPlan_03	Verify display floor plan function for first time user.	Open the FTMK Navigator with internet connection and change floor number by	1. Install the FTMK Navigator.	-	The floor plans is changed floor to floor

		pressing up and down button.	2. Open FTMK Navigator with internet connection. 3. Press on up and down button on top right corner.		based on the current floor's number.

Test Case for Search Venue by Venue's Name Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result
SearchVenue_01	Verify search venue by venue's name function.	Test search venue by venue's name function without entering anything.	1. Press "Search Location" button. 2. Press "Go" button without entering anything on the text field.	-	A small red icon and "Required" is appeared on the end of text field to remind user to enter venue name.
SearchVenue_02	Verify search venue by venue's name function.	Test search venue by venue's name function by entering a venue name which is not present in database.	1. Press "Search Location" button. 2. Enter a venue name which is not present in database. 3. Press "Go" button.	"Lobby A"	A message is shown to tell user the venue name entered is not found.

SearchVenue_03	Verify search venue by venue's name function.	Test search venue by venue's name by entering a venue name which is present in database.	1. Press "Search Location" button. 2. Enter a venue name which is present in database. 3. Press "Go" button.	"Lecture Room 9"	A pin point is drawn on the floor plan and the venue's name is drawn on top of the pin point.

Test Case for Search Venue by Filter Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result
FilterVenue_01	Verify search venues by filter function.	Test search venues by filter function by select one of the option in spinner.	1. Press on filter spinner on bottom left corner and pick one of the venue's category.	-	Pin points are drawn on the floor plans based on the venue's category picked.

Test Case for Draw Route from Source to Destination Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result
DrawRoute_01	Verify draw route from source to destination function.	Test draw route function without entering anything.	1. Press "Search Route" button. 2. Press "Go" button without entering anything on "Source" and "Destination" text field.	-	A small red icon and "Required" is appeared on the end of each text field.
DrawRoute_02	Verify draw route from source to destination function.	Test draw route function with entering a venue name on one of the text field only, either "Source" or "Destination" text field.	1. Press "Search Route" button. 2. Enter a venue name on "Source" or "Destination" text field and the other left empty.	"Recap Room"	A small red icon and "Required" is appeared on the end of empty text field.

			3. Press "Go" button.		
DrawRoute_03	Verify draw route from source to destination function.	Test draw route function with entering a venue name which is not present in database on "Source" and "Destination" text fields.	1. Press "Search Route" button. 2. Enter a venue name which is not present in database on "Source" and "Destination" text field. 3. Press "Go" button.	"Lab 1", "Software Engineering Room"	A message is shown to tell user the venue name entered is not found.
DrawRoute_04	Verify draw route from source to destination function.	Test draw route function with entering same venue name which is not present in database on "Source" and "Destination" text fields.	1. Press "Search Route" button 2. Enter same venue name which is present in database on "Source" and "Destination" text field.	"Virtual Reality Studio"	A small red icon and "Please enter different venue name" is appeared on the end of last focused text field.

			3. Press "Go" button.		
DrawRoute_05	Verify draw route from source to destination function.	Test draw route function with entering different venue name which is not present in database on "Source" and "Destination" text fields.	1. Press "Search Route" button. 2. Enter different venue names which is present in database on "Source" and "Destination" text field. 3. Press "Go" button.	"Programming Lab 1", "Gaming Lab"	Two pin points will be drawn on floor plan and a shortest route is drawn between the two pin points and estimated distance is calculated.

APPENDIX B

Test Result for Display Floor Plan Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Status
DisplayFloor Plan_01	Verify display floor plan function for first time user.	Open the FTMK Navigator with internet connection.	1. Install the FTMK Navigator. 2. Open FTMK Navigator with internet connection.	-	The ground floor's floor plan is displayed as the first floor plan.	The ground floor's floor plan is displayed as the first floor plan.	Pass
DisplayFloor Plan_02	Verify display floor plan function for first time user.	Open the FTMK Navigator without internet connection.	1. Install the FTMK Navigator 2. Open FTMK Navigator without internet connection.	-	A message is shown to inform user to check their internet connection.	A message is shown to inform user to check their internet connection.	Pass

DisplayFloor Plan_03	Verify display floor plan function for first time user.	Open the FTMK Navigator with internet connection and change floor to floor.	1. Install the FTMK Navigator. 2. Open FTMK Navigator with internet connection. 3. Press on up and down button on top right corner.	-	The floor plans is changed floor to floor based on the current floor's number.	The floor plans is changed floor to floor based on the current floor's number.	Pass

Test Result for Search Venue by Venue's Name Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Status
SearchVenue_01	Verify search venue by venue's name function.	Test search venue by venue's name function without entering anything.	1. Press "Search Location" button. 2. Press "Go" button without entering anything on the text field.	-	A small red icon and "Required" is appeared on the end of text field to remind user to enter venue name.	A small red icon and "Required" is appeared on the end of text field to remind user to enter venue name.	Pass
SearchVenue_02	Verify search venue by venue's name function.	Test search venue by venue's name function by entering a venue name which is not present in database.	1. Press "Search Location" button. 2. Enter a venue name which is not present in database.	"Lobby A"	A message is shown to tell user the venue name entered is not found.	A message is shown to tell user the venue name entered is not found.	Pass

			3. Press "Go" button.				
SearchVenue_03	Verify search venue by venue's name function.	Test search venue by venue's name by entering a venue name which is present in database.	1. Press "Search Location" button. 2. Enter a venue name which is present in database. 3. Press "Go" button.	"Lobby", "Lecture Room 9"	A pin point is drawn on the floor plan and the venue's name is drawn on top of the pin point.	A pin point is drawn on the floor plan and the venue's name is drawn on top of the pin point.	Pass

Test Result for Search Venues by Filter Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Status
FilterVenue_01	Verify search venues by filter function.	Test search venues by filter function by select one of the option in spinner.	1. Press on filter spinner on bottom left corner and pick one of the venue's category.	-	Pin points are drawn on the floor plans based on the venue's category picked.	Pin points are drawn on the floor plans based on the venue's category picked.	Pass

Test Result for Draw Route from Source to Destination Module

Test Case ID	Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Status
DrawRoute_01	Verify draw route from source to destination function.	Test draw route function without entering anything.	1. Press "Search Route" button. 2. Press "Go" button without entering anything on "Source" and "Destination" text field.	-	A small red icon and "Required" is appeared on the end of each text field.	A small red icon and "Required" is appeared on the end of each text field.	Pass
DrawRoute_02	Verify draw route from source to destination function.	Test draw route function with entering a venue name on one of the text field only, either "Source" or	1. Press "Search Route" button. 2. Enter a venue name on "Source" or "Destination"	"Recap Room"	A small red icon and "Required" is appeared on the end of empty text field.	A small red icon and "Required" is appeared on the end of empty text field.	Pass

		"Destination" text field.	text field and the other left empty. 3. Press "Go" button.				
DrawRoute_03	Verify draw route from source to destination function.	Test draw route function with entering a venue name which is not present in database on "Source" and "Destination" text fields.	1. Press "Search Route" button. 2. Enter a venue which is not present in database on "Source" and "Destination" text field. 3. Press "Go" button.	"Lab 1", "Software Engineering Room"	A message is shown to tell user the venue name entered is not found.	A message is shown to tell user the venue name entered is not found.	Pass

DrawRoute_04	Verify draw route from source to destination function.	Test draw route function with entering same venue name which is not present in database on "Source" and "Destination" text fields.	1. Press "Search Route" button 2. Enter same venue name which is present in database on "Source" and "Destination" text field. 3. Press "Go" button.	"Virtual Reality Studio"	A small red icon and "Please enter different venue name" is appeared on the end of last focused text field.	A small red icon and "Please enter different venue name" is appeared on the end of last focused text field.	Pass
DrawRoute_05	Verify draw route from source to destination function.	Test draw route function with entering different venue name which is not present in	1. Press "Search Route" button. 2. Enter different venue names which is present in	"Programming Lab 1", "Gaming Lab"	Two pin points will be drawn on floor plan and a shortest route is drawn between the two pin points	Two pin points will be drawn on floor plan and a shortest route is drawn between the two pin points	Pass

		database on "Source" and "Destination" text fields.	database on "Source" and "Destination" text field. 3. Press "Go" button.		and estimated distance is calculated.	and estimated distance is calculated.	