

HAMBA



Email Address: qinisodap@gmail.com

Supervisor: Mrs I.N Ezeji

Institution: University of Zululand

1. **Introduction**

**Abstract**

The Hamba project of the University of Zululand is a project that is a lift club and ride hailing system, project according to the needs of students. The system primary allows both students and staff members to request a lift from drivers that have registered on the platform. Because the system is for the school the school will have to appoint a suitable administrator.

The University of Zululand was selected as a case study because of its strategic significance to achieve architecture. A study of the structural structure and methods for design is adopted. In addition, Java was used to design the front end of the program and Firebase was used to design the back end. The software will make the lives of students simpler and safer.

The purpose of this software document is to describe in detail both the user requirements and the system requirements for the proposed system the “HAMBA” software and its functionality. Its intention is to show interaction between programmers and users within the proposed system from its inception to delivery by using requirements specification which is the first phase in the development of software. As aforementioned its intended audience are the users such as university students, car drivers, and university administrator. Other intended audiences are software developers which are deeply involved in software development from requirements specification to maintenance.

### The Background

The HAMBA is a mobile application which assists the students at the University of Zululand to travel around local areas and/or long distances if needs be seeing that sometimes it is quite difficult to find transport at any time in case of emergency. For example, a transport travelling to, and from Richards Bay is scarce in Kwa-Dlangezwa Taxi rank.

On the hand, it’s still troublesome to keep the Covid-19 regulations when travelling with a public transport such as taxis, and buses. Therefore, these public transports are, or can become another super spreader of the virus. Car drivers can provide their information using the mobile application by creating account.

This personal information will be displayed to the student webpage when logged in and the student will also create account and view the list of drivers and choose one to Driver. For safety reasons, car drivers while signing up will submit their information such as licence number, image of and car model, indicate whether a car is covered by insurance, or not. Administrator will use the HAMBA application to control the system’s functionality. The administrator can verify the authenticity of the driver’s information and manage their information. The software system will require both the GPS connection and the internet connection for the car driver to know where to pick the on campus and/or off campus student. GPS-navigator application will be embedded together with the HAMBA software system within the mobile application.

### The Objectives

The proposed system (through the Application) aims to simplify the procedures of the Application will be capable to lever many amenities to take care of all users in a quick manner. As a solution to a large quantity of student request handling going on in the institution.

This Application can furthermore also be used to fix all the weaknesses, security and ease of use and, furthermost significantly, the consistency of data reclamation are some errors and have subjective satisfaction with the general end-user.

Ensures that contact inside the requests scheme is successful and secure. The Application also allows for easy data access, data modification without any hindrances. The framework enforces security measures to prevent entry by persons who are not enrolled at the university.

In order to fulfil all stated set objectives – as in Systems Requirements Specifications (SRS) – the requirements analysis and assembly processes are crucial for the success of any software engineering scheme/project. Requirement’s analysis in software Development is a procedure that defines the duties that are essential to ascertain the requirements and conditions to create a new product or to make modifications to any current product/application.

**The requirements ought to be actionable, measurable, testable, and linked to the defined desires of the Software system design. From the software-engineering viewpoint, requirements evaluation is a three-step process.**

1. **Requirements Elicitation**: evocation of requirements (also known as requirements collecting) consist of the undertaking of identifying several requirement types from participants or project documentation.

2. **Requirements** Analysis: determines if the gathered requirements are clear, complete, and consistent. This analysis also handles any ambiguous requirements that do not clearly state what needs to be implemented, which could create a loss of resources and time if identified later in the development or testing phase. Requirement analysis requires identifying the stakeholders and considering their needs to help them understand the implications of designing the new system, along with the following:

* Everyone who uses the system,
* Everyone who benefits from the Application(system),
* Everyone who is Explicitly or implicitly involved in purchasing,
* Everyone or organizations argue with the Application(system), and
* Organizations responsible for the system design.

**3. Requirements Documentation**: This action entails documenting the requirements in several procedures, involving swift lists, normal language documents, visual documents, use cases, user stories, or method specifications. A requirement description document is classified in different ways according to the stakeholders’ needs, helping to create a clear contract between development and business.

### Project Purpose

The main objective of this project is to minimize the incidents of robbing students in and around campus, circumvent the spreading of the COVID virus and make it easy for students and lecturers to find lift clubs increasing their savings, providing job opportunities, circumvent human trafficking, save some people from the dreadful taxi maths and the altercations that may arise in public transportation. Inevitably saving peoples live and make a difference within the community. The plan is to create a specification, refining and execution of the current framework.

### Research Methodology

To ensure that the system meets all the expectations there was the data was collected through the following methods:

**Observation method**

The project closely observed the activities of student on a regular basis place. Then we also observed that there was a great number of potential users (students/lecturers) who were traveling from Richards Bay who don’t have a reliable mode of transportation, they’re several students that study till late and must sleep on campus since they can’t go out of the campus late and there is no transport available. We also factored in the taxi’s and saw that they drop people off by the main roads and it is not convenient especially if people are carrying groceries as well as goods, students are a times told to leave the campus residence due to the uproars and strikes that take place

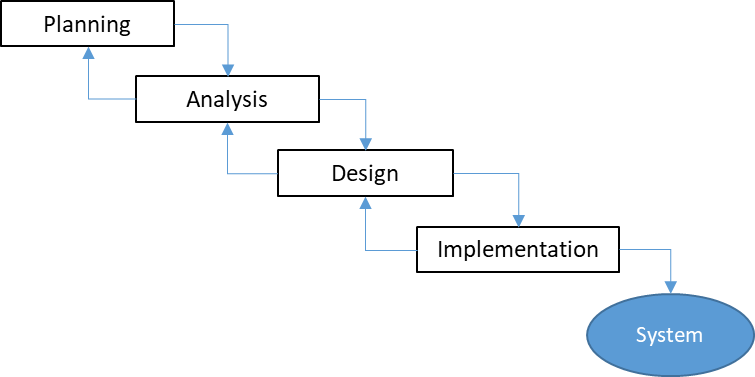
**Interview Method**

This was an exhaustible method of data collection. We asked the students about their experience at on campus. Some went as far as suggesting a few things that could be implemented. They specified their needs then I came up with the idea of making the system that will fulfil their needs.

**Computational research method**

I did my research over the internet about a similar system and looked at the current system at the university. It is not innovative as the one that I will develop and the current one is not fulfilling the needs of the users.

### Diagram of Waterfall Methodology



#### Data Flow Diagram

The graphical representation of data flow through an information system is a data flow diagram (DFD), modelling its process aspects. A DFD displays information that will be entered for output from the User Interface Block (UIB) system, where the data will come from and where it will go, and where the data will be stored in the UIBS database in this case. DFD'S is accomplished on several levels. In the next level, each process in the lower-level diagrams is broken down into a more detailed DFD. Context diagrams are often called top-level diagrams.

**Level 0 Context view of the Hamba System**

**User Searching**

**User details Driver reservation Maintain**

**Generate Administrator**

**LEVEL1: Admin View**

**Administrator Login Validate Login details**

**Maintain Table Database**

## Managerial processes

### **Project Deliverable**

All the items listed in this subsection are the deliverable requested by Project Managers.

* Project documentation
* Software Requirements Specification (SRS)
* Software Design Specification (SDS)
* Software Project Management Plan (SPMP)
* Software Test Plan (STP)
* Installation of the software program and Hamba binaries on target software.

### Evolution of the Plan

The plan is considered a complicated document and will be revised weekly, if necessary. Students and lecturers will be able to browse via an Android Application. The device will update the consumer about the return date of the item on time. Once a week, planned updates to the plan will occur. Notification of expected and unscheduled changes to the plan will be transmitted via desktop notification. When the initial plan is finalised, a benchmark of the plan will be set.

### Advantages of the system

* The system enables online access anywhere, anytime.
* Is an outlet that enables students as staff to apply

### Disadvantages of the system.

## Glossary

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| **User** | Student, driver, and/or administrator |
| **Database (DB)** | Data storage room |
| **OS** | Operating System |
| **GPS** | Global Positioning System |
| **HAMBA** | The proposed software system |
| **GB** | Giga Bytes |
| **GHz** | Giga Hertz |
| **UZ** | University of Zululand |
| **DFD** | Data flow diagram |

## References

[1] IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

## Software requirement document organizational structure

Section two deals with overall description in which the requirements overview is given. It looks at factors that impact the requirements. It consists of subsections in random order such as product perspective, design constraints, product functions, assumptions and dependencies, and apportioning of requirements. Sections three and four is the framework for system designers and deals with analysis of requirements to ensure that the development of system product is in-line with the requirements. Section five (physical design) gives an idea how will the system look like in reality. Section six (System models) deals relationship, association and activities performed by objects. Section seven deals with hardware and software requirements. Section eight (Conclusion and future directions) deals with maintainability and extensibility of the software system

# Software Requirement Specification

## Purpose

The purpose of this software document is to describe in detail both the user requirements and the system requirements for the proposed system the “HAMBA” software and its functionality. Its intention is to show interaction between programmers and users within the proposed system from its inception to delivery by using requirements specification which is the first phase in the development of software. As aforementioned its intended audience are the users such as university students, car drivers, and university administrator. Other intended audiences are software developers which are deeply involved in software development from requirements specification to maintenance.

The device developer that is developing Hamba and the university end users can use this SRS. To build the required program, the project developer can use the SRS to fully understand the expectations of this RIDE HAILING APPLICATION/ LIFT CLUB. The end-users of the university will be able to use this SRS as a "test" to see whether the system will be designed to their specifications by the building developer. The end users can decide if it is not to their taste if it is not to their standards and the developer will alter the SRS to meet the needs of the end-users.

## Intended Audience and Reading Suggestions

In order to refer and evaluate the material, the intended audience of this document will be the owner and specific staff such as administrators and students/lecturers and project developers. In any case, the SRS document may be used concerning the project specifications and the solutions that have been implemented. Ultimately, the paper would provide a clear picture of the framework being designed. A brief outline of the document is as follows:

1. Overall Description,

2. System Features,

3. External Interface Requirements, and

4. Non-Functional Requirements.

## Product Scope

The implementation of the Hamba software would simplify the university's main operations. The reservation system is designed to monitor Drivers and reservations for cancellation and to check availability. The notification system is intended for users to be informed of when Drivers should be returned and whether reserved Drivers are eligible. Will regulate. There are two HAMBA users. Students and managers are the end-users. The administrator is unable to control all device functions without limitations. Some of the functionalities of the features can also limit students. HAMBA can build different Login features to maintain restrictions for each End User Level (EUL).

The purpose of the Automated HAMBA is to simplify the university's day-to-day processes. The device would be capable of handling multiple resources in a fast way to take care of all users. This program would be used to resolve such limitations as a solution to the vast amount of file handling occurring at the university. Safety, easiness of using and most importantly the efficiency of information retrieval is some benefits the development team going to present with this system. The framework should be ideal for the user, easy to use, easy to recover errors and high subjective satisfaction for the overall end-user.

## Overall Description

### Product Perspective

The HAMBA transport software is an application developed to operate on both website and mobile application. The software is designed to assist the students who wish to travel at any time around local areas without being restricted by operation hours of public transport. The individual who wants to Driver a car will be able to choose different modes of transportation such as mini car for grocery shopping, and/or mini truck for furniture, to name a few.

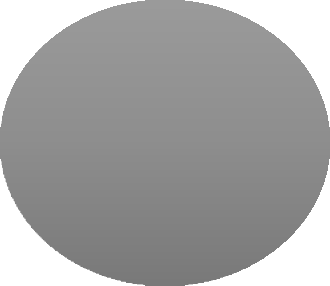
Since the mobile application will consist of a GPS-navigator application this will help in finding a customer’s physical location, in this case a student and driver’s location for who might to relocate to on campus residences or off campus accommodation.

A database will be needed to store user’s credentials since this project

The system that has been implemented will provide easy access to the system and will provide user-friendly functionality with appealing interfaces. The system will provide better options for the problem of large-scale handling of the physical file system, for calculation errors and all other required tasks specified by the client. The result of this project would in a very convenient way improve the efficacy of almost all the activities conducted at the university.

is data centric. Web portal will be used by the administrator to manage data of both student and drivers. The following diagram is intended to show the flow of data from one entity to another.

Make payment



Signup

Confirmation

Payment made

HAMBA

Sign up

Login

SOFTWARE

SYSTEM

Add

user

Driver a car

Car Drivered

Payment

Confirmation

Instant message for client car

Administrator

User

Figure 1. Level 0 of Data Flow Diagram (DFD) of HAMBA

## 2.6 Product functions

The user will have to register to get an account using his/her information such as username, identity number and password, and login thereafter. Account registration confirmation will be sent to mobile number. The user will check drivers for availability by viewing the list displayed on mobile application. After picking a driver his/her travelling details will be displayed with his/her time schedule detailing his/her availability depending on whether the driver has another occupation.

On the driver’s time schedule student will choose a date and time suitable to travel from one location area. While the system will enable its user to make a call or chat (preferably WhatsApp chat) with a car driver.

GPS-navigator application will help in finding a customer’s physical location, in this case a student and driver’s location for who might to relocate to on campus residences or off campus accommodation. On the other hand, on the web portal the system will add or delete cars

## User characteristics

The overall system consists of three entities that interact with it: university students, car drivers, and administrator. These entities each has its own requirements and use the system differently from each other.

The university students use the mobile application only to find a car driver, make Booking, and pay thereafter. The student will view the list of car drivers, choose a driver from that page and then view the profile of a driver which consist of his details (mobile number, social media account), the car models he/she owns, a weekly time schedule for booking.

The car drivers will use the mobile application to update their profile. Firstly, the driver can change a car model, indicate whether he/she is on/off duty.

The administrator will manage the overall system operation to ensure that there is no incorrect information within it. The administrator will verify both students’ credentials, drivers, changes made in the profile of, by the car drivers. Timetable for drivers will be made and modified,

if necessary, by the administrator.

## General Constraints

The internet connection: the software system will be effective and efficient if data move to, and from database. Therefore, it is important there is access to internet. There will be a long queue since all the aforementioned entities will make requests which will send to database server. Therefore, the system might take longer to respond than expected.

## Assumption and dependencies

This product will run on mobile phone with high performance. Insufficient hardware resources might cause the application to display semantic errors.

## Requirements apportioning

if the project is not finished on timespan of three months, then some requirements could be transfer to the next version.

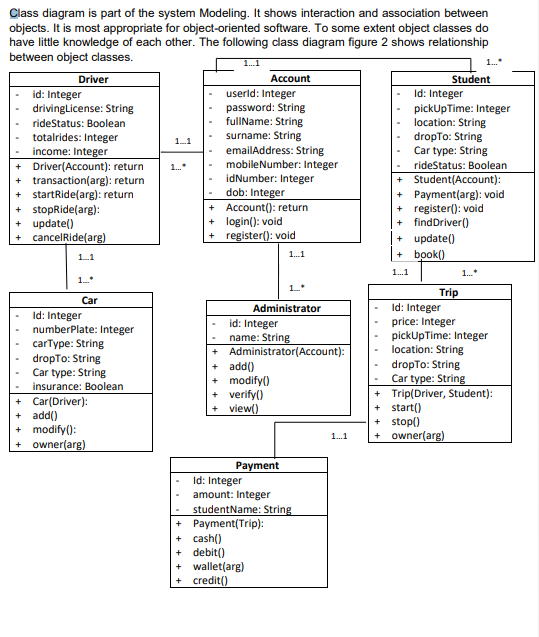
## Overall Description

## Product Perspective

Hamba is a self-contained software product that the developer of the project will produce to overcome the problems that have occurred. The system that has been implemented will provide easy access to the system and will provide user-friendly functionality with appealing interfaces. The system will provide better options for the problem of large-scale handling of the physical file system, for calculation errors and all other required tasks specified by the client. The result of this project would in a very convenient way improve the efficacy of almost all the activities conducted in and around campus.

## Product Functions

* Make Reservations
* Search Drivers
* Add Drivers
* Issue Notifications
* Manage Users (Add, Update users)
* Manage Driver Details (Add, Update, Delete)
* Manage User Details (Add, Update, Delete, Edit)
* Taking Backups
* SMS notifications



**Functional Requirement**

|  |  |
| --- | --- |
| Function 1 | Make Reservations |
| Input | Name, Student number, Cell phone number |
| Output | Database Record, Database successfully updated message |
| Processing | Validate the given details and record the information in the database. |

|  |  |
| --- | --- |
| Function 2 | Add User |
| Input | Last name, First name, Phone number, Student number, ID number, email address. |
| Output | Database Record, Database successfully updated message |
| Processing | Validate the given details and record the information into the database |

|  |  |
| --- | --- |
| Function 3 | Add Administrator |
| Input | Phone number, Administrator Name, Identity number, email address, staff no |
| Output | Database Record, Database successfully updated message |
| Processing | Validate the given details and record the information into the database |

|  |  |
| --- | --- |
| Function 4 | Search DRIVER |
| Input | Driver name, ISBN, author |
| Output | Display a message with available Driver details |
| Processing | Validate the given details, check for the available Drivers, and return its availability |

|  |  |
| --- | --- |
| Function 6 | Notifications |
| Input | text message |
| Output | SMS sent |
| Processing | Verify if the details match |

|  |  |
| --- | --- |
| Function 7 | New request |
| Input | Swapping Request Option or Cancel Option |
| Output | Database Record, Database successfully updated message |
| Processing | Validate the specified elements and evidence the knowledge into the DB |

### Search Application

|  |  |
| --- | --- |
| Use Case Name | Search Application |
| XRef | Section 2.2.1, Search for driver  SDD, Section 7.1 |
| Trigger | The User assesses the Hamba Application |
| Precondition | The Application is displayed with grids for searching |
| Basic Path | 1. The User decides how to search the Application. The choices are by Driver, by Route, and by location. 2. If the search is by Driver, the system creates and presents an alphabetical list of all Drivers in the database. In the case of an application with multiple Drivers, each is contained in the list. 3. The User selects Driver. 4. The system creates and presents a list of all requests by that Driver in the database. 5. The User chooses an application. 6. The system displays the Abstract for the Application. |
| Alternative Paths | In step 2, if the User selects to search by category, the system generates and presents a list of all categories in the database.   1. The User selects a category. 2. The system creates and presents a list of all Applications in that category in the database. Return to step 5.   In step 2, if the User selects to search by keyword, the system presents a dialog box to enter the keyword or phrase.   1. The User enters a keyword or phrase. 2. The system searches the Summaries for all Applications with that keyword or phrase and creates and presents a list of all such Applications in the database. Return to step 5. |
| Post-condition | The selected Application is downloaded to the client’s phone. |
| Exception Paths | The User may abandon the search at any time. |
| Other | The categories list is generated from the information provided when Application are published and not predefined in the database. |

## Activity diagram

* + 1. **activity diagram for user**



Registration

Login

No

successful

?

Yes

Chat

View profile

Pick a car

book

Pay

Online

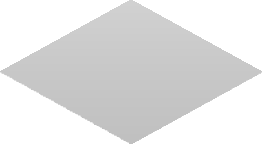
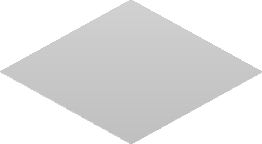
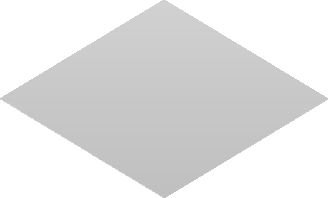
Cash

Confirmation

View drivers’ list

Figure 19. Activity diagram for user

Logout



Registration

Login

No

Successful

?

?

## Use case diagram

Update driver’s details

Edit cars

Confirmation

Online

Cash (receipt)

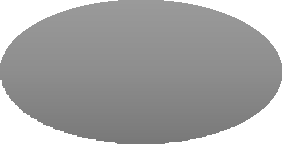
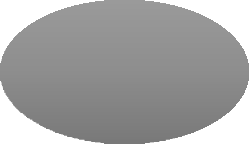
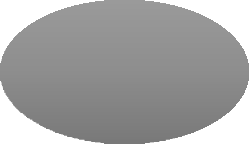
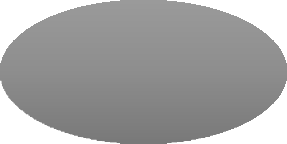
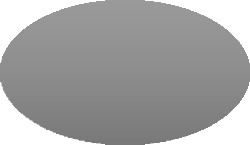
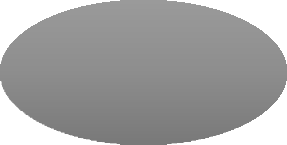
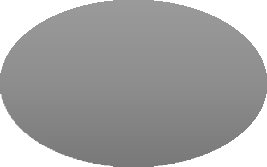
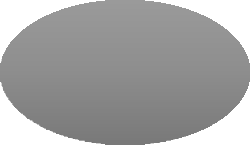
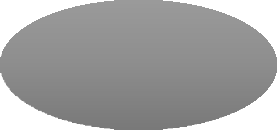
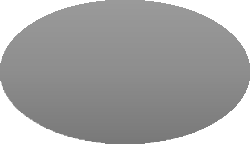
Manage payments

Update profile

Manage Booking

Update database

* + 1. **student use case diagram**



**HAMBA SYSTEM**

Registration

Error

message

<<Extend>>

Login

<<Include>>

Verify

View

Drivers

<<Include>>

Booking

View car

information

Student

Payment

Receive

Notification

Logout

* + - 1. **Use case description**
         1. **Registration**

|  |  |
| --- | --- |
| **Name** | Registration |
| **Actor** | Student |
| **Precondition** | None |
| **Description** | Describe the process for an account registration. |
| **Successful completion** | None. |
| **Alternative** | 1. Display error message if credentials are invalid. 2. Student can cancel or create new username and/or password. |

* + - * 1. **Login**

|  |  |
| --- | --- |
| **Name** | Login |
| **Actor** | Student |
| **Precondition** | Create account |
| **Description** | Describe the process for an account registration. |
| **Successful completion** | Student credentials are verified on the database. |
| **Alternative** | 1. Display error message if credentials are invalid. 2. Student can cancel or create new username and/or password. |

* + - * 1. **View drivers**

|  |  |
| --- | --- |
| **Name** | View drivers |
| **Actor** | Student |
| **Precondition** | Logged in |
| **Description** | Student can chat, and/or view driver’s  profile including rides. |
| **Successful completion** | Student can start booking. |
| **Alternative** | 1. Student can opt for online chat or call the driver of choice. |

* + - * 1. **Booking**

|  |  |
| --- | --- |
| **Name** | Booking |
| **Actor** | Student |
| **Precondition** | View drivers |
| **Description** | If ride request is successful, the student will book. |
| **Successful completion** | . |
| **Alternative** | 1. Display error message if credentials are invalid. 2. User can cancel or create new username and/or password. |

* + - * 1. **Payment**

|  |  |
| --- | --- |
| **Name** | Payment |
| **Actor** | Student |
| **Precondition** | Booking |
| **Description** | Student will choose the method of payment, online (payment gateway), or cash. |
| **Successful completion** | After paying, the student must have a proof of payment. |
| **Alternative** | If payment was made manually (by cash) then the student must have the receipt. |

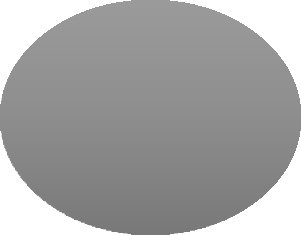
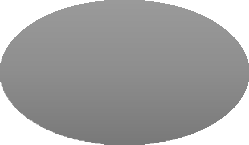
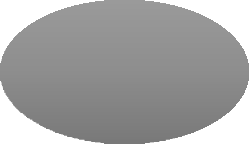
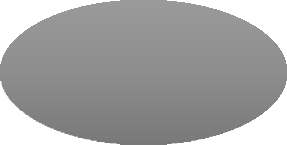
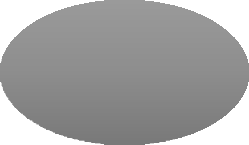
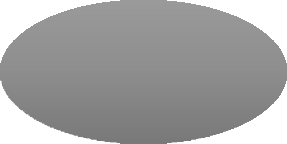
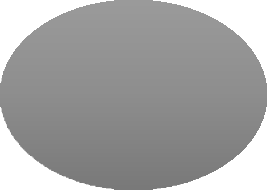
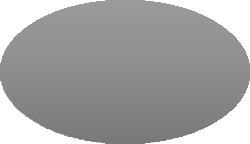
**6.3.1.6 Notification**

|  |  |
| --- | --- |
| **Name** | Notification |
| **Actor** | Student |
| **Precondition** | Payment |
| **Description** | None. |
| **Successful completion** | Student must receive message informing him/her of the ride made confirming his/her booking. |
| **Alternative** | None |

**6.3.1.1.7 Logout**

|  |  |
| --- | --- |
| **Name** | Logout |
| **Actor** | Student |
| **Precondition** | Logged in |
| **Description** | None |
| **Successful completion** | None |
| **Alternative** | None |

* + 1. **Drivers use case diagram**



**HAMBA SYSTEM**

Error

message

<<Extend>>

Login

<<Include>>

Verify

View trips

Search

pickup location

Payment

Driver

Receive

Notification

cash

Logout

Figure 22. Drivers use case diagram

<<include>>

gateway

* + - 1. **Drivers use case description**
         1. **Registration**

|  |  |
| --- | --- |
| **Name** | Registration |
| **Actor** | Driver |
| **Precondition** | None |
| **Description** | Describe the process for an account registration. |
| **Successful completion** | None |
| **Alternative** | 1. Display error message if credentials are invalid. |

* + - * 1. **Login**

|  |  |
| --- | --- |
| **Name** | Login |
| **Actor** | Driver |
| **Precondition** | Create account |
| **Description** | None. |
| **Successful completion** | Driver credentials are verified on the database. |
| **Alternative** | 1. Display error message if credentials are invalid. |

* + - * 1. **View trips**

|  |  |
| --- | --- |
| **Name** | View trips |
| **Actor** | Driver |
| **Precondition** | View drivers |
| **Description** | The user will view how many rides he/she must make a day. |
| **Successful completion** | None |
| **Alternative** | None |

* + - * 1. **Search pickup location**

|  |  |
| --- | --- |
| **Name** | Search pickup location |
| **Actor** | Driver |
| **Precondition** | View trips |
| **Description** | If request made by a student is successful, the driver will search for pickup location. |
| **Successful completion** |  |
| **Alternative** | 1. Display error message if credentials are invalid. 2. User can cancel or create new username and/or password. |

* + - * 1. **Payment**

|  |  |
| --- | --- |
| **Name** | Payment |
| **Actor** | Driver |
| **Precondition** | Booking |
| **Description** | Student will choose the method of payment, online (payment gateway), or cash. |
| **Successful completion** | After student had paid, the driver must receive notification confirming that the ride is on. |
| **Alternative** | If payment was made manually (by cash) then the driver must retain the receipt. |

* + - * 1. **Notification**

|  |  |
| --- | --- |
| **Name** | Notification |
| **Actor** | Driver |
| **Precondition** | Payment |
| **Description** | None. |
| **Successful completion** | After student had paid, the driver must receive notification confirming that the ride is on. |
| **Alternative** | None |

* + - * 1. **Logout**

|  |  |
| --- | --- |
| **Name** | Logout |
| **Actor** | Driver |
| **Precondition** | Logged in |
| **Description** | None |
| **Successful completion** | None |
| **Alternative** | None |

## Non-functional Requirements

### Performance Requirements

For device functionality, performance criteria determine appropriate response times. Although the system is configured for the lowest system performance, the performance of the system would depend heavily on the performance of the hardware and software components of the device mounted. The load time for user interface screens must take no longer than two seconds while considering the timing relationships of the device. It provides easy access to the functions of the device. The log-in details would check the efficiency of the device within five seconds. It makes the search feature more precise by returning query results within five seconds.

### Safety Requirements

In Hamba, there are many user levels, access to the different subsystems will be secured by a user login screen requiring a username and password. This offers various views and user-level functions accessible via the system. Device database security is maintained by maintaining backups. It is possible to recover the system in an emergency.

Managers of the Hamba and students would be able to log in to the device. Reservation/Ordering, searching and other subsystems will be required for students. Managers would have access to both the Management subsystem and the Subsystems for Adding Drivers. A user log-in screen requiring a username and password would secure access to the different subsystems.

### Software Quality Attributes

* • Availability: - The system must be accessible during the usual cycle of operation of the University, during exams.
* Correctness: - The degree to which the software follows expectations, meets the aims of the user's mission.
* • Efficiency: - How much less time and energy are needed to accomplish a specific task through the method.
* • Flexibility: - Ability to incorporate and easily manage new functionality to the framework.
* Integrity: - How the system will vulnerable the system's data and how it will prevent data losses. In database tables and interfaces, referential integrity
* • Maintainability: - How simple it is to maintain the system as it is and adjust fix defects.

• Portability: - Hamba runs on any android device.

* Reliability: -Define the factors needed to assess the software system's required reliability at the time of delivery. Meantime to recovery between failures and meantime
* Reusability: What is the ability to use the system components available in other systems, too?
* Testability: The effort necessary for testing to ensure performance as expected
* Usability: How quickly the advantages of the system and user-friendliness can be taken away from a person.
* Robustness: System strength to reliably control system operations and maintain the database without facing unexpected failures.
* Maintainability: What design, coding standards must comply with generated exclusions

### Business Rules

Hamba operates for Three users, Students and Administrators. The system is designed to reduce the burden and rights of students and administrators to reduce the position of administrators in order to free students from daily interference with the system. Most of the rights that the owner has are therefore granted to the administrator, especially those that are essential and relevant. The most used features of the system are offered to students who have less responsibility. Only the system administrator can remove any information on the system.

## User Classes and Characteristics

### User Classes

There are two user levels in Hamba (Ride Hailing) System:

1. Students
2. Administrator

### Characteristics of User Classes

**Administrator:**

The administrator has the responsibility of controlling and approving other system-managed activities. She can access most of the functions that the system performs. The administrator may access the admin panel that is considered the heart of the system. As the key approved person is the administrator who can get the ability to control the other users including their user levels and privileges. The Administrator may also perform backups of the system and restore the system. He would be able to take all the forms of reports available in the framework in the meantime. As the system and university administrator, he has the right to delete the user from the database of the system. His or her job would be to ensure that all users are handled with a high service level. He or she will perform several tasks, such as making reservations, sending SMS reminders for Drivers and reservations to users.

**Student:**

Students are responsible for using resources available in the Hamba (Ride Hailing) System. They also have the least of the privileges when it comes to the usage of the system. The user-level has the authority to take all the reports available in the system. Students have other abilities that Administrator has. Such as, adding their details to the system to the system, modifying them or removing them, cancelling their orders and reservations, giving their report to the system.

## Operating Environment

Hardware and software requirements for the system are outlined below:

**Hardware:**

1. **Operating System** Supports all known operating systems, such as Windows, Linux.
2. **Computer** 1087+ RAM, monitor with a minimum resolution of 1024x768, keyboard, and mouse.
3. **SMD of 250**
4. **ACER core i5**

**Software:**

1. Software is designed to run on any platform above Microsoft Windows 10 (64bit).
2. Android Studio

## Design and Implementation Constraints

Any design and implementation restrictions are implemented to ensure device reliability and longevity. Minimum memory of 512MB will be needed for the device but a memory of 1 GB is highly recommended. When developing device interfaces, new tools, such as Android Studio. Considering the budget, interfaces were produced using inexpensive technology in a clear, practical way.

Any of the software used to introduce the system is expensive and the consumer has decided to pay the sum of money required to buy it. It is expected that in the next stages of software development, the customer will not alter the decision. We believe the client uses Windows 7 or Windows 8, though. Otherwise, there is a need to adapt whether the client uses an open-source operating system.

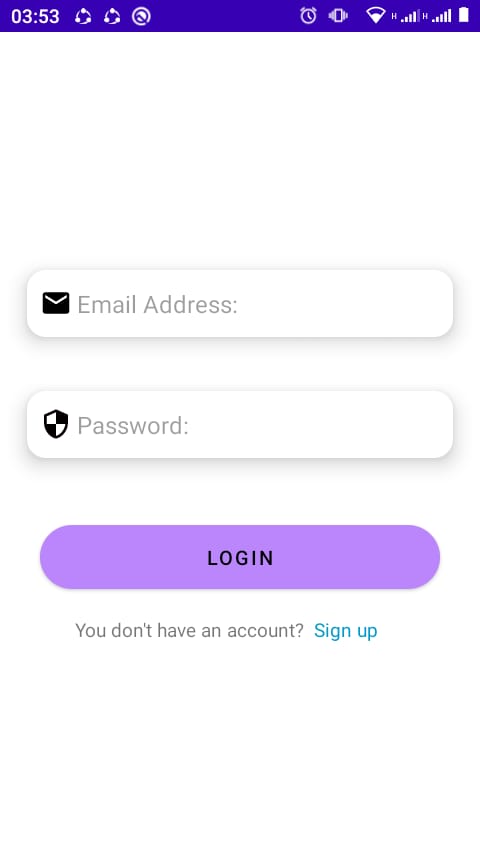
## External Interface Requirements

### User Interfaces

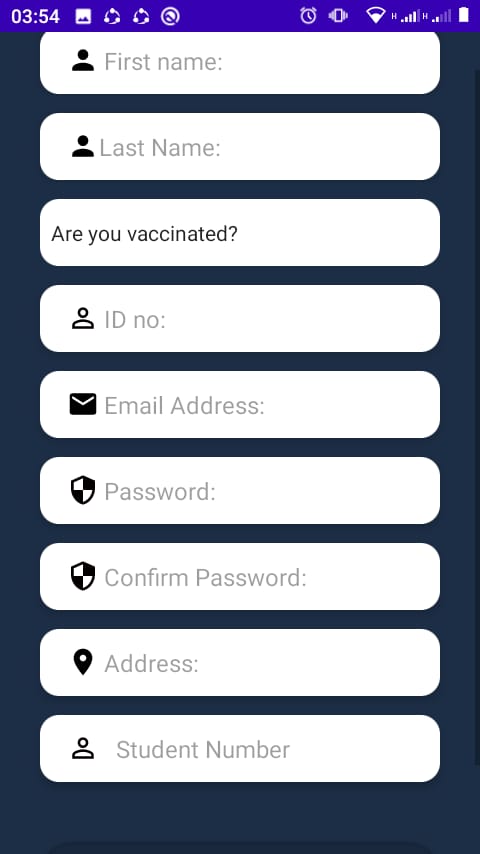
A user-friendly dashboard of the system



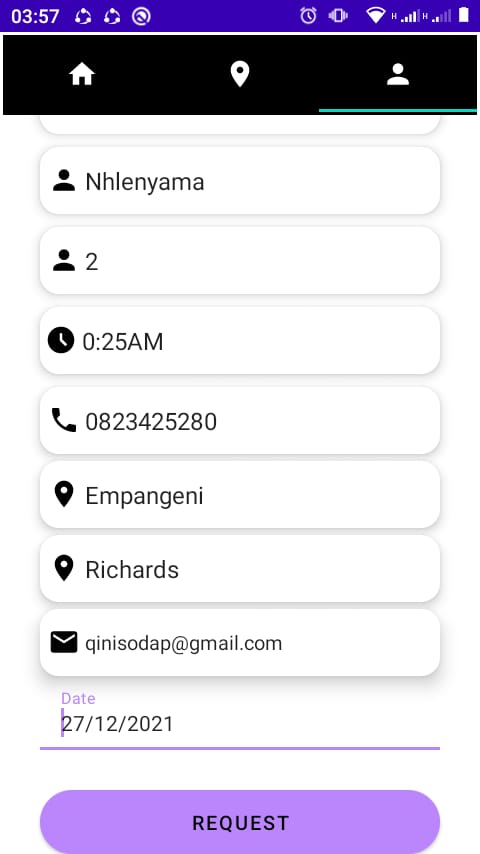
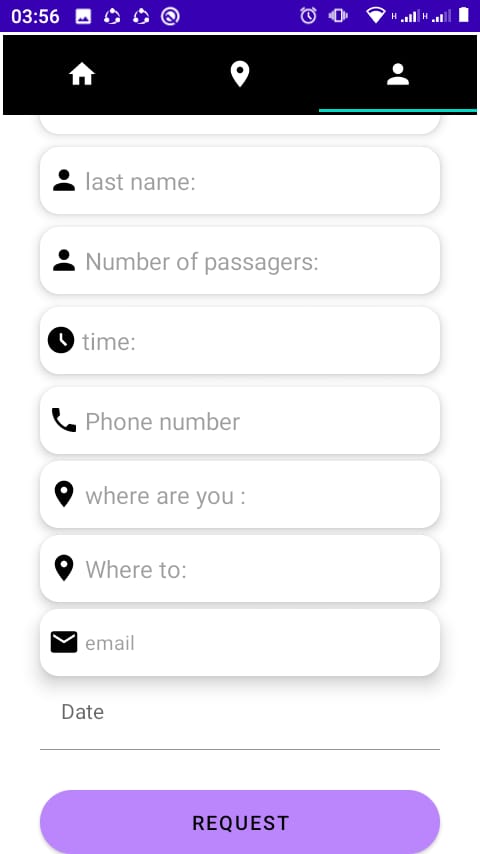
Splash screen

**

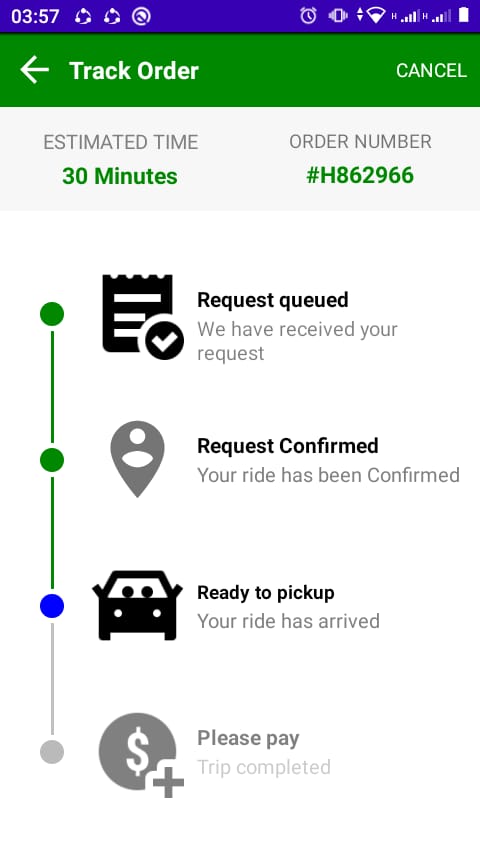
One login for Administrator, User and Driver

**

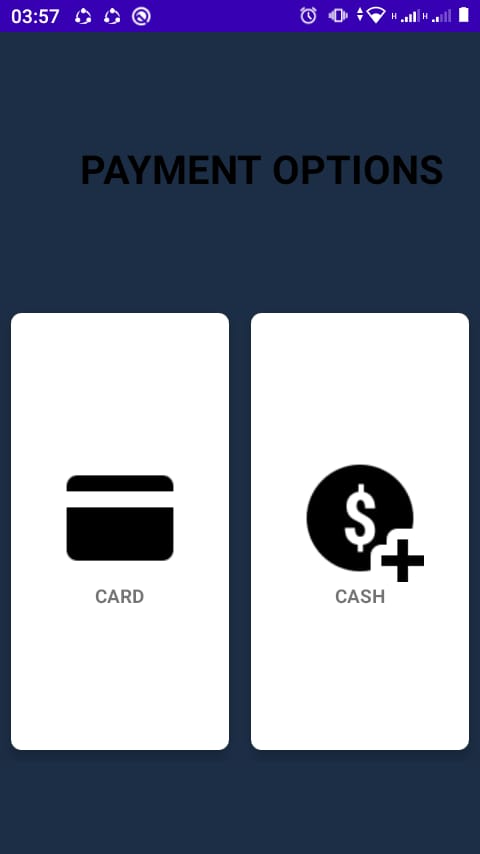
Registration



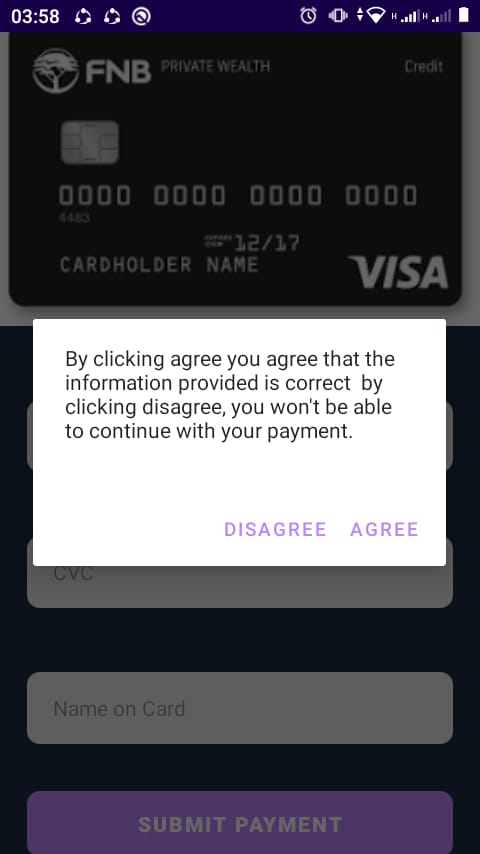
Bookings page



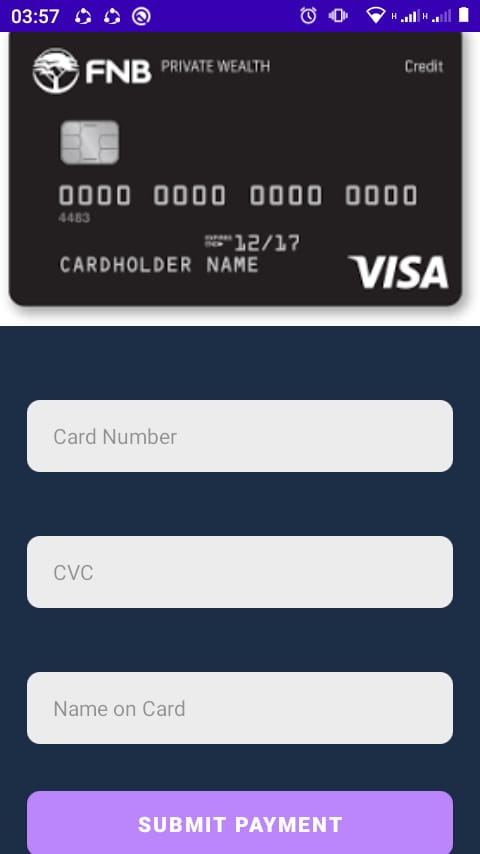
Track Request Page



Payment Options page



Confirm payement



Payment activity

A screenshot of a computer

Description automatically generated with low confidence

On Successful payment

A picture containing text, screenshot, electronics

Description automatically generated

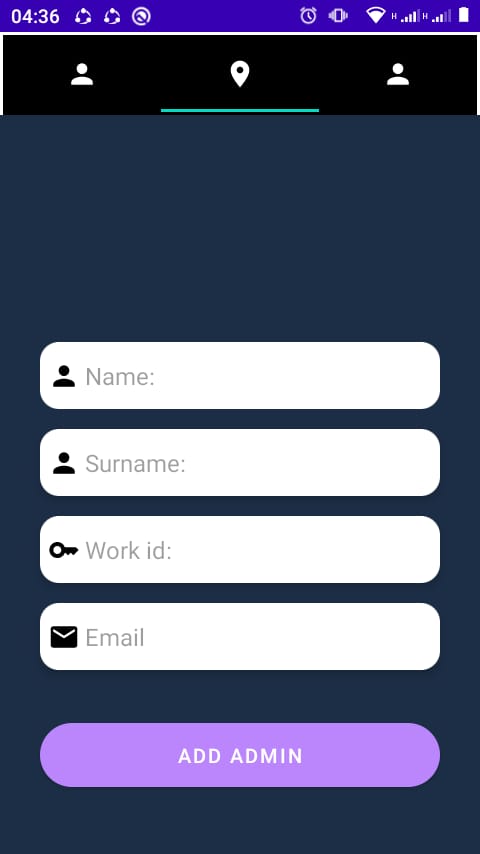
Rate driver page

‘

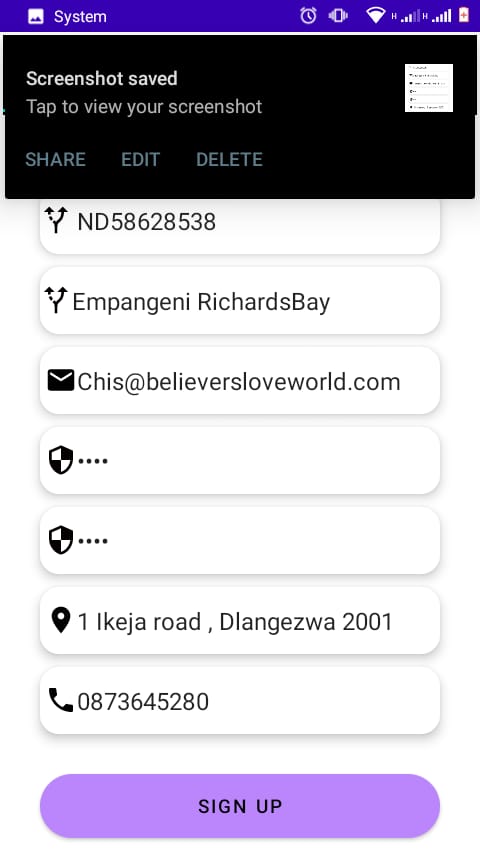
A screenshot of a phone

Description automatically generated with medium confidence

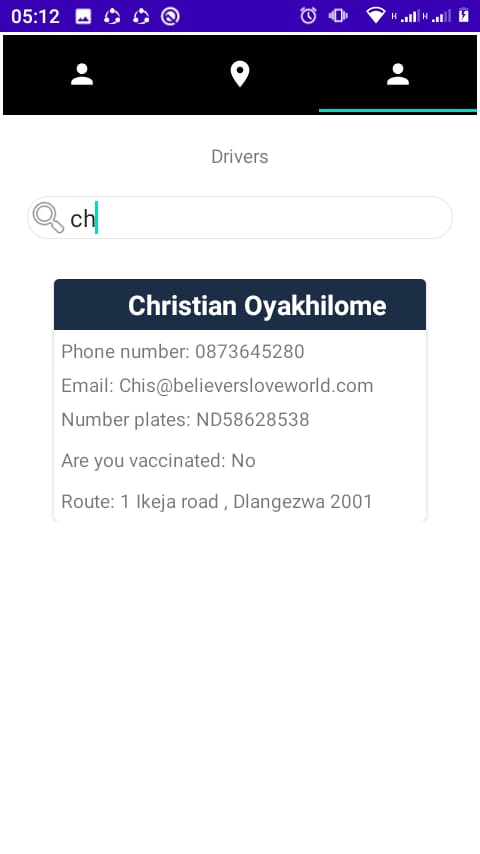
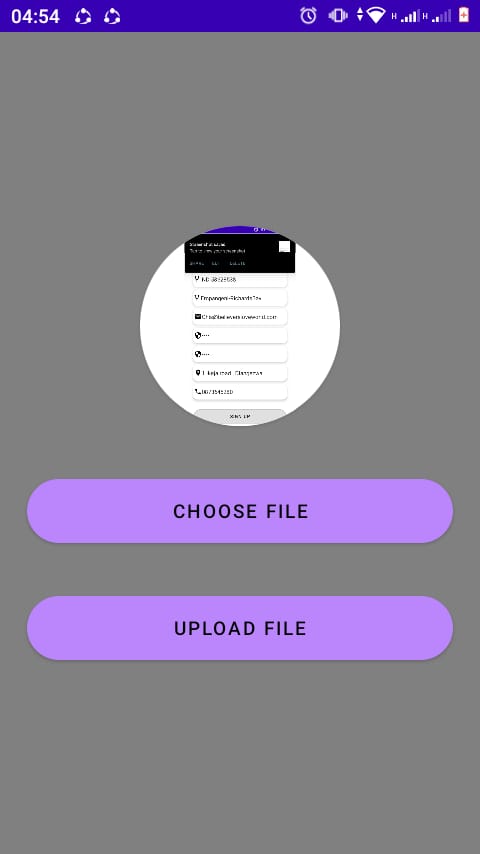
Driver review



*AdminTabs-Add administrator page*



*AdminTabs- add driver*

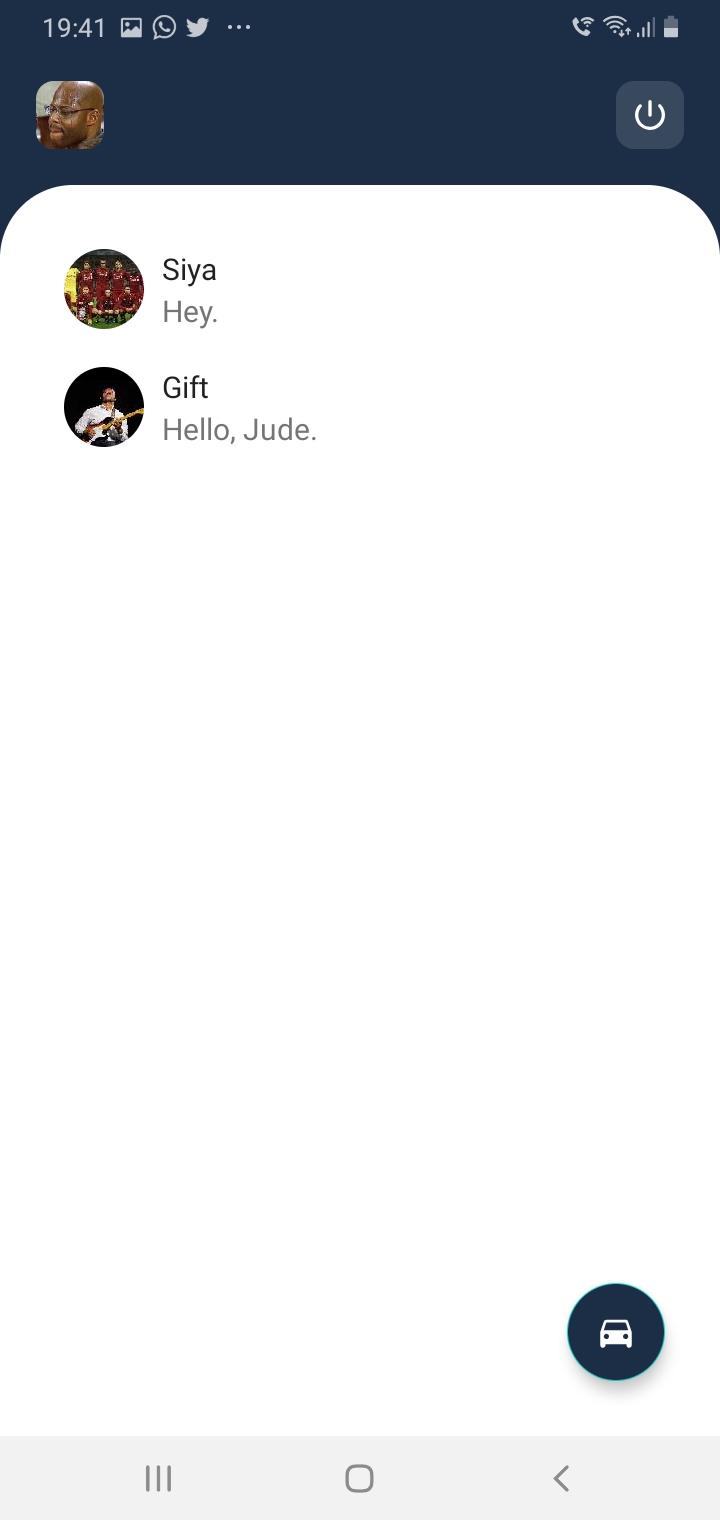


List drivers view

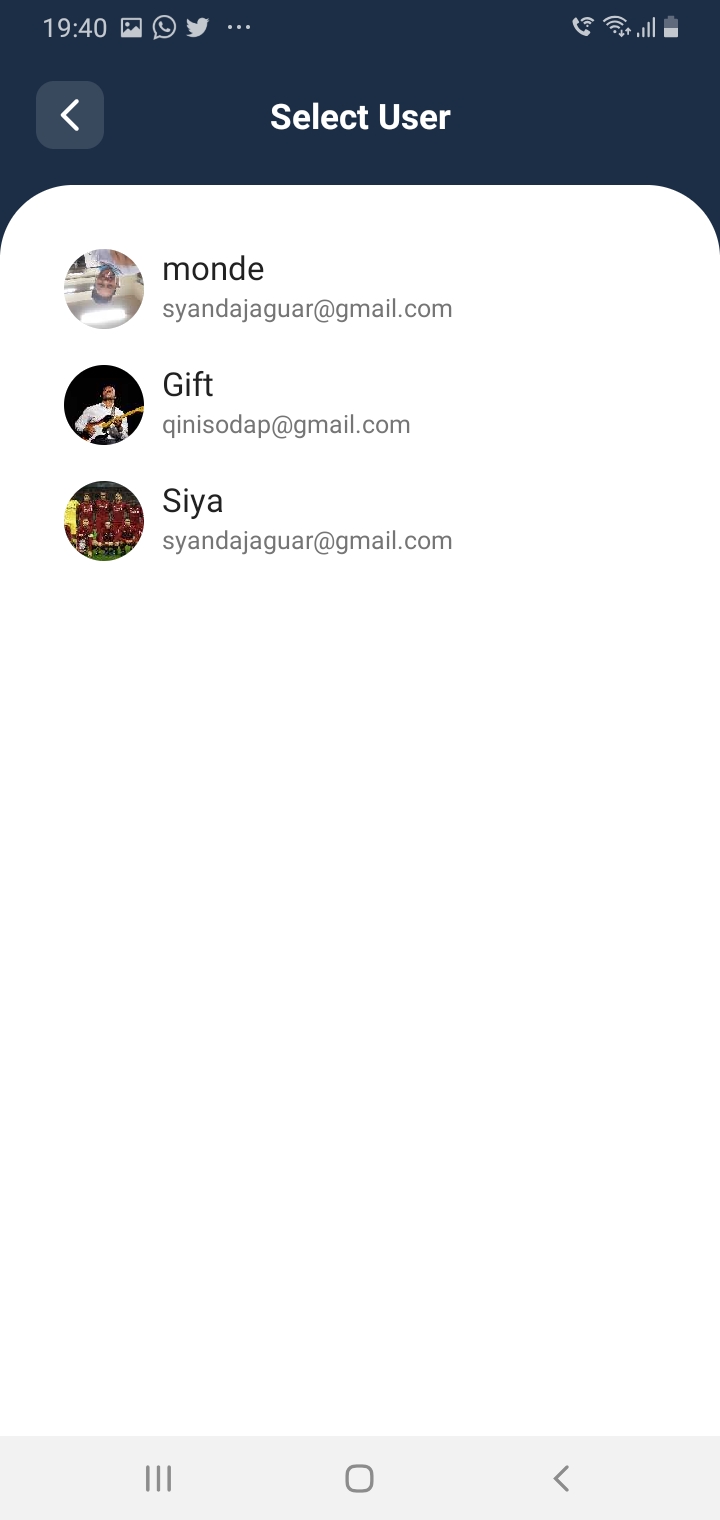
A picture containing graphical user interface

Description automatically generated

Drivers view of new client request



x x



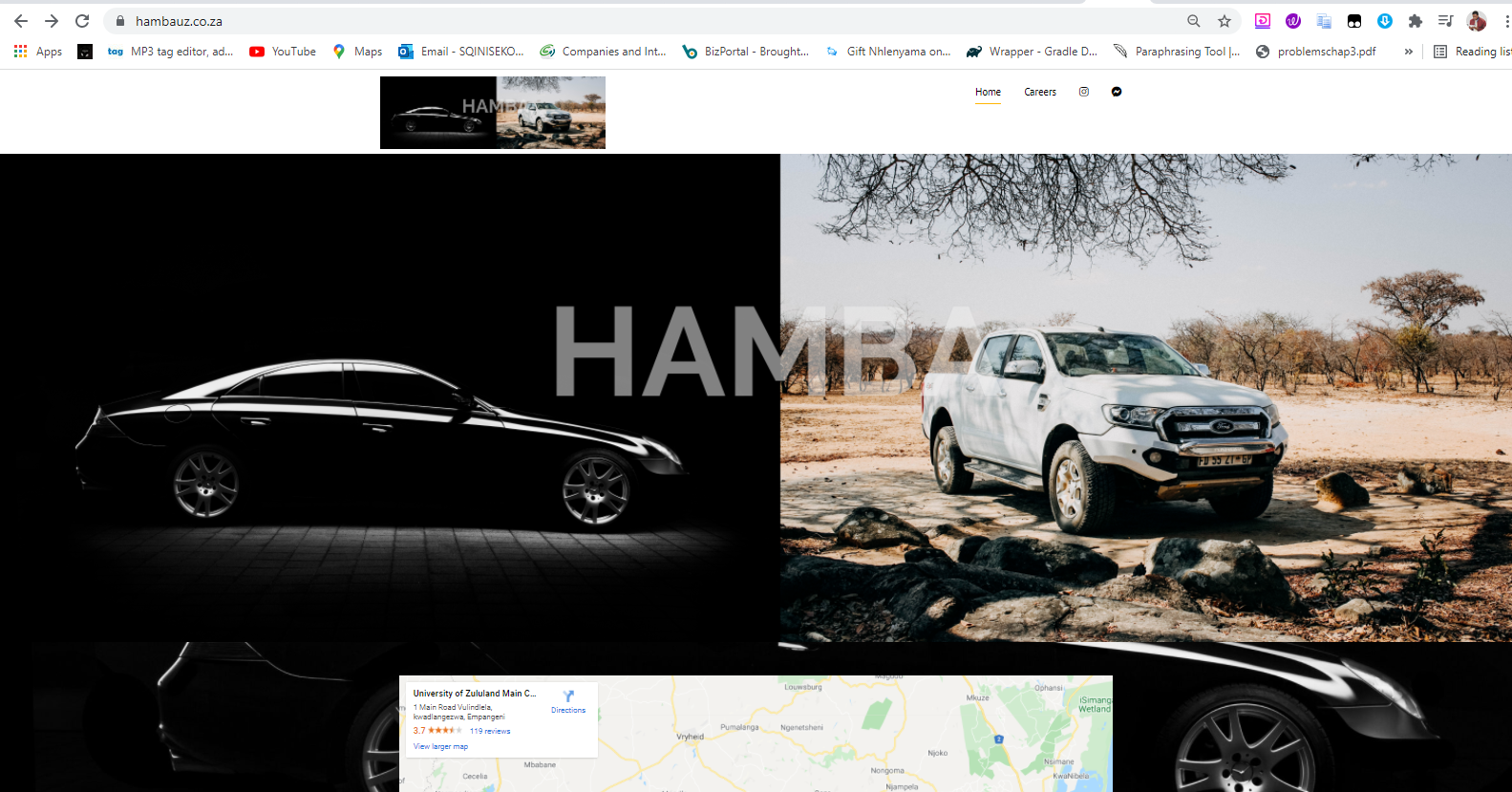
Chats page

A picture containing text, monitor, indoor, control panel

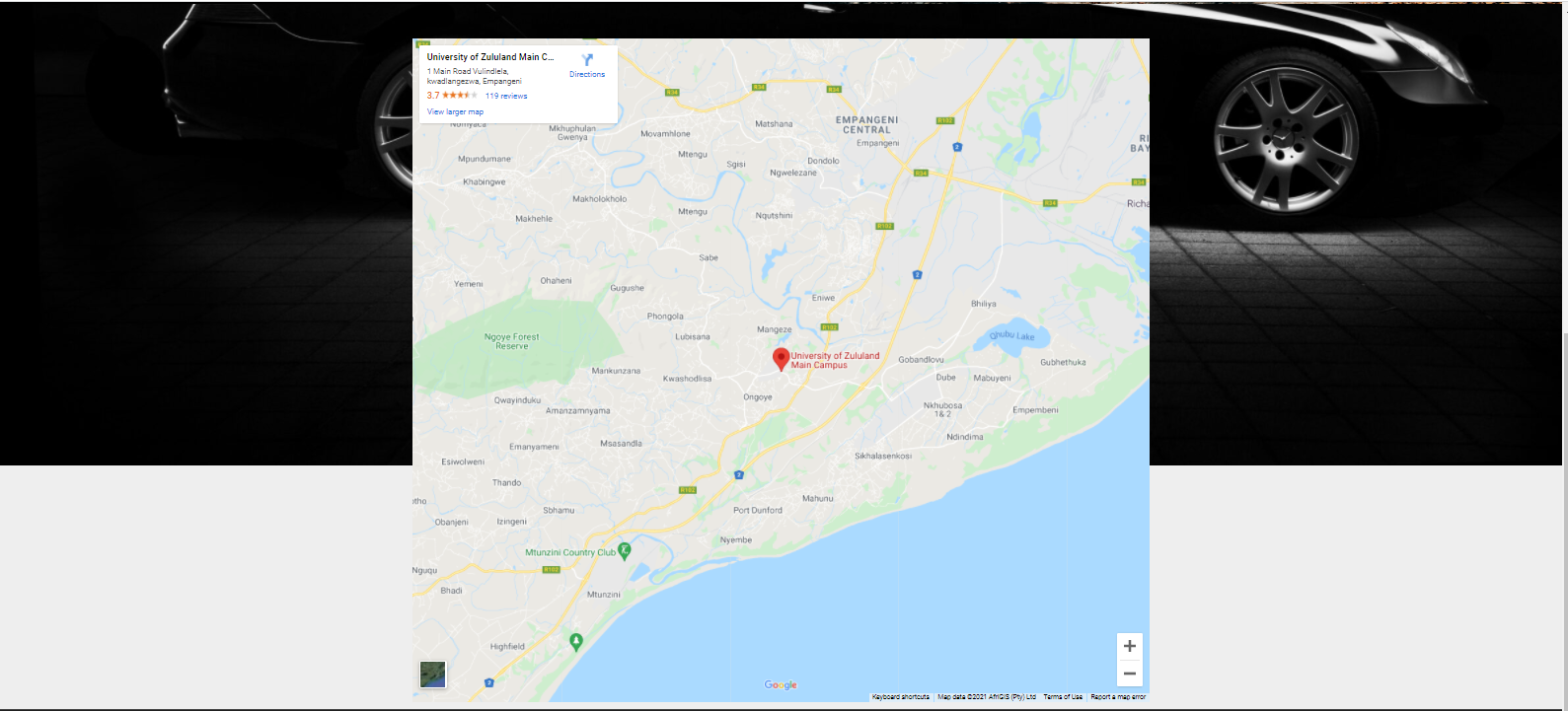
Description automatically generatedREQUEST FORM TIME PICKERA screenshot of a phone

Description automatically generated with medium confidence

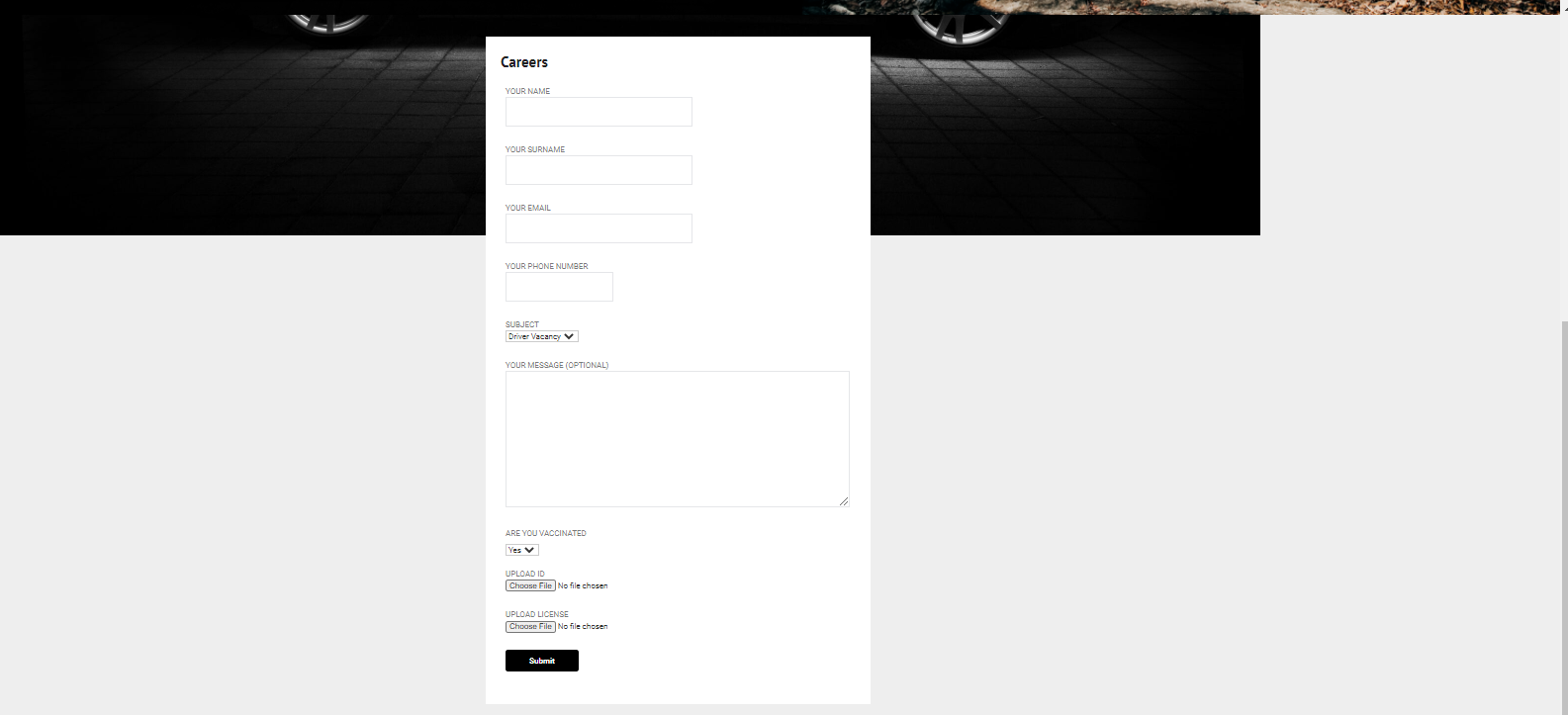
Request page-datePicker



Hamba Website



LOCATION



Vacancy Request form

# Software Design Description

## Purpose

The full design description of Hamba is included in this text. This involves the system's architectural features through descriptions of the operations performed by each code module and the layout of the database. It also demonstrates how, using this template, the use cases detailed in the SRS would be implemented in the framework. Software development is the primary focus of this paper.

## Scope

Hamba uses a standard client-server architecture with a server database. Hamba Manager is a database-using repository architecture. This system has an external system interface, as the administrator uses SMSs sent outside the system. These SMSs are used to warn students about their orders and the return of Drivers as a reminder. <<Since the SRS has defined Domain (Analysis) classes, we use them in the design.>>

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| HAMBA | Ride Hailing /Lift club application |
| Student | Undergraduate/postgraduate of the institution. |
| Administrator | A person who is in a position of authority or who manages people, practices and policies. |
| Return | To put, give or send something back to a place. |
| Keyword | Word or phrase used in searching by keyword. The case is not significant. |
| Reserve | Retain for future use |
| Notification | The act of telling someone something or the method in way notice is given. |
| searching | Trying to find something by looking or otherwise seeking thoroughly. |
| User | Administrator or Student |

## Deployment Diagram

Web Browser

management System Server

Internet

Hamba Manager Processor

Intranet

Database Server

Intranet

The system operates over the internet using a web browser that is not part of the system, while internet browsers are used to access the database. On a dedicated client service with a permanent Web connection, Hamba resides. The Administrator performs all the adding and preparing work on his/her personal computer (the Hamba Manager Processor), interacts with the Database Server Firebase Database when necessary, and when ready, uploads completed orders to the Client-Server. The Hamba Manager Processor provides a local file system and a phone number and email system for the Administrator.

# SECTION THREE: FUNCTIONAL REQUIREMENTS

This section looks at statements of service the system should provide. It also looks at how the system should respond and/or behave to input data. The overall behavior of the HAMBA software system hinges upon three entities, students, drivers, and administrator. Therefore, the system is expected to do the following:

* The system shall accept users’ credentials in a process of registration.
* If registration is successful message confirmation shall be generated

And sent to user’s mobile number or email.

* The system shall verify users’ information when logging in
* The system shall display error message if incorrect information were used
* After many attempts made the system shall allow the user to create new password
  + - The system shall enable user to navigate within it to edit profile.
    - The system shall enable user to modify profile
    - The system shall update database

1. After changes are saved notification shall be sent to user’s mobile

Number or email.

* + - The system shall display pickup location
    - The system shall display car types in each driver’s profile
      * Display each car’s routes with car fare.
    - The system shall enable administrator to add/delete car details
    - The system shall update database when new changes are made

## Design of system database

The primary purpose of building a database is to provide an environment that is easy and effective to use in information retrieval and storage. The following were considered while developing the database for the University of Zululand: a) Database purpose: The main purpose of this database is to store and retrieve student or administrator information and to create an environment that will allow the student and administrator to easily evaluate their details whenever appropriate. The tables needed to construct the database and the fields needed to create the tables were identified: the fields to be used to create the tables needed were identified and the tables were generated accordingly. The different table that was created.

Table login: This table has been labelled "Personal login”. It displays the students' login information. The user logs in with a username with a TEXT field type and a STRING password with a field type (Tables 1-3).

**1. Personal Details Table: Registration Table**

|  |  |  |
| --- | --- | --- |
| Field name | Field Type | Length |
| 1.id | Int | 11 |
| 2.Firstname | String | 50 |
| 3.Usename | String | 30 |
| 4.Password | String | 15 |
| 5.Retype Password | String | 15 |
| 6.Cellphone | String | 10 |
| 7.EmailAdddress | String | 20 |
| 8.LastName | String | 50 |

**2. notification**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Length** |
| 1.cellphone number | Int | 10 |

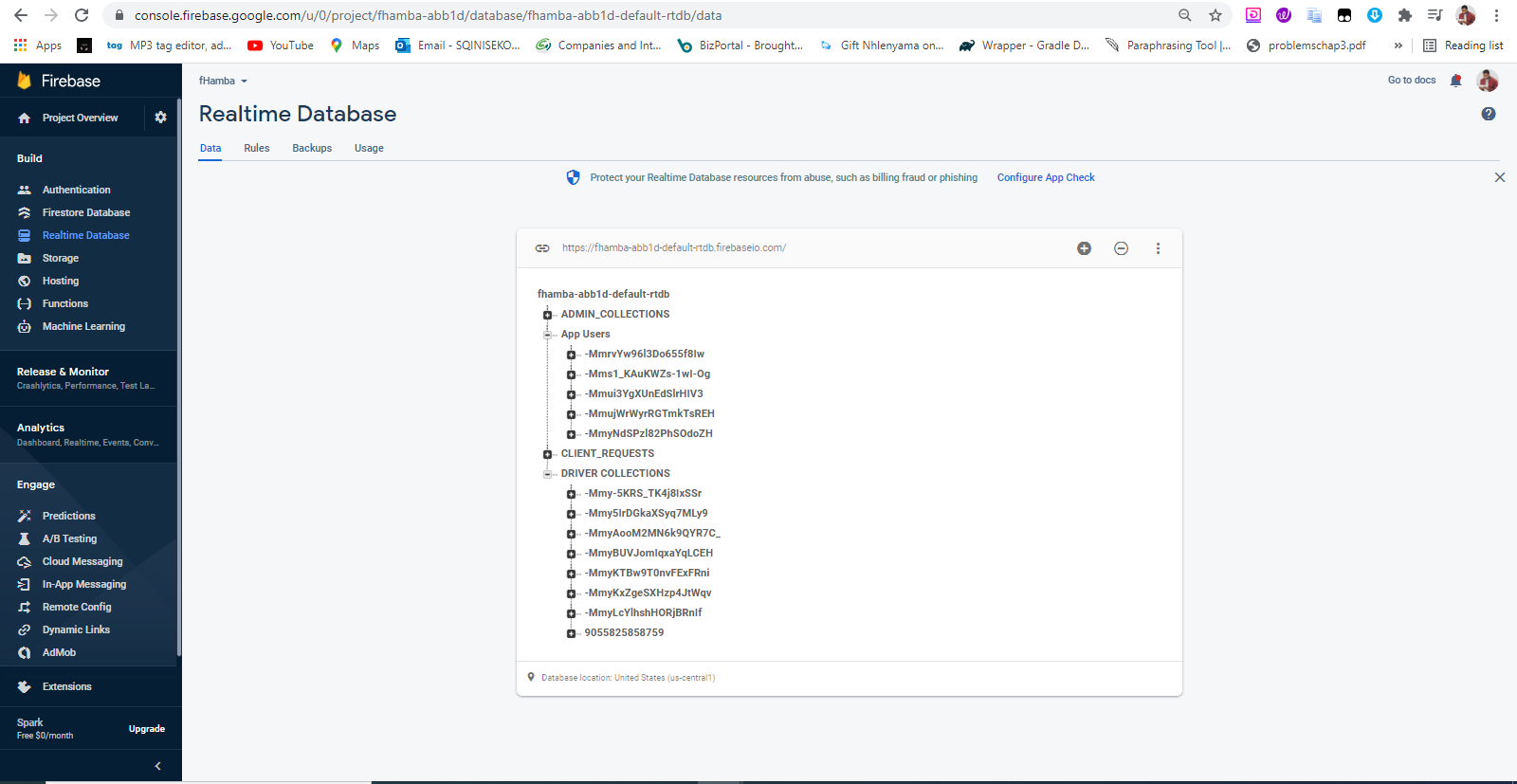


Figure: Database View

## 

## Architecturally Significant Design

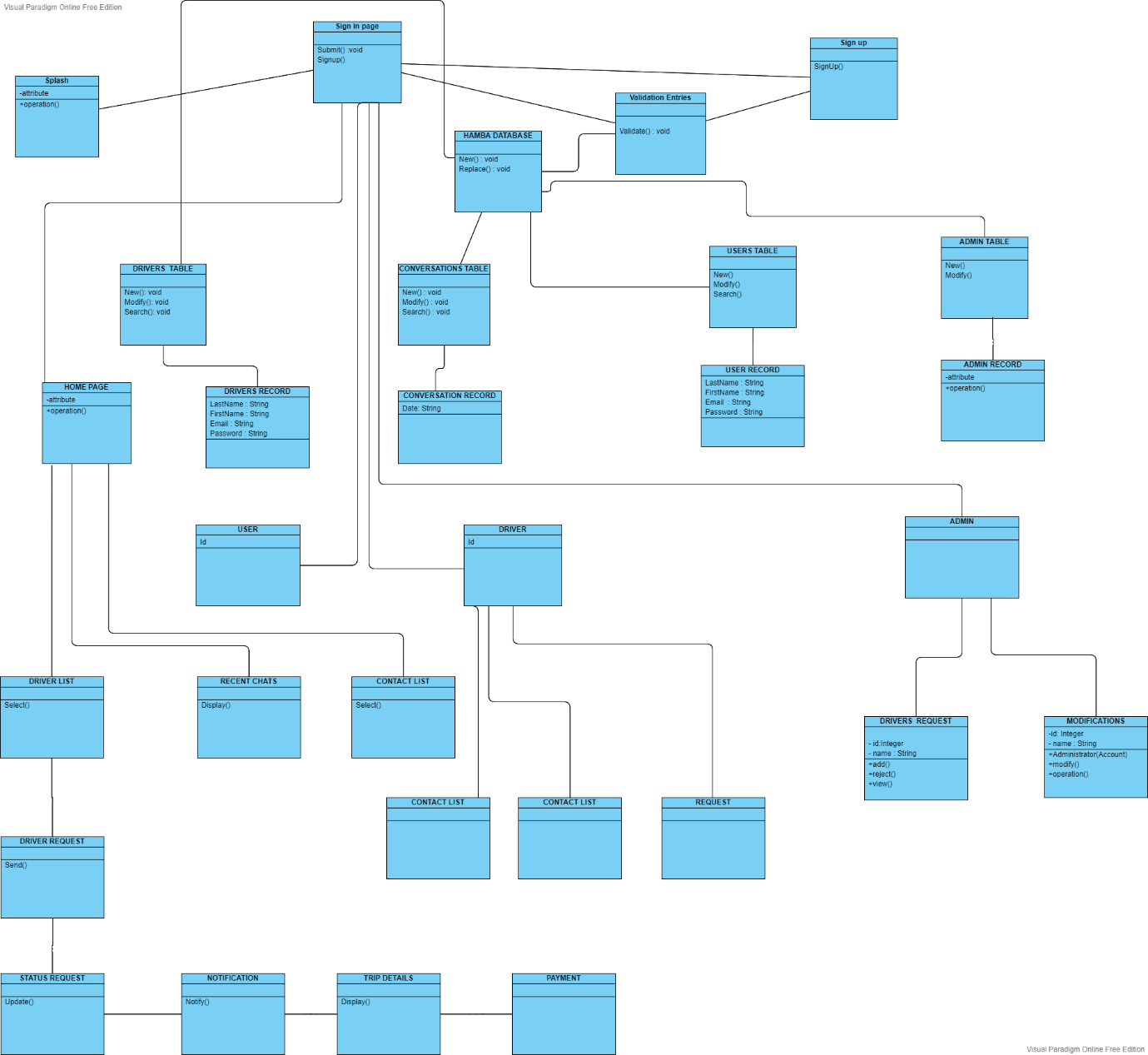


Figure 1 Architecture Design

Splash Screen

Name: Java Activity

Description: This Is the Initial Page the Client upon entering the Application

Attributes: None

Resources: None

Operations:

Name: Intent ()

Arguments: None

Returns: No return value

Pre-condition: Connected to the application

Post-condition: go to the sign in page

Exceptions: None

Flow of Events:

1. The client is presented with the Splash screen
2. The screen is automatically goes to the next screen
3. Client is connected to the sign in page.

**SIGN IN PAGE**

Name: Sign in page

Type: Activity

Description: When the Client clicks the Sign in button, he/she will be presented with a form. The form will be having a list blank field to be completed. All fields will be required. When the form is completed, the Client can submit by pressing the sign in button. If the form is cleared the Client goes to the home page. If the form is submitted the information is saved on the Hamba Server in the Client Database.

Attributes: None

Resources: None

Operations:

Name: Submit ()

Arguments: None

Returns: No return value

Pre-condition: Correct Information

Post-condition: Go to the home page

Exceptions: Wrong info

Flow of Events:

1. The client is presented with the Sign in screen is goes to the next screen
2. Client/ Admin / Driver fills in the page.
3. Client Clicks Sign in.
4. The Client goes to the Home Screen.

Name: Signup ()

Arguments: None

Returns: No return value

Pre-condition: does not have an account

Post-condition: Go to the Sign-up page

Exceptions: Wrong information

Flow of Events:

1. The client is presented with the Sign in
2. Client/Admin / Driver doesn’t have an account
3. Client Clicks Sign up.
4. The System goes to the Sign Up.

### Hamba Database

Name: Hamba Database

Type: Database

Descriptions: The Hamba Database will reside on the Firebase server.

The Client/Admin/Driver Database will be created and maintained in Firebase.

Attributes:

Last Name: String

First Name: String

E-mail Address: String

Receive e-mails: Boolean

Password: String

Operations:

Name: New ()

Arguments: None

Returns: None

Pre-condition: None

Post-condition: A new record is added to the /Admin/ Database Table

Exceptions: None

Flow of Events:

1. The Client is on the added.
2. The Client fields in the fields
3. The Client clicks submit
4. The Departmental server adds the new information as a table in the Client Database

Name: Search ()

Arguments: None

Returns: Record

Pre-condition: None

Post-condition: The Client/ Database has been searched

Exceptions: None

Flow of Events:

1. The Departmental Admin server searches the Client Database for a particular record
2. If the record is found the Departmental Server presents the record to the Client

**Name: Drivers Screen**

Type: Activity

Description: This page is displayed to the client

Attributes: None

Resources: None

Operations:

Name: Select ()

Arguments:

Returns: None

Pre-condition: Signed into the application

Post-condition: goes into the chatting activity with the Driver

Exceptions: None

Flow of Events:

1. Client is presented with a list of drivers.
2. Client Selects one driver
3. Client is goes into one driver

**5.0. Use case realizations**

**Diagram

Description automatically generated**

Figure 2 System Sequence Diagram

Diagram

Description automatically generated

Figure 3 Hamba Sequence Diagram

A picture containing graphical user interface

Description automatically generated

**Figure 4 Entries Sequence Diagram**Diagram

Description automatically generated

Figure 5 Add Entry Sequence Diagram

Diagram

Description automatically generated

**Update Entry**

**4.0. Data structure design**

The data is stored in a relational database using Access. The relations are described by the database administrator for the CIS.

The fields for transmitting to and from the database are given in the following table.

Data field types and sizes.

### Application Layer

The Application level/Layer delivers the business logic and links the Presentation layer to the data bank. The Application level is enclosed within Hamba web services (Firebase server). All interaction with the Presentation level is performed through web services (Google Firebase Web services). Console firebase.

### Data Layer

The Data layer comprises of the Firebase web server Database and the accumulated procedures contained within The Data layer offers continuity for the system and all communiqué is done via Firebase Database queries, stored procedures and views.

### 3.7.3. Presentation Layer

Presentation layer is comprised of a homepage with various login view standards for Students, Driver and Administrator post logging in they have appropriate tasks to achieve and not view all available task options.

Administrator Task

* Admin can register using details that are verified and are the only one who knows the enrolment key for security measures.
* Admin can add, remove, and update Drivers.
* Admin can allow Drivers to be ordered they give the final validation.
* Admin sends SMS notifications to a user.

Student Task

* A student must register and authenticated by the administrator before proceeding.
* Students can search for Drivers they want using the name.
* The student will be given an option to order a book if it is available.
* If the quantity of the book is at 0 then the student will be able to have the reserve option

# **Software Test Plan**

## Purpose

The primary aim of the Hamba (Ride Hailing) System test plan is to address the research specifics of the use cases. The goal, scope and method of the software testing initiative for the project are also defined in the software project test plan. Hamba test plan also defines the workers accountable for each assignment and indicates the risks linked with the test plan.

## **Objectives**

The key goals of the Hamba (Ride Hailing) System test plan are as follows:

* To classify the system's features that will be checked.
* Distinguish and describe all the actions needed to plan for and perform the system's testing process.
* Defining the pass/fail conditions for each object that will be tested.
* describe the testing stage deliverables.
* Characterize any conditions for suspension and techniques of resumption.
* To address the methods of research used for testing Hamba.

## DEFINITIONS

* Many of the terms and concepts associated with the Hamba (Ride Hailing) System's test plan include the following:
* Pass/Fail criteria: Rules of judgment that are used to decide whether a program is used.
* An object passes a test or fails.
* Test: A collection of one or more test cases
* Test Item: A software item that is a research target.
* Test Plan: A document outlining the scope, strategy, sources and timing of the research activities envisaged.
* Test Overview Report: A text that summarizes the events and outcomes of the studies.
* Testing: The procedure of analysing a software item to identify the variations between the conditions that exist and are necessary.

## TEST ITEMS

This section of the test plan lists all the items of the Hamba (Ride Hailing) Structure project that wish to be tested:

• Login

• Requesting of Drivers

• SMS Notifications

• Register

## APPROACH

The overall approach to testing the Hamba (Ride Hailing) System is outlined in this section of the test plan. The methodology taken to test Hamba ensures that the project's key features are properly tested. All the testing will be done with the assistance of the Android studio. The research will be conducted when logging into the system as an administrator or a regular system user on Hamba.

### UNIT TESTING

Unit Testing (UT) is a test that checks every individual software module to test for errors. This is done primarily to uncover errors in Hamba code. The primary objective of the testing of the unit would be to separate each part of the program and to verify the code's correctness. In the case of the Method of Hamba Application Supervision, All the web structures and classes in Java will be reviewed. There remain several advantages to the testing of this unit:

Unit checking allows the modification of the code. It enables testing to be carried out in a bottom-up manner. At the same time, there are certain drawbacks to unit testing, such as it does not detect each device error.

### INTEGRATION TESTING

The specific software modules are integrated and tested as an entire unit in Integration Testing. In general, integration testing approaches unit testing, in which each module is assessed as a separate unit. The main objective of the integration test is to test the functional and efficiency demands on the main project products. All the independently designed project modules will be integrated and evaluated in the integration testing as a whole system.

### REGRESSION TESTING

The Regression Test is normally conducted if changes are made to a project's source code. Regression Testing can also be viewed as the process of testing modifications made to the computer program and ensures that the new modifications are still working with the older programming.

prior to any new version of software being released, with the modifications made, the old test cases for the project will be run against the software to ensure that the old project functionalities still work.

## SYSTEM TESTING

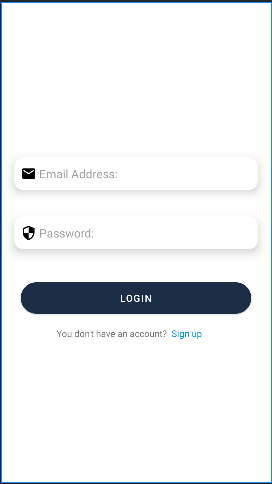
System testing is performed predominantly on the whole integrated system to ensure that all the specifications are met by the project that has been created. The mixture of device and integration tests would be the test case for system testing.

## TEST CASES

The following are the test cases for Hamba (Ride Hailing) System:

### **TEST CASE 1 – Student LOGIN**

* Incorrect input: Incorrect username, which is the username that the user has applied to the database for Hamba.
* Pass Criteria: To imply that an invalid username has been typed, an acceptable message should be produced.
* Correct input: A verifiable username of the user and a valid password associated with the username that he uses to log in will be the correct input.
* Pass criteria: After logging into the framework, the user should be guided to the website that the user is supposed to go to.



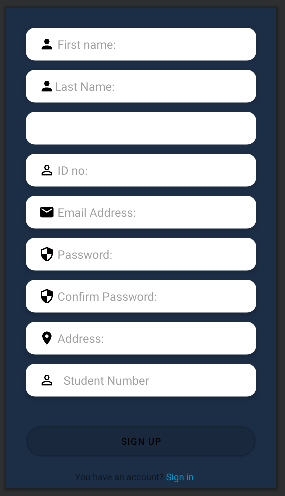
If a Use put correct Login Input

### **TEST CASE 2 – STUDENT REGISTRATION**

* Incorrect data: incorrect format entered in the registration page input fields.
* Pass Criteria: The user should produce an acceptable message indicating that he/she entered the incorrect format in the relevant input field.
* • Correct input: The correct input is the correct format entered by the user in the registration page input fields.

Pass criteria: For this test case, the pass criteria will be a good one.

Student enrolment for the Hamba Application Management Scheme. After this, the scheme will log the user into the system. If the user fails to fill in all the text fields,



A screenshot of a phone

Description automatically generated with medium confidence

**T**

**TION FOUR: NON-FUNCTIONAL REQUIREMENTS**

**SECTION FOUR: NON-FUNCTIONAL REQUIREMENTS**

## Availability

The system should be 99.99% available. This hinges upon on GPS connection, and strength of internet connection through which data can to, or from database. GPS will enable a driver to check pickup location.

## Performance

The application should not take long to respond. However, this will depend on the strength of internet connection. If the server takes longer than expected to respond it should display a message indicating the delay.

## Design constraints

There are some limitations imposed by the hardware on the system. The designed system may be advanced yet limited by the outdated hardware. The system is designed to run on android studio and windows operating systems.

## Portability

The software will be expressed in Java because it a robust object-oriented language. Java is flexible because it can move from one computer to another.

## Reliability

The HAMBA should display the desired outcome. Semantic errors are intolerable. Where numeric values are required, it should accept such data otherwise it should display error message.

## Security

Communication between the system and server should be encrypted so that user’s credentials are not stolen. After many failed attempts made to login to the system access should be denied and the user directed to create a new password. The locking period should be at 120 minutes and the system will be disabled during that duration.

## Maintainability

The application should be flexible for extensibility. It should be open for extension but closed for modification. This means it should support implementation of new functions.

Chapter Five:

Conclusion Introduction

This chapter is based on the findings on the data analyzed in the previous chapter and the findings based on the experiences, future enhancements of the system, and summary and experiences.

Experiences

The project study covers module search, sending the appointment,

, add schedule, set reminder, rating, and help deletion of appointment and schedule. Hamba exposed me into different types of dimensions and my coding skills were tested extremely new experience and memories were achieved.my project crashed one week before the final presentation, and I had to create a new project in one week.

Future enhancements of the system

Voice activated Hamba for blind and amputees

• Implementation of more modern online payment gateway that might help Drivers to generate income faster for themselves e.g., enabling student to pay using MasterCard etc.

• Integrating university registered student database which will be meant for tight security and proper documentation preventing the application from high traffic

That tutorials have their own session per module as it can sometimes be written on the course outline that is the traditional nature of how tutorials are supposed to be hold at the University of Zululand and to other universities. Hamba will be a very paramount software for the university of Zululand students as it will take the advantage of the traditional tutorial session and incorporate it with today’s modern technology this will be done through the entire processes that this app enhanced that met high demands of students which those demands were found through various methodologies. Hamba has been developed in a way that it can substitute the recognized nature of the tutorials that was achieved through the implementation of features and mechanisms that increase a steadfastness, security, performance, and efficiency resulting to the application being resiliently enough to rate amongst the best as written to the objectives and goals thus making the teaching and learning environment beneficial for all. The aim it is to launch of the application, lead the university to improve help/guidance to start new insurgencies and research that will appeal to the eye of the public consequently giving the university credit

**SECTION FOUR: NON-FUNCTIONAL REQUIREMENTS**