**CHAPTER 1 – INTRODUCTION**

**1.1 Background of study**

Taxi Business is as old as Nairobi itself. Before the advent of technology and emergence of taxi hailing, numerous taxi companies were operating in Nairobi Kenya. A taxi, also known as a cab or a taxicab, is a type of vehicle for hire with a driver, used by a single passenger or small group of passengers, often for a non-shared ride. A taxicab conveys passengers between locations of their choice. This differs from public transport where the pick-up and drop-off locations are decided by the service provider, not by the customers, although demand responsive transport and shared taxis provide a hybrid bus/taxi mode.

There are four distinct forms of taxicab, which can be identified by slightly differing terms in different countries:

• Hackney carriages, also known as public hire, hailed or street taxis, licensed for hailing throughout communities.

• Private hire vehicles, also known as minicabs or private hire taxis, licensed for pre-booking only.

• Taxi Buses, also come in many variations throughout the developing countries as jitneys or jeepney, operating on pre-set routes typified by multiple stops and multiple independent passengers.

• Limousines, specialized vehicle licensed for operation by pre-booking

**1.2 Problem statement**

Driving a taxi can be a hazardous job, and many drivers experience assault, robbery, and other forms of violence every day. Violence is often a leading cause of taxi driver deaths, along with motor vehicle crashes. Police officers are also involved in taxi driver’s killings in suspicious circumstances. Harassment from some of their customers who threaten to finish them because they feel like they are being swindled by the taxi drivers in the name of exorbitant prices.

Such cases result in loss of life, children becoming either partial orphans or orphans totally and also this leaves a dent in the economy as these same people do help to build the nation and pay taxes. Furthermore the loss of taxi drivers also affects the transport industry which is a compliment to matatus and buses.

The ride-hailing companies like Bolt, Little cab, etc. do not have any security measures to protect the driver. There are no security measures put in place by the government or unions to protect the taxi industry. This leaves the taxi drivers to trust their instincts and more or less rely on the police to protect them.

**1.3 Aim of the study**

The taxi driver's security system's main aim was to create a platform where drivers would constantly update the activities involved while picking up and driving their customer to their destination.

 Its simplicity was to enable and ensure that the drivers are able to use the system without straining and ensuring fast responses from the police. The interface of the system which only requires registration of the user and the driver can use it at simple touch of the button.

Within a time frame of 6 weeks development of a website that has a backend for the dispatcher and frontend part for the driver was developed. It was aimed to target independent taxi drivers.

**1.3.1 Research objective**

The main objective of this project was to enable the taxi drivers to keep themselves safe with the help of the police. It aimed the driver to conveniently and periodically update his whereabouts, the clients they are driving.

Other objectives were:

* Expand my knowledge base to increase security in the transport industry.
* Keep me up to date with the emerging security measures and trends.
* Build my credibility - This research and project provided me with a good foundation upon which I developed my thoughts and ideas
* Sparks connections - It helped me communicate better with others and convey why I pursue things.
* Encourage curiosity – It helped me spark curiosity as it encouraged me to explore and gain deeper insights into this subject matter. Curiosity leads to higher levels of positivity and lower levels of anxiety.
* Learnt through the project that included the literature review.
* Designed and developed a software.

**1.4 Significance/Justification of the study**

The taxi driver’s security system will benefit private taxi driver owners in that it will help alleviate harassment cases from customers and the police. It is intended to curb the number of deaths and robbery in the transport industry. With this system the drivers will be able to request help and report the threats posed to them in a timely manner and thus reduce the number of deaths and loss cases.

**1.5Scope**

The major stakeholders that were involved in this project were the supervisor, the taxi drivers and me (the developer).

The system entails a web application. The web application will enable the driver to:

* Create an account
* Notify a dispatcher when:
  + When picking up a passenger.
  + If a passenger changes their destination once they’re in the taxi.
  + If driving to a certain area feels potentially unsafe.
* Checking in regularly with the dispatcher.
* Displaying available police stations in Nairobi.

Technologies to be used will entail HTML, CSS, JavaScript, PHP and pivotal tracker to track my progress. All these activities were scheduled to happen in a period of 6 weeks.

The local Kenyan drivers were the main stakeholders in this system.

**1.6 Assumptions**

The main assumption while developing this system was that the drivers are either computer literate or know how to use a smartphone. This entails using the internet and dialing.

The driver must have a roadworthy car allowed by the transportation authorities (NTSA).

Another assumption is that the police and the judiciary system will be ready to cooperate and be able to respond immediately.

**1.7 Limitations**

A major concern for this system to work is to be integrated with the current security policies and protocols in place. As per now locally in Kenya the police system is that most things are done physically and manually. The call free toll system 999 and 911 rarely works and at times the police assume calls and don’t take them seriously to some extent. The response time is also not fast enough and it takes a second for the damage to be done.

Another limitation is that internet connection is required. Data bundles and internet connection is quite expensive which posed a challenge.

**Chapter 2 - Literature Review**

**2.1 Introduction**

This chapter reviewed literature related to the study objectives based on the high rise of insecurity taxi drivers in Kenya. Literature review sought to bring out the existing body of knowledge relevant to taxi drivers security, critic the work and identify the gaps. These gaps motivated the study to advance the body of knowledge on taxi drivers' security.

The purpose of this chapter was to review relevant literature information that was related and consistent with the objectives of the study. Important issues and practical problems were brought out and critically examined so as to determine the current facts. This section was vital as it determined the information that linked the current study with past studies and what future studies will still need to explore so as to improve the knowledge. The following are related work that has been done before on improving the security of taxi drivers.

**2.2 Related systems**

**2.2.1 Little cab**

Little cab is a fully owned taxi hailing technology of Craft Silicon Ltd, a Kenyan Company based in Westlands, Nairobi. (*Little - ride a little better*, n.d.) is a mobile phone app automated transport solution offered on the go for consumers and corporate client. It uses geolocation to be able to track the vehicle being used by the driver.

**2.2.2 Bolt Taxi**

Bolt is an Estonian ridesharing company founded and headquartered in Tallinn, Estonia. (*Bolt | Fast and Affordable Rides*, n.d.) The company develops and operates the Bolt mobile application, which allows people to request a taxi or private driver from their smartphone, as well as electric scooters and food delivery services.

**2.3 Limitations of these systems.**

The named systems aim to serve a purpose that is to transport the passengers. Nevertheless the systems have loopholes which still put the taxi driver at risk and threaten their lives.

One of the limitations in both Bolt and Little cab systems is that it tracks the taxi vehicle. This means that it can only be used to identify the location of the car if it is reported stolen. This only keeps tabs on the car but not the taxi driver. For the vehicle to be stolen it could be that the driver was attacked or drugged. Meaning the driver is always vulnerable.

Another limitation is that in case of any altercations or an attack between the driver and the passenger there is no real time reporting of the incidences. The incidences are reported after they happened which might be too late.

**2.4 Proposed solutions to these limitations.**

With the given limitations I aimed to solve some of these problems with the project where there will be a second party involved to enhance the driver’s security.

The main purpose of this project was the driver to constantly report or inform a dispatcher, preferably someone in the police of all the activities that the driver is involved in. It will involve picking passengers to the time they call it a day and head back home to their families.

In addition to that every time the driver picks up a passenger, they have to inform a dispatcher and they will pass on the details about the passenger for future references and for the security of the driver. In case the passenger wants to change their destination the driver has to relay the same information to the dispatcher.

The system will be a plug and play type of system. After the driver creates an account they can start using the system. It has have a simple interface and with the help of the dispatcher on call they will be able to navigate the system easily without interruptions on their daily activities.

A distress button will be available in the system to be used for emergency purposes. This may be involved when the driver is being attacked or is being held at gunpoint. Simply put if the driver’s life is in danger the distress button comes to play. When they press this button the dispatcher on the other end will immediately inform the authorities and thus effect an immediate response to save the driver.

**Chapter 3 – Methodology**

**3.1 Introduction**

This chapter described the methodology that was used in developing the Taxi driver’s security system. It entails description of tools and methods that were used in achieving the objectives of the proposed system.

**3.1.1 System development life cycle**

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. When it comes to software development SDLC is the process that takes place. It consists of a phase that system developers follow during the process of system development. There are different methods of system development life cycle and for this I used agile methodology.

**3.2 Methodology**

**3.2.2 Agile Methodology**

The term "agile methodology" refers to a practice that encourages continuous development and testing throughout the project's software development lifecycle. Unlike the Waterfall paradigm, both development and testing operations are contemporaneous under the agile model of software testing.

An agile methodology is an iterative approach to software development. Each iteration of agile methodology takes a short time interval of 1 to 4 weeks. The agile development process is aligned to deliver the changing business requirement. It distributes the software with faster and fewer changes.

The single-phase software development takes 6 to 18 months. In single-phase development, all the requirement gathering and risk management factors are predicted initially.

The agile software development process frequently takes the feedback of workable products. The workable product is delivered within 1 to 4 weeks of iteration.

**Roles in Agile**

There are two different roles in an Agile methodology. These are the Scrum Master and Product Owner.

**1. Scrum Master**

The Scrum Master is a team leader and facility provider who helps the team member to follow agile practices, so that the team member meets their commitments and customers requirements. The scrum master plays the following responsibilities:

* They enable the close co-operation between all the roles and functions.
* They remove all the blocks which occur.
* They safeguard the team from any disturbances.
* They ensure that Agile Inspect & Adapt processes are leveraged correctly which includes
  + Planned meetings
  + Daily stand-ups
  + Demo
  + Review
  + Retrospective meetings, and
  + Facilitate team meetings and decision-making process.

**2. Product Owner**

The Product Owner is one who runs the product from a business perspective. The Product Owner plays the following responsibilities:

* He defines the requirements and prioritizes their values.
* He sets the release date and contents.
* He takes an active role in iteration and releasing planning meetings.
* He ensures that the team is working on the most valued requirement.
* He represents the voice of the customer.
* He accepts the user stories that meet the definition of done and defined acceptance criteria.

**3.2.3 Justification of Agile Methodology**

Agile approach isn't only a set of rituals or a set of development techniques. Rather, it is a collection of approaches that demonstrate a commitment to continuous development and tight feedback loops.

To meet targeted delivery time frames, this methodology works in iterations, with each iteration lasting 10 to 15 days. The original Agile Manifesto, however, did not specify a two-week iteration timeframe or an optimum team size.

This methodology involves tremendous awareness of the project which enables catching and addressing issues quickly.

**3.3 Data collections**

Data collection is the process of gathering, measuring, and analyzing accurate data from a variety of relevant sources to find answers to research problems, answer questions, evaluate outcomes and forecast trends and probabilities.

This project is highly dependent on data, which underscores the importance of collecting it. Accurate data collection is necessary to ensure quality assurance and keep research integrity.

**3.3.1 Data collection Methods**

Data collection methods can be broken down into two methods.

1.      Primary – This is original, first-hand data collected by the data researchers.

2.      Secondary data is second-hand data collected by other parties and already having undergone statistical analysis. This data is either information that the researcher has tasked other people to collect or information the researcher has looked up.

The main primary data collection methods in this project entailed:

* Interviews – I asked questions from a large sample of people by means of direct interviews.
* Focus groups – It involves anywhere from a half-dozen people, led by a moderator, brought together to discuss the issue.
* Questionnaires – respondents got a series of questions, either open or close-ended, related to the matter at hand.

The main secondary Data collection methods in this project will entailed: The internet

**3.3.2 Data collection tools**

Data collection tools refer to the devices/instruments used to collect data, such as a paper questionnaire or computer-assisted interviewing systems. Tools involved in this project were:

* Word Association - Respondents will received a set of words and asked what comes to mind when they heard each word.
* In-Person Surveys – I asked questions in person.
* Observation – It was effective in small-scale situations

**3.3.3 Data sets**

A data set is a collection of data. The main sources of data sets will be from the internet which will include blogs and journals. Open police were also used.

**References -** *Extrajudicial Killings in Kenya*. (n.d.). Www.opensocietyfoundations.org. Retrieved January 3, 2022, from <https://www.opensocietyfoundations.org/voices/extrajudicial-killings-kenya>

*Police executions push Kenya to dark days*. (2016, July 18). New Internationalist. https://newint.org/features/web-exclusive/2016/07/18/police-executions-push-kenya-to-dark-days/

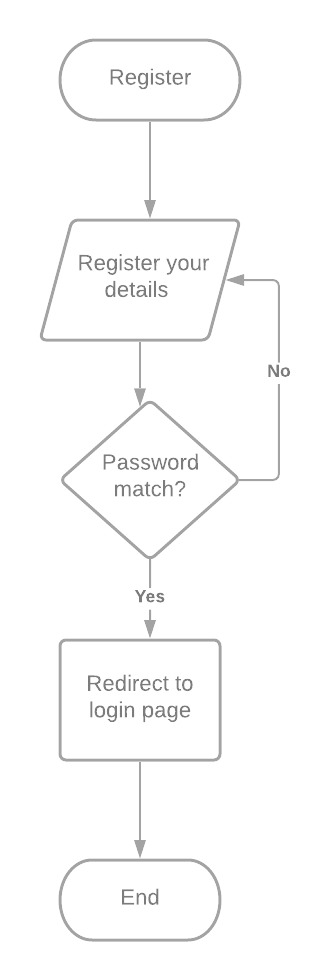
**Chapter 4 System Analysis**

**4.1 Detailed analysis of current system.**

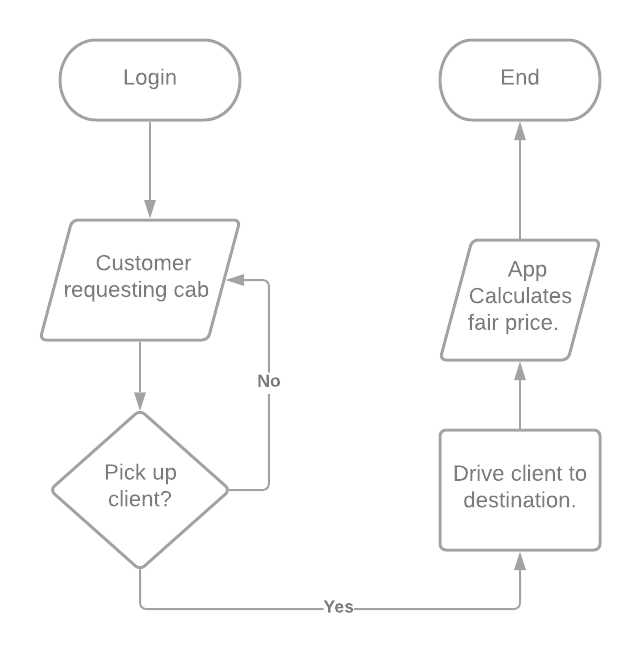
The essential components of an Information System, how data moves within the system, user-interaction points, and the Authorization Boundary are all captured in the Information System/Data Flow Diagram.

**4.1.1 Flowchart**

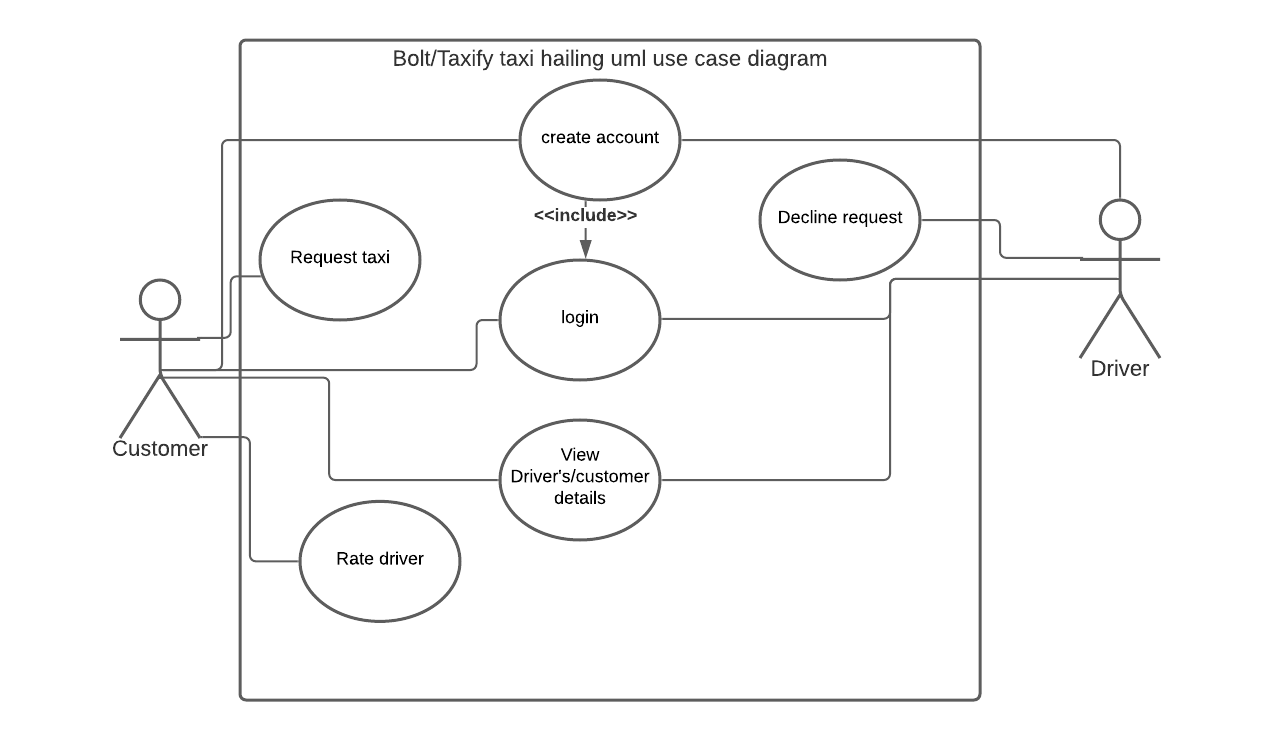
Registration flowchart

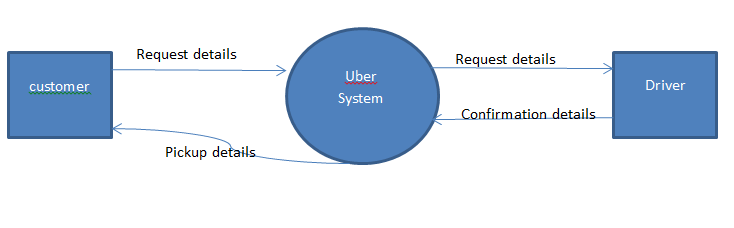
****

Picking up client procedure



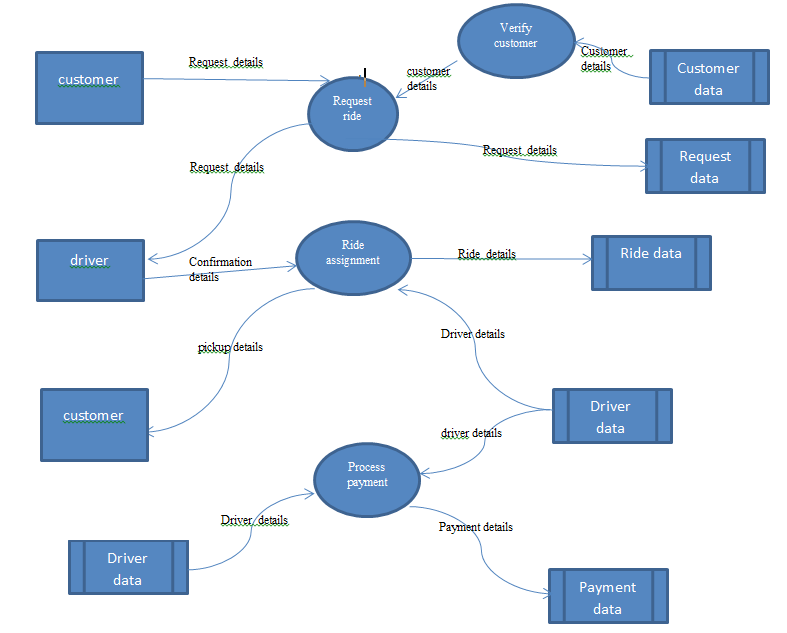
**4.1.2 Use case Diagram**

****

4.1.3 Context diagramlevel 0 DFD

System

4.1.4Flowdiagram



**4.2 System requirements**

The features and behavior of a system or software application are described in system requirements. It consists of a number of parts that seek to define the customer's intended functionality in order to please their various consumers. In addition to stating how the system should act, the specification also describes at a high-level the core business processes that will be supported, what simplifying assumptions have been made and what critical performance parameters will need to be met by the system.

**4.2.1 Functional requirements**

A Functional Requirement (FR) is a statement that describes the service that the software must provide. It refers to a software system or a component of one. A function is nothing more than the inputs, behavior, and outputs of a software system. A calculation, data manipulation, business procedure, user interaction, or any other unique functionality that defines what function a system is likely to execute can all be considered. Functional Requirements are also known as Functional Specification in Software Engineering. A Functional Requirement can range from a high-level abstract explanation of the sender's need to detailed mathematical functional requirement specifications in software engineering and systems engineering. Functional software requirements assist you in capturing the system's intended behavior.

The driver should be able to:

1. Register his/her credentials in the system.
2. Log in to the system.
3. See and change their profile details.
4. Call the police directly.
5. Interact with the dispatcher.
6. Fill in details when they pick up their passengers. This includes:
   1. Picking up a passenger.
   2. Fill in change of destination.
   3. Press the distress button.
   4. Press the arrived button.

The dispatcher should be able to:

1. View and update his profile
2. See the registered drivers in the system.
3. Interact with the driver.
4. Call the police.
5. Acquire driver’s locations.

**4.2.2 Non-Functional requirements**

The limits or requirements imposed on the system are known as non-functional requirements. They define the software's quality attribute. Scalability, maintainability, performance, portability, security, and dependability are all examples of non-functional requirements. Non-Functional Requirements address critical software quality issues.

The system needs:

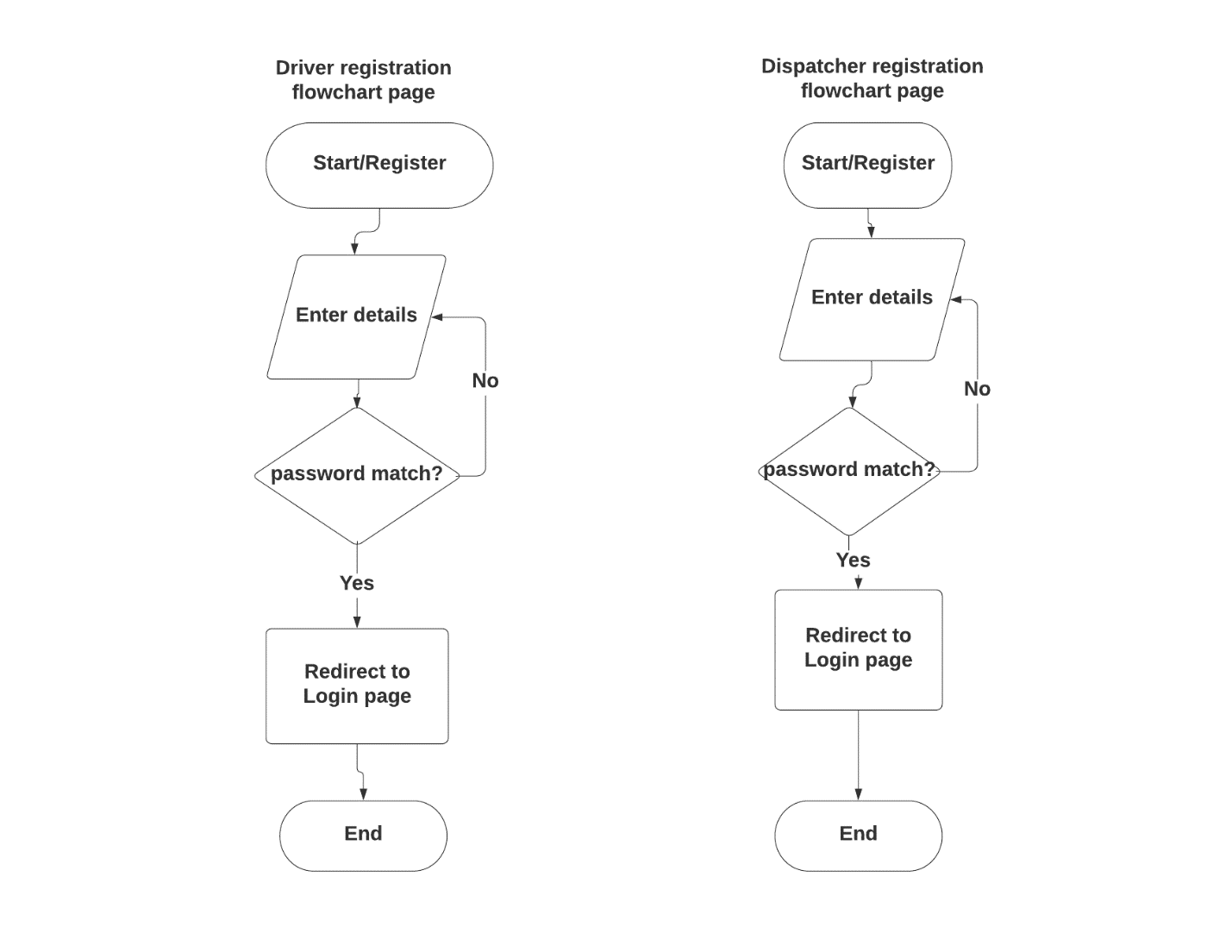
* A browser
* A computer with preferable with over 2gb ram
* A database host
* Access to google maps API
* Database security
* Working internet connection or locally hosted server XAMMP server.

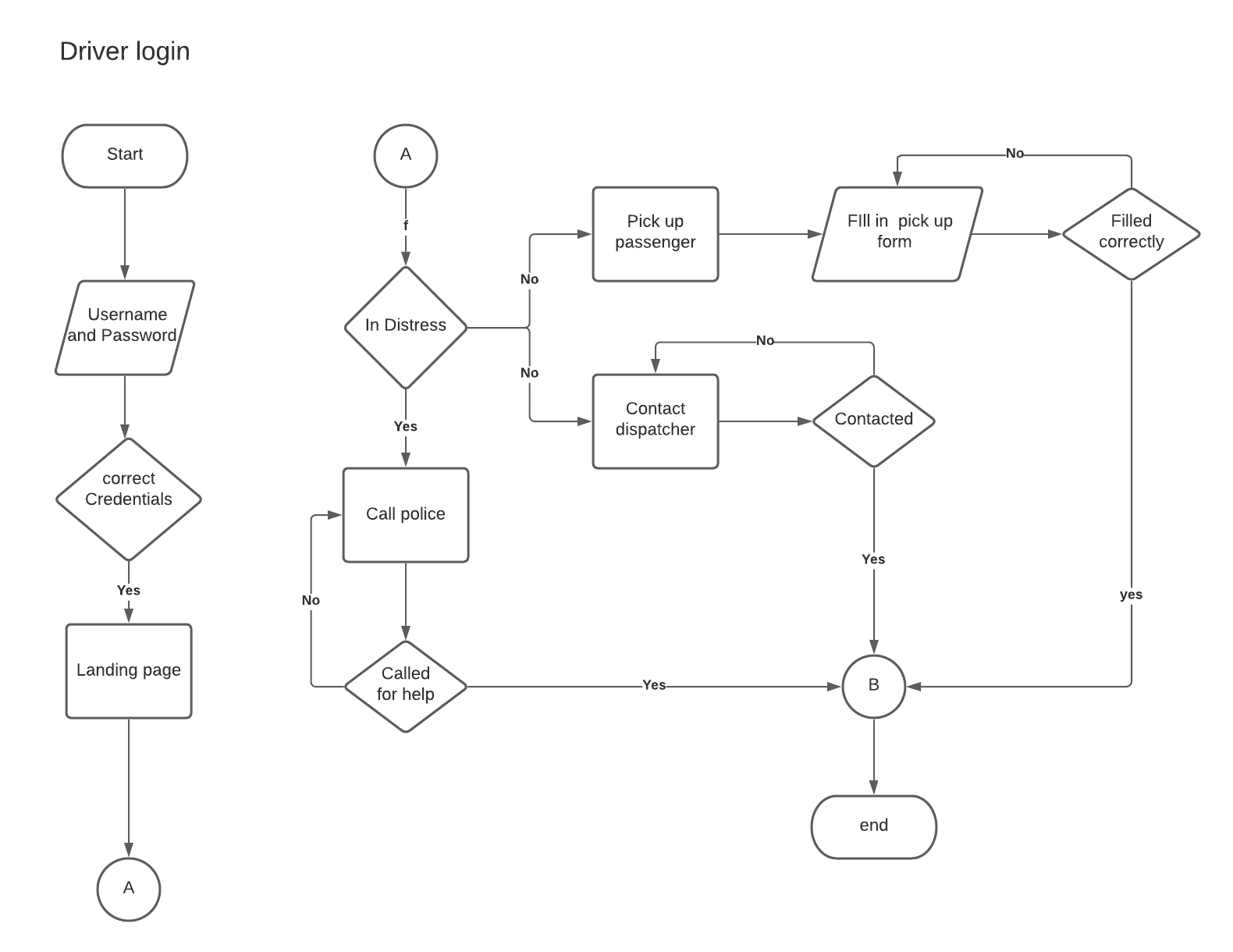
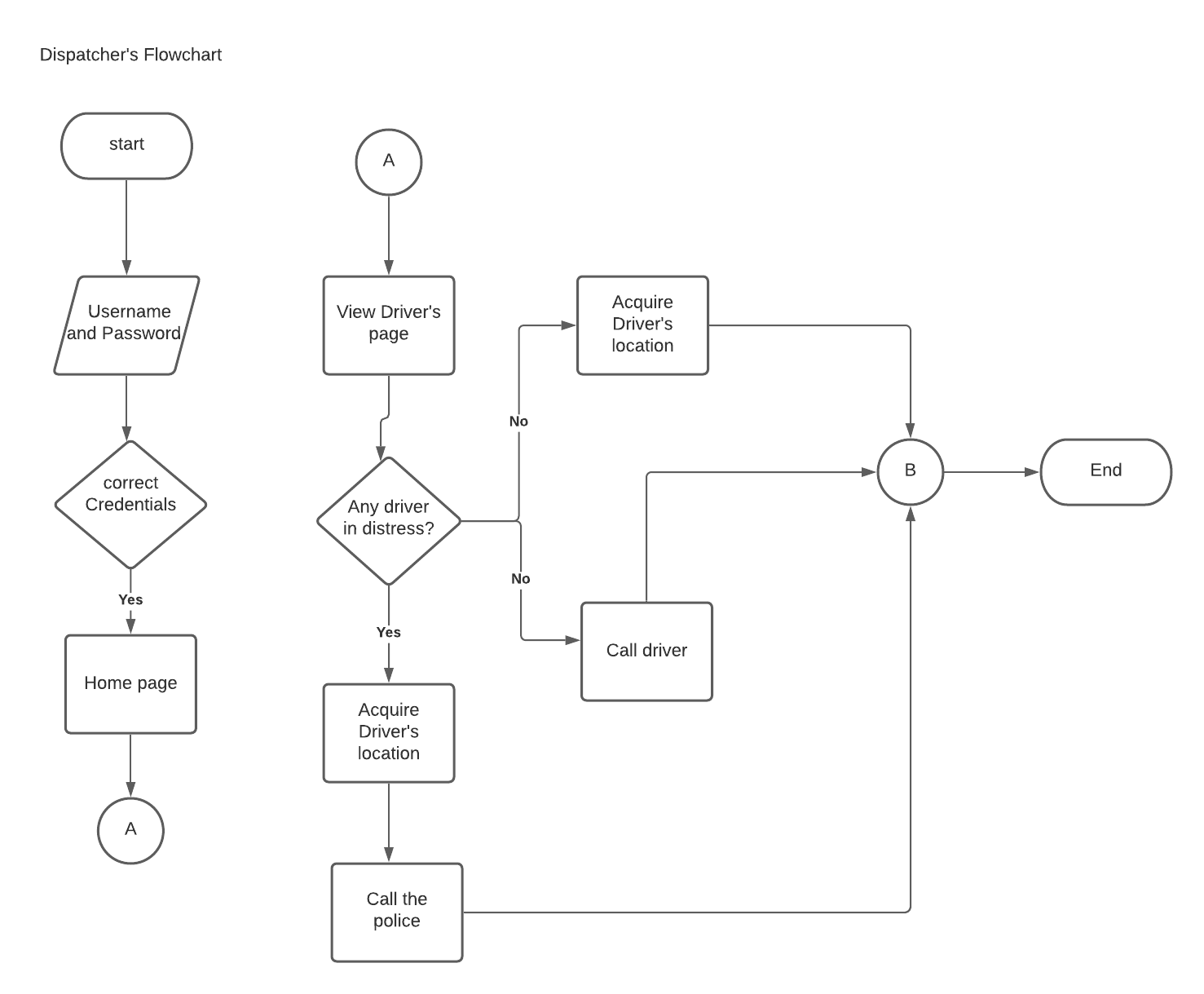
**CHAPTER 5 SYSTEM DESIGN**

**5.1 Architectural Design**

**5.1.1 Flowchart**

Flowchart is a diagrammatic representation of sequence of logical steps of a program. Flowcharts use simple geometric shapes to depict processes and arrows to show relationships and process/data flow.

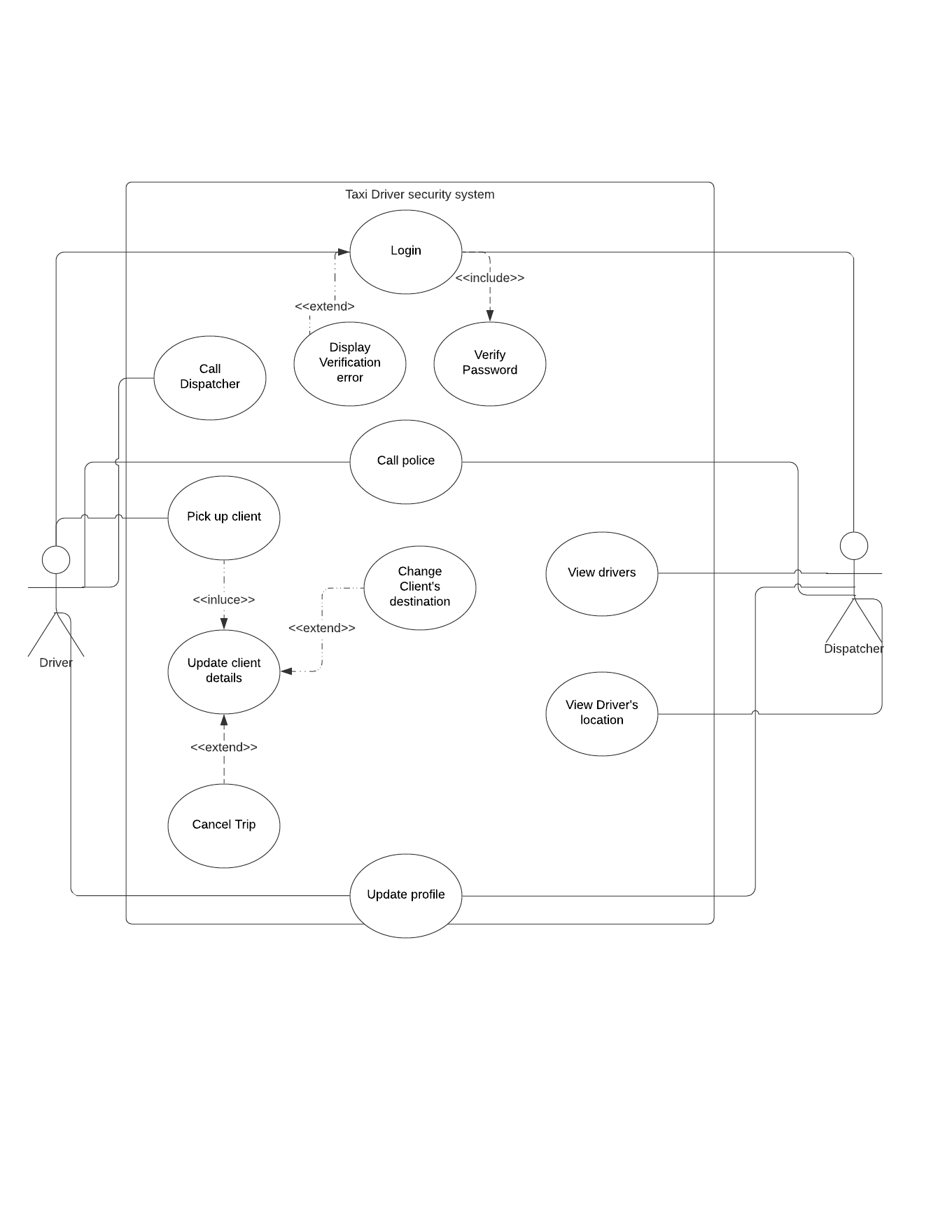


****

**­­­­­**

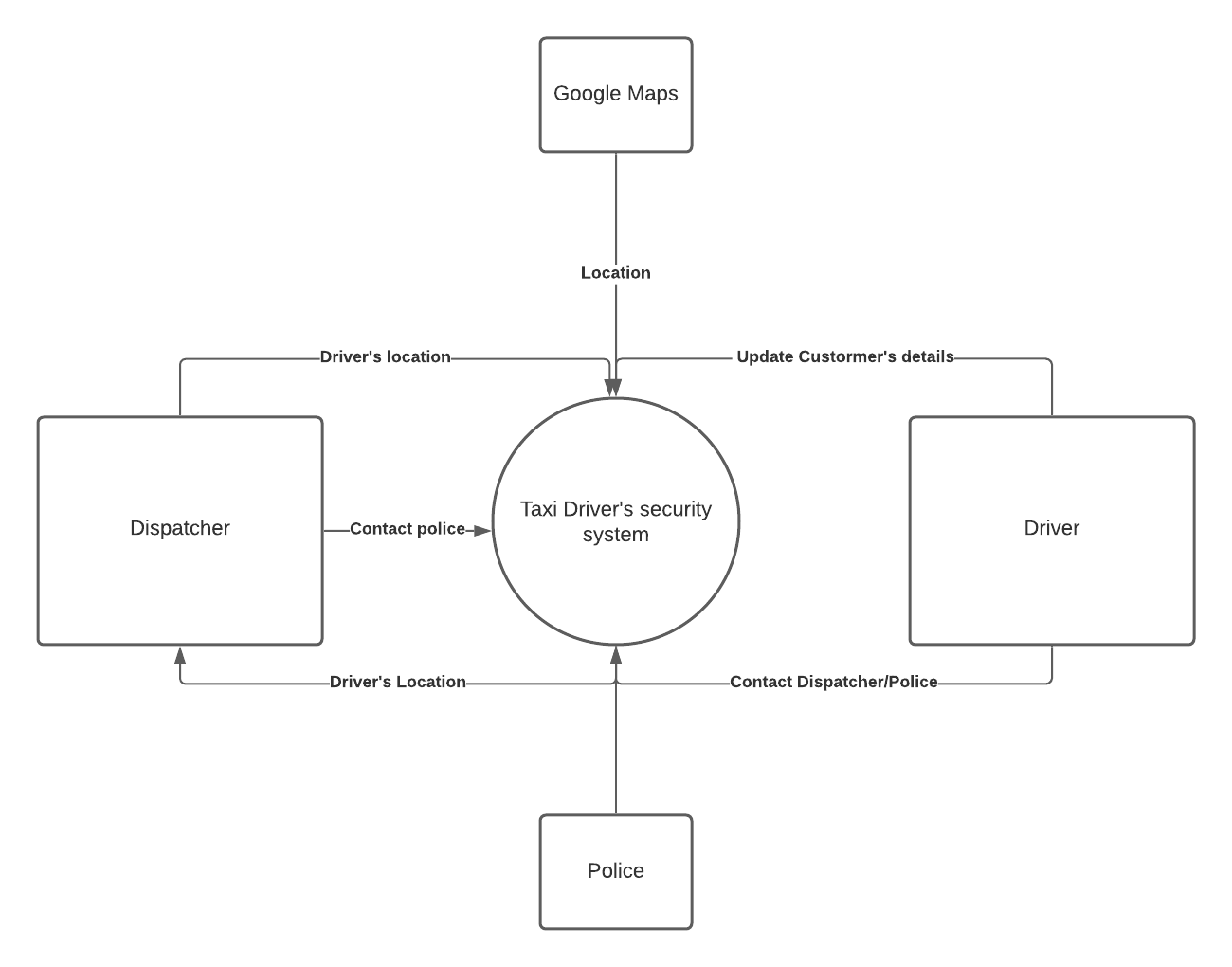
**5.1.2 UML Use case diagram**

The primary form of system/software requirements for an undeveloped software program is a UML use case diagram. The intended behavior (what) is specified in use cases, not the actual technique of achieving it (how).

****

**5.1.3 Context diagrams**

The Context Diagram depicts the system in question as a single high-level process, followed by the system's connection with other external entities (systems, organizational groups, external data stores, etc.).

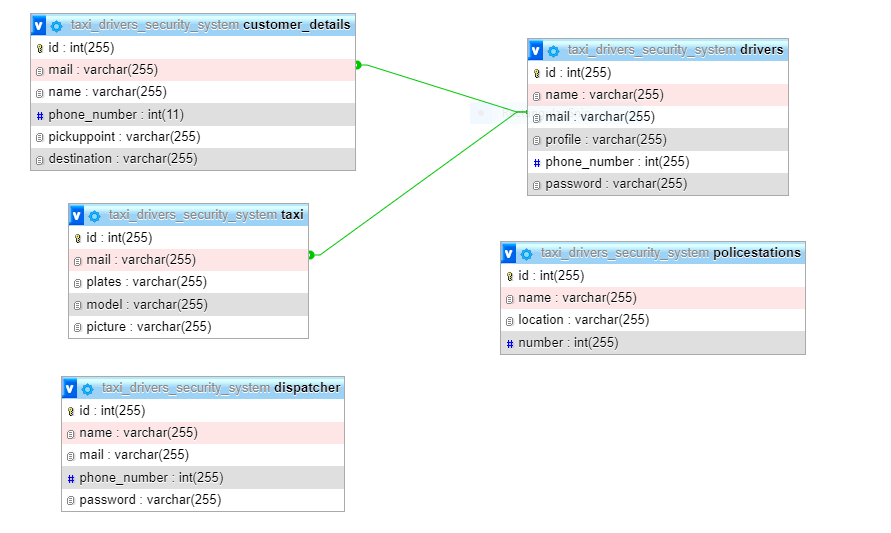


**5.2Database Design**

**5.2.1 Database Tables’ Names and Description**

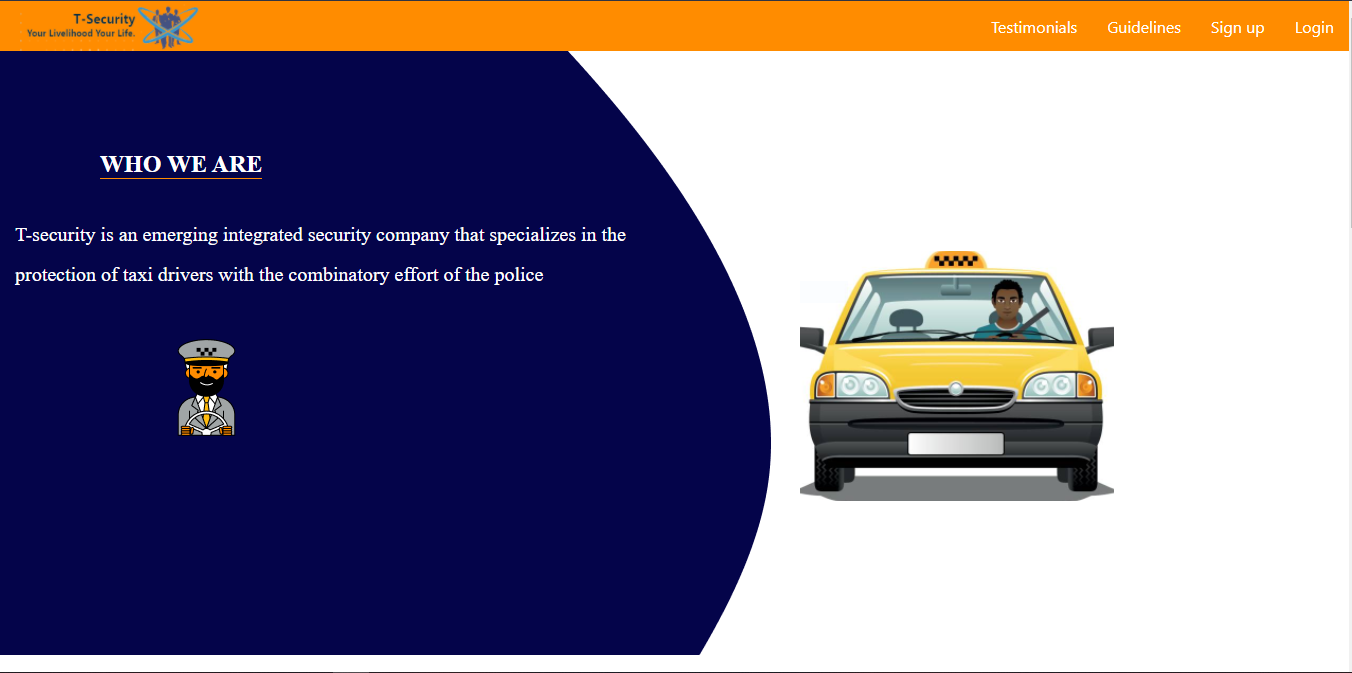
|  |  |
| --- | --- |
| **Table Name** | **Table Description** |
| Driver | Stores information about the registered drivers. |
| Taxi | Stores the information of the taxis owned by the drivers. |
| Police station | Stores information about all police stations integrated in the system |
| Dispatcher | Contains responder’s details. |

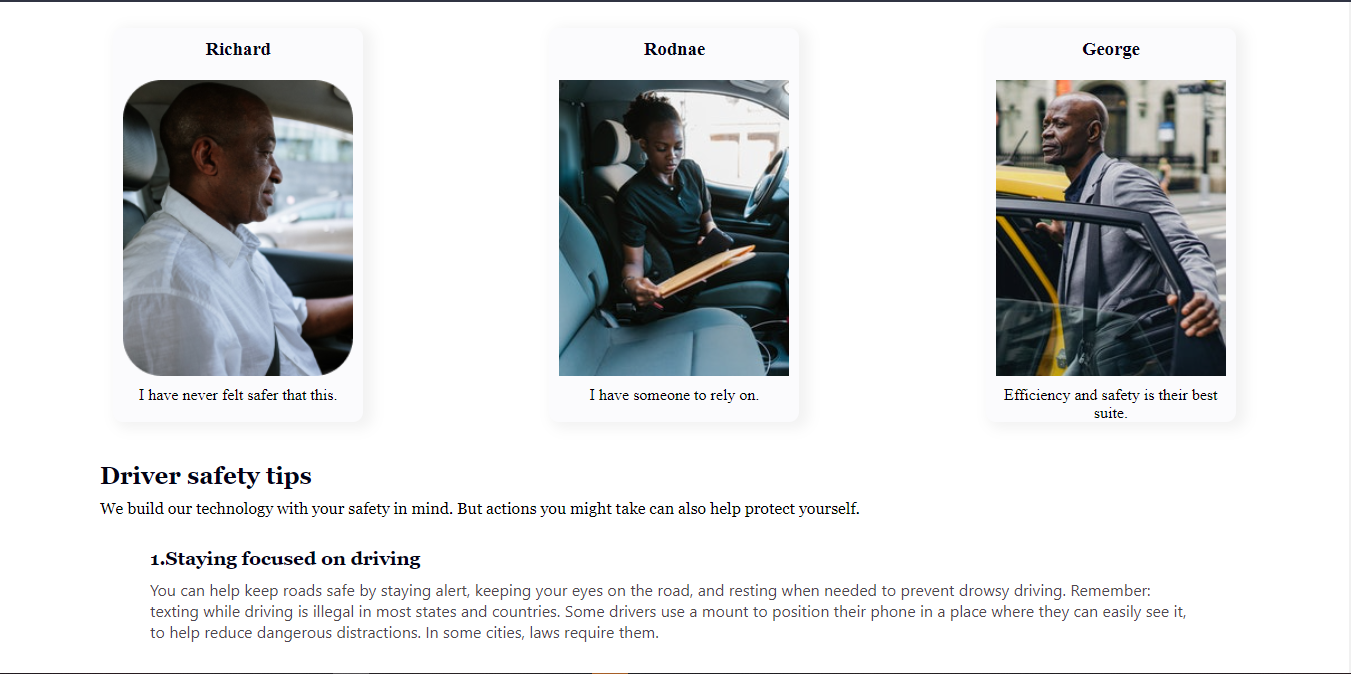
**5.2.2 Entity Relationship Diagram**



* 1. **User Interface Design**
     1. **Landing page**

The landing page highlights a couple of testimonies and gives a brief overview of what the system does.





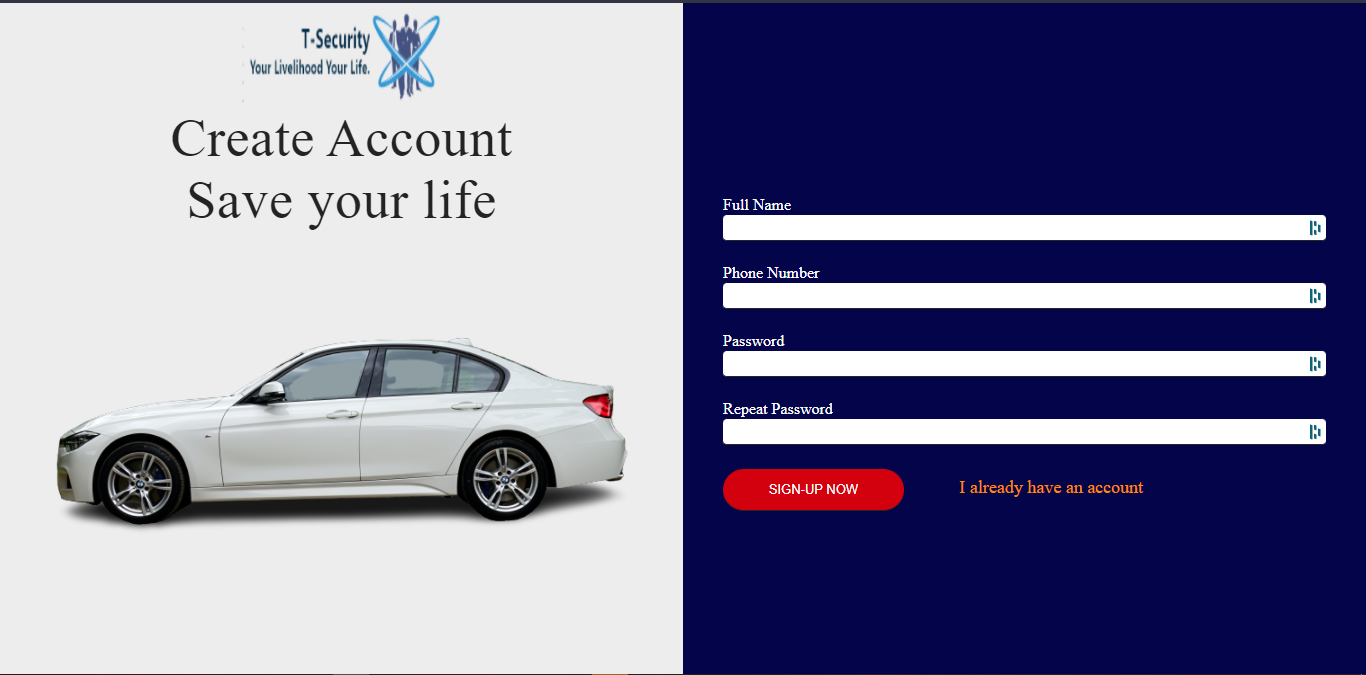
* + 1. **Sign up page**

The sign up page offers two options for either to register as a dispatcher or a driver.

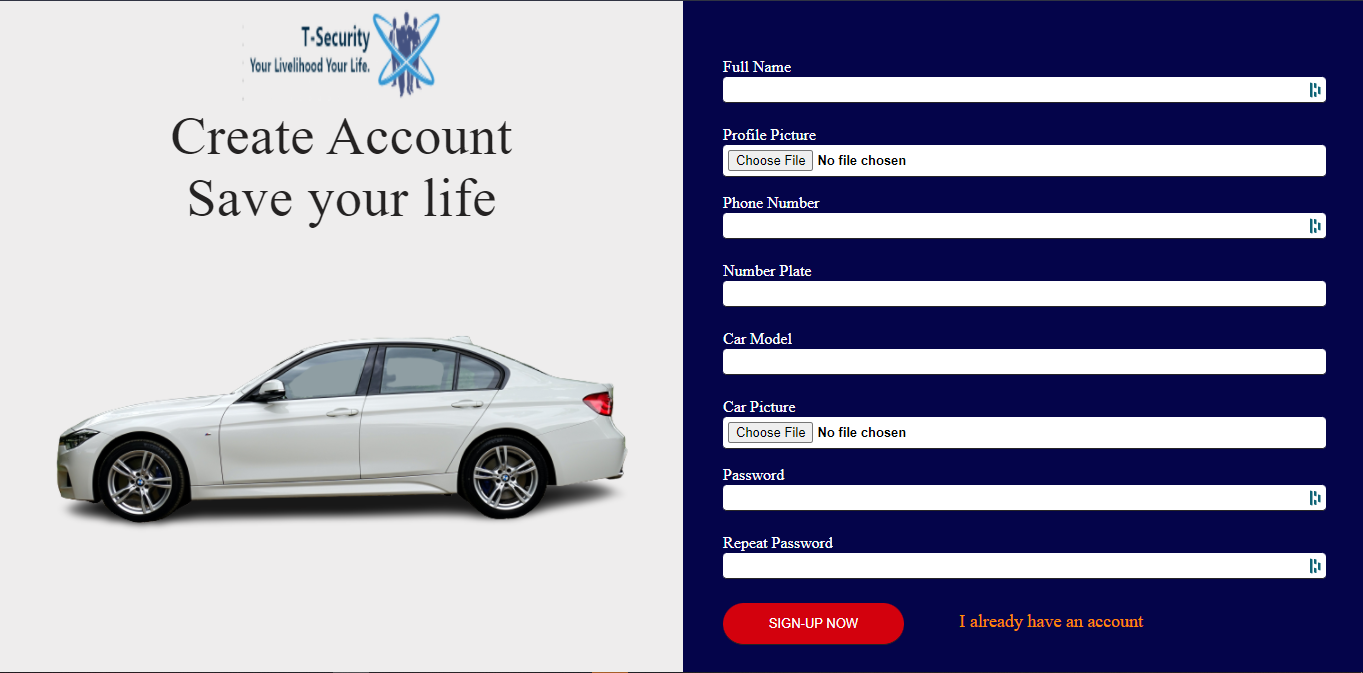


* + - 1. **Dispatcher sign up page**

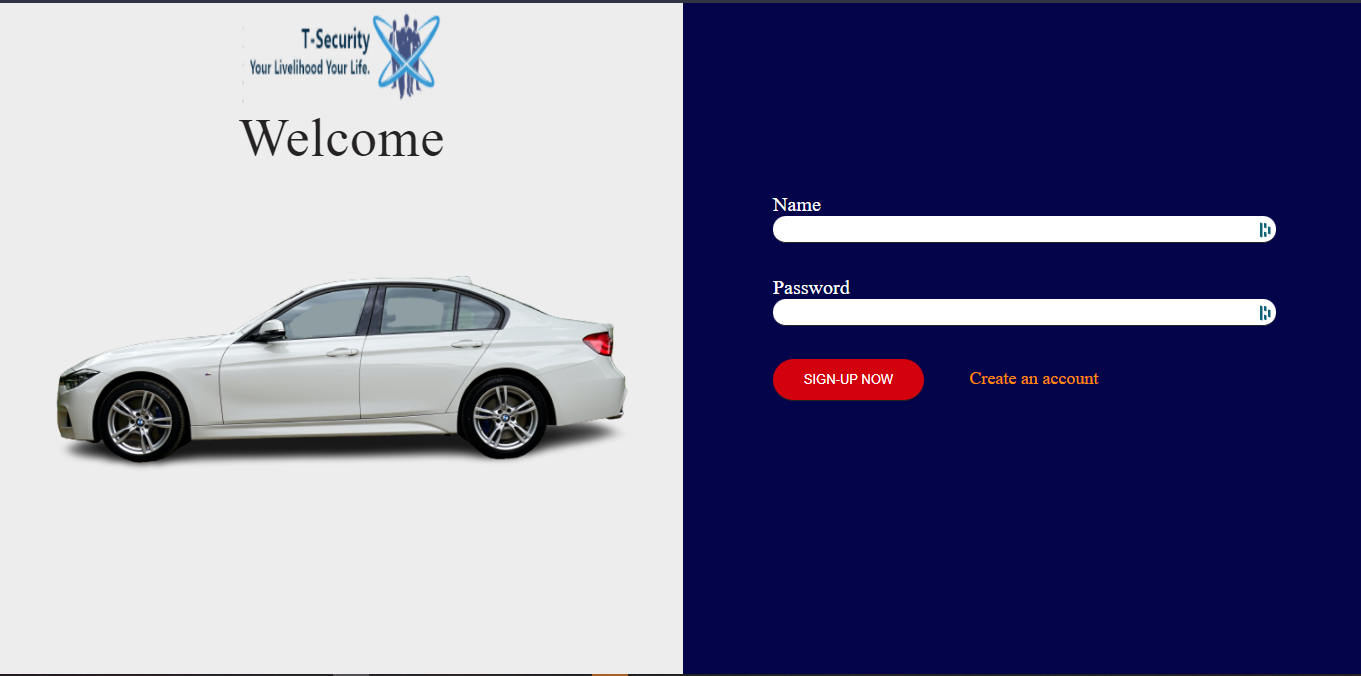
This page is used to register new dispatchers.



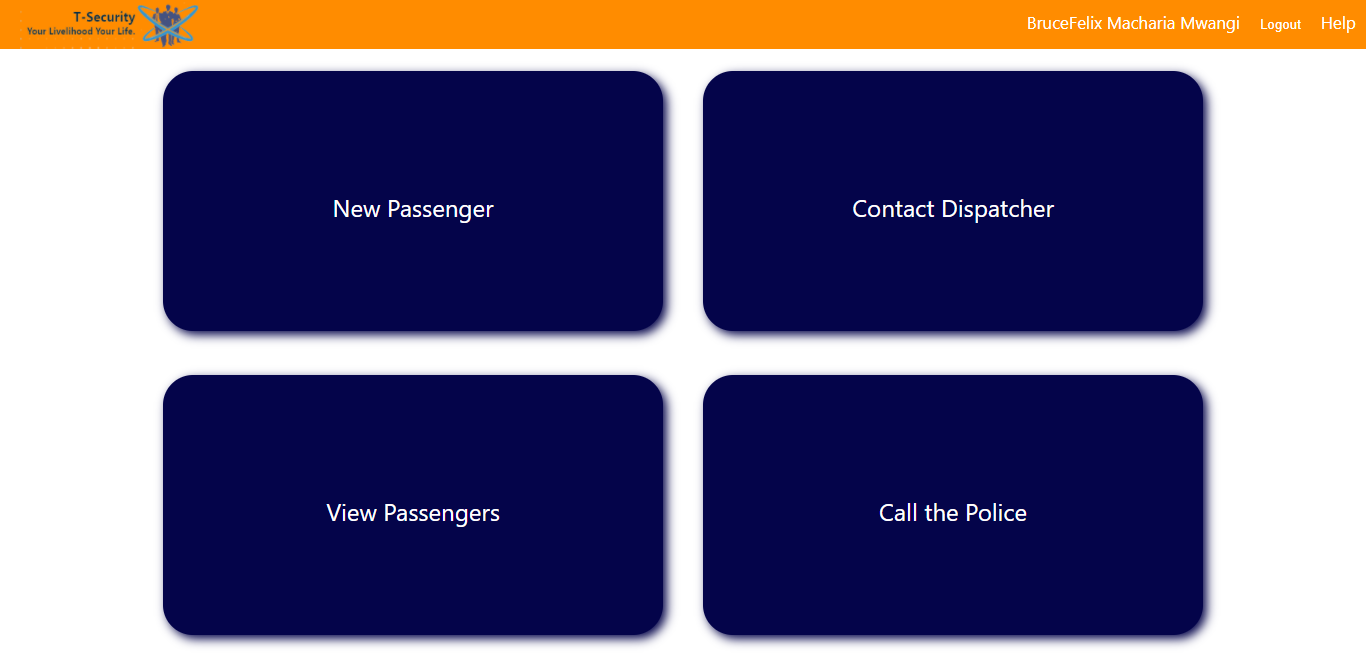
* + - 1. **Driver’s sign up page**



* + 1. **Login page**



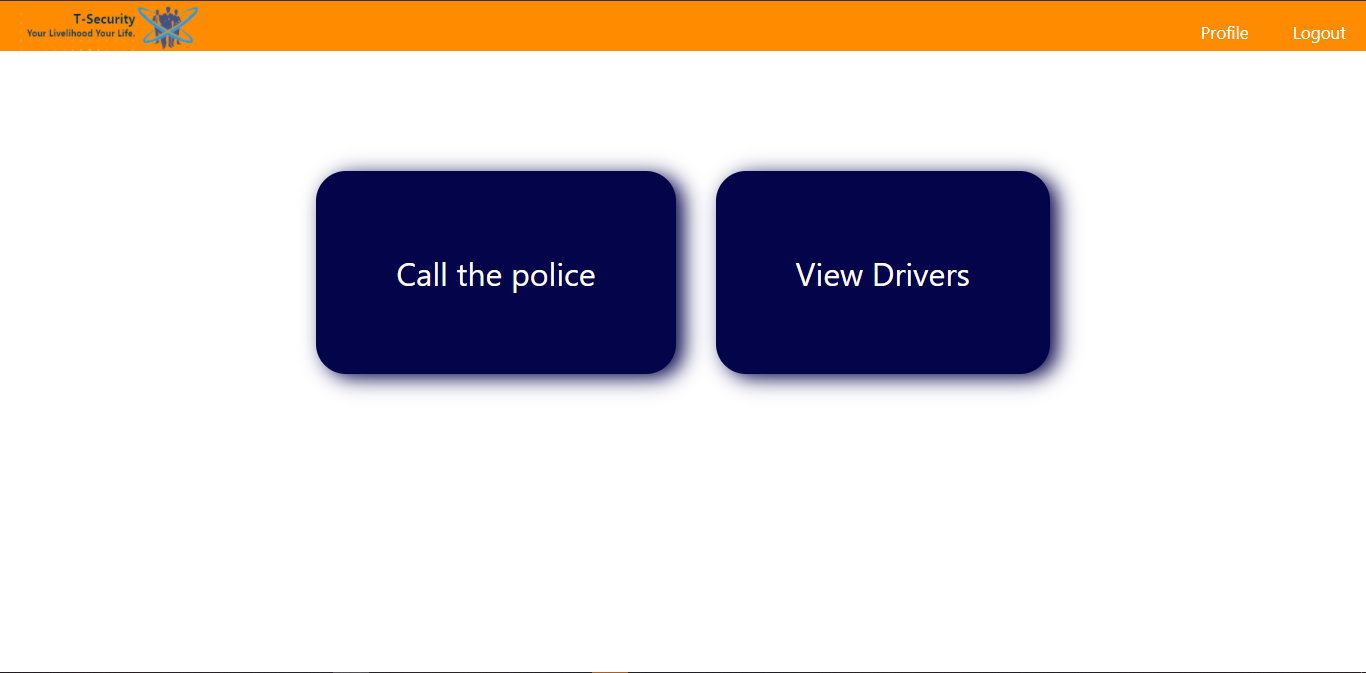
* + 1. **Drive’s page**



This page is used by drivers to add a new passenger details, contact a dispatcher and call the police.

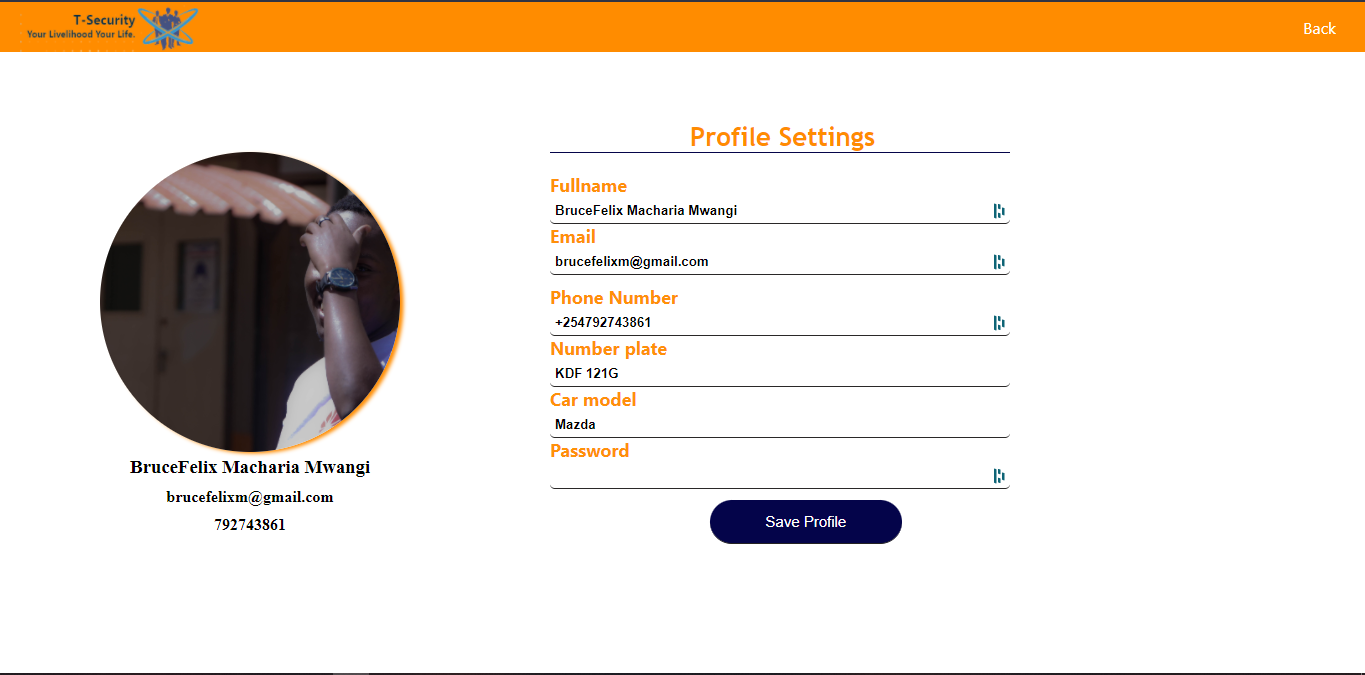
* + 1. **Dispatcher’s page**

This page is used by the by the dispatcher to call the police and view registered driver’s in the system.



* + 1. **Profile pages**

This page is used by both the dispatchers and drivers to update their details.



**CHAPTER 6: IMPLEMENTATION AND TESTING**

**6.1 Development environment.**

The main tools of development tools were: Visual studio code, Xampp control panel v3.3.0 and brave browser.

On the client side I used HTML, CSS, JAVASCRIPT technologies that assisted in designing and making the user interface. In addition Figma was also used in designing the outlook of the user interface before implementation started.

MySQL and PHP were the main technologies in implementing the backend server side. Google maps API will is also used in backend. Xampp also play a big role in providing apache servers locally for the development phase.

**6.2 System components.**

**6.2.1 Landing page**

The taxi driver’s security system has a landing page that gives a brief overview of the purpose of the system. On the navigation bar there is a testimonial link, a guidelines link, a sign up and login link. The testimonial section gives a number of testimonies from drivers whom have benefited from the system. The guidelines highlight some rules or advice to observe on the roads to protect the drivers from the dangerous people out there. The sign up page takes the users to a registration form depending on if the new user is a driver or a dispatcher. In case the user already has an account they can login in immediately.

**6.2.2 Dispatcher’s signup page**

This module allows for a dispatchers to register for an account in the system. The dispatcher needs to enter their basic details which are the Name, E-Mail, password, phone number and a confirmation password. On successful registration, the user is automatically redirected to the login page.

**6.2.3 Driver’s signup page**

The page allows new drivers to sign up and be monitored by the system. Details required in the registration form are the driver’s name, email, profile photo, phone number, number plate, car model, the car picture and password. After the details are verified by the server side he is redirected to a login page.

**6.2.4 Driver and Dispatcher login page**

These pages will take user’s username and password and after querying the database for validation of the credentials, it redirects a user to either the dispatcher’s homepage or driver’s homepage depending on the type of user.

**6.2.5 Driver’s homepage**

This is the main page for drivers where they have a navigation bar with a link to their profiles and a log out button. In addition, on the page there are three modules: the new passenger, Contact dispatcher and call the police. The new passenger module is used to capture the customer details that used the taxi services and the details are stored in the database and can be retrieved when needed. The contact dispatcher module and police module enables the driver to call dispatcher or police depending on the proximity of the driver to a police station for assistance.

**6.2.6 Dispatcher’s module**

The dispatcher’s home page has only two modules. One is the police module, which can be used to call the police in case of an incident. The view driver’s table is used to view the registered drivers in the system and view their details. It also has a navigation bar that has two links one to the dispatcher’s profile and a log out button.

**6.2.7 Driver’s and Dispatcher’s profile page.**

This page is used to by both the driver and dispatcher to edit their details. The dispatcher can edit their name, email, phone number and password. The Driver can edit their own details and the taxi they have.

**6.3 Test plan**

**6.3.1 Test data**

Test data will include user’s registration details, this includes the dispatcher and driver details. The details entail username, email, and phone number for both driver and dispatcher. While the car details will include the vehicle type, number plates and the image of the car. In addition to user details, their geolocation will also be used to determine the nearest police stations around their area.

**6.3.2 Test cases**

Cases will include if the user will be able to register in to the system. It will also include trying to obtain the location of the user using google maps.

**6.3.3 Test results**

The users were able to register successfully into the system. Their details were stored in the database and could be retrieved when needed. Geolocation is also working. The area of interest which is Nairobi could show the police stations available and their contact details.

**Chapter 7 Results & conclusion**

**7.1 Achievements and lessons learnt**

I was able to develop an interface for the system which would be used by the driver and dispatchers to communicate and give feedback on their progresses. I developed a web page with driver’s pages and dispatcher’s pages. I was also able to get the coordinates of the police stations of Nairobi and map them in to the Nairobi county map. I was also able to use geolocation to determine the driver’s location.

In the development process I have obtained some valuable skills that will be put to use in any other project I will partake in. The most important lesson I have learnt is time management. I was able to plan myself accordingly in the execution of the development stages and at the same time write the project documentation. Using pivotal tracker I was able to plan every stage and allocate ample time for each stage. Furthermore, I also learnt that project management is an important skill that one should aspire to have in software development

**7.2 Conclusions**

This application is made for all the drivers so that in cooperation of a group of dispatchers and the police to ensure their security is paramount. So, the main goal of communication and accounting gets achieved by this application.

Further, this application is made using web application programming technology. Hence, this application is also economic solution for the user.

Now, this is a user-friendly website so, any driver can use this website. Therefore, this application provides collaborative interface to the users.

**7.3 Recommendations**

* Integration with voice recognition system to be able to detect altercations between the driver and the passenger or any other person in contact with the driver.
* A secure recording closed circuit television system should be installed in this taxis.
* Installations of tracking chips in this vehicles would go a long way in improving the drives security.