**CHAPTER ONE**

**INTRODUCTION**

**1.1 Background of the Study**

The mobile has become a valuable part in recent years for the human beings. It has become necessary for humans to have a powerful device that will provide numerous facilities other than the simple facilities available on mobile phones.

A Geographic Information System (GIS) is a computer-based information system for storing, managing and analyzing, and invoking geographic reference data that has developed rapidly in the last five years (Wibowo, 2018), GIS is a computer system that captures, stores, validates and displays data about locations on the Earth's surface. GIS can display a wide range of data on a single map, including streets, buildings, and plants. This allows people to identify, evaluate, and comprehend patterns and correlations more quickly. which the aim of this project is to aid in simplifying the process of finding a location on campus through media mapping. This place finder application is expected to help and facilitate locating a place.

Android devices have become the first preference for many users and the demand for social applications is growing immensely. This research is a navigational assistance application which is for the main campus of Kaduna polytechnic. Android is used as a basis for this information system because Android has been known as one of the open-source systems where developers are free to develop various applications according to their own wishes. In addition, the reason why this geographic information system is built on android is to prioritize aspects of the high level of mobility that each android application has (Habibi, 2018).

Navigational assistance for students on the main campus has become an increasingly important issue in recent years. This is due to the fact that universities have become larger and more complex, making it difficult for new students to find their way around. Furthermore, the introduction of new technologies such as mobile apps and GPS tracking devices has further complicated the process of navigating the campus. As a result, I have been motivated to develop systems and strategies that can help students navigate the campus with ease. A higher institution such as Kaduna polytechnic has a lot of infrastructure inside the school, with different names and usage. As a new student in a new space, navigating to a different location on campus is quite challenging, asking other people around the area might seem helpful but not as accurate as when a geographic information system such as google map API is employed in achieving the task.

Since GPS is suitable for navigating outdoors, Users just have to choose or search for their desired destination from the interactive user interface and a route will be presented to them.

**1.2 Statement of the Problem**

As a new student in a new environment without a clear navigation plan is frustrating, not knowing where to go, or having the phobia of asking questions, even on asking a question most people would rather make jest of you by pointing you to the wrong direction all in the name of “in the past similar thing was done to me” thereby wasting one's valuable time, one issue is that the campus may be large and complex, making it difficult for the student to find their way around. In addition, the student may not be familiar with the campus, which could make it difficult for them to navigate to the different locations. Hence the need for this application. By using these applications, we can guide the new student or any student with difficulty in finding a location easily, therefore reducing the cost of parsing information and providing a platform that can be accessed anytime and from anywhere.

**1.3 Aim and Objectives of the Study**

The project is aimed at designing a working mobile-based application in Kaduna polytechnic main campus that will be used mainly by both newly admitted students and also returning students to help them locate places on campus.

**Objectives**

The objectives of this research work are as follows:

1. To design an android application that will guide students to their destinations on campus.
2. To develop a system that will store and retrieve location data on campus.
3. To evaluate how efficiently the system manages location records on campus to aid in eliminating time wastage in locating a place.

**1.4 Scope of the Study**

This research work is centered on providing navigational assistance for students on the main campus of Kaduna polytechnic, for effectiveness, it will provide assistance to a set of predefined locations on the main campus it will not cover providing navigation for students on other campuses.

**1.5 Limitations of the Study**

The scope of this study has been limited by a number of difficulties, including:

**Time** - The time allowed for research for this study was significantly constrained by the researcher's demanding academic obligations, The researcher only has a limited amount of time to conduct the study.

**Data collection**: In pursuit of high-quality information, assessing some google map endpoints for location service data is actually expensive, not to forget the difficulty of finding participants who are willing and able to take part in the study.

**Finance** - The requirement for a standard functioning personal computer unit to run and debug the application software slowed down the efficient and straightforward completion of the operation.

**1.6 Significance of the Study**

This study is to provide students with a navigational assistance system that can be used on campus to help them find their way around. This will help students become more familiar with the campus layout and reduce the stress of getting lost or not being able to find a particular building or area. Additionally, this system could provide an easy way for students to access important information about the campus, such as opening and closing times, available services, and directions to specific areas. One potential significance is that it could provide new insights into the challenges and obstacles faced by students when navigating a campus. This could help to identify areas where improvements are needed and could inform the development of new strategies and solutions for addressing these challenges. The navigational assistance system could help improve student safety and satisfaction with the campus by providing them with the information and tools they need to find their way around.

**1.7 Project Organization**

Project organization refers to how the chapters in a project are structured and basically what is contained in them. The project is divided into five chapters. The outlines are presented below:

**Chapter One: Introduction**

Chapter one introduces this project work, the background of the study, the statement of the problem, the aim and objectives, the scope of the study, limitations of the study, the significance of the study, project organization, and the definition of terms.

**Chapter Two: Literature review**

This chapter focuses on the literature review, and the contributions of other scholars on the subject matter being discussed.

**Chapter Three: Methodology and Design**

This chapter is concerned with the presentation of the end results of system analysis and design. It presents the research methodology used in the development of the system to facilitate an understanding and effective future implementation of the system.

**Chapter Four: System Implementation Evaluation**

This chapter describes the system implementation and documentation, analysis of modules, and system requirements for implementation.

**Chapter Five: Summary, Conclusion, and** **Recommendation**

The chapter provides a summary of major findings, conclusions, and recommendations based on the study conducted.

**1.8 Definition of Terms**

1. **Google Map**: A web-based mapping service called Google Maps was created by Google. It enables route planning for travel by foot, automobile, bicycle, or public transit as well as satellite images, street maps, 360-degree panoramic views of streets, and current traffic conditions.
2. **Database**: A database is a collection of arranged data that is electronically stored and accessible. A database often contains significant volumes of structured data that can be searched, sorted, and retrieved with ease by users.
3. **Django**: Django is a free and open-source web framework written in Python. It is designed to make the development of complex, database-driven websites and applications easier and more efficient.
4. **GIS**: Geographic information system is referred to as GIS. It is a system for gathering, conserving, processing, and managing geographic or geographical data. Users of GIS may produce and examine maps as well as analyze and comprehend geographical data.
5. **REST:** REST, or Representational State Transfer, is an architectural style for building web services and APIs. It is a set of principles and guidelines that define how a web service or API should be structured and how it should operate.
6. **API**: An API, or application programming interface, is a set of rules and protocols that allows different software applications to communicate with each other. An API defines the way in which different software components should interact and exchange data, allowing them to work together to accomplish a specific task or function.
7. **Flutter**: Flutter is a free and open-source mobile application development framework created by Google. It is used to build natively compiled applications for mobile, web, and desktop from a single codebase.
8. **UI**: UI stands for the user interface. It is the part of an application or website that the user interacts with directly. The user interface is the means by which the user and the application or website communicate with each other.

**CHAPTER TWO**

**Literature Review**

**2.1 Introduction**

This chapter's goal is to demonstrate how the topic under study relates to previous research, current practice, or other areas of knowledge by citing pertinent works by other scholars that have dealt with a similar issue. In addition, this chapter will provide a synthesis of the existing research on the topic, highlighting areas of agreement, disagreement, and gaps in the literature, to establish the importance of the project topic in the field and to identify areas for further research.

**2.2 Literature Review**

Low and Lukman (2021) recently made research on Campus Location Finder Using Mobile Application. The university campus is made up of several buildings and rooms, each with its unique name and function. Aside from that, moving from one building to another will take some time because the campus is not tiny like a primary or secondary school. A standard map does not appear to be useful because it does not identify the user's position. It may take some time to determine the user's present location. Finding a location inside a building is difficult since the halls are nearly identical. Asking other individuals in the area for directions may provide results, but it will be difficult when the route to the destination contains several turning points. Another issue was remembering all of the locations correctly.

More importantly, the application for indoor and outdoor navigation has been built using the Thunkable website. Outdoor navigation is linked to Google Maps, while indoor navigation uses QR codes for location and routing pictures. Outdoor navigation may take users to the main faculties of UTHM, whereas inside navigation is only available for the G1 building in UTHM. Thunkable website, Google Drive website, and Adobe Photoshop CC2015 software were all utilized.

However, the application begins with the main screen, which allows the user to select between indoor and outdoor travel. If the user selects outdoor, the next step is to click the placement button. If the user knows the name of the building, choose it and push the button on the page. If the user simply knows the name of the room or lecturer, enter it in the search box and select the name from the list. The user must then zoom out on the Google map until the blue pin and user position are visible. Then, for navigation, hit the blue pin.

In conclusion, using the Thunkable website and other tools, this indoor/outdoor navigation application was completed. Recommendations to increase the application's usefulness in the future include making indoor navigation available for all buildings in the use case of the study. replacement of   QR code indoor navigation with real-time navigation using several navigation methods such as Wifi, etc. Furthermore, because the Thunkable website has many limitations, the application may be improved by using Android Studio to develop the application.

Wei Chuan et al. (2017) recent study on Batu Pahat Car Workshops Finder, proposed an application in response to the frequent breakdowns that occur in automobiles; this problem is exacerbated when the driver is in an unfamiliar location. As a result, Batu Pahat Car Workshops Finder is an application that helps solve the aforementioned problems. The program enables users to fix their automobile while traveling if it breaks down and there is no local automotive service or repair center. This application will display the available vehicle workshops around the user's current location, along with their contact information.

Moreso, Android Studio is the software program used to develop Batu Pahat Car Workshops Finder. Java programming language is utilized in Android Studio, while Firebase is used for database storage.

In addition, ten participants participated in user acceptability testing. The testers are UTHM students with valid driver's licenses. Questionnaires were distributed to these students about the interface design and level of user-friendliness of this system. Eight users agreed that the program is moderate and user pleasant. While two people complain that the program is difficult to use.

In conclusion, the program is utilized to assist users in locating the nearest vehicle workshops in Batu Pahat to resolve a car breakdown problem. According to the results of the testing, the majority of the application's functions are working properly, but there is still room for improvement in the development of GPS navigation capability.

Nadeeshani. (2020) perform a study recently on “On-Road Vehicle Breakdown Assistance”. This article is primarily meant to minimize the amount of time drivers spend waiting. Most drivers endure difficulties on the road due to car breakdowns. The breakdown of a vehicle wastes the user's precious time, that is the worst experience they can have. The motorist must look for a mechanic and a spare parts store near their position. If the motorist is unable to locate a good mechanic, he or she must seek assistance from someone, which may or may not be technical assistance.

Furthermore, while developing an Android application, requirements are gathered as primary and secondary data. As a first phase in the project, determine if the suggested system is needed by society and whether it is available. As part of the market study for the "help ME" Android application, the researcher has established a Google survey form to collect primary data. As a result, the researcher was able to identify the need for Vehicle Breakdown Assistance in Sri Lanka. Secondary data collection refers to information that was stated or done by another person. To get secondary data, the researcher utilized the Internet to look up similar systems. Google Scholar, IEEE, websites, and online journals are all available. Look for comparable systems and compare them. When developing the Android application, the researcher used Android Studio as a tool. Android Studio handles all UI development and implementation. Firebase is a technology that is used for real-time databases. Image cropper dexter, glide, and circleImageView libraries were used as the main library for the android app. To crop the images the image cropper API was utilized and it is useful for editing the user profile.

Finally, when a breakdown occurs, the user can promptly repair their car. They will not waste their time on the journey. This allows them to save crucial time. This makes the person feel more at ease. There is a chat platform where users and mechanics may communicate. There, users may ask questions about automobile breakdowns. That can be answered by the mechanic. This will aid in improving the user's technological understanding of the car. The payment option for the project might be added in the future development of the project. This allows users to fix their vehicles at an affordable cost.

Adeosun and Melike (2021) made research on “Location Finder Mobile Application using Android and Google Spreadsheets” In this research, the researchers offer Location Finder, a mobile Web application that allows users to explore their surroundings by exploiting contexts that are significant to them. Someone who finds himself or herself in a city or town with which he or she is unfamiliar, such as North Cyprus, may decide to go to a restaurant and have no idea where to find one. He/she may decide to open the smartphone app and search for queries like "Restaurant".

Moreso, the Location Finder mobile application was created in Java with Android Studio and an online Google Spreadsheet. To facilitate understanding, the project has been kept as basic as feasible. It includes a few JAVA and XML files. The program was built on Google standard Google Maps to ensure maximum accuracy and portability, and it includes a dynamic backend based on Online Google Spreadsheet, which allows the administrator to update and manage backend data without the need for an external server or database.

In conclusion, the application works semantically because various places can be identified automatically. It is a complete native Android application for locating various destinations such as restaurants, hotels, stores, bus stations, ATMs, Universities, hospitals, petrol stations, and many more places. It offers innovative mapping technologies that display various points of interest for application users. Users can also change the search options (categories, searching area radius) and store a location for future reference. Experiments on Location Finder's performance also reveal that the proposed solution is substantially quicker and better than other popular mobile applications in the sector.

Shwethashree et al. (2019) recent study on Android-based Library Book Availability and Location Finder. In extremely large libraries having massive collections, identifying a certain book and understanding its availability of the book is quite difficult. In such a case, there must be a simple way to locate the specified book by just inputting its name. As a result, we suggest an Android application that assists the client in locating the book in a matter of seconds and even knowing the availability of the book in the library and, if not, the date when it will be accessible if it is issued to someone.

Furthermore, the standard book availability and location-finding mechanism necessitate searching for a longer length of time in all of the library's racks. In this prevalent technique, users encounter the challenge of time management. Even after a lengthy search, the chances of finding the book are slim. This inspired us to create a new system that would alleviate the hassle.

**2.3 Summary of Related Literature Reviews**

|  |  |  |
| --- | --- | --- |
| **Author & Year** | **Title & Description** | **Merit and Demerits** |
| Low and Lukman (2021). | In Campus Location Finder Using Mobile Application.  This project, an application for indoor and outdoor navigation at Universiti Tun Hussein Onn Malaysia created (UTHM). It is used to assist visitors and new incoming students by routing them from their current position to their destination utilizing the "U Finder" smartphone application. | The researcher achieved its stated objective as the indoor and outdoor navigation was fully effective and functional.  Indoor navigation lacks real-time navigation as it was built with QrCode indoor navigation. |
| Nadeeshani. (2020). | On-Road Vehicle Breakdown Assistance.  This article is primarily meant to minimize the amount of time drivers spend waiting. Most drivers endure difficulties on the road due to car breakdowns | There is a chat platform where users and mechanics may communicate.  The application lacks integration of a payment gateway. |
| Adeosun and Melike (2021). | Location Finder Mobile Application using Android and Google Spreadsheets  In this research, the researchers offer Location Finder, a mobile Web application that allows users to explore their surroundings by exploiting contexts that are significant to them | A new location is automatically identified with a fast response time.  Restricted to android mobile users only, and it has a high maintenance cost |
| Wei Chuan et al. (2017) | Batu Pahat Car Workshops Finder.  The proposed concept is to provide a real-time platform for detecting and tracking neighboring car workshops using GPS-enabled mobile phones. It is referred to as Batu Pahat Car | The developed application can assist users in locating the nearest vehicle workshops.  The GPS navigation needs improvement |
| Shwethashree et al. (2019). | Android-based Library Book Availability and Location Finder  To search for a certain book, this system employs an Android application that produces a user interactive interface with the library's database. | It even reveals the availability of the book before searching, which saves a significant amount of time.  The user interface is too colourful. |

**2.4 Analysis of the Current System**

The current system of providing navigational assistance to students on campus is the manual method which is through the use of wayfinding signs which are not interactive. Campus layouts and buildings can change over time, and it can be difficult to keep wayfinding signs current. This can lead to confusion for students who are relying on these resources to navigate the campus.

The current system makes it difficult for people or students unfamiliar with the campus to locate places, even when they decide to ask questions from any individual nearby, they might point them in the wrong direction or make jest of them

The wayfinding sign methods can be effective in helping students find their way around campus, but they have some limitations compared to digital navigational aid systems. Overall, while the manual way of providing navigational aid to students can be helpful in some cases, it may not be as effective or convenient as digital navigational aid systems, which can offer more up-to-date and interactive information to students.

**2.4.1 Problem Inherent in** **the Current System**

There are several problems inherent in the current system of providing navigational aid to students on campus which include:

1. **Time-consuming**: Providing students with navigational aid manually can be time-consuming, especially if the campus is large or if there are a large number of students in need of assistance.
2. **Limited availability**: There may not always be someone available to provide navigational aid.
3. **Inconsistency**: Different people may provide students with different directions or information, leading to confusion and inconsistency.
4. **Inaccuracy**: People can make mistakes or provide incorrect information when providing navigational aid manually.

**2.5.1 Advantages of the New Proposed System**

1. **Real-time information**: The proposed system can provide students with real-time information about their location and the fastest route to their destination, ensuring that they always have up-to-date and accurate information.
2. **Convenience**: The proposed system is available 24/7, so students can access navigational assistance at any time of day or night. This is particularly useful for students who are new to the campus or who are unfamiliar with the layout.
3. **Efficient**: The proposed system can process requests for navigational assistance quickly and efficiently, without the need for students to wait for assistance from a person.

**CHAPTER THREE**

**METHODOLOGY AND DESIGN**

**3.1 Introduction**

A methodology is a process of rigorous study or inquiry, particularly to unearth new facts or information; hence, research methodology should be good enough to enable the achievement of the specified objectives which are achievable using specific components, such as data collection and design procedures and system modeling (use case, activity, and class diagrams). This chapter contains the input/output specifications, and system requirements for the development of navigational assistance for students in the main campus of Kaduna polytechnic.

**3.2 Method of Data Collection**

Before developing any system, collecting data and facts about the existing system is critical to understand what is going on. This research was carried out using three methods.

1. Observation of the Work Environment
2. Interview
3. Documentation

**3.2.1 Observation of the Work Environment**

This method was employed to acquire information and data for this study by monitoring how the manual system worked. The most evident flaws in the existing system were discovered via detailed inspection. Using the observational approach, the context in which the observation is made can be modified in a variety of ways.

**3.2.2 Interview**

The main objective of using interviews as a method of data collection is to obtain information thoroughly and rigorously. Based on the questions the researcher provided, the researcher met meet with some students and acquired reliable information.

**3.2.3 Documentation**

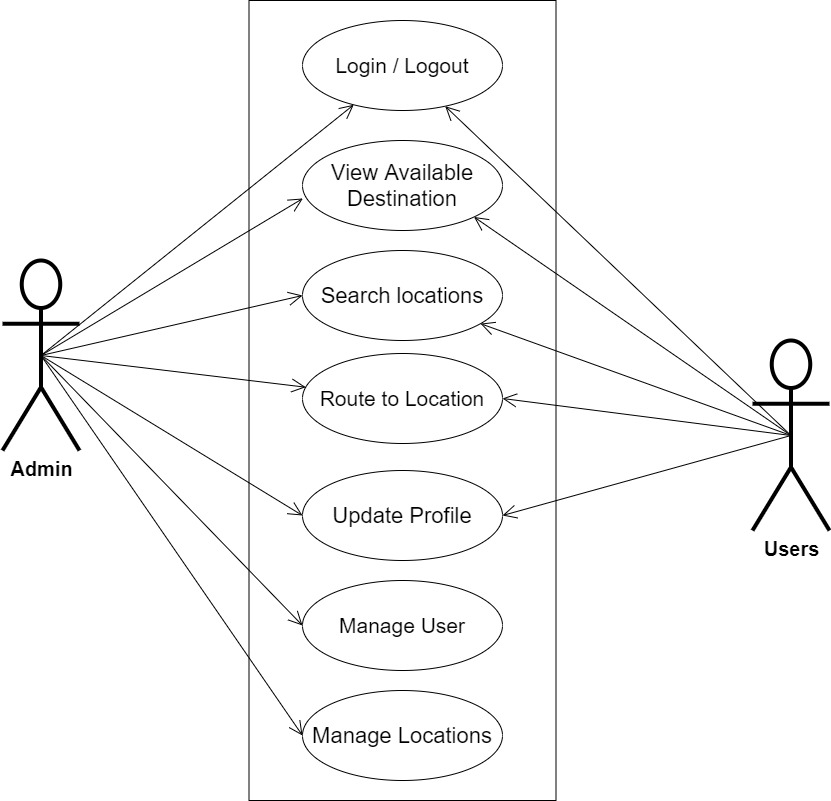
Documentation is a type of secondary data collection. This method makes use of journals, manuals, past work, publications, and other sources. This method of data collection is used because it allows for comparison with past studies. This includes the internet, which is a data collection tool. The internet was used to find information on difficult or ambiguous issues.

**3.3 System Modeling**

A system model is a conceptual model of a system that explains and represents it. A system is any interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the creation of this contemporary system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

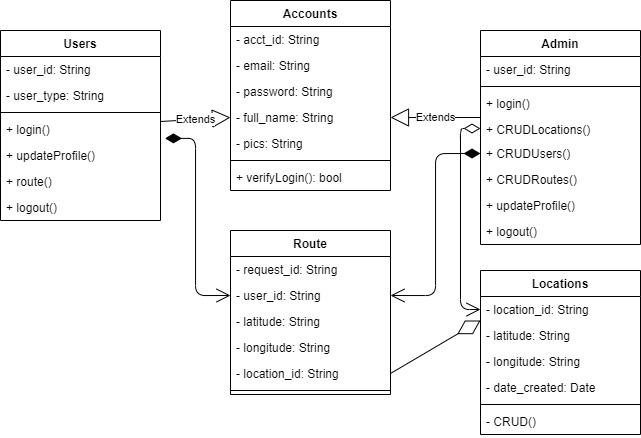
Use cases are collections of system-to-user interactions. Use case diagrams are used to graphically characterize the functionality of a system in terms of its actors, goals (represented as use cases), and dependencies between those use cases.



**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagrams**

The Unified Modeling Language (UML) class diagram is an implementation of an independent perspective of how the system interface would look, with each class having its own set of attributes and demonstrating how they interact with one another. Class diagrams employ the Unified Modeling Language standards to visually portray a given system's static structure and composition (UML).



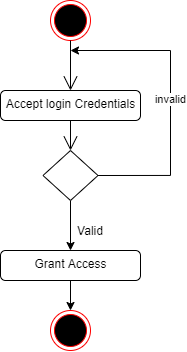
**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

An activity diagram, like a flowchart or a data flow diagram, visually illustrates a series of events or the flow of control in a system, but it acts more like an enhanced version of both.

**Login**

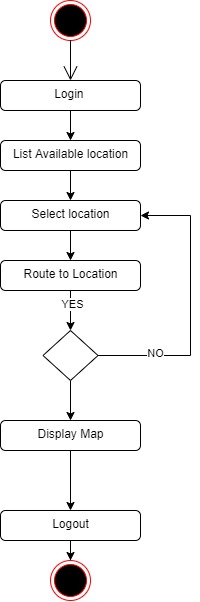
The process for gaining access to the system is depicted in the diagram below; the email address and password must be accurate to gain access.



**Fig 3.3 Login Activity Diagram**

**Routing**

The process for routing to a destination is depicted below, to route one has to check for the availability of the location.



**Fig 3.4 Routing Activity Diagram**

**3.4 Database Design**

The logical explanation of how data is kept in the computer's memory is called input specification. The freedom experienced in using the system, as well as the convenience of retrieving and reading the data and assuring applicability across the internet, make SQL standards essential for ensuring that structured data is uniform and independent of applications. Some of the input specifications employed in this project work are presented below.

1. Users Table: contains basic information about all system users.
2. Location Table: contains every system-saved location information.

**Table 3.1 Users Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| user\_id | Varchar | No | PK | 32 | Unique string for identifying users |
| email | Varchar | No |  | 100 | User email address |
| password | Varchar | No |  | 128 | User Password |
| full\_name | Varchar | No |  | 60 | User full name |
| user\_type | Varchar | Yes |  | 20 | User type (student/others) |
| pics | Varchar | Yes |  | 100 | User profile picture |

**Table 3.2 Location Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| location\_id | Varchar | No | PK | 32 | Unique string for identifying locations |
| location\_title | Varchar | No |  | 60 | Title of the Location |
| latitude | Varchar | No |  | 100 | Saved location latitude |
| longitude | Varchar | No |  | 128 | Saved location longitude |
| date\_created | Varchar | No |  | 60 | Date the location was saved |

**3.5 Output Design**

This declares and displays the outcome of the given input. This automated system's output is dependent on its input. The output specification is listed below.

**Table 3.3 Users** **output design table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User\_id** | **Email** | **Password** | **Full\_name** | **User\_type** | **pics** |
| XXXX | XXXX | XXXX | XXXX | XXXX | 833 No Image Available Stock Photos and Images - 123RF |
| XXXX | XXXX | XXXX | XXXX | XXXX | 833 No Image Available Stock Photos and Images - 123RF |
| XXXX | XXXX | XXXX | XXXX | XXXX | 833 No Image Available Stock Photos and Images - 123RF |
| XXXX | XXXX | XXXX | XXXX | XXXX | 833 No Image Available Stock Photos and Images - 123RF |

**Table 3.4 Location** **output design table**

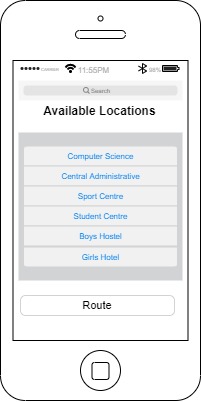
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **location\_id** | **Location\_title** | **Latitude** | **Longitude** | **Date\_created** |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |

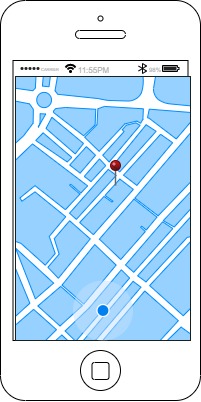
**3.6 Input & User Interface Design**

This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be fully secured, thus authentication will be required to see various levels of the information. To help with the designs, a mid-fidelity wireframing program called Draw.io is employed.



**Fig 3.5 User Login Screen**



**Fig 3.6 Available Locations Screen**

**Fig 3.6.2 Routing Screen**

**3.7 System Requirement**

Every piece of software generated has predefined system requirements that it must fulfill in order to function properly. The system requirements, on the other hand, are the bare minimum of hardware and software required for the system's intended operation.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 8 GB of RAM (Random Access Memory) installed.
2. Minimum of intel core i3 processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 10 OS (Operating System).
2. Flutter Installation.
3. Vs. Code / Android studio installation.
4. Emulator installation.

**3.8 Choice of Programming Language**

This research work will be a mobile-based application where flutter will be employed in designing the front-end; SQLite will be used as the database, Django will be used as the backend, Django REST Framework as the web API, Google Maps API is used to interpret the set-out coordinates. The combination of the above will help build a very robust platform that will be useful, fast, and handy.

**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION EVALUATION**

**4.1 Introduction**

This section describes in detail how the new system will be implemented in order to assure its efficacy. It illustrates instances of functional (new) systems as well as how the system will be implemented.

* 1. **System Testing and Evaluation**

The developed system should be tested for a variety of reasons. For example, only via testing will we be able to detect and address any problems in the new system. Unit and integration testing were used in this project to confirm the design's efficacy and efficiency, as well as to ensure the new system satisfies its functional requirements and is error-free.

**Unit Testing**

specific units or single components of the system are examined individually in this part to confirm that specific phases function properly and without problems.

**Integration Testing**

The program was tested via integration testing, in which all of the components were integrated and worked as one. The connection between the different components was examined to ensure that they are correctly integrated and that the units can function as a unit.

**4.3 System Installation**

In order to use the proposed application on any computer system, the following steps need to be taken:

1. Make sure, android studio, JDK, and Android emulator are installed on the system.
2. Copy your project folder to any location of your choice.
3. Open the project folder in Visual Studio Code
4. In the terminal run “flutter pub get” to get all the dependencies in the pubspec.yaml file
5. Select the Android emulator as the device to be used.
6. Locate the main.dart file and run the file in debug mode.

**4.4 Security Measures**

Since the scope of the application is public, literally all the information is made available to any user (students and admin), but some functionalities are restricted to the admin, functionalities that have to do with creating the student accounts, creating the candidates, managing the voting periods etc are restricted from the general student. The restriction is carried out by using passwords when the application is accessed.

**4.5 Sample Outputs**

These describe and give the pictorial representation of the program or software; it shows and gives clear understanding of the design, and displays all the interfaces.

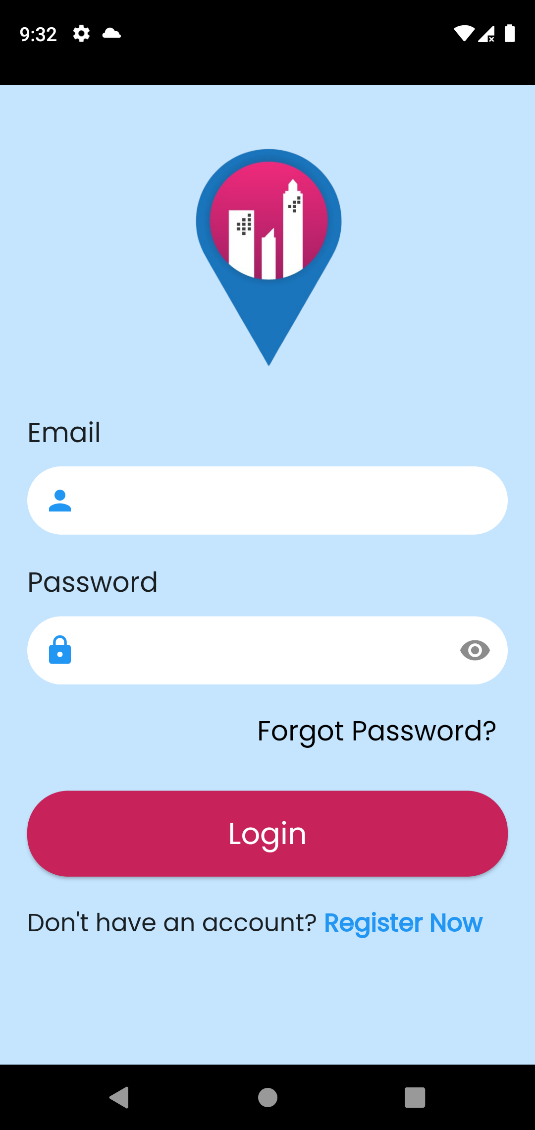


Fig 4.5.1: Splash Screen Fig 4.5.2: Login Screen

**Fig 4.5.1 Splash Screen**: This is the first screen displayed to every user that wishes to make use of the application.

**Fig 4.5.2 Login Screen**: The screen grants users access to the application only if the correct credentials are provided.

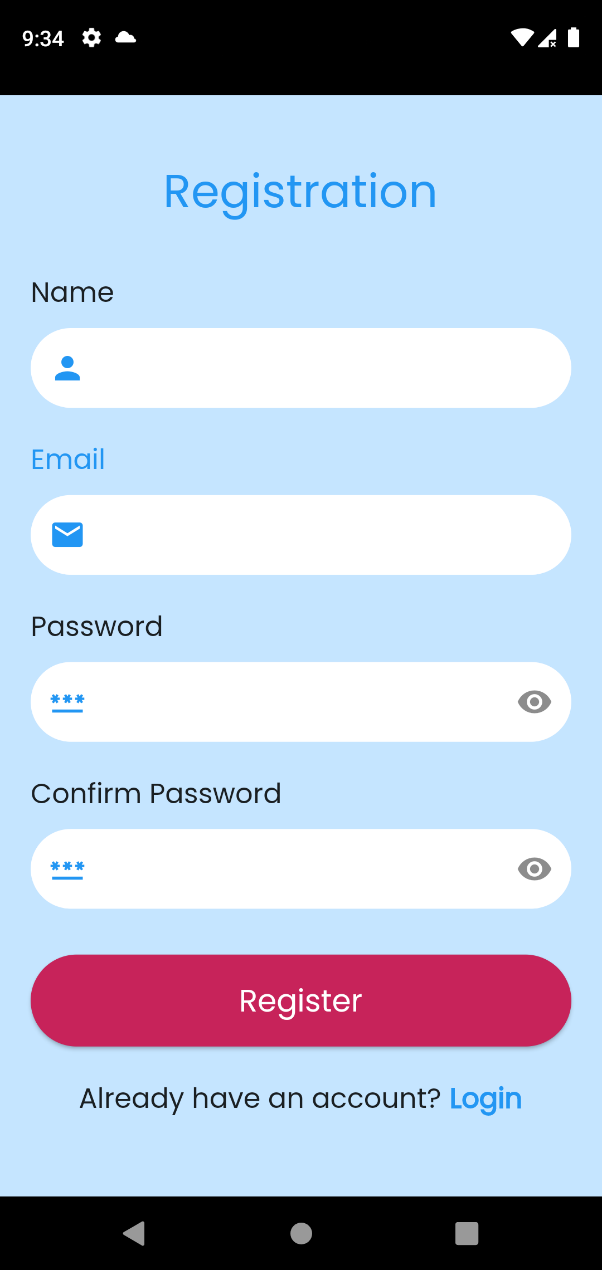
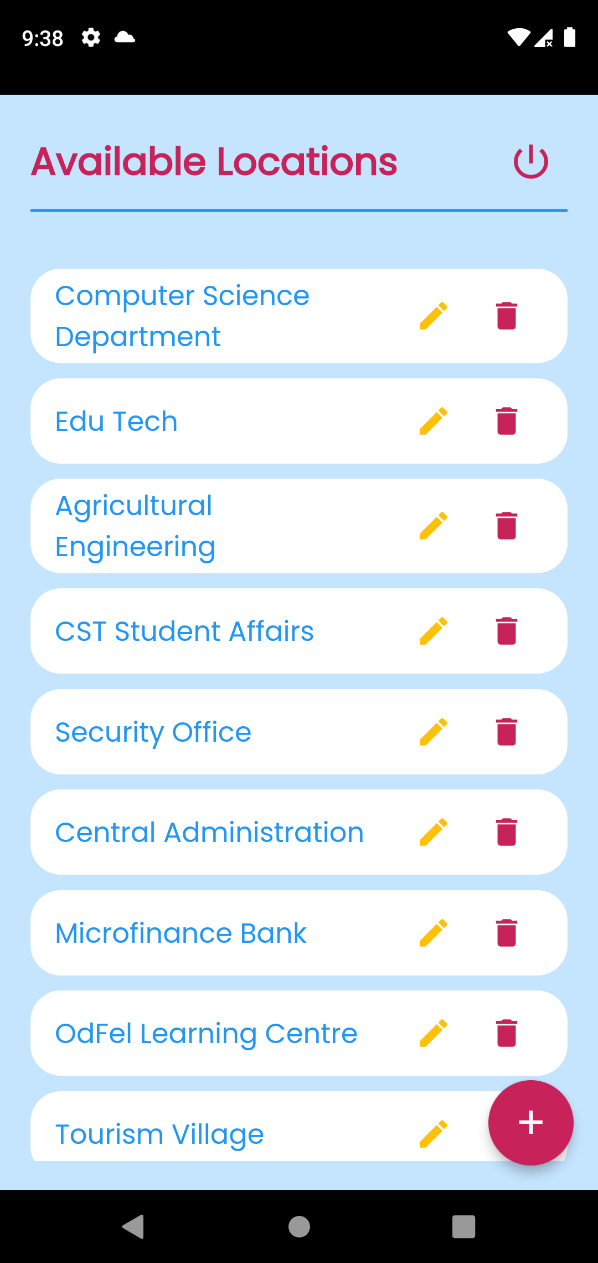


Fig 4.5.3: Signup Screen Fig 4.5.4: Available Location Screen

**Fig 4.5.3 Signup Screen**: The application ensures that only registered users can have access to the system, the screen enables the creation of accounts for new users.

**Fig 4.5.4 Available Location Screen**: This is an admin page where the option to create, delete and edit location is made possible.

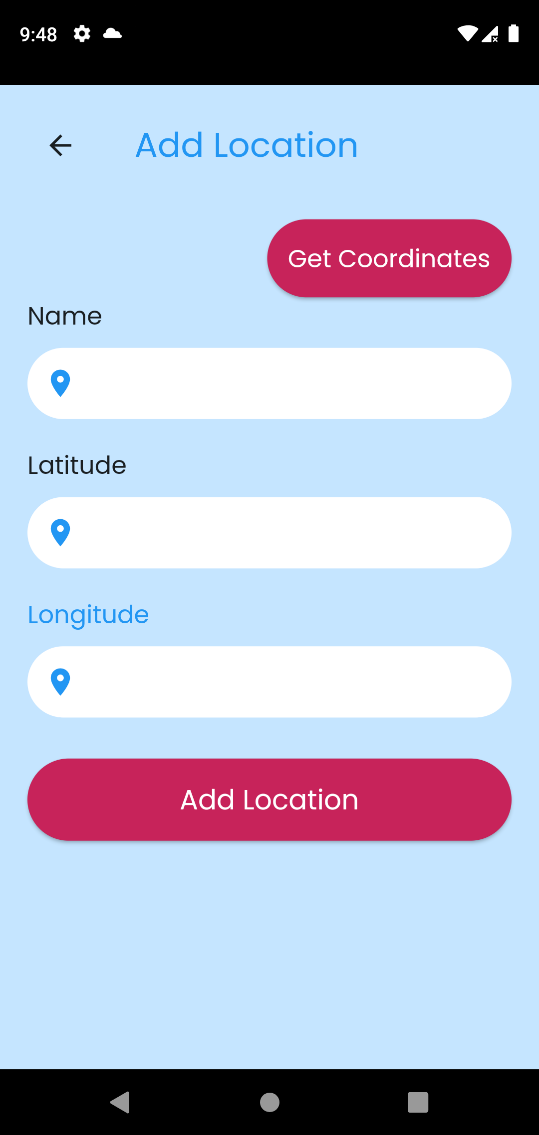
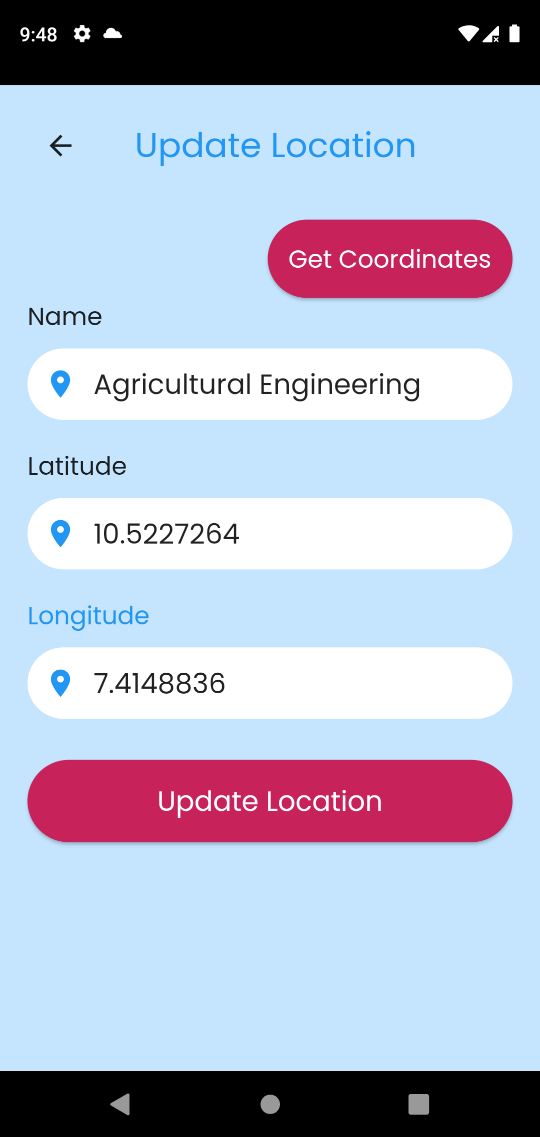


Fig 4.5.5: Add Location Screen Fig 4.5.6: Update Location Screen

**Fig 4.5.5 Add Location Screen**: The admin makes use of this screen to add location to the application.

**Fig 4.5.6 Update Location Screen**: The application admin can update an already exisiting location cordinates or title.

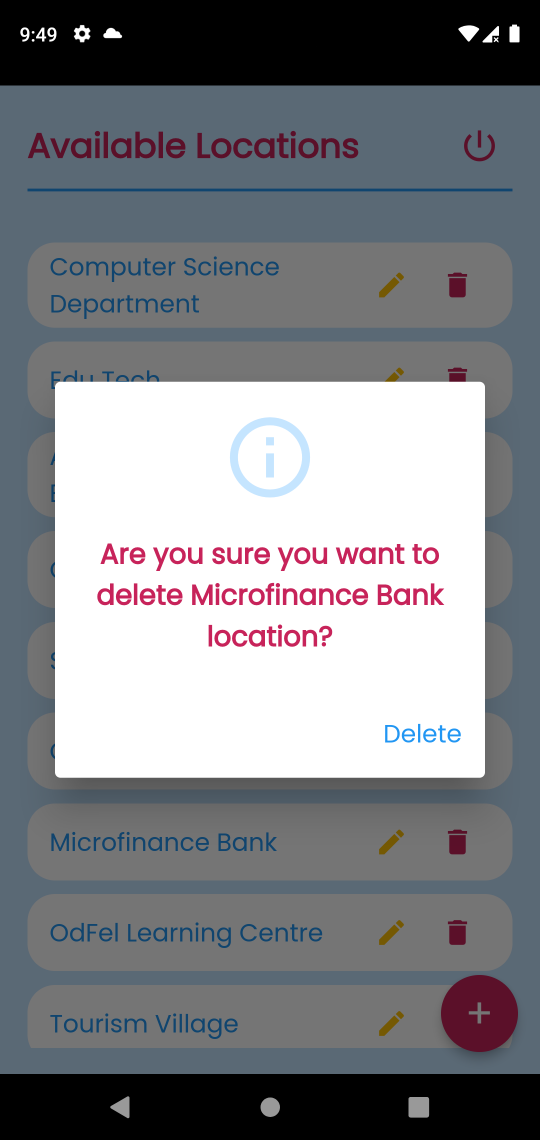
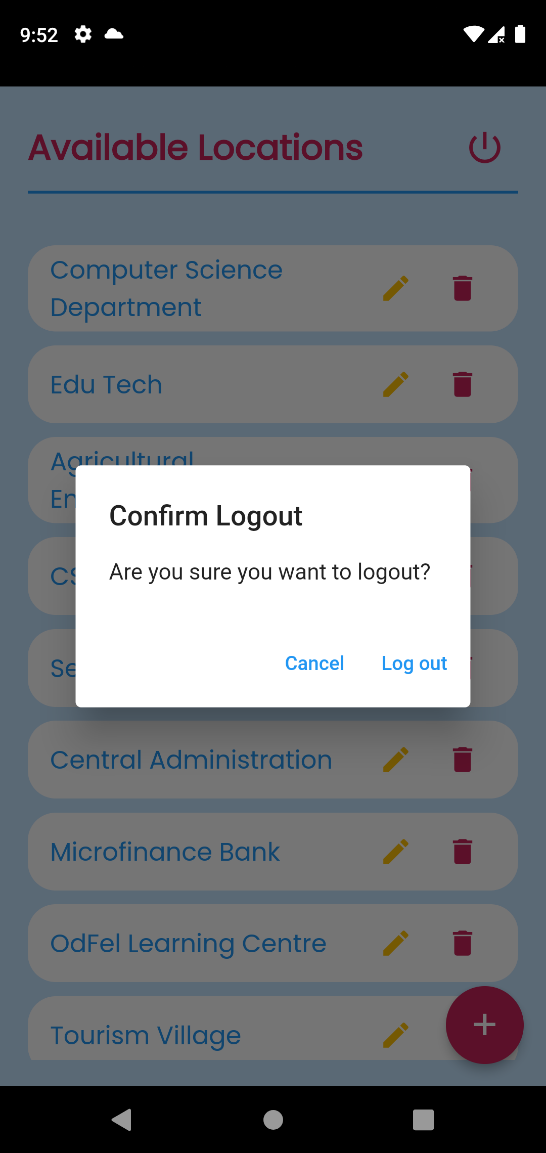


Fig 4.5.7: Delete Location Screen Fig 4.5.8: Logout Option

**Fig 4.5.7 Delete Location Screen**: The admin can delete any existing location.

**Fig 4.5.5 Logout Option**: This option logs out the user from the application, the user has to reauthenticate again in order to gain access to the application.

**User Screens**

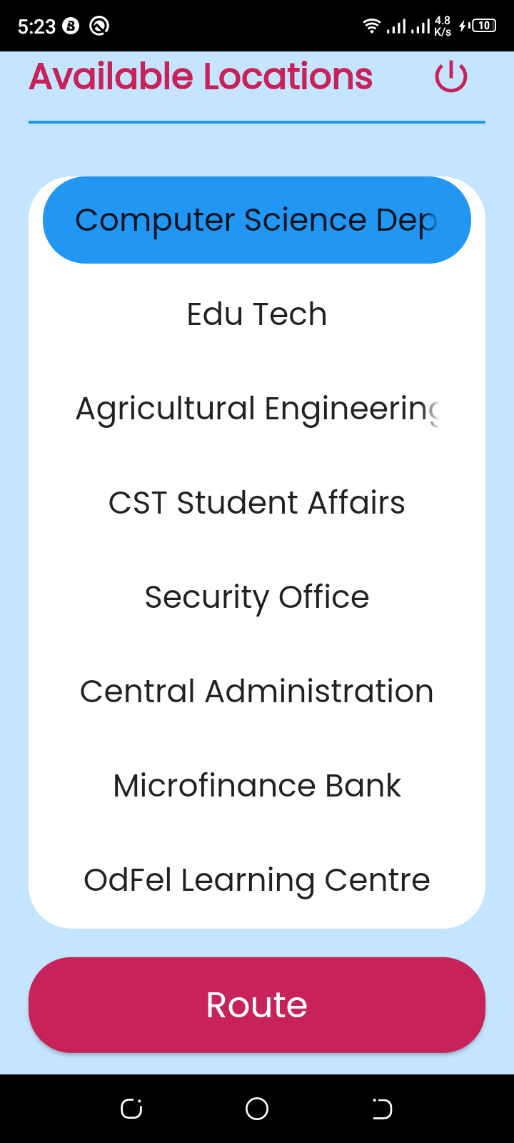
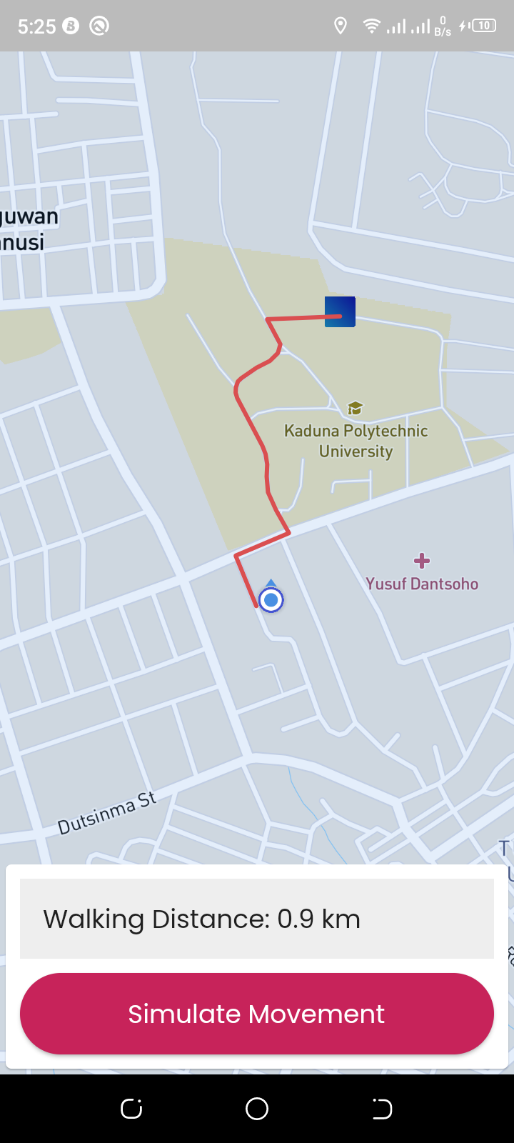


Fig 4.5.9: All Available Locations Fig 4.5.10: Map Marker

**Fig 4.5.9 All Available Locations**: The user can view all the locations added by the admin and can route to any of the locations.

**Fig 4.5.10 Map Marker**: MapBox provides the interface to show the current user position and destination on a map using markers and it changes as the user moves or attempts to move to their destination.

**CHAPTER FIVE**

**SUMMARY CONCLUSION AND RECOMMENDATION**

**5.1 Summary**

This research project focuses on developing a navigational assistance application for the main campus of Kaduna Polytechnic. The application aims to simplify the process of finding locations on campus through Mapbox API, utilizing the Android platform for its high mobility. By providing an intuitive and user-friendly interface, the application helps students navigate the campus efficiently, reducing the stress of getting lost or struggling to find specific areas. This study acknowledges the challenges posed by complex campus layouts and limited access to accurate information. However, it aims to overcome these limitations by providing a reliable and convenient solution for students to easily locate places on campus, enhancing their overall campus experience.

**5.2 Conclusion**

In conclusion, this research study has effectively addressed the requirement for a navigational application for Kaduna Polytechnic's main campus. The application uses the Android platform with media mapping technologies to give a user-friendly interface for students to quickly discover locations on campus. The concept recognizes the problems students have when navigating the complicated campus and strives to relieve these obstacles by providing a dependable and effective alternative. The created application provides students with a quick method to find their way about campus, minimizing the stress of being lost. Further testing and adjustments are recommended in the future to assure the application's usefulness and usability.

**5.2 Recommendation**

Based on the findings and conclusions of this research project, the following recommendations are proposed:

1. Further Testing and Evaluation: It is recommended to conduct thorough testing and evaluation of the developed navigational assistance application. This includes gathering feedback from users, identifying any potential issues or areas for improvement, and making necessary adjustments to enhance the functionality and user experience.
2. Integration of Real-Time Updates: To provide up-to-date information to users, the application should be integrated with real-time data sources. This could include incorporating live updates on building locations, route changes, and any other relevant campus information. Real-time updates would ensure that students have the most accurate and current information at their disposal.
3. Regular Updates and Maintenance: To ensure the continued effectiveness of the application, regular updates and maintenance should be carried out. This includes addressing any reported bugs, incorporating new campus developments or changes, and keeping the application compatible with the latest mobile device platforms and technologies.

By implementing these recommendations, the navigational assistance application can provide an invaluable resource for students, enhancing their campus experience and facilitating seamless navigation on the main campus of Kaduna Polytechnic.

**REFERENCES**

Balba, N. P., Rosas, N. M. F., Drilon, P. J. Z., Fallaria, S. J. F., & Fallaria, J. F. (2019).

“ISAKAY: Android-based routing system”. *International Journal of Innovative Technology and Exploring Engineering, 8(*10), 1265–1269. https://doi.org/10.35940/ijitee.I7535.0881019

Bari, Md & Efroymson, Debra. (2022). “Efficient Use of Routing of Door-

to-Door Mobility: Suggestions for Improvements in Dhaka.”

C. Habibi, S. Efensi, And Ramli, “Sistem Informasi Geografis Lokasi Restoran Di Kota Medan

Berbasis Android,” 2018.

Duraisamy, S., & Abuhuraira, J. U. (2018). “Android Mobile Application for Online Bus

Booking System”. *International Journal of Information System and Engineering, 6*(July), 34–56.

Fai, L. W., & Audah, L. (2017). In campus location finder using mobile application services. In AIP Conference Proceedings (Vol. 1883). American Institute of Physics Inc. https://doi.org/10.1063/1.5002040

K. M. Wibowo, I. Kanedi, And J. Jumadi, “Sistem Informasi Berbasisc Web Dengan Php Dan Mysql,” J. Media Infotama, Vol. 11, No. 1, Pp.51–60, 2018

Nevon Projects. (2021, August 13). *Web-Based Place Finder Using Django Python*. https://nevonprojects.com/web-based-place-finder-using-django-and-geodjango/

Nadeeshani, I.H. (2020). On-Road vehicle Breakdown Assistance. 10.13140/RG.2.2.11833.39521.

Olufemi, A. N., & Sah, M. (2022). Location Finder Mobile Application Using Android and Google SpreadSheets. In Smart Innovation, Systems and Technologies (Vol. 237, pp. 723–736). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-981-16-3637-0\_51

Okpara R., Otugeme E. and Osuagwu O. (2018). Development of a Mobile Android Routing App for Tertiary Institutions in Nigeria. *Routing Android App.*

https://www.researchgate.net/publication/344256892\_development\_of\_a\_mobile\_android\_routing\_app\_for\_tertiary\_institutions\_in\_nigeria

Sherine, A., Peter, G., Stonier, A. A., Leh Ping, D. W., Praghash, K., & Ganji, V. (2022). Development of an Efficient and Secured E-Roting Mobile Application Using Android. *Mobile Information Systems*, *2022*. https://doi.org/10.1155/2022/8705841

Wei Chuan, L., Kasim, S., Choon Sen, S., Hassan, R., & Mohd. Ali, N. (2017). Batu Pahat Car Workshops Finder. *International Journal of Innovative Computing, 7*(2). https://doi.org/10.11113/ijic.v7n2.146

**APPENDIX**

**Hompage**

import 'package:flutter/material.dart';

import 'package:mapbox\_gl/mapbox\_gl.dart';

import 'package:place\_finder/main.dart';

import 'package:place\_finder/services/remote\_service.dart';

import 'package:place\_finder/utils/constants.dart';

import 'package:place\_finder/utils/defaultButton.dart';

import 'package:place\_finder/utils/defaultText.dart';

import 'package:place\_finder/utils/defaultTextFormField.dart';

import 'package:place\_finder/utils/mapbox\_handler.dart';

class HomePage extends StatefulWidget {

  const HomePage({super.key});

  @override

  State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

  int \_selectedIndex = -1;

  late String destLat;

  late String destLng;

  \_getRoute(double destinationLat, double destinationLng) async {

    await sharedPreferences.setDouble("destinationLat", destinationLat);

    await sharedPreferences.setDouble("destinationLng", destinationLng);

    LatLng sourceLatLng = Constants.getSourceDestLatLng('source');

    LatLng destinationLatLng = Constants.getSourceDestLatLng('destination');

    Map? modifiedResponse = await getDirectionsAPIResponse(

        sourceLatLng, destinationLatLng, context);

    Navigator.pushNamed(context, '/map',

        arguments: {'modifiedResponse': modifiedResponse});

  }

  @override

  Widget build(BuildContext context) {

    Size size = MediaQuery.of(context).size;

    return SafeArea(

      child: Scaffold(

        backgroundColor: Constants.backgroundColor,

        body: SingleChildScrollView(

          child: Padding(

            padding:

                const EdgeInsets.symmetric(horizontal: 20.0, vertical: 20.0),

            child: SingleChildScrollView(

              child: Column(

                children: [

                  Row(

                    mainAxisAlignment: MainAxisAlignment.center,

                    children: [

                      DefaultText(

                        text: "Available Locations",

                        size: 25.0,

                        weight: FontWeight.bold,

                        color: Constants.altColor,

                      ),

                      const Spacer(),

                      IconButton(

                          onPressed: () {

                            showDialog(

                              context: context,

                              builder: (context) {

                                return AlertDialog(

                                  title: const Text('Confirm Logout'),

                                  content: const Text(

                                      'Are you sure you want to logout? '),

                                  actions: [

                                    TextButton(

                                      onPressed: () {

                                        Navigator.pop(context);

                                      },

                                      child: const Text('Cancel'),

                                    ),

                                    TextButton(

                                      onPressed: () {

                                        Navigator.pop(context);

                                        sharedPreferences.clear();

                                        Navigator.pushNamedAndRemoveUntil(

                                            context,

                                            '/login',

                                            (route) => false);

                                      },

                                      child: const Text('Log out'),

                                    ),

                                  ],

                                );

                              },

                            );

                          },

                          icon: Icon(

                            Icons.power\_settings\_new,

                            color: Constants.altColor,

                            size: 30.0,

                          ))

                    ],

                  ),

                  Divider(

                    color: Constants.primaryColor,

                    thickness: 2.0,

                  ),

                  const SizedBox(height: 30.0),

                  Container(

                    width: size.width,

                    height: size.height / 1.8,

                    decoration: const BoxDecoration(

                      color: Colors.white,

                      borderRadius: BorderRadius.all(Radius.circular(30.0)),

                    ),

                    child: FutureBuilder(

                      future: RemoteService.locationResponse(context),

                      builder: (context, snapshot) {

                        if (snapshot.hasData && snapshot.data!.isEmpty) {

                          DefaultText(

                            text: "No Locations",

                            color: Constants.primaryColor,

                          );

                        }

                        if (snapshot.hasData) {

                          var data = snapshot.data;

                          return ListView.builder(

                              itemCount: data!.length,

                              itemBuilder: (context, index) {

                                return Container(

                                  margin: const EdgeInsets.only(

                                      bottom: 5.0, left: 10.0, right: 10.0),

                                  width: MediaQuery.of(context).size.width,

                                  decoration: const BoxDecoration(

                                    borderRadius:

                                        BorderRadius.all(Radius.circular(20.0)),

                                    color: Colors.white,

                                  ),

                                  child: ChoiceChip(

                                    label: Container(

                                      padding: const EdgeInsets.all(10.0),

                                      width: size.width,

                                      child: DefaultText(

                                        align: TextAlign.center,

                                        text: snapshot.data![index]!.name,

                                        size: 22.0,

                                      ),

                                    ),

                                    shape: RoundedRectangleBorder(

                                        borderRadius:

                                            BorderRadius.circular(30.0)),

                                    backgroundColor: Colors.white,

                                    selected: \_selectedIndex == index,

                                    selectedColor: Constants.primaryColor,

                                    onSelected: (bool selected) {

                                      setState(() {

                                        if (selected) {

                                          \_selectedIndex = index;

                                          destLat =

                                              snapshot.data![index]!.latitude;

                                          destLng =

                                              snapshot.data![index]!.longitude;

                                        }

                                      });

                                    },

                                  ),

                                );

                              });

                        }

                        return const Center(child: CircularProgressIndicator());

                      },

                    ),

                  ),

                  const SizedBox(height: 20.0),

                  SizedBox(

                      width: size.width,

                      child: DefaultButton(

                          onPressed: () {

                            print("selected index: $\_selectedIndex");

                            if (\_selectedIndex < 0) {

                              ScaffoldMessenger.of(context).showSnackBar(

                                  const SnackBar(

                                      content: DefaultText(

                                          text:

                                              "You haven't made any selection",

                                          size: 18.0)));

                            } else {

                              \_getRoute(

                                  double.parse(destLat), double.parse(destLng));

                            }

                          },

                          text: "Route",

                          textSize: 25))

                ],

              ),

            ),

          ),

        ),

      ),

    );

  }

}

Views.dart

import 'package:flutter/material.dart';

import 'package:place\_finder/main.dart';

import 'package:place\_finder/services/remote\_service.dart';

import 'package:place\_finder/utils/constants.dart';

import 'package:place\_finder/utils/defaultButton.dart';

import 'package:place\_finder/utils/defaultText.dart';

import 'package:place\_finder/utils/defaultTextFormField.dart';

import 'package:http/http.dart' as http;

class AdHomePage extends StatefulWidget {

  const AdHomePage({super.key});

  @override

  State<AdHomePage> createState() => \_AdHomePageState();

}

class \_AdHomePageState extends State<AdHomePage> {

  \_deleteLocation(String id) async {

    await RemoteService.deleteLocation(context, id);

    await RemoteService.locationResponse(context);

    Navigator.pop(context);

    setState(() {});

    ScaffoldMessenger.of(context).showSnackBar(SnackBar(

        content:

            DefaultText(text: "Location successfully deleted!", size: 18.0)));

  }

  @override

  Widget build(BuildContext context) {

    Size size = MediaQuery.of(context).size;

    return SafeArea(

      child: Scaffold(

        backgroundColor: Constants.backgroundColor,

        body: SingleChildScrollView(

          child: Padding(

            padding:

                const EdgeInsets.symmetric(horizontal: 20.0, vertical: 20.0),

            child: SingleChildScrollView(

              child: Column(

                children: [

                  Row(

                    mainAxisAlignment: MainAxisAlignment.center,

                    children: [

                      DefaultText(

                        text: "Available Locations",

                        size: 25.0,

                        weight: FontWeight.bold,

                        color: Constants.altColor,

                      ),

                      const Spacer(),

                      IconButton(

                          onPressed: () {

                            showDialog(

                              context: context,

                              builder: (context) {

                                return AlertDialog(

                                  title: const Text('Confirm Logout'),

                                  content: const Text(

                                      'Are you sure you want to logout? '),

                                  actions: [

                                    TextButton(

                                      onPressed: () {

                                        Navigator.pop(context);

                                      },

                                      child: const Text('Cancel'),

                                    ),

                                    TextButton(

                                      onPressed: () {

                                        Navigator.pop(context);

                                        sharedPreferences.clear();

                                        Navigator.pushNamedAndRemoveUntil(

                                            context,

                                            '/login',

                                            (route) => false);

                                      },

                                      child: const Text('Log out'),

                                    ),

                                  ],

                                );

                              },

                            );

                          },

                          icon: Icon(

                            Icons.power\_settings\_new,

                            color: Constants.altColor,

                            size: 30.0,

                          ))

                    ],

                  ),

                  Divider(

                    color: Constants.primaryColor,

                    thickness: 2.0,

                  ),

                  const SizedBox(height: 30.0),

                  SizedBox(

                    height: size.height,

                    child: FutureBuilder(

                      future: RemoteService.locationResponse(context),

                      builder: (context, snapshot) {

                        if (snapshot.hasData && snapshot.data!.isEmpty) {

                          return DefaultText(

                            text: "No Location",

                            size: 25.0,

                            color: Constants.altColor,

                          );

                        } else if (snapshot.hasData) {

                          var data = snapshot.data;

                          return ListView.builder(

                            shrinkWrap: true,

                            scrollDirection: Axis.vertical,

                            itemCount: data!.length,

                            itemBuilder: (BuildContext context, int index) {

                              return Container(

                                margin: const EdgeInsets.only(bottom: 10.0),

                                width: MediaQuery.of(context).size.width,

                                decoration: const BoxDecoration(

                                  borderRadius:

                                      BorderRadius.all(Radius.circular(20.0)),

                                  color: Colors.white,

                                ),

                                child: ListTile(

                                  onTap: () {},

                                  title: DefaultText(

                                    size: 18,

                                    text: snapshot.data![index]!.name,

                                    color: Constants.primaryColor,

                                    weight: FontWeight.w500,

                                  ),

                                  trailing: Row(

                                    mainAxisSize: MainAxisSize.min,

                                    children: [

                                      IconButton(

                                          onPressed: () {

                                            Navigator.pushNamed(

                                                context, '/updateLocation',

                                                arguments: {

                                                  'id': snapshot

                                                      .data![index]!.locationId,

                                                  'name': snapshot

                                                      .data![index]!.name,

                                                  'latitude': snapshot

                                                      .data![index]!.latitude,

                                                  'longitude': snapshot

                                                      .data![index]!.longitude

                                                });

                                          },

                                          icon: const Icon(

                                            Icons.edit,

                                            color: Colors.amber,

                                          )),

                                      IconButton(

                                          onPressed: () {

                                            Constants.dialogBox(

                                                context,

                                                "Are you sure you want to delete ${snapshot.data![index]!.name} location?",

                                                Colors.white,

                                                Constants.altColor,

                                                Icons.info\_outline,

                                                buttonText: "Delete",

                                                buttonAction: () async {

                                              \_deleteLocation(snapshot

                                                  .data![index]!.locationId);

                                            });

                                          },

                                          icon: Icon(

                                            Icons.delete,

                                            color: Constants.altColor,

                                          )),

                                    ],

                                  ),

                                ),

                              );

                            },

                          );

                        }

                        return const CircularProgressIndicator();

                      },

                    ),

                  ),

                ],

              ),

            ),

          ),

        ),

        floatingActionButton: FloatingActionButton(

          backgroundColor: Constants.altColor,

          onPressed: () {

            Navigator.pushNamed(context, '/addLocation');

          },

          child: const DefaultText(text: "+", size: 30.0),

        ),

      ),

    );

  }

}