



**Green University of Bangladesh**  
Department of Computer Science and Engineering (CSE)  
Faculty of Sciences and Engineering  
Semester: (Spring, Year:2023), B.Sc. in CSE (Day)

## **Project Report**

Course title: **Integrated Design Project II**  
Course Code: CSE406              Section: 201D7

# **Emergency Ambulance Booking App**

### **Student Details**

| Name |                  | ID        |
|------|------------------|-----------|
| 1.   | Joy Munshi       | 201002143 |
| 2.   | Md. Jahid Hassan | 201002463 |
| 3.   | Md. Mubin Khan   | 201002375 |

Course Teacher's Name: Mr. Palash Roy

Lecturer  
Dept. of CSE  
Green University of Bangladesh  
Dhaka, Bangladesh

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>4</b>  |
| 1.1      | Introduction . . . . .   | 4         |
| 1.1.1    | Statement of the Project: . . . . .                                | 4         |
| 1.1.2    | Scope of the Work: . . . . .                                       | 5         |
| 1.1.3    | Design Goals: . . . . .  | 5         |
| 1.1.4    | Outline: . . . . .   | 5         |
| 1.2      | Literature Review . . . . .  | 6         |
| 1.2.1    | Introduction: . . . . .  | 6         |
| 1.2.2    | Why People Should Choose Our System: . . . . .                     | 6         |
| 1.2.3    | Collection of Data from Uber And Pathao: . . . . .                 | 6         |
| 1.2.4    | Effective Ambulance Service: . . . . .                             | 7         |
| 1.2.5    | Global Positioning System(GPS): . . . . .                          | 7         |
| 1.2.6    | Conclusion: . . . . .  | 7         |
| 1.3      | Objectives and Motivation . . . . .                                | 7         |
| 1.3.1    | Motivation: . . . . .  | 7         |
| 1.3.2    | Objective: . . . . .   | 8         |
| 1.4      | Problem Description . . . . .                                      | 8         |
| 1.5      | Methodology . . . . .  | 9         |
| 1.5.1    | Introduction: . . . . .  | 9         |
| 1.5.2    | Methodology of Emergency Ambulance Booking app: . . . . .          | 10        |
| 1.5.3    | Architectural Design of Emergency Ambulance Booking app: . . . . . | 11        |
| 1.5.4    | Working Procedure of the Project: . . . . .                        | 11        |
| 1.5.5    | Summary: . . . . .   | 11        |
| <b>2</b> | <b>Requirement Analysis and Design</b>                             | <b>13</b> |
| 2.1      | Feasibility Study . . . . .  | 13        |
| 2.1.1    | Technical Feasibility . . . . .                                    | 13        |
| 2.1.2    | Operational Feasibility . . . . .                                  | 13        |
| 2.1.3    | Economical Feasibility . . . . .                                   | 13        |
| 2.2      | Budget and Schedule . . . . .                                      | 13        |
| 2.2.1    | Introduction: . . . . .  | 13        |
| 2.2.2    | Gantt Chart of Project Development Timing . . . . .                | 14        |
| 2.2.3    | Detailed Budget . . . . .  | 15        |
| 2.2.4    | Summary: . . . . .   | 15        |
| 2.3      | Software Requirement Specification (SRS) . . . . .                 | 15        |

|          |   |           |
|----------|---|-----------|
| 2.3.1    | Introduction: . . . . .   | 15        |
| 2.3.2    | Software Requirement Specification of the project: . . . . .      | 16        |
| 2.3.3    | Non-Functional SRS for Emergency Ambulance Booking app: . . . . . | 16        |
| 2.3.4    | Summary: . . . . .  | 17        |
| 2.4      | SDLC Model Selection and Data Flow Diagram . . . . .              | 17        |
| 2.4.1    | Introduction: . . . . .   | 17        |
| 2.4.2    | Reasons of Incremental Model using in this project . . . . .      | 18        |
| 2.4.3    | Summary: . . . . .  | 19        |
| 2.5      | Data Flow Diagram: . . . . .                                      | 19        |
| 2.5.1    | Introduction: . . . . .   | 19        |
| 2.5.2    | Data flow diagram(Level 0) . . . . .                              | 19        |
| 2.5.3    | Data flow diagram(Level 1) . . . . .                              | 21        |
| 2.5.4    | Summary: . . . . .  | 22        |
| 2.6      | Class, Use-case and Sequence Diagram . . . . .                    | 22        |
| 2.6.1    | Introduction: . . . . .   | 22        |
| 2.6.2    | Class Diagram: . . . . .  | 22        |
| 2.6.3    | Use-Case Diagram: . . . . .                                       | 23        |
| 2.6.4    | Sequence Diagram for this project: . . . . .                      | 24        |
| 2.6.5    | Summary: . . . . .  | 25        |
| 2.7      | System Architecture . . . . .                                     | 25        |
| 2.7.1    | Introduction . . . . .  | 25        |
| 2.7.2    | Objectives: . . . . .   | 25        |
| 2.7.3    | High Level Architecture . . . . .                                 | 26        |
| 2.7.4    | Low level Architecture . . . . .                                  | 26        |
| 2.7.5    | Conclusion . . . . .  | 27        |
| 2.8      | ER diagram . . . . .  | 28        |
| 2.8.1    | Introduction . . . . .  | 28        |
| 2.8.2    | ER-Diagram Figure . . . . .                                       | 29        |
| 2.8.3    | Ambulance Management System ER diagram . . . . .                  | 29        |
| 2.8.4    | Entity . . . . .  | 32        |
| 2.8.5    | Conclusion . . . . .  | 32        |
| 2.9      | Performance Evaluation . . . . .                                  | 32        |
| 2.9.1    | Introduction: . . . . .   | 32        |
| 2.9.2    | Checking from User side: . . . . .                                | 32        |
| 2.9.3    | Evaluating from Driver side: . . . . .                            | 32        |
| 2.9.4    | Summary: . . . . .  | 32        |
| 2.10     | Impacts on societal/health/legal/cultural issues . . . . .        | 32        |
| 2.10.1   | Social Impact: . . . . .  | 33        |
| 2.10.2   | Health Impact: . . . . .  | 33        |
| 2.10.3   | Cultural Impact: . . . . .  | 33        |
| <b>3</b> | <b>Interface Design and Implementation</b> . . . . .              | <b>34</b> |
| 3.1      | UI/UX Design . . . . .  | 34        |
| 3.1.1    | Registration . . . . .  | 34        |
| 3.1.2    | Booking Process . . . . .   | 35        |

|          |  |           |
|----------|--|-----------|
| 3.1.3    | Ending Ride . . . . .                        | 37        |
| 3.1.4    | Ride Confirmation . . . . .                  | 37        |
| 3.1.5    | Navigation bar . . . . .                     | 39        |
| 3.1.6    | Payment & Ride History . . . . .             | 40        |
| 3.1.7    | Other design . . . . .                       | 40        |
| 3.2      | Working procedure . . . . .                  | 41        |
| 3.2.1    | Registration . . . . .                       | 41        |
| 3.2.2    | Booking Request Process . . . . .            | 42        |
| 3.2.3    | Booking Accept Process . . . . .             | 43        |
| 3.2.4    | Payment and Rating . . . . .                 | 43        |
| 3.2.5    | History . . . . .                            | 44        |
| 3.3      | Discussion/Analysis/Conclusion . . . . .     | 44        |
| <b>4</b> | <b>Test Cases</b>                            | <b>45</b> |
| 4.1      | Introduction . . . . .                       | 45        |
| 4.2      | Objectives . . . . .                         | 45        |
| 4.3      | Login/registration test cases: . . . . .     | 45        |
| 4.4      | Location from map test cases: . . . . .      | 47        |
| 4.5      | Ambulance Booking Test Cases: . . . . .      | 47        |
| 4.6      | Total rent calculation test cases: . . . . . | 49        |
| 4.6.1    | Payment module: . . . . .                    | 50        |
| 4.6.2    | Conclusion of Test Cases . . . . .           | 50        |
| 4.7      | Discussion . . . . .                         | 51        |
| <b>5</b> | <b>Conclusion and Future Work</b>            | <b>52</b> |
| 5.1      | Introduction: . . . . .                      | 52        |
| 5.2      | Concluding Remarks: . . . . .                | 52        |
| 5.3      | Practical Implementation: . . . . .          | 52        |
| 5.4      | Future Plan of the Work: . . . . .           | 52        |
| 5.5      | Summary of the Project: . . . . .            | 53        |

# Chapter 1

## Introduction

### 1.1 Introduction

Today's "Emergency Ambulance Booking System" is an important project. Today, ambulances play an important role when there is an accident on the road, when an emergency is needed or when people feel the need to consult a licensed doctor to save their life. In an emergency, manually ordering an ambulance can waste valuable patient time because manually ordering an ambulance is a time-consuming process. In addition, due to severe traffic congestion between the place of care and the hospital, the risk of death for the victim can be increased. So we came up with a app System called "Emergency Ambulance Booking System".

#### 1.1.1 Statement of the Project:

Our project is "Emergency Ambulance System". It is a Android-based system. We all know that; Software products can be generic or specific (personalized). Generic software developed to sell to various customers such as Microsoft Word, and Microsoft Excel. On the other hand, software developed for a client according to their specifications is called native software. Since we have launched it for some spec users like ambulance drivers and emergency ambulance users, so we can put our project as software custom.

Here is some key question which will we try to solve:

1. How can we ensure that users can easily request an ambulance during emergencies?
2. What features and functionalities should the app offer to facilitate a seamless booking experience?
3. How can we accurately determine the availability of ambulances in real-time?
4. What methods can be implemented to prioritize the nearest available ambulance for emergency cases?
5. How can users accurately specify their pick-up and drop-off locations to ensure timely assistance?

6. What security measures should be implemented to protect user data and ensure secure transactions?
7. How can the app integrate with emergency services or helplines to facilitate faster response times?
8. What payment options should be available to users, and how can we handle the billing process efficiently?
9. How can users track the location and ETA (Estimated Time of Arrival) of the assigned ambulance?
10. What mechanisms can be put in place to handle multiple ambulance requests simultaneously?
11. How can users provide feedback and rate the ambulance service to maintain accountability?
12. What support channels should be available to users in case of issues or queries?

### **1.1.2 Scope of the Work:**

This “Emergency ambulance booking System” project has just been started. Today, this is a rigorous project and also a project without errors. In this system, we just create options like users can register, login, and check which driver is available and can check hospital info on Wikipedia, and also get translation service from google map. On the driver’s side, he can view user requests, hospital lists and Google Map. But many future updates are waiting for this project. Here we will add user ratings and digital payment options to make the transaction process even better.

### **1.1.3 Design Goals:**

We discussed in this chapter (Chapter 1.1, Part 2: 1.1.1 - Project Summary and in Part 3: Scope of Work) that this project is a preset and it is a bug-free project. . It is also a demanding project. It helps to deliver the emergency ambulance and also makes it easy for the ambulance driver to receive the patient. But this is not the end. We’ve got a lot of updated ideas. Since this is a very popular System, we will engage our users by providing a wide range of services. If we add payment option, rating option and user distance calculator, it will be more popular.

### **1.1.4 Outline:**

The title of our project is “Emergency Ambulance Booking System”. In this System we have created features like user registration, user login, check ambulance driver list, view medical information, view google map, connect driver, display patient information display, etc. We all know that when patient feels sick, we need an emergency ambulance service for better treatment. That’s why we launched this initiative. By using this System user will be able

to get emergency ambulance and also get medical information quickly. The driver will also benefit as he will become patient in the digital system. We have many plans to update our System like digital payment option using mobile banking and debit or credit card payment. We also planned to set rating option as users can choose their driver by rating rating first, then user will benefit more.

## **1.2 Literature Review**

### **1.2.1 Introduction:**

Our project's tile is "Emergency Ambulance Booking Software." It's app-based software. We are all aware that software packages might be generic or customized (custom). Generic software is designed to be offered to a wide range of users, such as Microsoft Word and Microsoft Excel. Bespoke software, on the other hand, is designed for a particular customer according to their specifications. Because we launched it for specific users such as ambulance drivers and emergency ambulance users, we may classify our product as Bespoke software.

### **1.2.2 Why People Should Choose Our System:**

All the ambulance booking systems in Bangladesh cannot provide emergency facilities like Uber and Pathao. This system will help to call the nearest available ambulance. Drivers will show their availability to the system, the system will show the user accordingly. From this app site "<https://www.sheba.xyz/ambulance-service>" we can see that this site is only dealing with user sites but we are integrating drivers and hospitals.

If you want to compare our app with other apps, our app has extra security and user,driver can contact each other. User can easily find near available ambulance.

### **1.2.3 Collection of Data from Uber And Pathao:**

The choice to stagnate the transportation sector of 18 million people is ineffective in the face of 40,000 CNG auto-rickshaws and 40,000 trailers a day, where ride-sharing is transforming the transportation industry (Kamal , 2018; Sadat, 2018). Responding to local demand for economic dynamism warrants the expansion of email services (Kamal, 2018). This service is not a "silver bullet" to solve traffic congestion, but a "medium urban solution" to serve a large volume (Hassan, 2017). A global survey indicates that a personal car is only used for the owner's consumption 4% of the time, with an average of 50 to 60 trips per month, while the rest of the time. again it is not used in the garage (Ahmed 2018). The carpooling market in Dhaka is estimated at 2,000,000 people and accounts for 23% of the market share of transport services in one year, signaling rapid expansion in Dhaka; Uber-Pathao contributed the most (Sadat, 2018). System-based ride-sharing services, primarily Uber and Pathao, are two game-changers transforming Dhakacity's scheduled transportation sectors. Kamal and Ahsan (2018) have shown that the market currently operated by these services will flourish. More than 18 million people and a growth rate of 4.2% raise the worst living standards in Dhaka and according to a recent global report published in on some news portals it turns out to be the second worst city to live in (<https://thefinancialexpress.com>, 2018),

#### **1.2.4 Effective Ambulance Service:**

We are constantly seeing the usage of Uber and Pathao increasing day by day. In Dhaka's traffic jam, it takes a long time for an ambulance to arrive from a distant hospital, which can be detrimental to the patient's life. Our system will help the patient to call the nearest ambulance.

#### **1.2.5 Global Positioning System(GPS):**

The shortest path based GPS for ambulance interprets the GPS as a grid of spacecraft, transmitting example information about the changing position of the device to and from the satellite to the planet, which is then captured by the satellites. Global Positioning System transceivers, such as navigation objects, and used to estimate location, speed, and time at the location of service providers.

#### **1.2.6 Conclusion:**

It takes a lot of time for an ambulance to come from this congested city. Our system will try our best to reduce the time. Now most of the people in the country have smart phones, so everyone can take maximum advantage of our system.

### **1.3 Objectives and Motivation**

The “Emergency Ambulance Booking Software” is the title of our project. We have included functions in this software such as user registration, user log-in, ambulance driver list check, medical information view, a google map view, driver log-in, and patient information view, among others. Everyone knows that when a patient feels ill, we need an emergency ambulance service for better care. We have started this project because of that. A user of this software will be able to quickly get medical information and an emergency ambulance. Drivers will gain from the digital system as they learn to be patient. We have several plans to upgrade our software, including the ability to make digital payments using mobile banking and make payments using debit or credit cards. Additionally, we intend to include a rating option so that users can choose their drivers based on past ratings. This would benefit users more.

#### **1.3.1 Motivation:**

The human body is a natural machine. Machines tend to break down and sometimes need maintenance. Doctor at God's side! They save lives in times of crisis. However, it can be difficult to get System operate medical care at the time of an emergency. Unfortunately, many people lost their lives due to not having an ambulance in time. Although basic and advanced needs like food, clothes, houses etc everything is available online and in minimal time. But not a perfect System for ambulance service.

### **1.3.2 Objective:**

The need of the hour is an ambulance service that is easily accessible and can provide the best possible assistance in an emergency with minimum time and cost. Don't wait! This ambulance service system is a must have on your smartphone so that you can save a life in one tap.

- To fill the gap between users and drivers and enable smooth transportation of emergency patients to the closest hospital.
- To ensure fast transfer to the hospital after timely and efficient on-site paraclinical treatments for severely unwell patients.
- To eliminate any ambiguity surrounding the price of emergency transportation services by offering clear and simple pricing regulations.
- To provide consumers more control over their transportation by letting them select highly rated drivers from our system, which will ensure dependable and effective emergency medical transportation.
- To speed up response times in emergency situations by making it simple for users to find and call the closest available ambulance, providing quick access to life-saving medical assistance.

### **1.4 Problem Description**

There are many challenges or problems in any kind of work. There are also some problems that we face. The main problem is finding a list of hospitals with specific conditions. We want use gps for track user and driver. So we have to get permission from user and driver. We want to add google map. The difficulty is adding goole map on our system. We have to collect so many driver from every part of our city.

There are some problem will occur!

- How the driver will find the user?
- The payment method problem of the user.
- We get information about various hospital for blogspot.
- Now a days private ambulance have different contact number.
- How user will communicate with driver?

There are various problem will occur. For contact driver we will add a chat system. They can chat each other. The system will track user location using gps and give location info to driver. So the driver will easily find the proper address of user. The system will show hospitals. Here is the mindmap of our project:

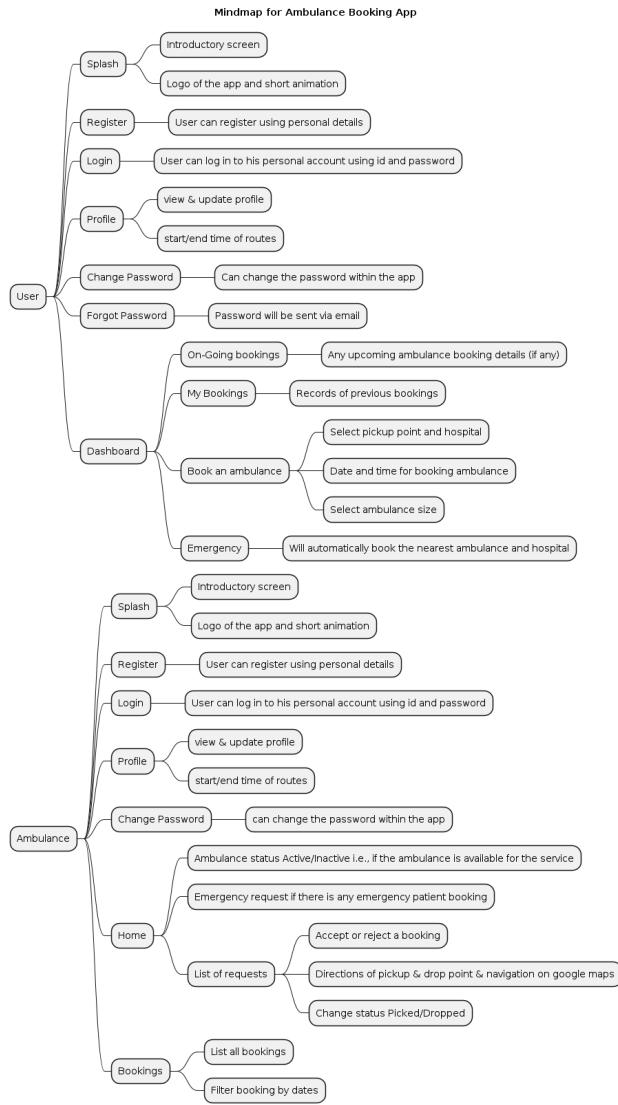


Figure 1.1: Mindmap

## 1.5 Methodology

### 1.5.1 Introduction:

In the previous chapter 1.1, we discussed the project summary and the future scope of our project. We also discuss the design goals and at the end we get a brief overview of our project. In this chapter 1.5, we will discuss the design methods, the architectural design that we Systemlized in this project, and the working processes - how this project works. We will discuss and describe the methodology and map of this project, how we pre-plan our project, and how we execute it.

### 1.5.2 Methodology of Emergency Ambulance Booking app:

We map the project and note our project methodology. The methodology of “Emergency Ambulance System” can be described as a sequence as it is a serial procedure and it is also done by fulfilling certain steps. If we show the Method sequence with a flowchart, we can execute the procedure. For a clear design, our methodology is designed by a diagram given below:

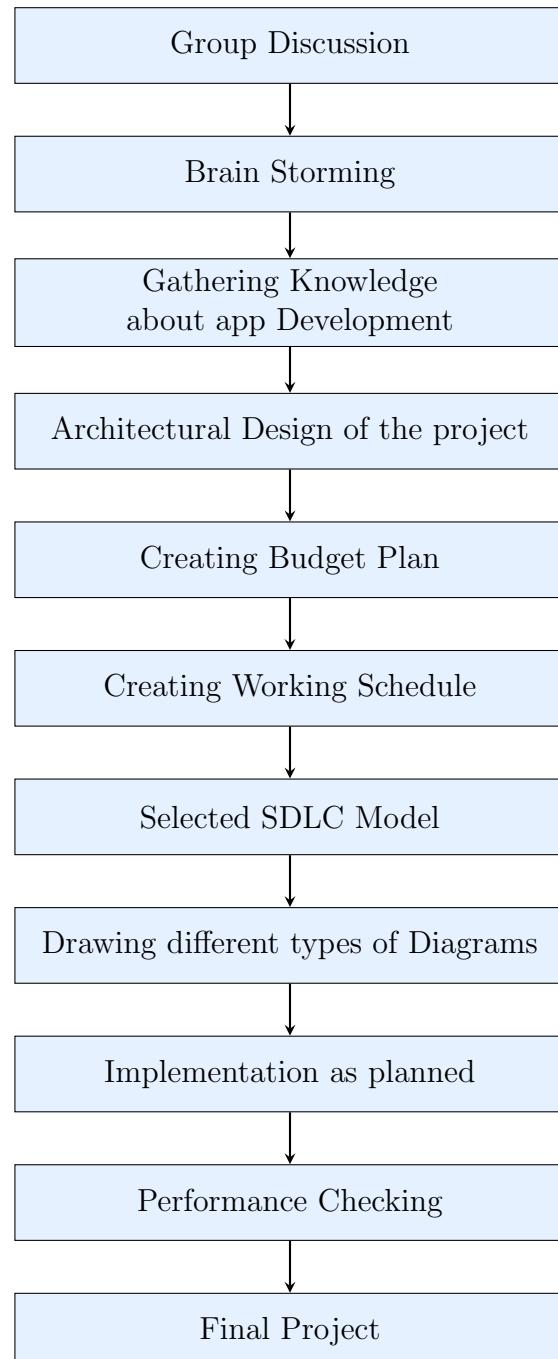


Figure 1.2: Flow chart for Methodology of Emergency Ambulance Booking System

### 1.5.3 Architectural Design of Emergency Ambulance Booking app:

After Mind mapping, we have drawn an architectural design of our planned project which is given below for a clear conception about our project.

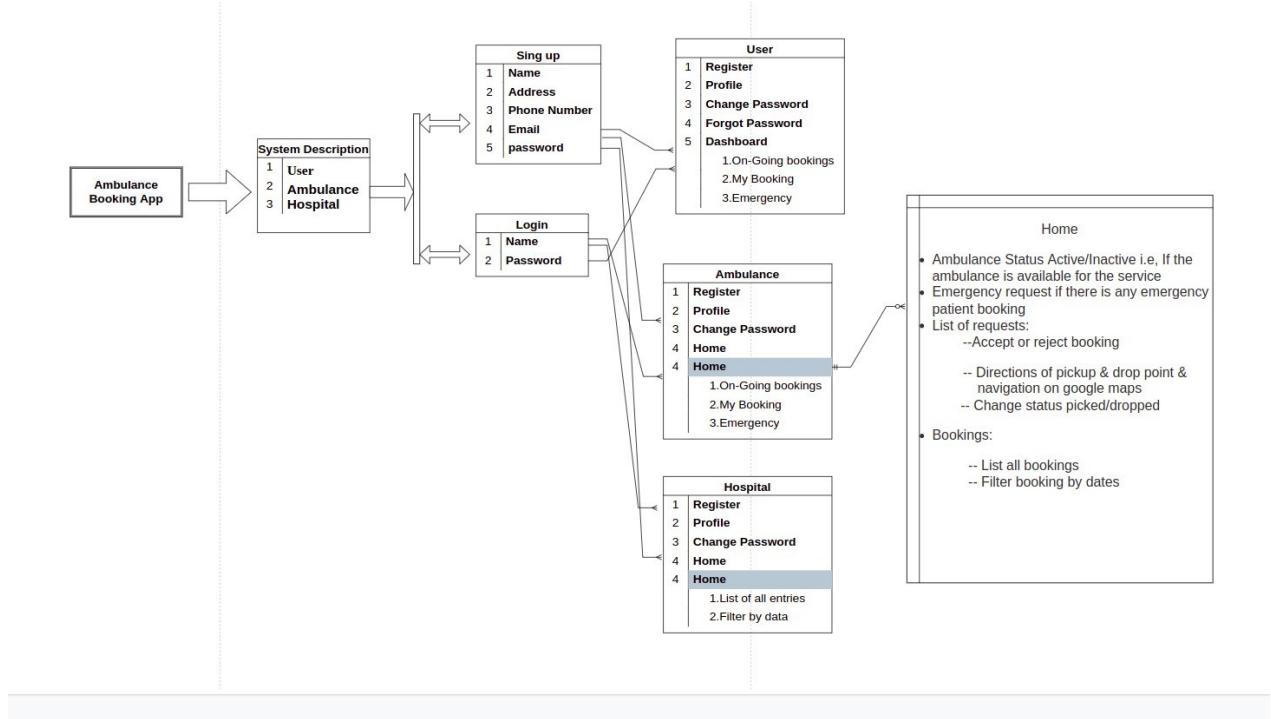


Figure 1.3: Architectural Design of Emergency Ambulance Booking app

### 1.5.4 Working Procedure of the Project:

In fact, our project is a System based on emergency ambulance booking. So here our main aim is to find the nearest ambulance driver and also to search the best hospital according to our disease. That's why at first we created login pages like user login page, and driver login pages and if they don't have an existing account because they can sign up we created a registration page and also store data in our database. . The login page and registration page are just a form for a System, but our main job is to find the driver, final confirmed passenger information and find medical information such as our illness, Which medical and which doctor will be the best or not.

### 1.5.5 Summary:

In this chapter we described the mind map of the "Emergency Ambulance Booking System" project, what knowledge we used here to create it, and methodology of this project, and how we designed the total work and implementation of the complete project. This chapter is the focus of this project report. If someone reads this chapter and tries to realize our methodology, people can understand the main topic and can have a clear concept about this

project, what is the main goal of our project, how we are implementing our features, what are the future plans and how we satisfy our users by providing services and how we recognize the seriousness of our patients and how we treatment by managing the ambulance in the digital system.

# **Chapter 2**

## **Requirement Analysis and Design**

### **2.1 Feasibility Study**

A feasibility analysis assesses the probability of project success; Therefore, perceived objectivity is an essential factor in research credibility for potential investors and lending institutions. There are five types of feasibility studies – the separate areas that a feasibility study examines, are described below.

#### **2.1.1 Technical Feasibility**

The system will be very lightweight and can run on every Android, iOS, and Linux-based system. The customer can take the service even if the internet is weak. The software will take less internet for booking an ambulance. If there is no internet, the user will select his area then our software will suggest the driver number, the user will contact the driver and the driver can confirm the ride through our software.

#### **2.1.2 Operational Feasibility**

Most users have smartphones, so booking an emergency ambulance is very easy.

#### **2.1.3 Economical Feasibility**

The software will return a very good profit. Commission will be taken from driver and passenger. If a customer has a good internet connection it will show some banner ads for revenue.

## **2.2 Budget and Schedule**

### **2.2.1 Introduction:**

Our project name is “Emergency ambulance booking app” and it is software based project and it is deployed on app System platform. Finally, to complete this project we need some

| Activities           | Months |   |   |   |   |   |   |   |
|----------------------|--------|---|---|---|---|---|---|---|
|                      | 1      | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Planning             | ■      | ■ | ■ | ■ | ■ | ■ |   |   |
| Employee Recruitment | ■      | ■ |   |   |   |   |   |   |
| Design               |        | ■ | ■ | ■ | ■ | ■ |   |   |
| Coding               |        |   | ■ | ■ | ■ | ■ |   |   |
| Testing              |        |   |   | ■ | ■ | ■ |   |   |
| Delivery             |        |   |   |   | ■ |   |   |   |

money and we also need to set a timetable of how long we need and when we can release our project to our beloved users. mine. In this chapter 2.2, we will discuss the human resource plan, budget details, and work schedule for our project. This is also an important chapter because if we create an interim budget and work plan we can guarantee our users the amount we require for this project and the number of times we need it for the job. this.

### 2.2.2 Gantt Chart of Project Development Timing

A planned project titled “Secure Ticket Booking System” will be launched within 4 months (16 weeks). This software development timing is described below by a Gantt Chart:

### **2.2.3 Detailed Budget**

The budget details of this project titled “Emergency Ambulance Booking System” is given below by a table. Our Accounts office will get a clear idea about the budget details by seeing this table.

| <b>SL No.</b> | <b>Criteria</b>       | <b>Cost Specification</b> | <b>New System (Taka)</b> |
|---------------|-----------------------|---------------------------|--------------------------|
| 01.           | Office Cost           | Team Meeting              | BDT 15,000               |
|               |                       | Project Meeting           | BDT 20,000               |
|               |                       | First Aid                 | BDT 1,000                |
| 02.           | Office Equipment Cost | Computer                  | BDT 4,70,000             |
|               |                       | Laptop                    | BDT 13,00,000            |
|               |                       | Printer                   | BDT 60,000               |
|               |                       | CC Camera                 | BDT 50,000               |
| 03.           | app Cost              | app Maintenance           | BDT 4,000                |
| 04.           | Salary Cost           | Team Leader               | BDT 2,05,000             |
|               |                       | System Designer           | BDT 77,000               |
|               |                       | Software Engineer         | BDT 1,00,000             |
|               |                       | Animator                  | BDT 45,000               |
|               |                       | Developer                 | BDT 47,000               |
|               |                       | Officer                   | BDT 40,000               |
|               |                       | Guard                     | BDT 20,000               |
| Total Cost    |                       |                           | BDT 24,54,000            |

Table 2.1: Detailed Budgets

### **2.2.4 Summary:**

This chapter (Chapter 2.2: Budget and Schedule) is an important chapter of this master report because in this chapter we presented our human resource plan how we distribute work to our staff and then we detail the budget how much money might be needed to start the project and a detailed work plan, our total work is 16 weeks and which week we will use it. for what tasks and how we can get our work done in an organized manner.

## **2.3 Software Requirement Specification (SRS)**

### **2.3.1 Introduction:**

Our main goal in this chapter 2.3 is to describe the software requirements specification for this project. We all know that there are two types of software requirements specifications: functional and non-functional. The functional requirements specification will be described in Part 2 (Section ?? of Section 1, and the non-functional part of the requirements specification will be described in Part 2 of Part 2 (Section 2.3.3).

### **2.3.2 Software Requirement Specification of the project:**

A software requirements specification (SRS) includes in-depth descriptions of the software that will be developed. It is defined as two categories like-

- (i) functional software requirement specification and
- (ii) non-functional software requirement specification.

Functional SRS is describing in the Chapter-?? and Non-functional SRS is describing in the chapter-

**User Interface:** A user interface describes the logical nature of each interface between a software product and its users. This can include sample screens, UI/UX standards or suite style guides to follow, screen layout constraints, standard buttons and functions displayed on each screen, keyboard shortcuts, error message display standards, and more defines software components. User interface design details should be documented in a separate user interface specification.

**Hardware Interfaces:** Hardware interfaces describe the logical and physical characteristics of each interface between a software product and the hardware components of a system. This may include the supported device types, the nature of the data and control interactions between software and hardware, and the communication protocols used.

**Software Interfaces:** Software Interfaces describes the connections between the product and other specific software components (name and version) including databases, operating systems, tools, libraries, and integrated commercial components. It identifies the data items or messages coming into the system and going out and also describes the purpose of each. It also describes the services needed and the nature of communications and refer to the documents that describes the detailed System programming interface protocols. It also identifies data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), it specifies this as an implementation constraint.

**Communications Interfaces:** Communication Interface describes requirements related to all communication functions required by this product, including e-mail, app browser, network server communication protocols, electronic forms, etc. Define the Systemappropriate message format. It also defines any communication standard to use, such as File Transfer Protocol (FTP) or Hypertext Transfer Protocol (HTTP). Specifies any communication security or encryption concerns, data rates, and synchronization mechanisms.

### **2.3.3 Non-Functional SRS for Emergency Ambulance Booking app:**

1. **Performance Requirements:** If there are performance requirements in our project for the product in various scenarios, list them here and explain their rationale to help developers understand their intent and make Systemappropriate design choices. It specifies time relationships for real-time systems. It makes these requirements as specific as possible. We may need to state performance requirements for individual feature or functional requirements.
2. **Safety Requirements:** Safety requirements specify requirements that address loss,

damage or harm that may result from the use of the product. It defines safeguards or actions that must be taken as well as actions to be avoided. It refers to any external policy or regulation that outlines safety issues affecting the design or use of a product in our project. It also defines the security certificates that must be met.

3. **Security Requirements:** Security requirements specify any requirements relating to security or privacy issues surrounding the use of the product or the protection of data used or generated by the product in your project. we have the title "Emergency Ambulance Booking System". It identifies all user identity authentication requests. It also addresses any external policies or regulations that contain safety issues affecting our project products. It also identifies any security or privacy certifications that must be met.
4. **Software Quality Attributes:** Software quality attributes specify any additional quality characteristics of the product that will be important to the customer or developer. Some things to consider are: adaptability, usability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, scalability robustness, testability and usability.

#### **2.3.4 Summary:**

This is Chapter 2.3: Defining Software Requirements. Here, we discussed defining individual software requirements in our project. We have clarified functional software requirements specification and non-functional software requirements specification in our project. If someone reads the functional and non-functional requirements section, he will have a clear idea of the user needs and the general needs of any app System.

### **2.4 SDLC Model Selection and Data Flow Diagram**

#### **2.4.1 Introduction:**

Every activity has a lifecycle, and the software development process is no exception. Even if we don't know about SDLC, We still follow it subconsciously. However, if the software engineer knows the SDLC, the project can be completed in a more controlled way. One of the great benefits of this awareness is that hot developers don't run (code) themselves, which can actually lead to an uncontrolled release of the project. Second, it helps customers and software professionals avoid confusion by predicting problems and problems ahead of time. Simply put, SDLC defines the various stages of the software lifecycle. But before understanding what SDLC is. We need to understand the beginning and end of SDLC more broadly. If every running project has no start and no end, then there is already a problem. If we go out on the road, we have to know the beginning and the end, otherwise, we will be moving endlessly. Below is a picture showing the typical flow of SDLC with 5 main patterns.

SDLC (Software Development Life Cycle) is a framework that describes the activities performed at each stage of a software development project. SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how

to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. A typical Software Development Life Cycle consists of the following stages –

1. Planning and Requirement Analysis,
2. Defining Requirements,
3. Designing the Product Architecture,
4. Building or Developing the Product,
5. Testing the Product,
6. Deployment in the Market and Maintenance.

There are different types of SDLC Model. Following are the most important and popular SDLC models followed in the industry –

1. Waterfall Model
2. Incremental Model
3. Spiral Model
4. V-Model

In our project we have used Incremental Model. The details of Incremental Model is given below for clear conception.

**Incremental Model:** Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

#### **2.4.2 Reasons of Incremental Model using in this project**

In our project titled “**Emergency Ambulance Booking System**” we have used Incremental Model. The reasons of using Incremental Model are given below for clear conception-

1. The cost of accommodating changing customer requirements is reduced.
2. It is easier to get customer feedback on the development work that has been done.
3. More rapid delivery and deployment of useful software to the customer is possible.
4. Errors are easy to be recognized.
5. Simple to manage risk because it is handled during its iteration.
6. The client gets important functionality early.

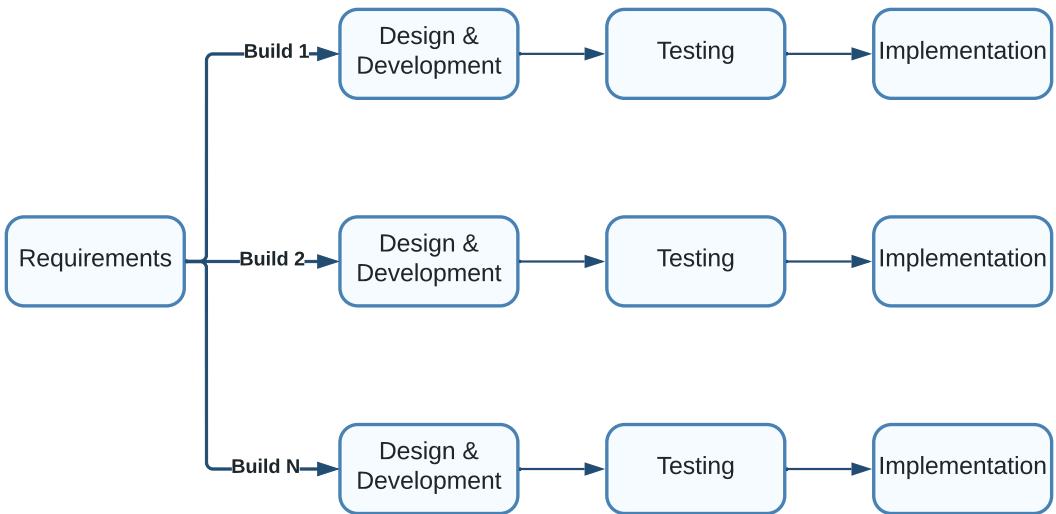


Figure 2.1: Incremental Model

### 2.4.3 Summary:

Here at first, we got an idea about what's the SDLC Model and what are the renowned SDLC Model. Then we discussed which SDLC model we have used in our project and what's the reason for using that SDLC Model in our project.

## 2.5 Data Flow Diagram:

### 2.5.1 Introduction:

DFD and use case diagram are playing very important role in our project.

**Data flow diagram:** A data flow diagram is a graphical representation of the data flow in an information system. It can represent incoming data stream, outgoing data stream, and stored data. The DFD doesn't mention anything about how data flows through the system. It is a traditional visual representation of the information flows in a system. A clear and sharp DFD can graphically represent the System appropriate amount of system requirements. It tells how data enters and leaves the system, what changes the information and where the data is stored. There is an important difference between a DFD and a Flowchart. Flowchart describing control flow in program modules. DFDs describe the data flow in the system at different levels. The DFD does not contain any control or branch elements.

### 2.5.2 Data flow diagram(Level 0)

In this level there are two entities (Passenger and driver) and the system will have a central database. When the user and driver will log in both information will store in the database. When a user requests a ride it will go to the driver and the profile of the user will visible

to the driver. After accepting ride request the driver's profile will visible to the passenger. After finishing driver will request payment and the passenger will confirm payment and the driver will notify payment confirmation.

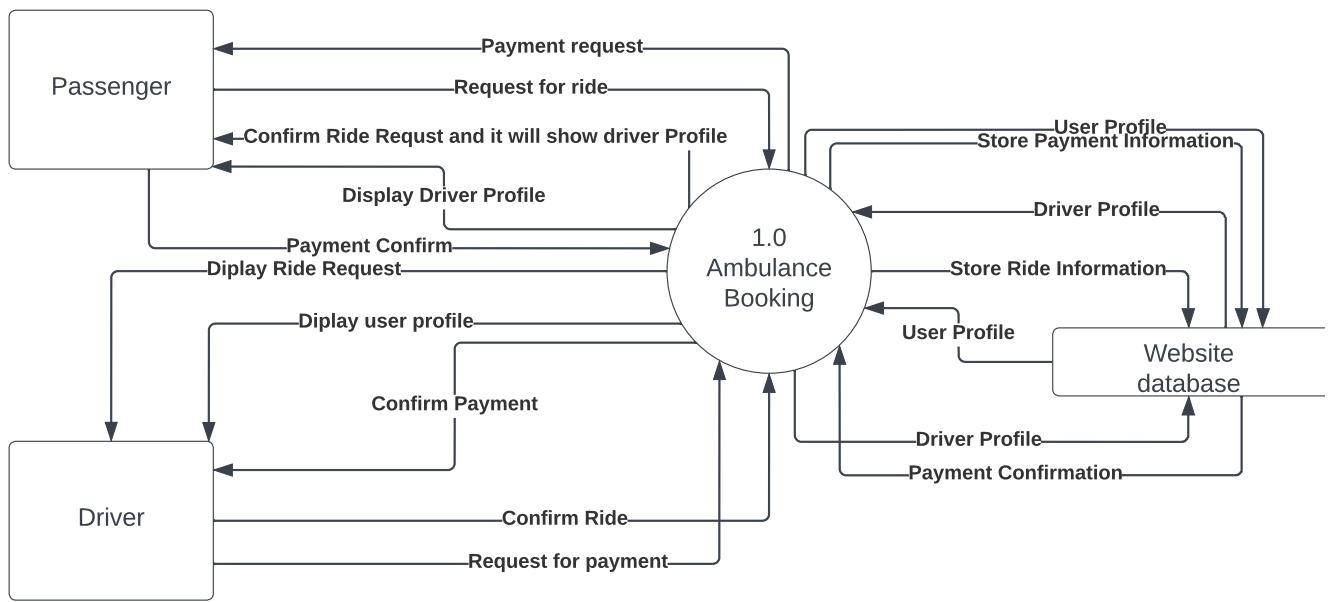


Figure 2.2: Data Flow Diagram level 0

### 2.5.3 Data flow diagram(Level 1)

Data flow diagram will describe the way of flow of data. In user section user will request for login or Registration to our system. Then system will confirm user log in. After successfully log in system will store user data to passenger database. In driver section driver will also give login request and system will confirm the request and store the driver data in driver database. In Ride confirmation process user will request a ride and driver will confirm the ride. Driver and user both can view driver and user profile and necessary information. This process will perform data maintenance system. After booking data will store to booking database.

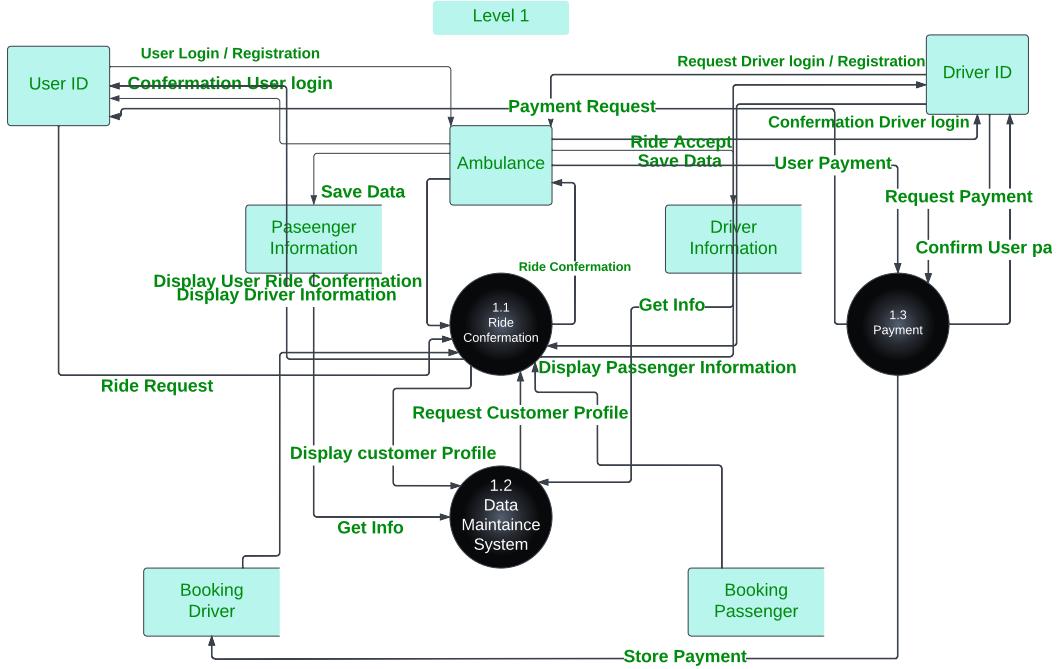


Figure 2.3: Data Flow Diagram level 1

#### 2.5.4 Summary:

We got the idea of data flow diagram and its different levels. As level 0 is for beginner level and level 1 is for mid-level people which means by looking at Level-1, a person can infer working processes and data storage systems and data Manipulating system. So we have drawn level-1 diagram in our lab.

## 2.6 Class, Use-case and Sequence Diagram

### 2.6.1 Introduction:

In this chapter (Chapter-2.6), we will show three (3) types of diagrams of our project titled “Emergency Ambulance Booking System” which are- (i) Class Diagram, (ii) Use-case Diagram and (iii) Sequence Diagram.

## 2.6.2 Class Diagram:

In this section (Chapter 2.6, Part 2.6.2) we will show a class diagram of a project titled “Emergency Ambulance Reservation System”. In this system, we have added some classes named passenger, admin, driver, hospital, register, booking, driver confirmation, driver payment, and log-out.

The Ambulance Booking System class diagram shows the structure of the information or data processed by the system. This data or information is represented by classes. Each class

has attributes depending on the methods it uses. So the UML class diagram was represented by a box with three partitions, the name of the class on top, the attributes in the middle, and the methods on the bottom. The arrows represent their relationship to each other.

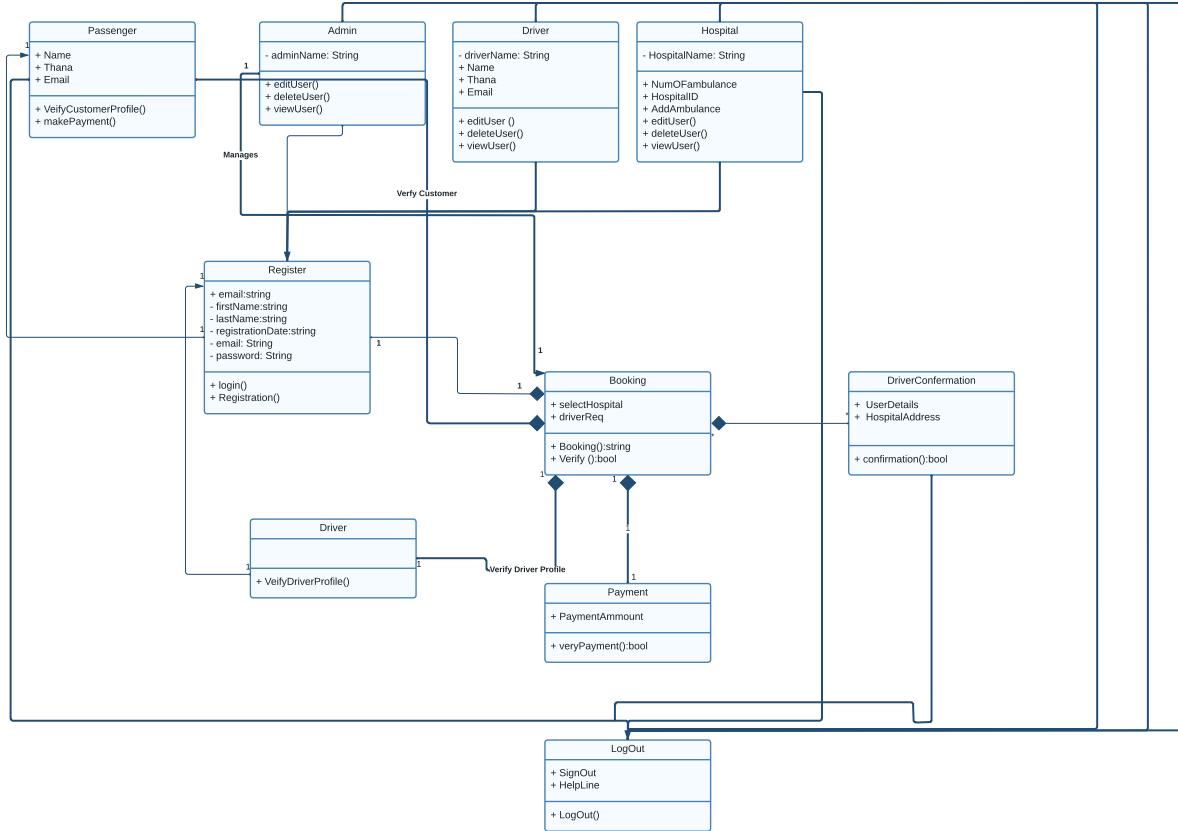


Figure 2.4: Class Diagram for “Emergency Ambulance Booking app”

### 2.6.3 Use-Case Diagram:

The report showing a use case diagram of a project titled ” Emergency Ambulance booking System”, which is the main form of our new undeveloped software. These use cases specify the expected behavior (what), not the exact method to achieve it (how). Once specified, it can be referenced with both text and images. This use case model is that it helps us to design a system from the point of view of the end user. It is an effective technique to communicate system behavior according to user conditions by specifying all externally visible system behaviors. It just summarizes some of the relationships between use cases, actors, and systems. It does not show the order in which the steps are taken to achieve the goal for each use case.

In this software have a login system. Users or drivers have to log in to our system. So for login, we use email and password. So, email and password will include in our system. After login user will search for a driver and request a driver. Then the user will find a driver on the map and the driver can also find the user on the map. Users can also find the hospital list in

our system and choose which hospital users want to go and it will set the map to the driver's side. In your driver section user and driver, can contact each other. User and driver both should fill up a short registration form for security purposes which includes Name, Email, Thana, and Phone Number.

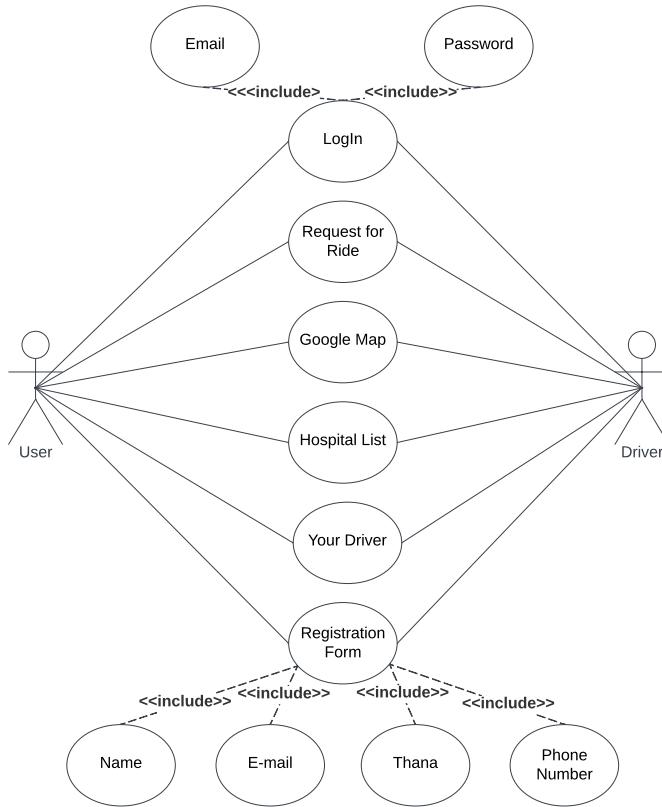


Figure 2.5: Use-Case Diagram of “Emergency Ambulance Booking app”

#### 2.6.4 Sequence Diagram for this project:

In this part (Chapter-2.6, part-2.6.4) we will show the sequence diagram of our project titled “Emergency Ambulance Booking System” which is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. It consists of a group of objects that are represented by lifelines, and the messages that they exchange overtime during the interaction. This diagram shows the sequence of messages passed between objects. It can also show the control structures between objects. This diagram shows the objects and the messages between the objects.

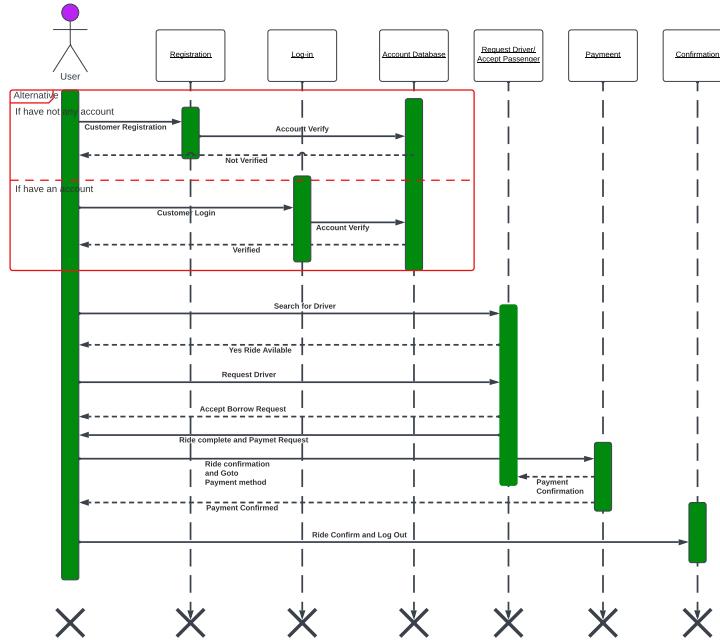


Figure 2.6: Sequence Diagram of “Emergency Ambulance Booking app”

### 2.6.5 Summary:

This chapter (Chapter 2.6) is all about diagrams, because in this chapter we have shown 3 different diagrams to have a clear concept of actors and their responsibilities as well as summarize some of them. relationships between use cases, actors, and systems. We also illustrated the flow of messages between objects in a sequence interaction diagram of this project.

## 2.7 System Architecture

### 2.7.1 Introduction

The “Emergency Ambulance Booking Software” is a significant initiative nowadays. Nowadays, ambulances play an important role when a road accident occurs, a medical emergency arises, or a human feels the need to visit a medical registered doctor in need to save his or her life. Manually booking an ambulance in an emergency can waste a patient’s valuable time because ambulance booking is a time-consuming process. Also, because of the heavy traffic congestion between the pickup location and the hospital facility, the victim’s risk of death may be exacerbated. As a result, we have developed a software titled “Emergency Ambulance Booking Software”.

### 2.7.2 Objectives:

- To provide an abstract view of the system components and their interactions

- To illustrate the main functions and features of the Ambulance Booking System
- To identify the major subsystems and their roles in the system
- To provide a detailed view of the system components and their interactions.
- To explain the data models and schemas used in the system database.

### 2.7.3 High Level Architecture

The User Interface, Ambulance Booking System, Database, and Notification System make up the four primary parts of the Ambulance Booking App's high-level architecture.

The point of interaction between users and the app is the User Interface component. It enables customers to enter ambulance requests, read booking confirmation and information, and monitor notifications and updates about the progress of their reservations. The main

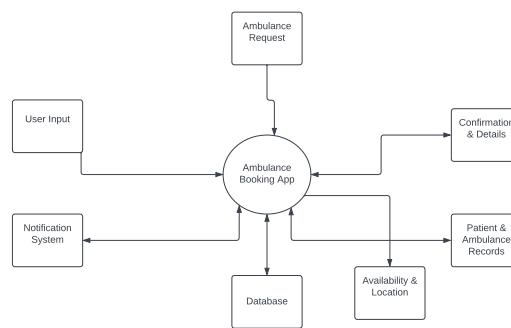


Figure 2.7: USER SignUP Page

features of the app are handled by the Ambulance Booking System component. It controls the booking of ambulances, including responding to user requests, accessing and updating the Database, and allocating the available ambulances.

Important information about patients, ambulances, availability, locations, and user preferences is stored and managed by the database component. It offers quick access to user settings, availability and location data, patient and ambulance records, and other data.

Sending users pertinent alerts is within the purview of the notification system component. It gives updates and delays on the assigned ambulance, lets users know how their ambulance booking is progressing, and, if required, sends out emergency warnings.

### 2.7.4 Low level Architecture

The User Interface, Ambulance Booking System, Database, and Notification System make up the four primary parts of the Ambulance Booking App's high-level architecture. The point of interaction between users and the app is the User Interface component. Users may enter their ambulance requests, examine confirmation and booking data, and get notifications and updates on the status of their bookings.

The main features of the app are handled by the Ambulance Booking System component.

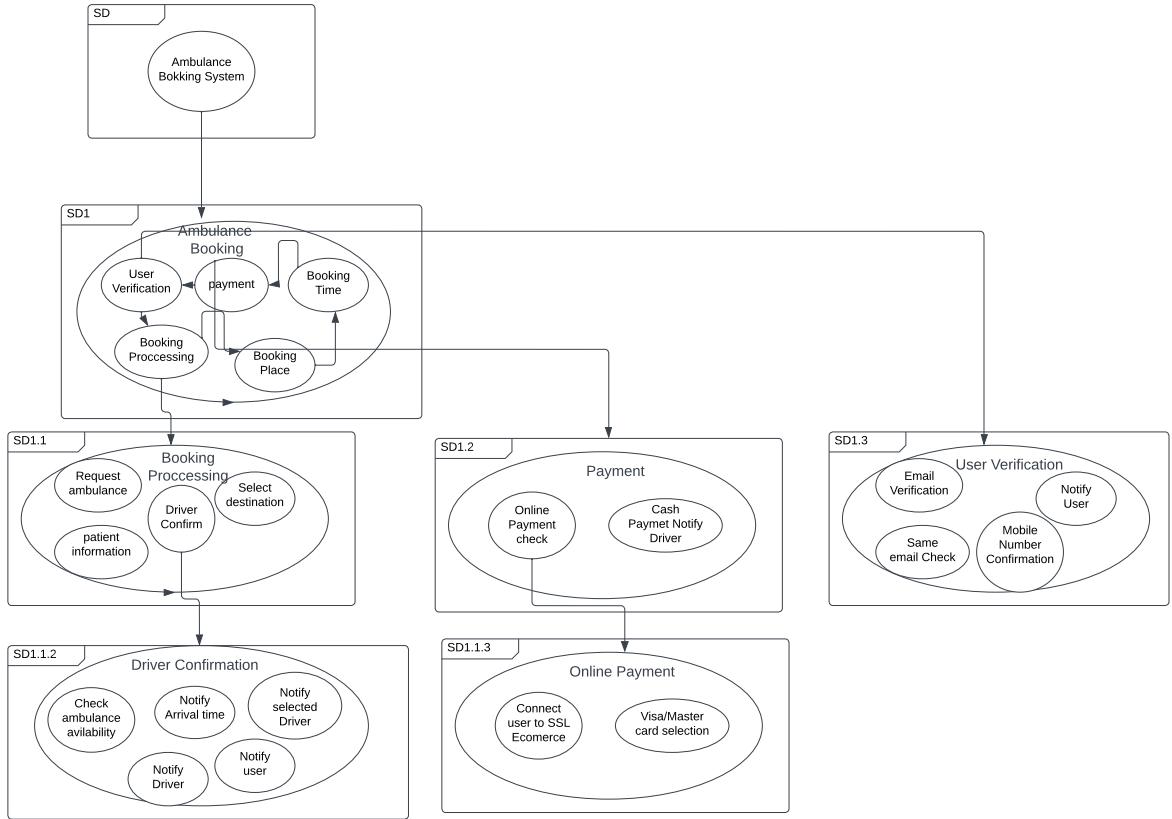


Figure 2.8: Low-Level Architecture

It controls the booking of ambulances, including responding to user requests, accessing and updating the Database, and allocating the available ambulances.

Important information about patients, ambulances, availability, locations, and user preferences is stored and managed by the base component. It offers quick access to user settings, availability, and location data, patient and ambulance records, and other data.

Sending users pertinent alerts is within the purview of the notification system component. It provides consumers with status information on.

### 2.7.5 Conclusion

The User Interface, Ambulance Booking System, Database, and Notification System are the four primary components of the Ambulance Booking App's high-level architecture. A more thorough understanding of the underlying workings and relationships of each component is provided by the low-level architecture. This architecture provides effective emergency ambulance booking and administration, ensuring that patients in need receive quick medical attention.

Despite these difficulties, the Ambulance Booking App is a viable platform for customers to swiftly book emergency ambulances. The seamless coordination between components is ensured by the high-level and low-level design, enabling effective ambulance allocation and

prompt user updates.

Continuous monitoring and performance improvement will be crucial in subsequent revisions of the app in order to further improve user experience and guarantee the program's dependability. To handle new risks and ensure data integrity, frequent security assessments and upgrades will also be made.

## 2.8 ER diagram

### 2.8.1 Introduction

An ER diagram is a data modeling tool used to create visual diagrams representing different entities and their relationships. An ambulance-specific ER diagram can be used to document the various entities involved in the ambulance service, such as patients, doctors, paramedics, and ambulances, as well as their individual attributes and interconnections. This diagram can provide a detailed view of the system being modeled and can be used for analysis and optimization of the ambulance services.

## 2.8.2 ER-Diagram Figure

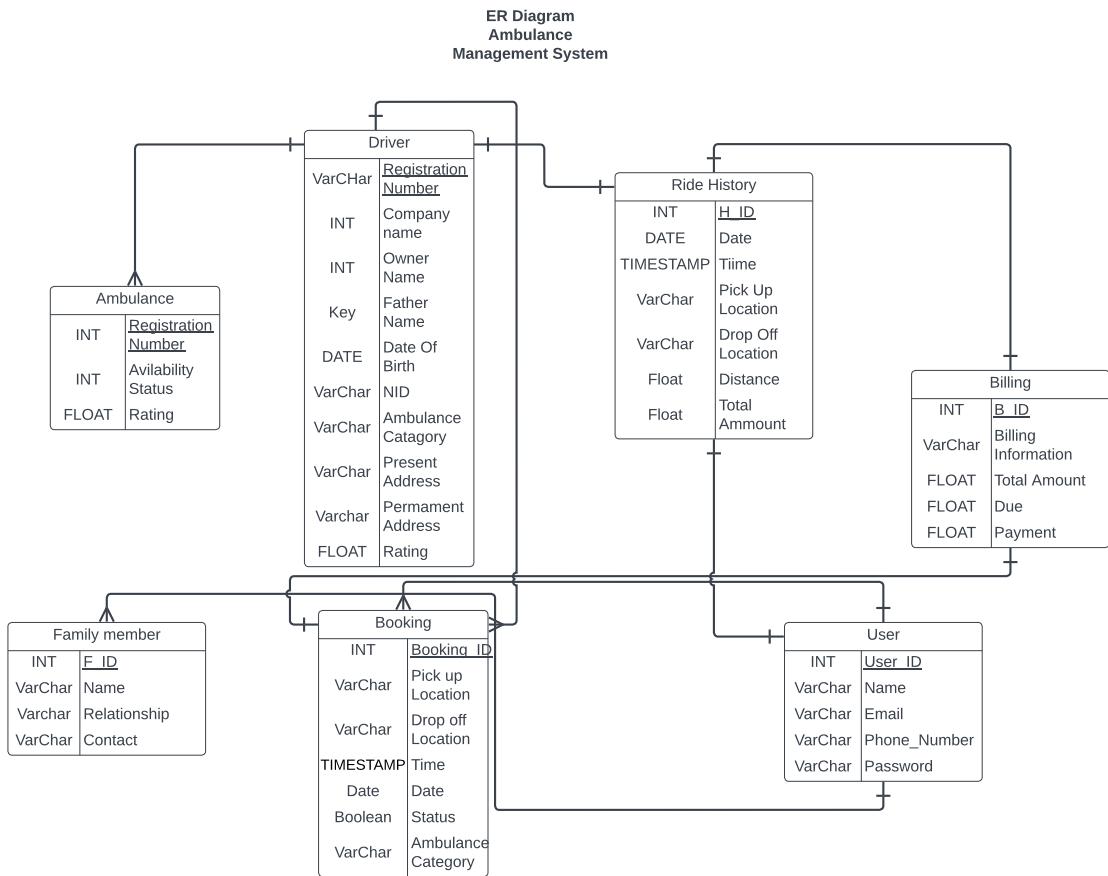


Figure 2.9: ER- Diagram

## 2.8.3 Ambulance Management System ER diagram

### 1. Ambulance:

- Registration Number (INT): Each ambulance is assigned a unique registration number for identification purposes.
- Availability Status: Indicates whether the ambulance is currently available for booking or not.
- Rating (FLOAT): Represents the rating or feedback given to the ambulance service.
- Ambulance Category: Specifies the category or type of ambulance, such as basic life support (BLS) or advanced life support (ALS).

- Driver Registration Number (INT): Refers to the registration number of the driver assigned to the ambulance. An ambulance can have one driver associated with it.

## 2. Family Member:

- Family Member ID (INT): A unique identifier for each family member associated with an ambulance.
- Name (VARCHAR): Stores the name of the family member.
- Relationship (VARCHAR): Represents the relationship of the family member with the patient or ambulance crew.
- Contact (VARCHAR): Stores the contact information (phone number, email, etc.) of the family member.

## 3. Booking:

- Booking ID (INT): A unique identifier for each booking made by a user.
- User ID (INT): Refers to the ID of the user who made the booking.
- Pick-up Location (VARCHAR): Specifies the location where the ambulance will pick up the patient.
- Drop-off Location (VARCHAR): Indicates the destination or location where the patient will be dropped off.
- Time (TIMESTAMP): Records the time at which the booking was made.
- Date (DATE): Stores the date of the booking.
- Status (BOOLEAN): Represents the status of the booking, whether it is confirmed or pending.

## 4. User:

- User ID (INT): A unique identifier for each user of the ambulance service.
- Name (VARCHAR): Stores the name of the user.
- Email (VARCHAR): Represents the email address of the user.
- Phone Number (VARCHAR): Stores the phone number of the user.
- Password (VARCHAR): Stores the password associated with the user account.

## 5. Billing:

- Billing ID (INT): A unique identifier for each billing record.
- User ID (INT): Refers to the ID of the user associated with the billing record.
- Total Amount (FLOAT): Represents the total amount to be paid for the ambulance service.
- Due (FLOAT): Indicates the amount that is yet to be paid.
- Payment (FLOAT): Represents the amount already paid by the user.

## 6. Driver:

- Driver Registration Number (INT): A unique identifier for each driver.
- Name (VARCHAR): Stores the name of the driver.
- Company Name (INT): Refers to the ID or name of the company the driver is associated with.
- Owner Name (VARCHAR): Stores the name of the owner of the driver's company.
- Father's Name (VARCHAR): Represents the father's name of the driver.
- Date of Birth (DATE): Stores the driver's date of birth.
- NID (VARCHAR): Refers to the National Identification Number or any other identification number associated with the driver.
- Present Address (VARCHAR): Indicates the current or present address of the driver.
- Permanent Address (VARCHAR): Stores the permanent address of the driver.

## 7. Ride History:

- History ID (INT): A unique identifier for each ride history record.
- Date (DATE): Represents the date of the ride.
- Time (TIMESTAMP): Indicates the time at which the ride took place.
- Pick-up Location (VARCHAR): Specifies the location from where the ambulance picked up the patient.
- Drop-off Location (VARCHAR): Indicates the destination or location where the patient was dropped off.
- Distance (FLOAT): Represents the distance traveled during the ride.
- Total Amount (FLOAT): Indicates the total amount charged for the ride.

## **2.8.4 Entity**

An ambulance has a one-to-many relationship with a family member, as an ambulance can have multiple family members associated with it. The user has a one-to-many relationship with Booking, as a user can make multiple bookings. A user has a one-to-many relationship with Billing, as a user can have multiple billing records. A driver has a one-to-many relationship with an ambulance, as a driver can be associated with multiple ambulances. A user has a one-to-many relationship with Ride History, as a user can have multiple ride history records.

## **2.8.5 Conclusion**

These entities, relationships, and attributes provide a basic overview of an ambulance management system. The ER diagram can be further expanded and refined based on specific requirements and additional functionalities desired for the system. It's essential to involve stakeholders and domain experts during the requirement analysis process to ensure all necessary aspects are considered.

# **2.9 Performance Evaluation**

## **2.9.1 Introduction:**

This is chapter 2.9, where we will evaluate the performance of our project. In Part 2 (Chapter 2.9.2) we will test user-side performance and in Part 3 (Chapter 2.9.3) we will evaluate driver-side performance, in Part 4 (Chapter 2.9.3) we will test the overall performance of our System, then part 5 (chapter 2.9.4), we will summarize this chapter.

## **2.9.2 Checking from User side:**

In this part (Chapter-2.9.2), we will check the user side that, is it working perfectly or not.

## **2.9.3 Evaluating from Driver side:**

## **2.9.4 Summary:**

This chapter (chapter 2.9) is used to evaluate our project. In this chapter we checked all the parts of the project and finally decided that our project is working perfectly and here we have no problem. All features are correct in our project.

# **2.10 Impacts on societal/health/legal/cultural issues**

In this chapter-2.10 we will discuss about various impacts of our system in social, health cultural.

### **2.10.1 Social Impact:**

Ambulance is an important transport method for life saving. Using ambulance any emergency patient can reach hospital in time. We know that every minute is important for a emergency patient.

Our system will save time for get an ambulance in time. The death rate will be decrease, it will save many life which will give a positive impact in our society.

### **2.10.2 Health Impact:**

The main goal of the system is save life of a emergency patient.

Many patients are dying day by day due to not being able to go to the right hospital on time. Our system works to deliver ambulance on time. As a result, the patient will reach the hospital on time. As a result, the patient's death rate will decrease a lot.

### **2.10.3 Cultural Impact:**

The system will make positive effects on culture. The system will make trust between user and driver. The employment would be increase. Driver can earn some money using our system.

Culture is difficult to define properly, and it is unclear how it affects behavior. For example, for emergency services, does it make sense to talk about a single culture within or across organizations? Likewise, culture cannot change so easily.

# Chapter 3

## Interface Design and Implementation

### 3.1 UI/UX Design

The UI/UX design of the Emergency Ambulance Booking System will follow a simple and intuitive approach to make it easy for users and ambulance drivers to use the system.

#### 3.1.1 Registration



Figure 3.1: First page & Login page for all type of user

In figure(3.12a) we can see here any user can choose the option to access this application. For existing users click on the "SIGN IN" button. Then we can see the figure 3.12b to "SIGN IN". Everyone must fill up this field to sign in.

In figure(3.13) we can see that here is the "SIGN UP" page for users. Every user must sign up to access this application.

Above we can see that there is another "SIGN UP" page. don't be confused by figure(3.14) sign-up steps for ambulance drivers.

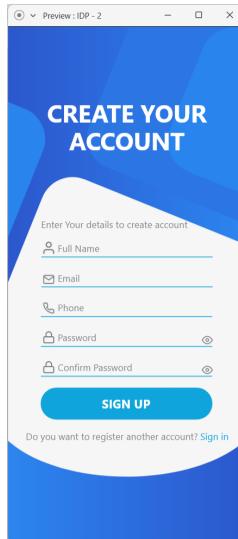


Figure 3.2: USER SignUP Page

(a) Step - 1
(b) Step - 2
(c) Step - 3

Figure 3.3: Steps for driver registration

In figure(3.4) is for "SING UP" for the driver also. Figure (3.4a) is for confirmation of privacy policy after that add the payment method that how drivers want to receive their payments. Drivers also add this latter by clicking "DO THIS LATER" and then show the figure(3.4c).

### 3.1.2 Booking Process

For booking an ambulance follow the steps shown in figure(3.5 & 3.6). Here user can book an ambulance for themselves, friends & family also for others that is selected in figure(3.5b). Then in the next step select the Ambulance category that indicates which type of ambulance is best for the user. Here are three categories of ambulances.

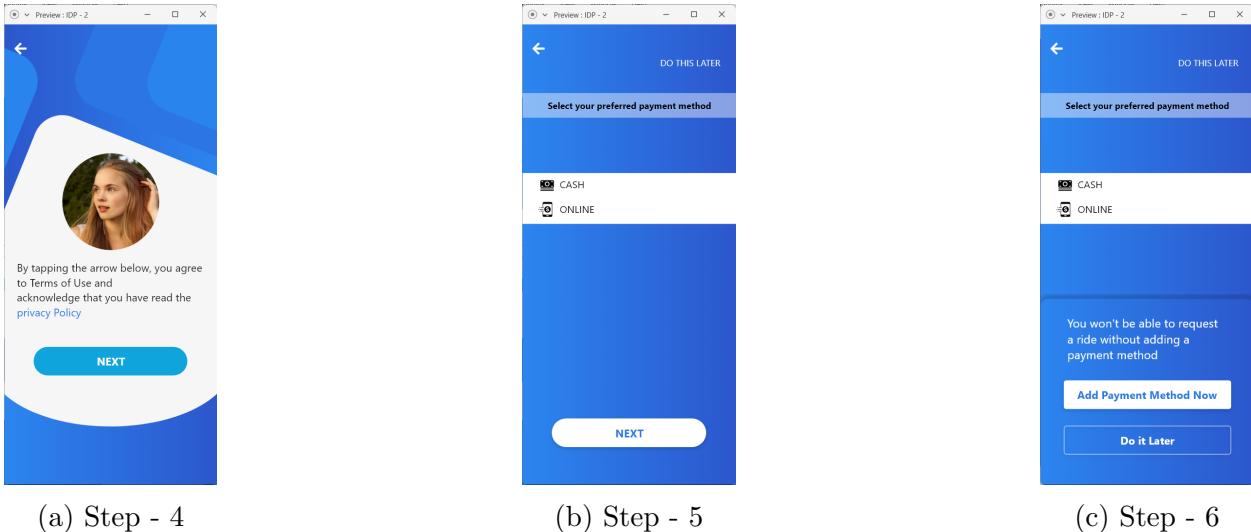


Figure 3.4: Steps for driver registration



Figure 3.5: Booking Process

- BLS - Basic Life Support
- PTS - Patient Transport Service
- ALS - Advanced Life Support

After selecting the ambulance category then select where is your pickup location and which hospital you want to go to by clicking "Where to?", these pages show in figure(3.6a & 3.6b). Then select a schedule for an ambulance that is visible by clicking on the "Clock symbol" in the figure(3.6c). For receive ride request driver must be online like figure(3.6d)

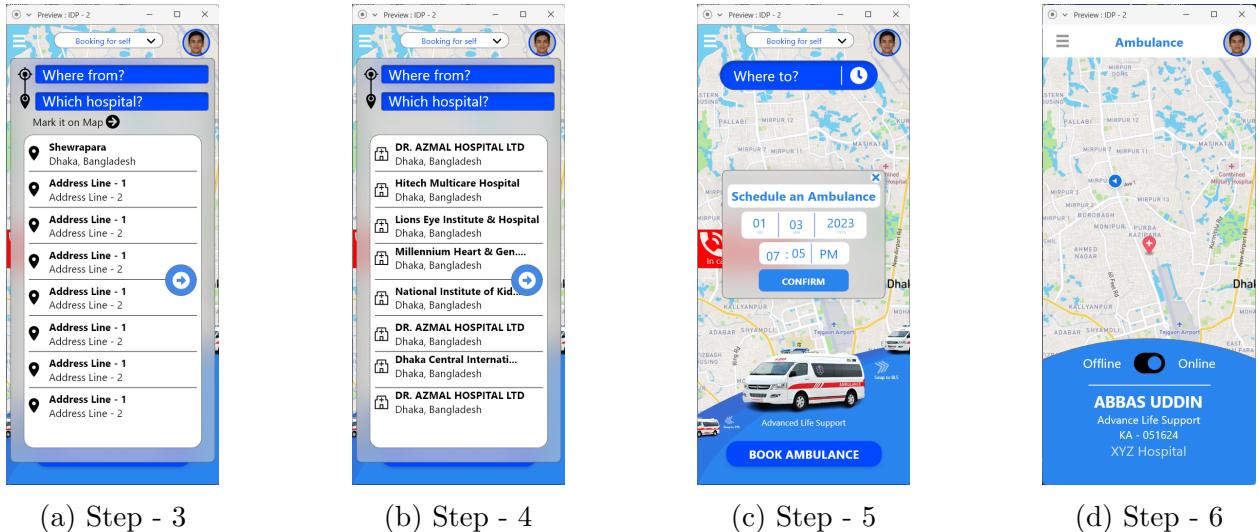


Figure 3.6: Steps for driver registration

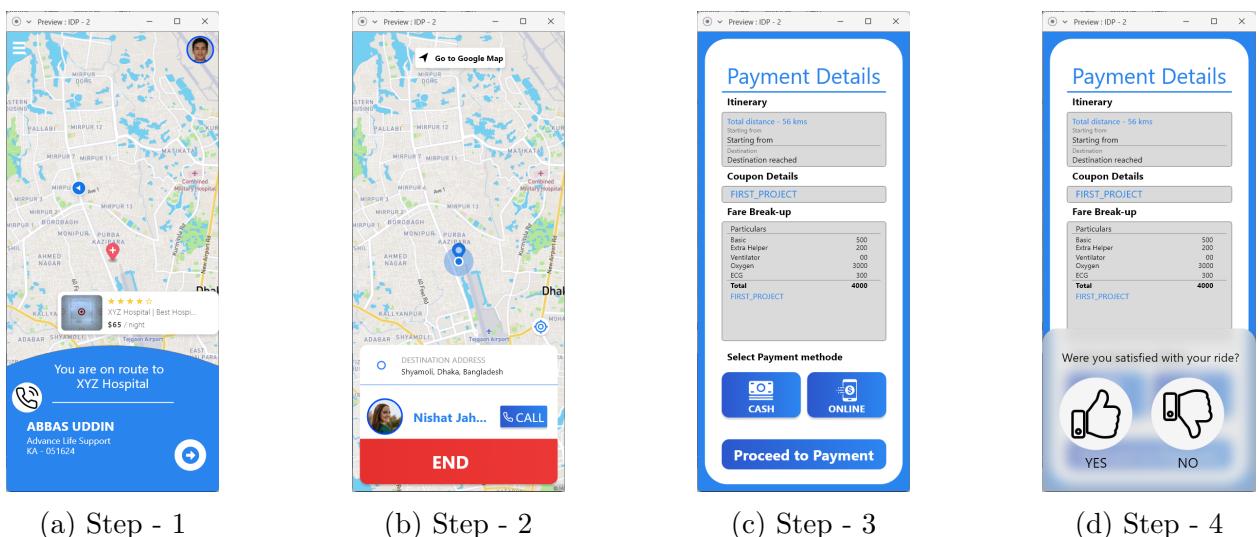


Figure 3.7: Steps for end ride

### 3.1.3 Ending Ride

When traveling an ambulance to their destination the user and driver show like figure(3.7a & 3.7b) respectively. Finally when they reach the destination location then the driver will click on the "END" button then user see the payment details and select any payment method between cash or online payment finally click on the "Proceed to Payment" button for payment like figure(3.7c). End of a ride user can rate this ride in figure(3.7d).

### 3.1.4 Ride Confirmation

When any user clicks on the "Book Ambulance" button the driver receives a ride request. If the driver accepts the ride then click on the "ACCEPT" button before timeout that is

shown in figure(3.5b). After accepting the ride driver reaches the user till the driver shows an interface like figure(3.8b). When reaching near the user then show an interface like figure(3.6a).

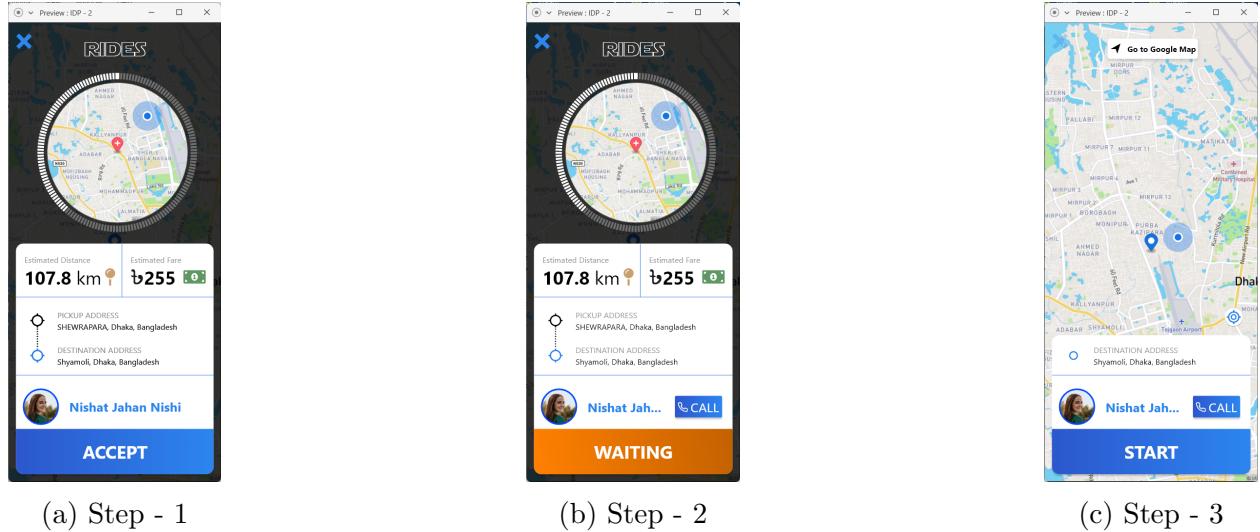


Figure 3.8: Steps for driver registration

### 3.1.5 Navigation bar



Figure 3.9: NavBar

In the Above figure(3.9) there are two interfaces. One is for the User and another is for the driver. That is designed for helping orient themselves in this application.

### 3.1.6 Payment & Ride History

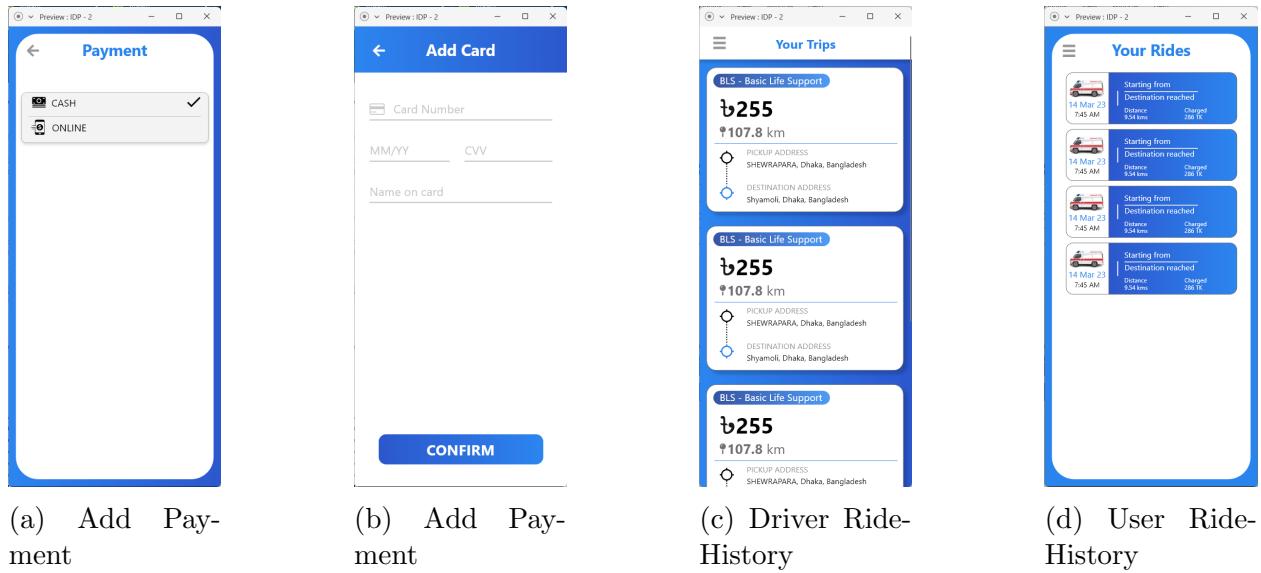


Figure 3.10: Add Payment & Ride History

Above in figure(3.10) two is for Add Payment steps in figure(3.10a & 3.10b) and another two is for to show the ride history for the user in figure(3.10d) and for driver in figure(3.10c).

### 3.1.7 Other design

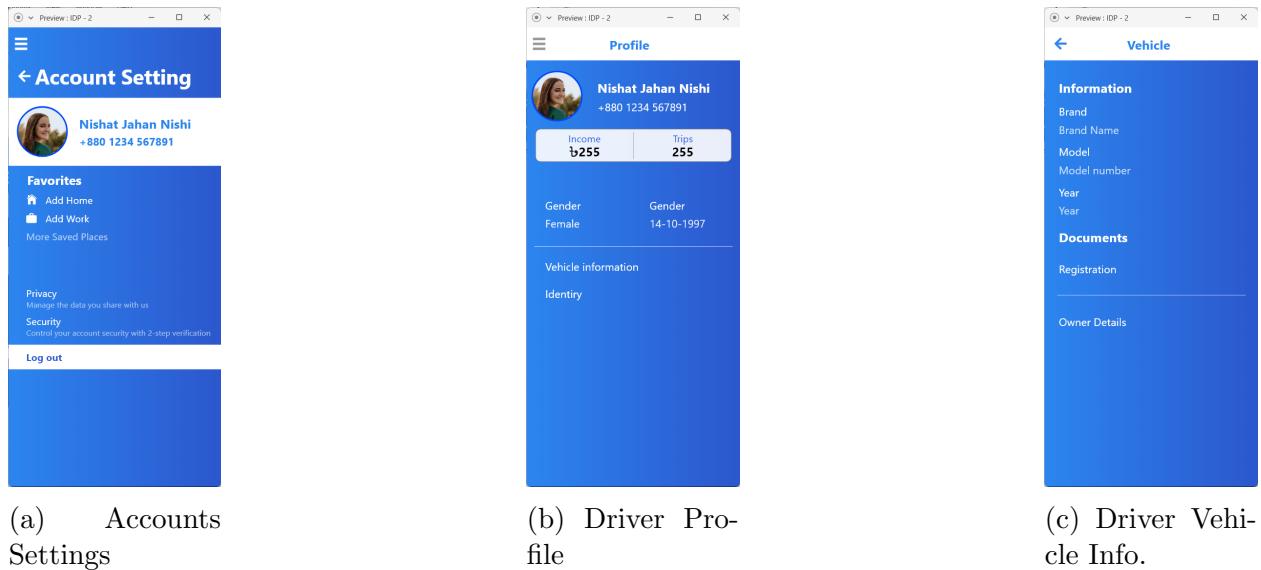


Figure 3.11: Other design

in figure(3.11) there are four figures. Figure(3.11a) is for account setting, figure(3.11b) is for driver profile and the last is the figure(3.11c) for driver vehicle information [?].

## 3.2 Working procedure

### 3.2.1 Registration



Figure 3.12: First page & Login page for all type of user

In figure(3.12a) user have to first sign up to our database. User does not have to give extra information. After clicking sign up it takes to “create your account section”

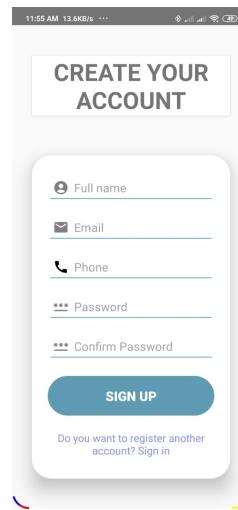


Figure 3.13: USER SignUP Page

In figure(3.13) we can see that user have to give full name, email address and phone number for sign up. User have to give password and confirm password for accounts security.

In figure(3.14) we can see that there are three page for driver registration. In first page driver have to give some basic information. In next page driver have to give company address and in third page driver have give address and NID information and also have to select the ambulance category.

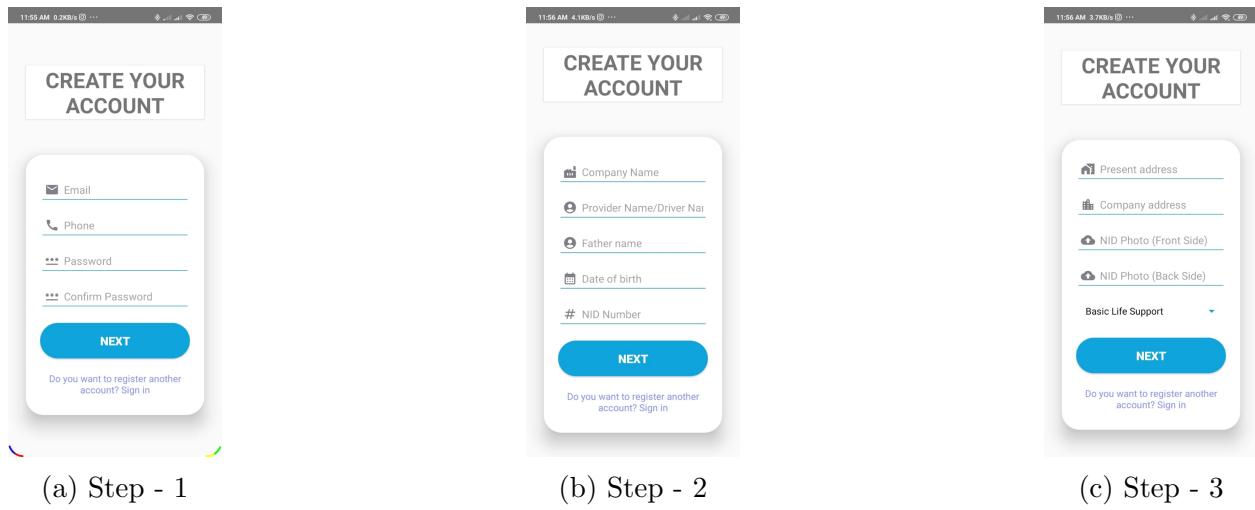


Figure 3.14: Steps for driver registration

### 3.2.2 Booking Request Process

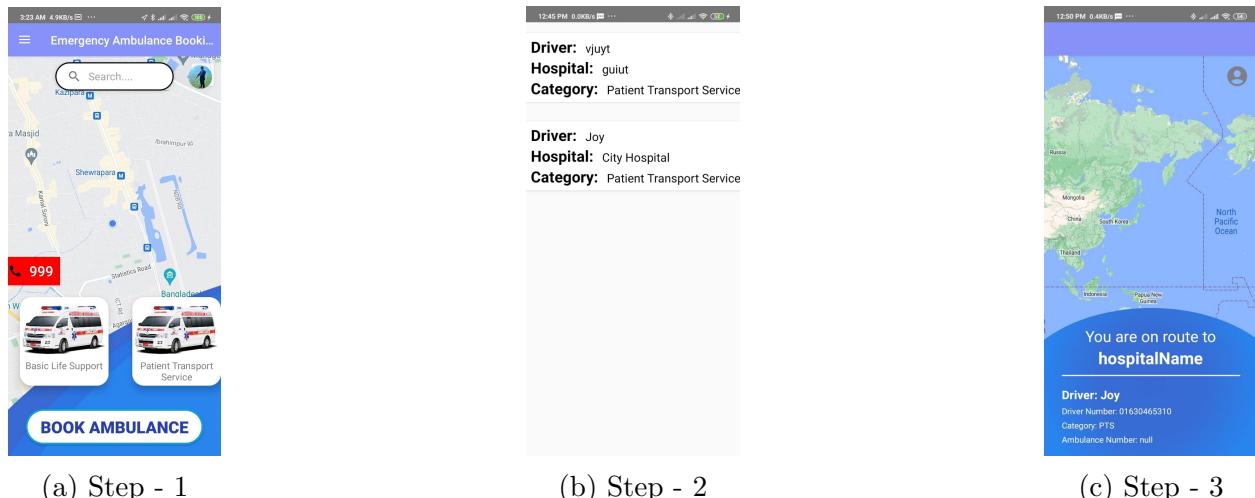


Figure 3.15: Booking Process

In figure(3.15 we can see the user interface of booking. In step- 1 user will search the pick up and drop off location then user will select the ambulance category, after clicking book ambulance it will take to the step- 2. In step-2 it will show the driver available list and after clicking a driver it will give request to the driver.

### 3.2.3 Booking Accept Process



Figure 3.16: Booking Process

In figure(3.16) shows the driver interface. In step-1 driver will be offline. In step-2 driver will see the booking information and he can accept the booking in 20s. After clicking accept it will go to the step-3. When he takes the patient he will click the button start after finish the ride he will click the end. After clicking end it goes to payment portion.

### 3.2.4 Payment and Rating



Figure 3.17: Booking Process

In figure(3.17) we can see the payment portion of driver and rating portion of the user. After getting payment driver will click received then user can give rating of the driver.

### 3.2.5 History



Figure 3.18: Booking Process

In figure(3.18) user can see the ride history after clicking history button.

## 3.3 Discussion/Analysis/Conclusion

The “Emergency Ambulance Booking Software” [?] is a project where users may register, log in, check the driver list, and book ambulances. Both the user and the driver benefit from utilizing this software because it allows them to read hospital information and the driver to receive trip information in a digital system.

# Chapter 4

## Test Cases

### 4.1 Introduction

The importance of test cases in a software project cannot be overstated. Test cases help ensure that the software functions as intended, meets the requirements, and behaves correctly under various scenarios. They play a crucial role in identifying and fixing defects, improving software quality, and enhancing user satisfaction. In this lab report, we present the test cases for the Emergency Ambulance Booking App.

### 4.2 Objectives

The objectives of this test case generation are as follows:

- To validate the login and registration module.
- To test the selection of pickup and drop-off locations.
- To verify the ambulance booking process.
- To validate the total rent based on distance and ambulance category.

### 4.3 Login/registration test cases:

| Step # | Name Step Details | Email Step Details        | Phone Number Step Details | Password Step Details | Confirm password Step Details | Actual Results  | Pass / Fail / Not executed / Suspended | Remarks                               |
|--------|-------------------|---------------------------|---------------------------|-----------------------|-------------------------------|---|--|---------------------------------------|
| 1      | Given name=j@y    | joymunshi46@gmail.com     | 11111111111               |                       | 123                           | 321   | Fail                                   | Name and Number field does not checks |
| 2      | jahid             | jahid .com                | 1814521457                |                       | 123456                        | 123456  | Pass                                   |                                       |
| 3      | Mubin@.com        | Mubin@gmail.com           | 15112345678411            | Mubin@1234            | Mubin@1234                    | Name can not use special character,Email should have @.Phone number should have 11 digit. | Fail                                   |                                       |
| 4      | Anna_Jose_Samul   | Anna_Jose_Samul@yahoo.com | 12121212121211q           | Pst_98                | Pst_98                        | Name can not use special character,Email should have @.Phone number should have 11 digit. | Fail                                   |                                       |
| 5      | Pet               | exampleexam ple.com       | 0121212121211             | %XYZabc%              | %XYZabc%                      |   | Fail                                   |                                       |
| 5      | Pet               | exampleexam ple.com       | 0121212121211             | %XYZabc%              | %XYZabc%                      |   | Fail                                   |                                       |

|    |              |                                  |                |          |          |  |      |  |
|----|--------------|----------------------------------|----------------|----------|----------|--|------|--|
| 7  | NULL         | @domain.com                      | 48448848715487 | Pst_9890 | Pst_9890 | Not As Expected, entered Special Character | Pass |  |
| 8  | Jippy_45     | email@domain.com                 | 454541248487   | tgress   | tgress   | Not As Expected, entered Special Character | Fail |  |
| 9  | Rose_merry9  | email@domain.com (Joe Smith)     | 1545784578     | 68778#5  | 68778#5  | Not As Expected, entered Special Character | Fail |  |
| 10 | robin_32     | rbn.email@domain.com             | 154789567      | 54667#5  | 54667#5  | Not As Expected, entered Special Character | Fail |  |
| 11 | lowra_12     | email.domain.com(lilas)          | 76230651       | 11230#5  | 11230#5  | Not As Expected, entered Special Character | Fail |  |
| 12 | markram_28   | email@domain.com(alideinM)       | 147345277      | 28341#5  | 28341#5  | Not As Expected, entered Special Character | Fail |  |
| 13 | Harry_brook9 | email@domain.com(HB)             | 769873520      | 90219#5  | 90219#5  | Not As Expected, entered Special Character | Fail |  |
| 14 | depok_HD2    | email@domain.com(hooda)          | 53970654       | 17432#5  | 17432#5  | Not As Expected, entered Special Character | Fail |  |
| 15 | cammeron_    | email@domain.com                 | 34860122       | 49259#5  | 49259#5  | Not As Expected, entered Special Character | Fail |  |
| 17 | Joy          | email@domain.com(sarah ali khan) | 65295266356    | 68458#   | 68458#5  | Password and confirm password should match | Pass |  |
| 18 | peter_10     | email@domain.com(perker)         | 5739525739     | 24874#5  | 24874#5  | Not As Expected, entered Special Character | Fail |  |
| 19 | mike_110     | email@domain.com (mitchell)      | 574833648      | 42956#5  | 42956#5  | Not As Expected, entered Special Character | Fail |  |
| 20 | _BLANK_      | _BLANK_                          | _BLANK_        | _BLANK_  | _BLANK_  |  | Pass |  |

Figure 4.1: Login/Registration

- Test Case 1: Attempt to register with a name containing a special character, an email without '@', and a phone number with less than 11 digits. Expected result: Name should not allow special characters, email should have '@', and phone number should have 11 digits. Actual result: Fail (Name and number fields do not check for these conditions).
- Test Case 2: Register with a valid name, email without any special characters, and a phone number with 11 digits. Expected result: Successful registration. Actual result: Pass.
- Test Case 3: Register with a name containing a special character, a valid email, a phone number with more than 11 digits, and valid passwords. Expected result: Name should not allow special characters, phone number should have 11 digits. Actual result: Fail.
- Test Case 4: Register with a long name, a valid email, a phone number with 13 digits, and valid passwords. Expected result: Name should not allow more than a certain number of characters, phone number should have 11 digits. Actual result: Fail.
- Test Case 5: Register with a short name, an invalid email format, a phone number with 12 digits, and valid passwords. Expected result: Name should not allow less than a certain number of characters, email should have a valid format, phone number should have 11 digits. Actual result: Fail.
- Test Case 6: Register with a blank name, an invalid email format, a phone number with 12 digits, and valid passwords. Expected result: Name should not be blank, email should have a valid format, phone number should have 11 digits. Actual result: Fail.
- Test Case 7: Register with a null name, an email with a special character at the beginning, a long phone number, and valid passwords. Expected result: Not as expected (Entered special character in the email). Actual result: Pass.
- Test Case 8 to 16: Register with different combinations of names, emails, phone numbers, and passwords, checking for special characters and password matching. Expected result: Not as expected (Entered special characters, invalid email format, or passwords don't match). Actual result: Fail.

## 4.4 Location from map test cases:

| Step # | Pick Up Location     | Drop Off Location    | Ambulance Category | Actual Results   | Pass / Fail / Not executed / Suspended | Remarks |
|--------|----------------------|----------------------|--------------------|--|--|---------|
| 1      | Inner Bangladesh     | Inner Bangladesh     | BLS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 2      | Inner Bangladesh     | Inner Bangladesh     | PTS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 3      | Inner Bangladesh     | Inner Bangladesh     | ALS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 4      | Inner Bangladesh     | Inner Bangladesh     | Not Selected       | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 5      | Inner Bangladesh     | Not Inner Bangladesh | BLS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 6      | Inner Bangladesh     | Not Inner Bangladesh | PTS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 7      | Inner Bangladesh     | Not Inner Bangladesh | ALS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 8      | Inner Bangladesh     | Not Inner Bangladesh | Not Selected       | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 9      | Not Inner Bangladesh | Inner Bangladesh     | BLS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 10     | Not Inner Bangladesh | Inner Bangladesh     | PTS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 11     | Not Inner Bangladesh | Inner Bangladesh     | ALS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 12     | Not Inner Bangladesh | Inner Bangladesh     | Not Selected       | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 13     | Not Inner Bangladesh | Not Inner Bangladesh | BLS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 14     | Not Inner Bangladesh | Not Inner Bangladesh | PTS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 15     | Not Inner Bangladesh | Not Inner Bangladesh | ALS                | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |
| 16     | Not Inner Bangladesh | Not Inner Bangladesh | Not Selected       | Should INNER Bangladesh and should have the ambulance category | Pass                                   |         |

Figure 4.2: Location

- Test Case 1 to 16: Selecting pickup and drop-off locations within and outside Inner Bangladesh, choosing different ambulance categories. Expected result: The pickup and drop-off locations should be within Inner Bangladesh, and the selected ambulance category should be displayed. Actual result: Pass.

## 4.5 Ambulance Booking Test Cases:

| Step # | Test Steps   | Expected Results  | Actual Results   | Pass / Fail / Not executed / Suspended | Remarks |
|--------|--|---|--|--|---------|
| 1      | Select the "Book Ambulance" option from the home screen.                               | The "Book Ambulance" option is successfully selected.   | Selected   | Pass                                   |         |
| 2      | Enter the required details (pickup location, destination, contact number, date, time). | The required details (pickup location, destination, contact number, date, time) are accurately entered. | without Pick up and Drop off are not selected          | Pass                                   |         |
| 3      | Select the preferred ambulance type (e.g., Basic Life Support, Advanced Life Support). | The preferred ambulance type is successfully selected.  | The preferred ambulance type is successfully selected. | Pass                                   |         |
| 4      | Click on the "Book Now" button.  | The "Book Now" button is clicked.   | The "Book Now" button is clicked.                      | Pass                                   |         |

Figure 4.3: Ambulance Booking

- Test Case 1: Select the "Book Ambulance" option from the home screen. Expected result: The "Book Ambulance" option is successfully selected. Actual result: Pass.
- Test Case 2: Enter the required details (pickup location, destination, contact number, date, time). Expected result: The required details are accurately entered. Actual result: Not executed (without pick-up and drop-off locations).

- Test Case 3: Select the preferred ambulance type. Expected result: The preferred ambulance type is successfully selected. Actual result: The preferred ambulance type is successfully selected. Pass.
- Test Case 4: Click on the "Book Now" button. Expected result: The "Book Now" button is clicked. Actual result: The "Book Now" button is clicked. Pass.

## 4.6 Total rent calculation test cases:

|        |                 | BLS<br><5km | 1200               | BLS<br>>5km               | Distance*300                           |         |
|--------|-----------------|-------------|--------------------|---------------------------|--|---------|
|        |                 | PTS<br><5km | 1200               | PTS<br>>5km               | Distance*500                           |         |
|        |                 | ALS<br><5km | 1200               | ALS<br>>5km               | Distance*700                           |         |
| Step # | Pickup Location | Destination | Ambulance catagory | Expected Results          | Pass / Fail / Not executed / Suspended | Remarks |
| 1      | Kazipara        | Shahbag     | PTS                | 6.9km and bill 3465.68 tk | Pass                                   |         |
| 2      | Mirpur 10       | Farmget     | BLS                | 5.9km and 1780.72         | Pass                                   |         |
|        | ECB Chattar     | Uttara      | ALS                | 6.14km and 4296.93        | Pass                                   |         |
| 4      | Mirpur12        | Airport     |                    | Not show the Bill         | Pass                                   |         |
| 5      | Dhan            | Ghulshan    | BTS                | Not show the Bill         | Pass                                   |         |
| 6      | Narayongonj     | Shahbag     | AAA                | Can not select AAA        | Pass                                   |         |
| 7      | Gazipur         | Mohammadp   | PTS                | Not show the Bill         | Pass                                   |         |
| 8      | Mohakhali       |             |                    | Can not book              | Pass                                   |         |
| 9      |                 | Dhanmondi   |                    | Can not book              | Pass                                   |         |
| 10     | NULL            | NULL        | NULL               | Can not book              | Pass                                   |         |
| 11     | kawron bazar    | Mirpur      | BLS                | 8.37 and 2509.87          | Pass                                   |         |
| 12     | Gulistan        | shahbagh    | ALS                | 2.7km and 1200            | Pass                                   |         |

Figure 4.4: Rent Value

The fixed rent for a distance of up to 5km is 1200 taka. For distances exceeding 5km, the rent for Basic Life Support (BLS) is calculated as the distance multiplied by 300 taka. For patient Transport Service (PTS), the rent is calculated as the distance multiplied by 500 taka. And for Advanced Life Support (ALS), the rent is calculated as the distance multiplied by 700 taka.

- Test Case 1 to 3: Enter different pickup locations, destinations, and ambulance categories. Expected result: The calculated distance and rent should match the given values for each ambulance category. Actual result: Pass.
- Test Case 4 and 5: Enter pickup and drop-off locations without selecting an ambulance category. Expected result: The bill should not be displayed. Actual result: Pass.
- Test Case 6 to 9: Enter different pickup and drop-off locations, including invalid ambulance categories. Expected result: The system should not allow selecting an invalid ambulance category. Actual result: Pass.
- Test Case 10 to 16: Enter different pickup and drop-off locations, including null values. Expected result: The system should not allow booking with null values. Actual result: Pass. User profile test cases: This module is not implemented, so there are no test cases.

#### **4.6.1 Payment module:**

This module is not implemented, so there are no test cases.

#### **4.6.2 Conclusion of Test Cases**

These test cases cover various scenarios related to login/registration, location selection, ambulance booking, and total rent calculation. The expected results are mentioned for each test case, and the actual results are provided to determine if the test case passed, failed, was not executed, or suspended.

## 4.7 Discussion

Based on the test cases conducted, the following observations can be made:

The login and registration module incorporates a validation mechanism to ensure the proper usage of special characters in the name field. This feature helps maintain data integrity and prevents potential issues arising from invalid inputs.

Regarding the selection of pickup and drop-off locations, the system operates smoothly, ensuring accurate booking processes, except in cases where the same location is chosen for both pickup and drop-off. In such instances, the system recognizes the error and prompts the user to select distinct locations, guaranteeing the reliability of the booking process.

The ambulance booking process functions flawlessly for valid categories, ensuring that users can make bookings efficiently. The system efficiently handles the various categories available for ambulances, providing a seamless experience to users while ensuring their needs are appropriately met.

Furthermore, the system accurately calculates the total rent based on the distance and selected ambulance category. The integration of these factors ensures that users receive accurate pricing information and eliminates any discrepancies in the calculation process. This precision enhances user satisfaction and transparency within the ambulance booking system. Further testing and improvements can be made to enhance the application's robustness and user experience.

# **Chapter 5**

## **Conclusion and Future Work**

### **5.1 Introduction:**

This project is an "emergency ambulance booking System". So from the title of the project it is clear to everyone that this is a app-based project. Here, we not only create an ambulance booking system but also maintain a strong connection between the driver and the user.

### **5.2 Concluding Remarks:**

In this project, since we have completed our project without any errors and in Performance Review (Chapter 2.9), we have tested our System and project performance and they i have found that our project is working perfectly. So we can say that our project is perfect.

### **5.3 Practical Implementation:**

In this project we see that our project is an emergency ambulance booking System. By using this System we can check hospital information, we can book ambulances and also we can check driver information. So this project is very practical and very important these days.

### **5.4 Future Plan of the Work:**

Our project is the "Emergency Ambulance app" and in Chapter 2.9, we evaluated our project and confirmed that our project was completely error-free and in perfect working order. We want to add family members for emergency contact, and it will send the location to every family member. We also want to add hospital suggestion system. No problem, but if we update some features like the digital payment section and driver review, then this project will be more efficient.

## **5.5 Summary of the Project:**

This project is an “Emergency ambulance booking System” and here the user can log in, register and check the driver list and also order an ambulance. Users can also view hospital information through this System and drivers also get rides in a digital system so using this System both benefits. Some needed updates in this System like the user reviews section, digital payment section, etc. We have discussed these options and have decided that in a future update we will add these features.

# References

- [1] Cardio Care BD. Retrieved from <https://cardiocarebd.com/>
- [2] Ambulance Service Bangladesh. Retrieved from <https://ambulanceservicebangladesh.com/>
- [3] Daffodil International University. (2013). A Study on Emergency Ambulance Service in Dhaka City. Retrieved from <http://dspace.daffodilvarsity.edu.bd:8080/handle/123456789/7167>
- [4] TeX Stack Exchange. (2011). A layout for writing a budget. Retrieved from <https://tex.stackexchange.com/questions/36017/a-layout-for-writing-a-budget>
- [5] Nevon Projects. Emergency Ambulance Booking Android App. Retrieved from <https://nevonprojects.com/emergency-ambulance-booking-android-app/>
- [6] ResearchGate. Retrieved from <https://www.researchgate.net/>