

SHIPX - AN EXPRESS DELIVERY SYSTEM

BY

FAHIM RAHMAN

ID: 173-15-927

AND

MD. ARMAN HOSSAN

ID: 173-15-923

This Report is Presented in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Nadira Anjum Nipa

Lecturer

Department of CSE

Daffodil International University

Co-Supervised By

Mohammad Monirul Islam

Lecturer (Senior Scale)

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

September 2022

APPROVAL

This Project/internship titled “**SHIPX - AN EXPRESS DELIVERY SYSTEM**”, submitted by Fahim Rahman (ID:173-15-927), Md. Arman Hossan (ID:173-15-923) to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 12-09-2022.

BOARD OF EXAMINERS

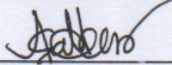
Dr. S M Aminul Haque

Associate Professor & Associate Head

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



Chairman

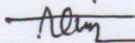
Mr. Abdus Sattar

Assistant Professor

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



Internal Examiner

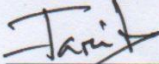
Al Amin Biswas

Senior Lecturer

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



Internal Examiner

Dr. Dewan Md. Farid

Professor

Department of Computer Science and Engineering

United International University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Nadira Anjum Nipa, Lecturer, Department of CSE**, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Nadira Anjum Nipa
Lecturer
Department of CSE
Daffodil International University.

Co-Supervised by:



Mohammad Monirul Islam
Senior Lecturer
Department of CSE
Daffodil International University.

Submitted by:

fahimrahman

Fahim Rahman
ID: 173-15-927
Department of CSE
Daffodil International University.

Akash

MD. Arman Hossan
ID: 173-15-923
Department of CSE
Daffodil International University.

ACKNOWLEDGMENT

First, we express our heartiest thanks and appreciation to Almighty God for His divine Blessing makes it possible to finish last year's venture effectively.

We are really grateful and wish to express our profound indebtedness to **Nadira Anjum Nipa, Lecturer**, Department of CSE, Daffodil International University, Dhaka. This project is carried out with the deep knowledge and keen interest of our supervisor in the field of learning model structure. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages, have made it possible to complete this project.

We would like to express our heartiest gratitude to Prof. Dr. Touhid Bhuiyan, Head, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire coursemates at Daffodil International University, who took part in this dialogue while completing the course work.

Finally, we should recognize with due regard the consistent help and patience of our work.

ABSTRACT

ShipX - An Express Delivery System is a Peer 2 Peer (P2P) parcel delivery service. Peer to Peer (P2P) parcel delivery is a rapidly emerging trend where travellers or crowdsourced travellers carry parcels from the senders to the receivers while undertaking a trip they already intended to take and receive a fee to contribute towards their travel costs. Our mission is to quickly and cheaply deliver goods from one place to another that you have always wanted. Basically, we connect you to travellers willing to carry your requested goods by the sender, all the while ensuring that your ShipX - An Express Delivery System experience is reliable and hassle-free. The way it works is that the sender selects their preferred destination and recipient through the app, and the app automatically searches for nearby travellers. As this is a pay-first service, the payment must be completed through the app before handing over the parcel to the traveller. Then the traveller shall make sure that the item(s) is/are correct, the service charge is paid, and that they are not violating any rules of ShipX - An Express Delivery System. After reaching the destination, the recipient will check if the item is correct and confirm it as received on the app. The traveller will immediately receive the payment. The sender and receiver will be able to know the location of the item or items while shipping. ShipX - An Express Delivery System is secure because we collect all the necessary credentials of senders, travellers, and receivers while they sign up.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of Examiners	ii
Declaration	iii
Acknowledgement	iv
Abstract	v
Table of contents	vi-viii
List of figures	ix-x

CHAPTER	PAGE
CHAPTER 1: Introduction	1- 2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
CHAPTER 2: Background	3-4
2.1 Introduction	3
2.2 Related work	3
2.3 Comparative Studies	3
2.4 Challenges	4

CONTENT	PAGE
CHAPTER 3: Requirement Specification	6-11
3.1 Requirement Collection and Analysis	5
3.2 Server Requirement	5
3.3 Client Requirement	5
3.4 Operating System	5
3.5 Developing Tools	6
3.6 Use Case Modelling and Description	10
3.7 Design Requirement	11
CHAPTER 4: Design Specification	12-13
4.1 SwiftUI Component: Views and Control	12
4.2 SwiftUI-View Layout and Presentation	13
CHAPTER 5: Backend Specification	16-18
5.1 baseURL	16
5.2 Authentication	16
5.3 User	17
5.4 Location	17
5.5 Posts	18
5.6 Admin	18

CHAPTER 6: User Interfaces	20-40
6.1 Login Register Selection View	20
6.2 Register View	21
6.3 Login View	22
6.4 Forgot Password View	23
6.5 Traveller Travel Post View	24
6.6 Requests View	30
6.7 Details View	31
6.8 Profile View	33
6.9 Find Traveller View	34
6.10 Shipping Details View	39
6.11 Others	40
CHAPTER 7: Working Procedure	41-42
7.1 Common user	41
7.2 Traveller	41
7.3 Sender	42
7.4 Receiver	42
CHAPTER 8: Conclusion and Future Scope	43
8.1 Conclusion	43
8.2 Future Scope	43
References	45
Plagiarism Report	47

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.6.1: Use case model	10
Figure 3.6.2: Use case model for sender, traveller and receiver	11
Figure 6.1: Login Register Selection View	20
Figure 6.2: Register View	21
Figure 6.3: Login View	22
Figure 6.4: Forgot Password View	23
Figure 6.5.1: Traveller's Travel Post view	24
Figure 6.5.2: Traveller's Travel Post view	25
Figure 6.5.3: Search Division View	26
Figure 6.5.4: Search District View	27
Figure 6.5.5: Search for Upazila View	28
Figure 6.5.6: Search City Corp. Area View	29
Figure 6.6: Request View	30
Figure 6.7.1: Accepted Request Details View	31
Figure 6.7.2: Accepted Request Details View	32
Figure 6.8: Profile View	33
Figure 6.9.1: Find Traveller View	34

Figure 6.9.2: Available Travellers View	35
Figure 6.9.3: Send Request View	36
Figure 6.9.4: Product Category Selection View	37
Figure 6.9.5: Send Request Alert	38
Figure 6.10: Shipping Details View	39
Figure 6.10.1: Shipping Details View	40

CHAPTER 1

INTRODUCTION

1.1 Introduction

The mobile application industry is a multi-billion dollar industry and one of the fastest-growing industries all over the world. In 1994, the first mobile app was built (1). Nowadays, there are a lot of operating systems for smartphones, such as - Android, iOS, Tizen, Sailfish OS, Ubuntu Touch, Plasma Mobile, PureOS, PostmarketOS, and KaiOS. Among all of them, the most popular operating systems are Android and iOS. 70.05% of people use the Android OS and 29.21% use iOS (2). There are several types of mobile applications. For example Lifestyle, Social Media, Utility, Games / Entertainment, Productivity, News / Information, Maps, and Navigation, Education, Business, Finance, Food, etc. Business is one of the most popular and useful categories of all types of apps. Any courier or shipping-related apps are under the Business category. In this chapter, we will discuss why we were motivated to develop this project, which falls under the Business category, and the contribution we will make towards the sector of the shipping industry.

1.2 Motivation

The motivation for ShipX - An Express Delivery System came from Backpack (3). The backpack is a P2P (Peer to Peer) product delivery service. But Backpack offers to deliver items only from abroad to locals. So we thought we could use the same kind of service inside a country. All over the world, 704.7 million people use traditional courier services every day (4). We need to depend on a courier service for this process. Here comes the P2P service. No courier service is needed, and the shipping process is much faster. The most amazing part is that anyone can earn from ShipX - An Express Delivery System, by becoming a traveller. There are no registration fees or any extra charges. The traveller just has to carry a small and light package with him towards his / her destination. We

could lower the courier costs by 50% or less through our P2P shipping service. Packages are anonymous to the travellers, and we collect all government IDs and other credentials for a secure experience

1.3 Objectives

1.3.1 Improved User Satisfaction

Our first objective is to provide the easiest, fastest, and most cost-effective service. By providing unique features and the most advanced mobile applications, we aim to make our users satisfied with our service.

1.3.2 Users' Experience

The main attraction of our service is the mobile application. We have used the latest and greatest technology throughout the app. Our backend service is also customizable and scalable. So as the user grows, our servers can easily handle every single request.

1.4 Expected Outcome

After passing the maximum part of the feasibility case study, we are at a stage that we could only imagine.

Sender: The sender is one of the actors in the service. The duty of a sender is to request a package handover and pay for it.

Traveller: The traveller is the most important actor in ShipX - An Express Delivery System. The traveller takes the package from the sender, then visually verifies it, delivers it to the receiver, and gets paid.

Receiver: The receiver has no work other than receiving the product from the selected location and confirming it.

CHAPTER 2

BACKGROUND

2.1 Introduction

The way it works is that the sender selects their preferred destination and recipient through the app, and the app automatically searches for nearby travellers. As this is a pay-first service, the payment must be completed through the app before handing over the parcel to the traveller. Then the traveller shall make sure that the item(s) is/are correct, the service charge is paid, and not violate any rules of ShipX - An Express Delivery System. After reaching the destination, the recipient will check if the item is correct and confirm it as received on the app. The traveller will then immediately receive the payment. The sender and receiver will be able to know the location of the item or items while shipping.

ShipX - An Express Delivery System is secure because we collect all the necessary credentials of senders, travellers, and receivers while signing up.

2.2 Related Works

There are a lot of P2P shipping services all over the world. Peer5, Bring4You, and SendThruMe are the most renowned companies. In Bangladesh, the only P2P shipping service is eCourier, but they haven't started their service yet.

2.3 Comparative Studies

Our service has some advantages and disadvantages. The following advantages and disadvantages are

Advantages:

1. User-friendly Mobile Application.
2. Easiest and Fastest service.
3. Most cost-efficient service.
4. Use of crowdsourcing.
5. We have 0% carbon emissions since we are not directly using any motor vehicles for the service.

Disadvantages:

1. Fewer travellers may delay the shipping process.
2. Users can't send packages over 5 kg.
3. Packages must be in the selected categories.

2.4 Challenges

We faced a few challenges when we were developing our project, as it was huge, and we had developed a custom backend for it.

1. Fully Customizable and Scalable The backend and cloud services are fully customizable and scalable.
2. Brand New SwiftUI Framework.
3. Optimizing both the frontend and the backend for a smooth experience.
4. User-friendly UI-UX

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Requirement Collection and Analysis

This part covers the prerequisites of our project "ShipX - An Express Delivery System". For the server-side, we needed to use a framework that can handle a lot of requests and maximum uptime. For the client part, we figured out that the latest framework from Apple would work best. After doing some research, we decided to use a NoSQL database, which is perfect for our job. We wanted a superfast response time, so instead of storing photos on the main database, we separated them and used a dedicated cloud server.

3.2 Server Requirement

Software: Linux Operating System, NodeJS, MongoDB Subscription, Cloudinary Subscription, Node Dependencies.

Hardware: 2.0 GHz Multi Core CPU (Minimum), 2 GB of RAM (minimum), 10 GB of Storage (Minimum).

3.3 Client Requirement

Software: macOS 12.0 (minimum),
iOS 15.0 (minimum).

Hardware: Macintosh 2017 (minimum),
iPhone 6S (minimum).

3.4 Operating System

macOS

macOS is an Apple-developed proprietary graphical operating system since 2001. It is the primary OS for Apple's computers. It is the second most widely used desktop operating system on the market for desktop and laptop computers, after Microsoft Windows and Chrome OS. In March 2001, the first desktop version, Mac OS 10.0, was released. In late

2020, Apple began the Apple Silicon transition, using self-designed, 64-bit ARM-based Apple M1 processors on new Mac computers (5).

Linux

Linux is a Unix-like and primarily POSIX-compliant portable computer operating system (OS) that is assembled under the model of free and American Standard Code for Information Interchange document package development and distribution. The method part of the OS is that the OS kernel, associated OS kernel, was initialized on a day in 1991 by Linus Torvalds (6).

Web Server

A Web server is a program that utilizes the Hypertext Transfer Protocol to serve documents to clients. Furthermore, committed PCs and machines could likewise be referenced as web servers. In light of the solicitations that are sent by their PC's HTTP customers, a web server also uses different communication protocols according to the requirements (7).

3.5 Developing Tools

Xcode (IDE)

Xcode is an Integrated Development Environment (IDE) applicable for macOS only. This IDE is at the center of the Apple development experience. It was released by Apple in late 2003. Currently, the latest version, 13.2.1, was released on December 17, 2021, and is available in the Mac App Store for free via the new macOS Monterey. Xcode is used to develop software for macOS, iOS, iPadOS, tvOS, and watchOS. It supports source code for the programming languages C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit (Rez), and Swift, with different types of programming models. It also supports Cocoa, Carbon, and Java. It has integrated built-in support for source code management systems, which use the git version control system and protocol. Xcode uses a different type of SDK to compile and debug the different platform applications. For iOS applications, it uses the iOS SDK; for macOS it uses the macOS SDK; for watchOS it uses the watchOS SDK; for iPad iPadOS SDK, and for tvOS it uses the tvOS SDK (8)

VS Code (Code Editor)

VS Code is also known as Visual Studio Code. It's a source code editor made by Microsoft. It's available for Windows, Linux, and macOS. Visual Studio Code was initially released on April 29, 2015. It's a powerful and lightweight source code editor. In

VS Code, users can customize it by customizing the theme, changing the keyboard shortcuts, using the preferences, and installing extensions that add extra functionality. There are many features, like support for syntax highlighting, debugging, snippets, intelligent code completion, code refactoring, snippets, and embedded Git. It allows users to work in multiple directories and save them for future use. It works as a language-agnostic code editor for other languages as well. When a code is run in it, its output, or if there is an error, is displayed in its terminal. The user can fix it by looking at the error message in the terminal (9).

JavaScript (Programming Language)

JavaScript is abbreviated to JS. JS appeared on December 4, 1995. It's a programming language, and that's a core technology of the WWW (World Wide Web). Around 97% of websites use JavaScript for client sites to maintain web page behavior. It supports third-party libraries. Nowadays, major browsers have a dedicated JavaScript engine to execute code on their devices. JavaScript is a high-level language as well as a just-in-time compiled language that has ECMAScript standards. It has an application programming interface. With that, the user can work with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). The JavaScript engine is mainly used in web browsers, but now it is being used in servers and some varieties of applications (10).

Swift (Programming Language)

Swift is a powerful programming language. It is a general-purpose, multi-paradigm, compiled programming language that was developed by Apple Inc. Swift first appeared on June 2, 2014. Swift was a replacement of objective-C. It works with the help of Apple's Cocoa and Cocoa Touch frameworks. It's built-in with the Xcode and LLVM compiler framework. Swift uses the runtime library of Objective-C, which helps to run Swift, C, C++, and Objective-C in one program, which is Xcode. This is a case-sensitive language, and in Swift, there hasn't been any issue with the syntax. If the user wants to use syntax, they can, but if the user doesn't want to use syntax, it's the user's wish. The latest version of Swift is Swift 5.5, which was announced by Apple in 2021 at WWDC (11).

Express (Framework)

Express is open-source software that is published under the MIT licence, and it's free for all. It is the most popular backend web application framework for Node.js, also known as the de facto standard server framework for Node.js. It is designed to create web applications and APIs. That is also known as the de facto standard server framework for Node.js, which is described (by the original author, TJ Holowaychuk), as a Sinatra-inspired server. Some popular stacks, such as MEAN, MERN, or MEVN, component database software and a JavaScript library, which are collectively called

Express. Some features of this are robust routing, concentrated on high-performance, content negotiation, HTTP helpers (redirection, caching, etc.) (12).

SwiftUI (Framework)

Apple introduced SwiftUI as a brand-new framework for Apple products in 2019. It's available in Xcode, which is built-in. In this framework, a user can create or develop projects or apps for iOS, macOS, tvOS, and watchOS. So it is an internal cross-platform for Apple products. SwiftUI became popular rapidly after its release. In this framework, the user can see a live preview of his/her product. User app or product interface controls, views, and layout structures are provided by SwiftUI.

This framework provides some amazing features like gestures, taps, animation, and different types of input, which make the app or product very beautiful. SwiftUI also provides the tools that manage the necessary data flow for the app. Because of this, it has some powerful modifiers that are helpful for a user. In SwiftUI, users can customize and create modifiers as needed (13).

MongoDB (Database)

MongoDB is a document-oriented cross-platform database used to construct highly available and scalable internet applications. This was developed by MongoDB Inc. It's classified as a NoSQL database program. It is licensed by the Server Side Public Licence. It provides drivers for all major programming languages. Furthermore, it's popular for developing scale applications, It's built on a scale-out architecture. Developers can store structured or unstructured data easily. It uses the JSON format to store data. Nowadays, applications can directly read that data in this format. If the user wants, MongoDB does not require spending time configuring the database. Instead, the user can start building the application (14).

Cloudinary (SaaS)

Cloudinary is a SaaS tech company that provides cloud-based data management services. This company was founded in 2012 and Santa Clara, California, United States is its headquarters. It allows users to save photos and videos, as well as manage photos and videos for manipulation and distribution when the user needs them for websites and apps. It is used for about 1 million webs and mobile applications, and around 8,000 companies use this service to maintain their websites and applications (15).

Heroku (PaaS)

Heroku is a cloud platform based on containers. It was founded in 2007. At that time, it used to only support the Ruby programming language, but it is currently supported by several programming languages like Node.js, Python, Scala, PHP, Clojure, and Go. This can be called a polyglot platform for supporting multiple languages. Heroku provides

some features for developers. They can run, build, and scale their applications similarly across other languages. It is fully managed. With this, it gives liberty to the user to focus on their project without any kind of distraction like maintaining servers, hardware, or configuration (16).

Git (Version Control)

Version control mentions that it makes tracking and managing when a change occurs to the software code easier, and git has this feature. That's the reason git is a version control distributed framework. It is used for tracking; it is used to track changes when a user makes any changes in his files. The original author of Git is Linus Torvalds. It was initially released on April 7th, 2005. The stable version will be available from January 24, 2022. It's open-source and free software under a GPL-2.0-only licence. Git is very useful and popular with software developers because it allows them to do collaborative work. Most of the users of it use the terminal to control and access their GitHub account, and that's very fast. Speed, data integrity, and support for distributed, non-linear workflows are the aims of git. Git has some useful features like fully distributed VCS, data integrity, and security, Support Non-linear Development, and branching (17).

GitHub (Code Hosting Platform)

GitHub is a code hosting platform. Users need to install git in order to use GitHub. GitHub was founded on the 8th of February 2008. That platform provides internet hosting for software development and version control using git. Because of this, it's the most popular platform for storing git repositories and a standard site for coders. It's a cloud-based service that allows users to manage and maintain their open source projects when they collaboratively work. Each user needs an account to get access to GitHub, and that is the place where the user stores their codes by git commands. Some users use the GUI. When developers collaboratively work there, each and every developer can see the changes made by others. They can download it and contribute (18).

Docker (Virtualization Platform)

Docker is one kind of software development tool and also a visualization technology. The originator of it is Solomon Hykes. It was initially released on March 20th, 2013. The stable version of it is available, released on November 18th, 2021. It's a platform as a service. It is available for free, which is for the basic and paid option, which is the premium. In Docker, every software package is called a "container," and Docker uses OS-level virtualization to deliver software containers. Containers or packages are isolated from each other and bundle their own software, config files, and libraries, and they communicate with each other through well-defined channels. The containers are hosted by a piece of software called Docker Engine. Some benefits of using Docker are lightweight resources, Microsoft compact services, well-scaled, consistent development, and offering version control (19).

NodeJS (Runtime)

Node.js is a runtime environment for JavaScript that creates a non-blocking action for serving requests. For this, it does not waste time returning I / O requests. It's an open-source, cross-platform backend. It executes code using the V8 engine. Not only that, but it allows users to use the command-line tool for scripting and server-side to make dynamic web page content. Node.js is an excellent example of JavaScript. For that, it's called a single programming language. The developers of Node.js are the OpenJS Foundation. It was released on May 27th, 2009. The stable version is available from January 18th, 2022. It uses the .js extension for filenames. It has the capability of asynchronous I/O through an event-driven architecture. Furthermore, it was previously controlled by the Node.js distributed development project; now it has merged with the JS Foundation. It supports open-source libraries. Thousands of open source libraries are available for it, most of them provided from npm websites. Node.js supports many cloud-hosting platforms like Google Cloud Platform, Joyent, Jelastic, and others (20).

Others (Express & SwiftUI Dependencies)

Dependencies mean third-party code or packages that an application depends on. Express is a dependency for JavaScript. Applying dependency is known as dependency injection, which is a technique to make independent and stable modules. Express is a popular JavaScript framework. A user needs to install the NPM packages to use this. Users have to install and download the NPM packages by using the terminal.

SwiftUI is a new framework by Apple. SwiftUI also has some dependencies, which means it requires some third parties libraries to be injected into Swift when the developers need them. The reasons for that are:

1. Helps to test demo data.
2. Help to maintain Swift standard APIs
3. Don't need to use force to unwrap
4. The package does not reveal anything personal or internal through which the standard is maintained.

3.6 Use Case Modelling and Description

UML use case diagrams can be used to describe the functionality or process of a system in a horizontal way and represent the details of individual features of a system.

3.6.1 Use Case Model

A use case chart is the essential type of framework necessary for another product program to be immature. Use cases indicate the normal course of conduct (what), and not

the definite strategy for getting it going (how). Use cases, once indicated, can be meant as both literary and visual portrayals (for example, UML). A key idea of utilization case demonstration is that it causes us to plan a framework from the end client's point of view.

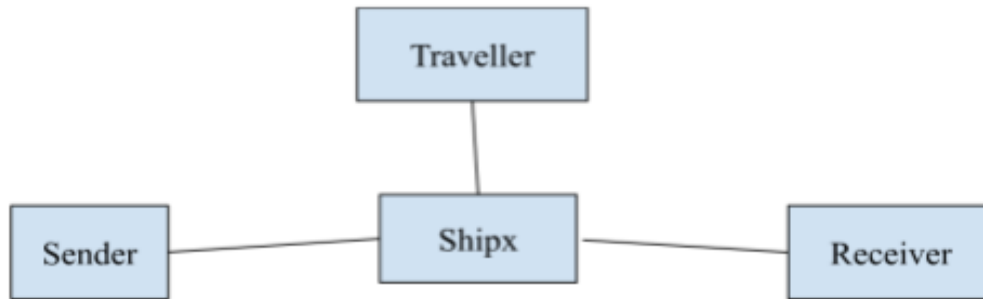


Figure 3.6.1: Use case model

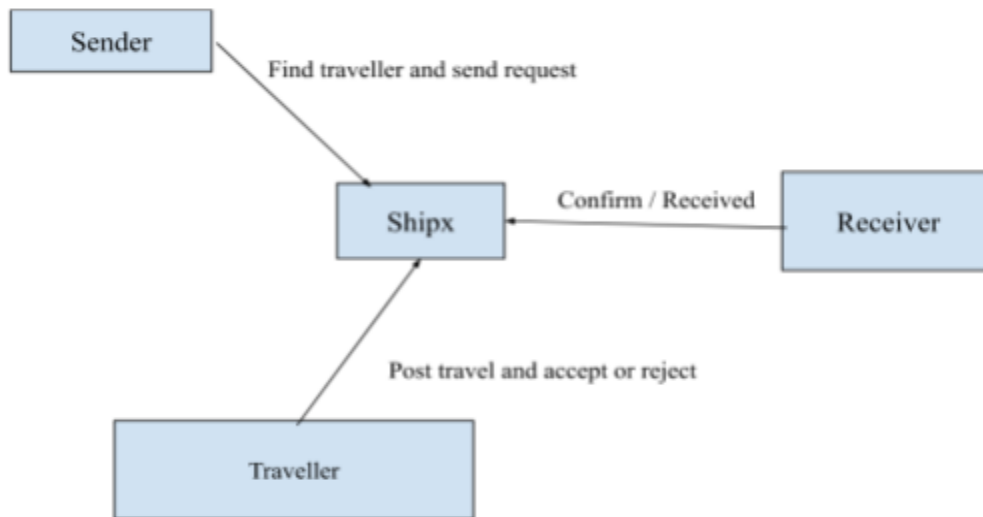


Figure 3.6.2: Use case model for sender, traveller and receiver

3.7 Design Requirements

Design requirements mean how friendly our user interface is to the users and how to easily implement the design on the front end. During app development, we try to maintain:

1. Make the app simple and flexible.
2. Efficiency.
3. Easy to use.
4. Optimize the performance.

CHAPTER 4

DESIGN SPECIFICATION

Users of ShipX will interact using an iOS App. In short, the client is an iOS app that will communicate with the server. The iOS App has a user interface for the user. Some UI pages and component descriptions are included below.

4.1 SwiftUI Component: Views and control

Text

In its use of read-only text, that text can be one or multiple lines. It has some modifiers. One of them is `.multilineTextAlignment`.

Label

It's a convenient component of view. The label has an image, and beside the image is text. It's fit for menu items or in settings.

TextField

The TextField is an important component of the user interface. Through this, input is taken from the user. It's required to have a placeholder that contains hints for the user. With this, it uses a state variable to take input from the user.

TextEditor

A TextEditor is similar to a TextField, but a TextEditor is used to edit or take input for long-form text.

SecureField

SecureField is an input view that is used to take the password from the user. In SecureField, when a user types the password, it shows up as a bold dot in the interface.

Image

An image is an element of SwiftUI. It's used to show images, logos, and icons in the UI. It's also used in buttons. Likewise, it uses some modifiers to control the size, like: `resizable`, `frame`.

Button

Functional control of UI elements. A button has two default modifiers, one for UI and another for function, or what to do.

Link

The link is the same as the button. Its create link-style button is used for the open associated app.

NavigationLink

It is another kind of button that presents navigation when it is tapped. The destination is set according to need.

ToolbarItem

It's used in the navigation bar to provide some option or item. The toolbar is used in the maximum application, and the toolbar is used with button, info, or arrow item functionality.

Toggle

A toggle is another kind of control system. It's a kind of button that works between on and off. Here, on means true and off means false.

Map

It's a view that shows the location and direction as an embedded interface on the display. It uses coordinates of latitude and longitude.

Picker

A control for selection from a list or set. Two types of picker views are available in SwiftUI.

1. Picker: For Select one value or option.
2. DatePicker: It's used for selecting the date.

ProgressView

It's a view for showing the progress of a task. It can be used as a slider. This view can be designed as a circle for a good-looking UI, and the slider is used for controlling a bounded linear range of values for selection.

4.2 SwiftUI - View Layout and Presentation

Stacks

Stacks are a complex container view in SwiftUI. In SwiftUI, five types of stacks are used. There are

1. HStack: It's a view that organizes its children or items in a horizontal line.
2. LazyHStack: That is a dynamic version of HStack that also arranges its children in a horizontal line when the item is needed.

3. **Vstack:** This view organizes its children or items in a vertical line.
4. **LazyVStack:** That is a dynamic version of VStack that also arranges its children in a vertical line when the item is needed.
5. **ZStack:** It's a container view that overlays its children or items with this, it aligns the items in both axes.

List

It's a container view that provides a single column to display its rows of components. The component can be text or an image. It can be used dynamically by using an array. It can also use sections to make sections in a list.

ScrollView

It is a view that provides the ability to scroll, so that a user can scroll to find his or her choice. Furthermore, it can be used to scroll horizontally, vertically, or both. The zoom ability is not functional in this view.

Grids

A grid is a component of SwiftUI that helps organize the UI. The grid has two types of variation in SwiftUI.

1. **LazyHGrid:** Container view used to make a grid of its item in horizontal line. It can create items only when needed.
2. **LazyVGrid:** It's the same as LazyHGrid, the only difference is that it creates a vertical line in the grid.

Form

The form is groping control data input container view. It is mostly used for inspection or taking input of data from users. It can be used by section.

Spacer

The Spacer is a very interesting component of SwiftUI. These spacer views fill up all the available space automatically on both axes.

Divider

This view makes a division of its surroundings by drawing a line. Suppose two stacks are in view, and it should look beautiful, or there should be a divider. There should be a divider.

NavigationView

It's the most important component because it provides some ability to pop and push screens with others. This view is used to make navigation titles. This view can unlock

the features of some advanced subcomponents. Like: navigation bar, navigation title, toolbar.

TabView

The tab view is used to create multiple tabs or buttons in a view that switch to a different view or screen when tabbed in a tab or button. Each button or tab in the tab view uses an icon and a title.

Alert

Alerts are used to show the user some alerts or warning messages. The boolean state is created for the functionality of the property, which is false by default. If the condition is true, a warning is displayed on the screen.

CHAPTER 5

Backend Specification

Backend is that portion of a project that users can't see, and it's a very sensitive part of a project (21). It's making sure everything client-side works. It communicates with the frontend to send and receive information to be displayed. To communicate with the frontend for different views, it uses a link called API. The ShipX APIs are given below:

5.1 baseURL:

baseURL is the host link or domain. It can be called the entry point because it contains the IP and port number of the server.

5.2 Authentication:

Register

Request - JSON BODY API (POST): {baseURL}/api/v1/register.

Register is the most important process that takes user data from a user to create an account on the ShipX server. It's a POST request API to make a request to the server to register an account for the individual.

Login

Request - JSON BODY API (POST): {baseURL}/api/v1/login.

Users need to access their accounts after creating an account by registering. After registering, after registering to access the account, the user needs to log in where the user He or she has to put in his or her username, email, or phone number and password. For login, ShipX uses this POST request login API.

Logout

Request - JSON BODY API (POST): {baseURL}/api/v1/logout.

Logout is a feature of every server-based app these days. It's used to temporarily disconnect from the server. For that, ShipX uses the POST request to logout.

Forgot Password

Request - JSON BODY API (POST): {baseURL}/api/v1/forgotPass.

It's a feature that if any user forgets his or her password that was used in ShipX at that time, it will be used to recover the account. That's to make a POST request to the server.

Send Register OTP

Request - JSON BODY API (POST): {baseURL}/api/v1/sendRegisterOTP.

It's a POST request to the server. When a user wants to register on ShipX, the server sends the user an OTP for verification. If the user has the right OTP, then he or she can complete the registration.

Send Forgot Password OTP

Request - JSON BODY API (POST): {baseUrl}/api/v1/sendForgotPassOTP.

Forgot Password An OTP is used when a user forgets a password and he or she wants to recover it or create a new password. This POST request sends an OTP to the user for verification and validation.

RefreshToken

Request - JSON BODY API (POST): {baseUrl}/api/v1/refreshToken.

This POST request makes a request to the server to generate a new access token after a certain time for user and server safety.

5.3 User:

Upload Profile Picture

Request - Form Data API (POST): {baseUrl}/api/v1/profilePicture.

It is a POST request to the server to upload and save the profile picture for the user of his account.

Get Profile Picture

Request - API (GET): {baseUrl}/api/v1/profilePicture.

It's a GET request to the server to get the uploaded profile picture in the user interface on the user's profile.

Switch Role

Request - JSON BODY API (POST): {baseUrl}/api/v1/switchRole.

This POST request is used to select the current role of a user because, in ShipX, a user can be a traveller, sender, or receiver.

ShippingCost

Request - API (GET): {baseUrl}/api/v1/shippingCost.

It shows the cost of shipping a product to the user interface, and it's a GET request.

5.4 Location:

Get Divisions

Request - API (GET): {baseUrl}/api/v1/divisions?division={divisionName}.

This is a GET request to get, find, or select the user division of the backend.

Get Districts

Request - API (GET): {baseUrl}/api/v1/districts?district={districtName}.

The GET request is used to get, find, or select the district name from the server. The GET request shows the user the list of districts and gives them the option to select.

Get Upazilas

Request - API (GET): {baseUrl}/api/v1/upazilas?upazila={upazilaName}.

It is another GET request that provides the power to the user to select the Upazila name.

Get Dhaka City Corporations

Request - API (GET): {baseUrl}/api/v1/cityCorpAreas?area={areaName}.

This GET request gives the ability for the user to select the area names under the Dhaka city corporations.

5.5 Posts:

Post Traveller Shipping Request

Request - API (POST): {baseUrl}/api/v1/travellerShippingRequest.

The user or traveller can make a shipping request on the server, and the server will send it to the user interface. In this way, a post can be seen by other users or senders.

Post Sender Shipping Request

Request - API (POST): {baseUrl}/api/v1/senderShippingRequest.

This POST request is similar to the post traveller shipping request, but the difference is that it's a sender making a post. The traveller can see the sent request post on the user interface.

Post Request Accept

Request - JSON BODY API (POST): {baseUrl}/api/v1/acceptRequest.

The sender's request can be accepted by using this API request. It's done by the traveller, and the sender will be notified about that.

Post Request Reject

Request - JSON BODY API (POST): {baseUrl}/api/v1/rejectRequest.

The sender's request can be rejected by using this API request. It's done by the traveller, and the sender will be notified about that.

5.6 Admin:

Post Shipping Cost

Request - JSON BODY API (POST): {baseUrl}/api/v1/shippingCost.

The shipping cost for the category is posted by the admin. It's fixed for now, but in the future, an AI will replace it.

Other admin operations can be done from the MongoDB web interface (22).

CHAPTER 6

USER INTERFACES

6.1 Login Register Selection View

The following figure shows Login Register View:

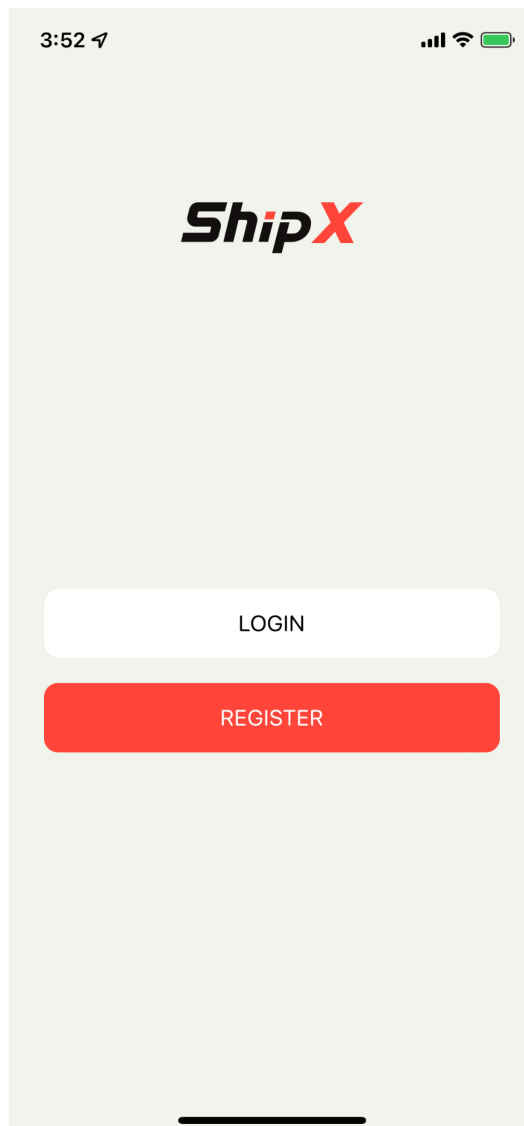


Figure 6.1: Login Register Selection View

6.2 Register View

The following figure shows Register View:

The screenshot displays a mobile application interface for registration. At the top, the status bar shows the time 3:52, signal strength, Wi-Fi, and battery icons. Below the status bar is a navigation bar with a back arrow and the text 'Back', and the title 'Register'. The main content area is a light green background with several white input fields. The first field is labeled 'Name'. Below it are two stacked fields labeled 'Phone' and 'Email'. Further down are two stacked fields labeled 'Password' and 'Confirm Password'. Below these fields is a note: 'Your password must be at least 6 characters long and also alphanumeric'. Below this note is a section titled 'Select Document Type' in red text. Underneath are two stacked fields labeled 'National ID' and 'NID or Birth Certificate ID'. At the bottom of the form is an 'Enter OTP' field with a red 'Get OTP' button to its right. Below the OTP field is a note: 'Press the Get OTP button and check your Email'. At the very bottom is a large red button labeled 'Register'.

Figure 6.2: Register View

This view takes input from the user and sends OTP to the user's given email address. When the OTP is correct and the user clicks the Register button, it creates a new account for that particular user.

6.3 Login View

The following figure shows Login View:

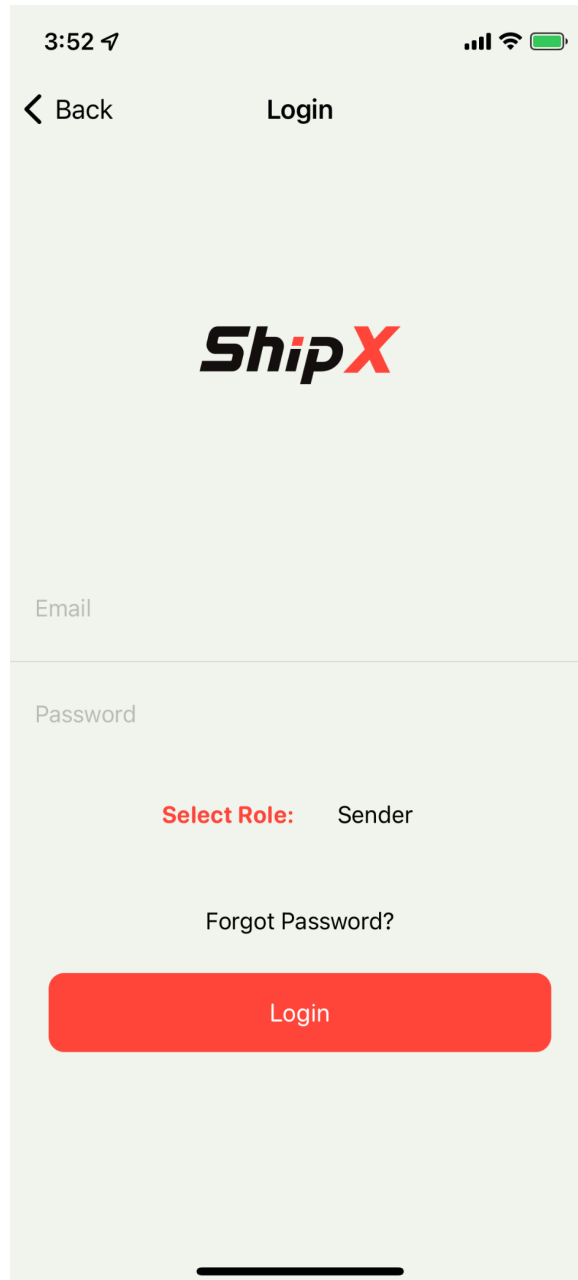


Figure 6.3: Login View

Here, a user will provide his email and password to log in to his or her account, and they can select their role as sender, traveller, or receiver.

6.4 Forgot Password View

The following figure shows Forgot Password View:

The screenshot displays the 'Forgot Password' screen of the ShipX application. At the top, the status bar shows the time as 3:53, along with signal strength, Wi-Fi, and battery icons. Below this, a navigation bar contains a back arrow and the text 'Login', followed by the page title 'Forgot Password'. The ShipX logo is prominently displayed in the center. The form consists of four input fields: 'Email', 'New Password', 'New Confirm Password', and 'OTP'. A red button labeled 'Get OTP' is located to the right of the OTP field. At the bottom of the form, there is a large red button labeled 'Update Password'.

Figure 6.4: Forgot Password View

If the user forgets their password, this interface will be useful. Here users have to enter the user's email, new password, and new confirm password and tap on the Get OTP button. An OTP will be sent to the user's email. Enter the OTP from the user's email to the Update Password to update the password or recover the account.

6.5 Traveller Travel Post View

The following figure shows Traveller Travel Post View:

The screenshot shows the 'Traveller Travel Post View' interface. At the top, the status bar displays the time 3:55, signal strength, Wi-Fi, and battery icons. The 'ShipX' logo is centered. Below it, the heading 'I am going to travel, From:' is followed by four dropdown menus for 'Division:', 'District:', 'Upazila:', and 'City Corp. Area:', each with a right-pointing arrow. An 'Additional Note:' field is present below these. The 'To:' section follows with another set of four dropdown menus for 'Division:', 'District:', 'Upazila:', and 'City Corp. Area:', also with right-pointing arrows. At the bottom, a navigation bar contains four icons: 'Travel' (a bus icon), 'Requests' (a person with a checkmark), 'Details' (a document icon), and 'Profile' (a person icon). The 'Travel' icon is highlighted in red.

Figure 6.5.1: Traveller Travel Post view

The following figure shows Traveller Travel Post View:

3:55

To:

Division: →

District: →

Upazila: →

City Corp. Area: →

Additional Note:

Travelling Time and Date

16 Jul, 2022 9:53 AM

Post

Travel Requests Details Profile

Figure 6.5.2: Traveller Travel Post view

To create a post, travellers have to fill out a form. To fill in the form, users or travellers have to tap on the right side arrows. It navigates the user to the Division name, District name, and Upazila's name pages. The user has to select from there. With this, users have to set the travelling time and date and tap the "Post" button to make a post.

When a user taps on the Division right side arrow, the given below view will appear, Here, users can search for and select their divisions.

The following figure shows Search Division View:

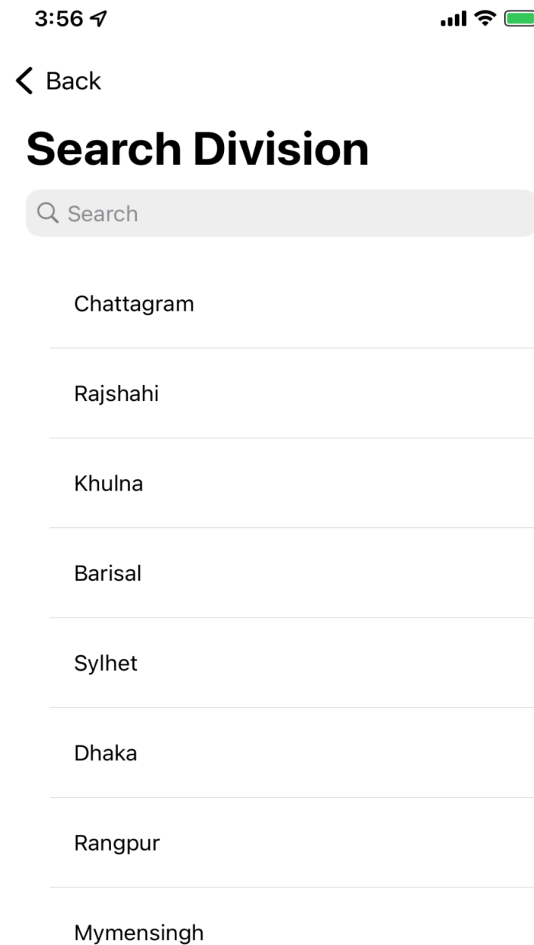


Figure 6.5.3: Search Division View

When a user goes to a district, similarly, the Search District view will come up on the screen where the user can search and select their district.

The following figure shows Search District View:

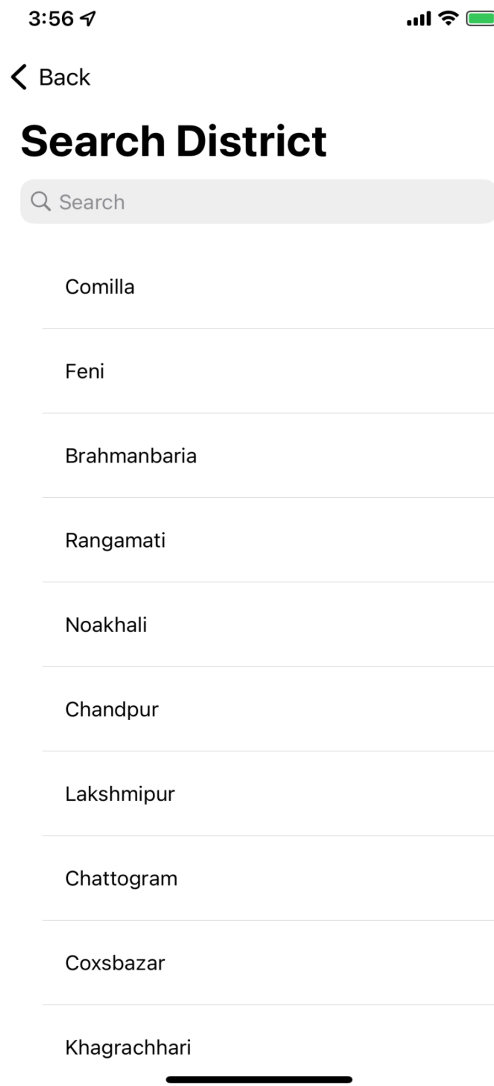


Figure 6.5.4: Search District view

To get to Upazila and the City Corporate Area, users have to hit the right side, and then they will be navigated to Search Upazila (Figure 6.5.5) and Search City Corp. Area (Figure 6.5.6) views:

The following figure shows Search Upazila View:

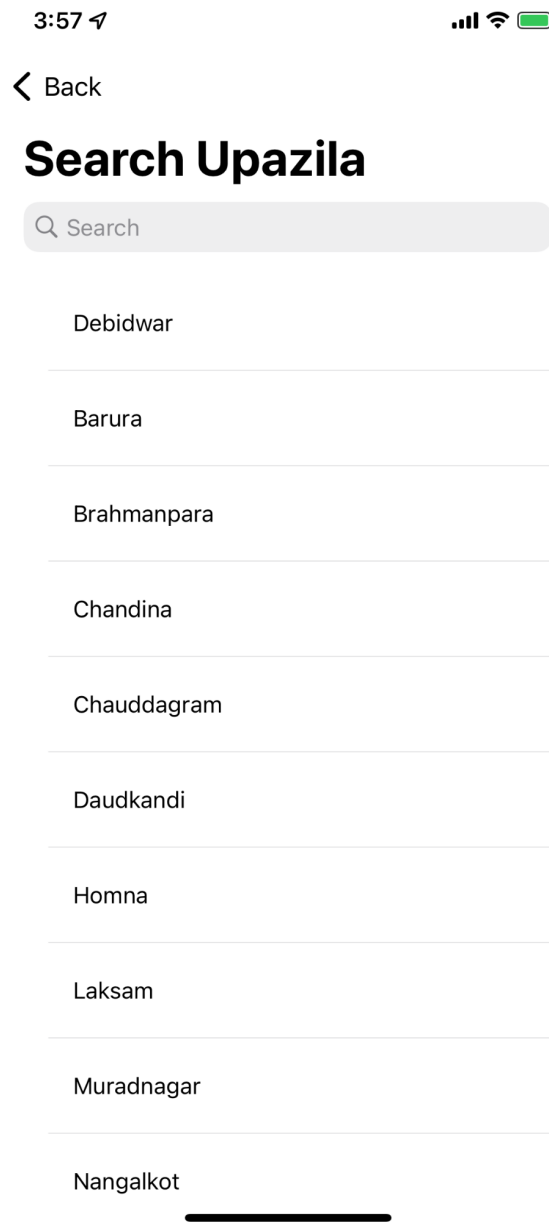


Figure 6.5.5: Search Upazila view

The following figure shows Search City Corp.Area View:

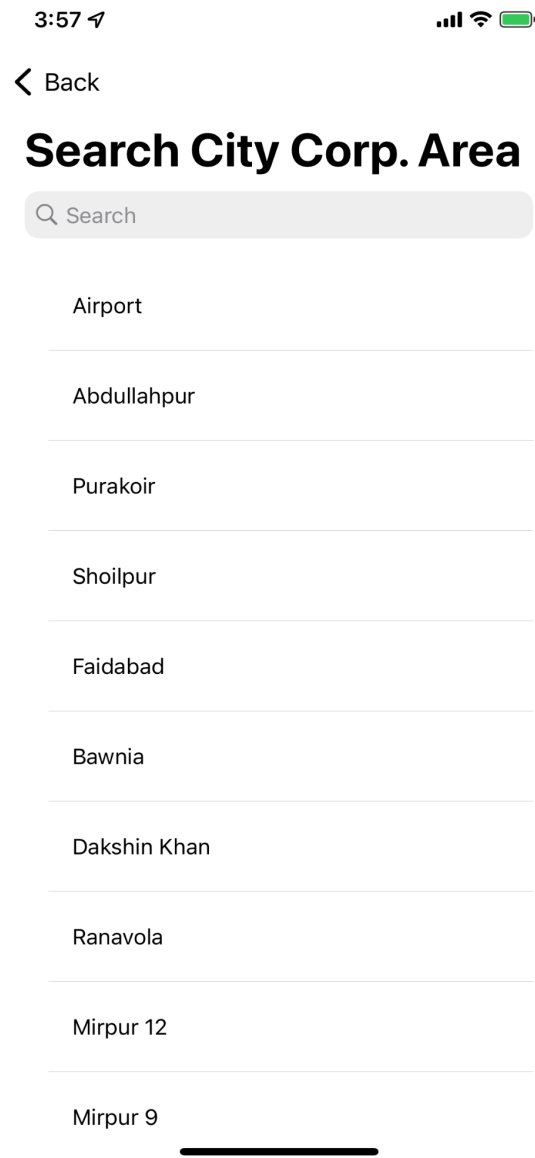


Figure 6.5.6: Search City Corp. Area view

6.6 Requests View

The following figure shows Requests View:

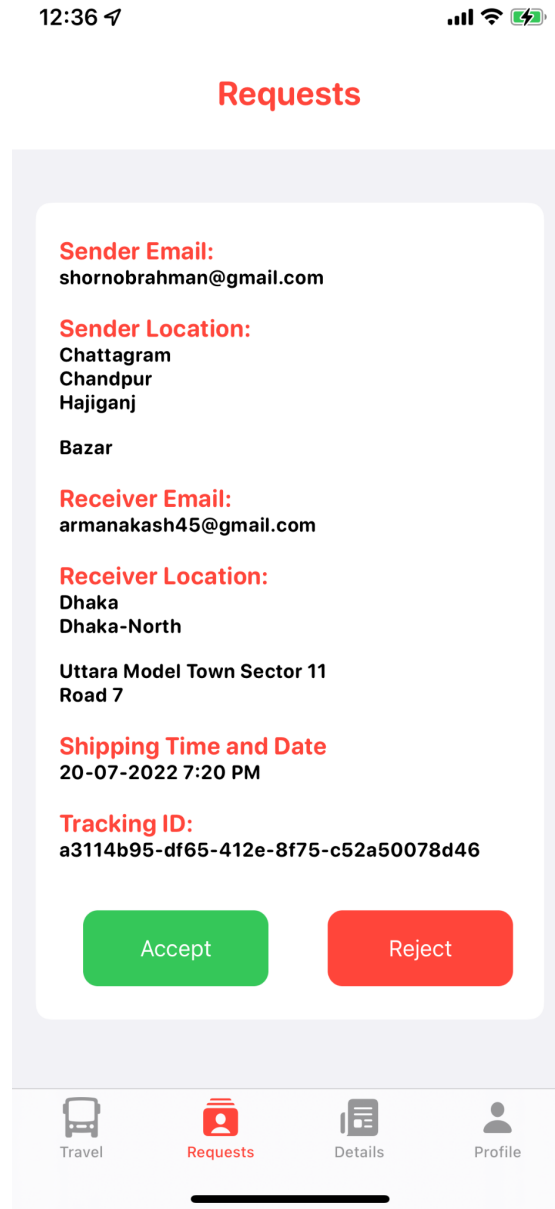





Figure 6.6: Requests View

In this view, travellers can see the requests of the sender and get the necessary details. They can either accept it by tapping on the Accept button, or reject it by tapping on the Reject button.


6.7 Details View

The following figure shows Details View:


10:00   

Accepted Request Details

Sender Information

 **Borno**
01838564443
shornobrahman@gmail.com

Receiver Information

 **Arman Akash**
01600008930
armanakash45@gmail.com

Product Information

Shipping Status:	Accepted
Payment Status:	Paid
Product Type:	Electronics
Shipping Cost:	60 BDT

Delivery Location: Dhaka-North
Dhaka-North
Uttara Model Town Sector 11





 Travel  Requests  **Details**  Profile

Figure 6.7.1: Accepted Request Details View

The following figure shows Details View:

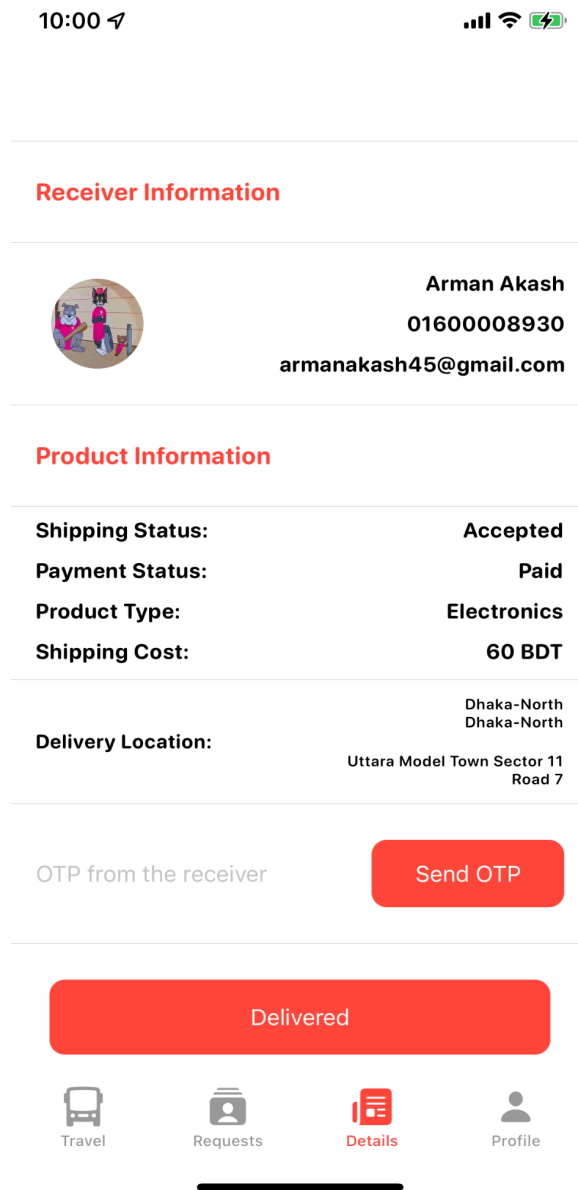


Figure 6.7.2: Accepted Request Details View

This view will appear when the user taps on the Details button. Here users can see the accepted request details like sender information (name, phone number, and email address), Receiver information (name, phone number, and email address), and product information. When the traveller reaches the receiver, the traveller has to open this view and tap on the "Send OTP" button to get the OTP from the receiver. After getting the OTP, the traveller has to press the "Delivered" button to complete the process.

6.8 Profile View

The following figure shows Profile View:

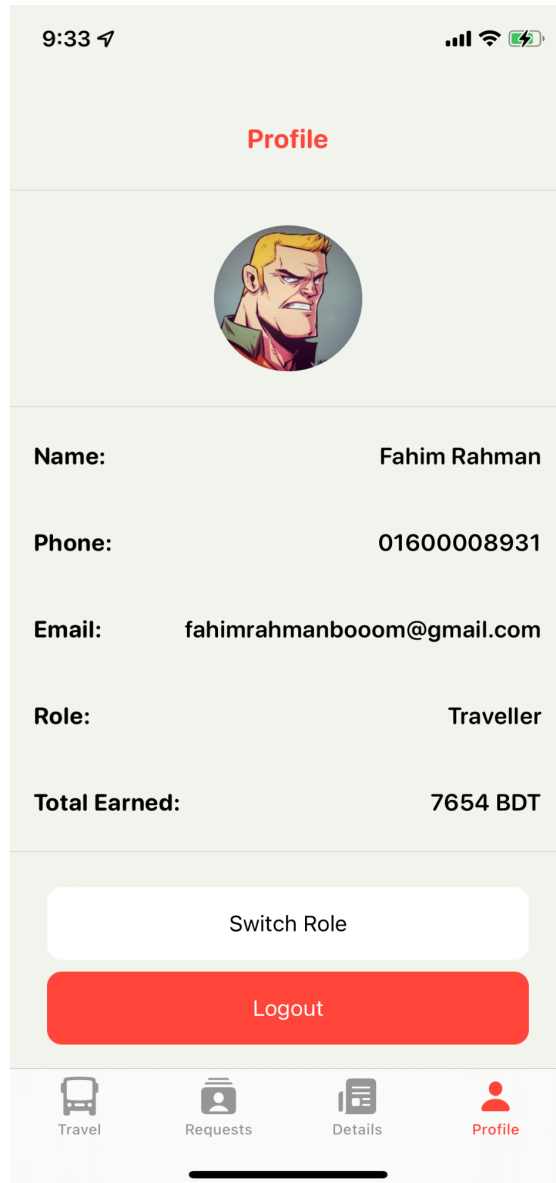


Figure 6.8: Profile View

This tab will appear on the user's screen when they press the Profile tab. It shows the user's information. From here, users can change or switch roles or select roles by tapping on the Switch Role button. If a user wants to log out of the ShipX server, they can do it from here by tapping on the Logout button.

6.9 Find Traveller View

The following figure shows Find Traveller View:

5:38

ShipX

Find Traveller, From:

Division: **Chattagram**

District: **Chandpur**

Upazila:

To:

Division: **Mymensingh**

District: **Mymensingh**

Upazila:

Find Travellers

Find Travellers Shipping Details Profile

Figure 6.9.1: Find Traveller View

When a user role is selected as the sender, then this Find traveller view will appear first. In this form, to find the traveller, the sender has to fill out the form and press the Find Travellers button.

The following figure shows Available Travellers View:

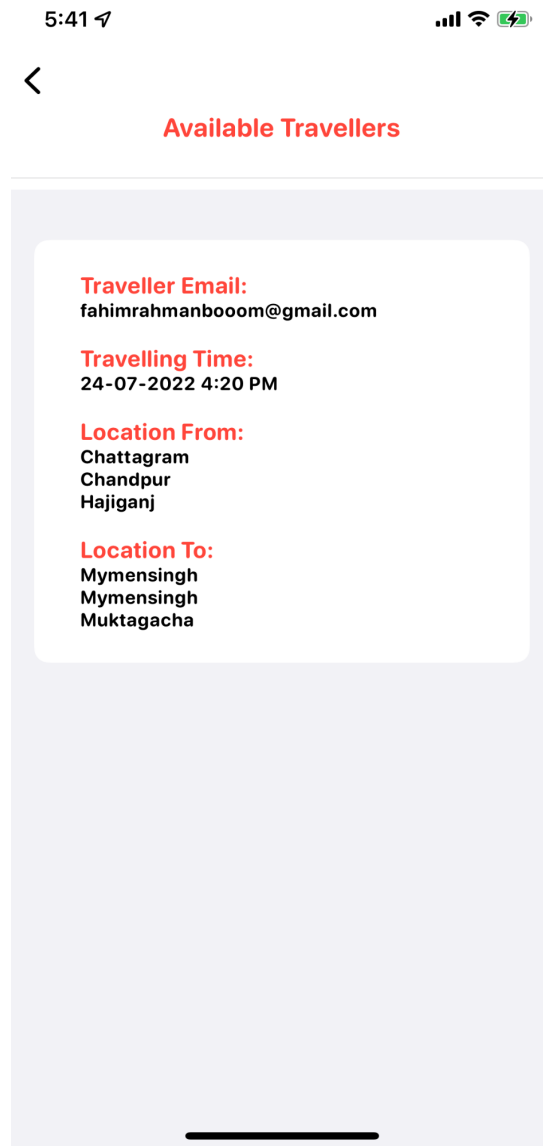


Figure 6.9.2: Available Travellers View

After tapping the Find Travellers button, the sender can see this view named Available Travellers. Here the sender can see the available travellers.

The following figure shows Send Request View:

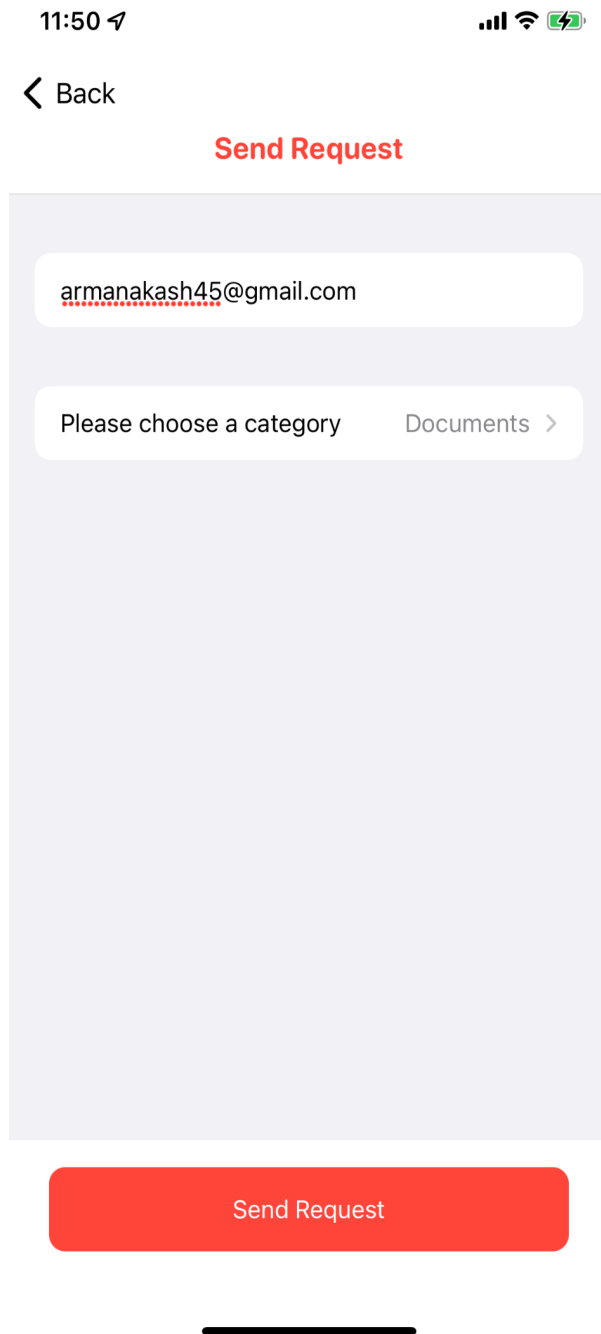


Figure 6.9.3: Send Request View

Here, the sender can select the category of the product and send a request to the selected traveller by tapping the Send Request button.

The following figure shows Product Category Selection View:

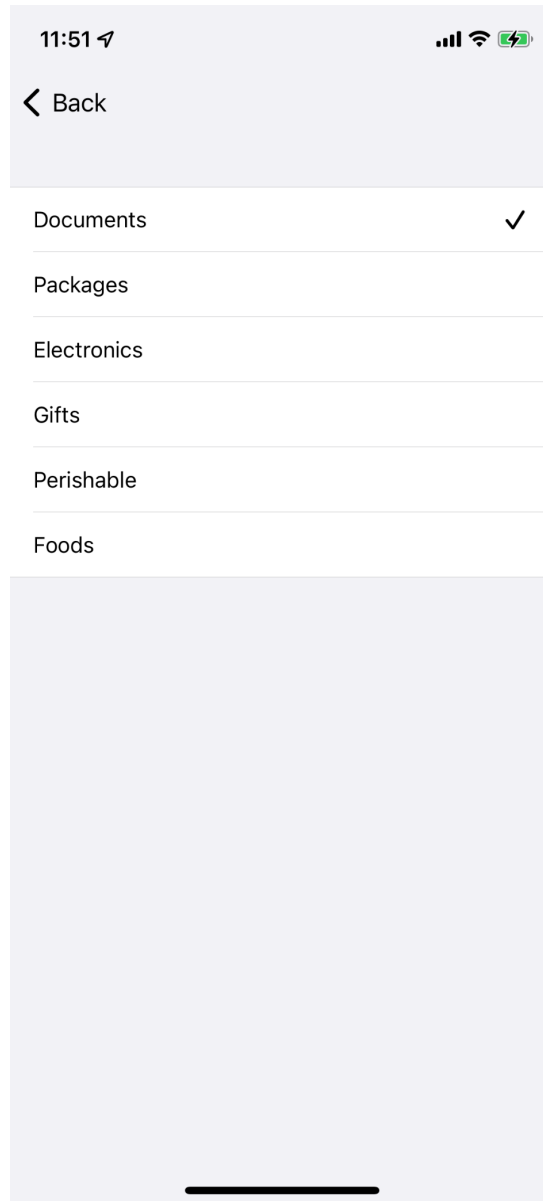


Figure 6.9.4: Product Category Selection View

The sender has to choose a category from this list. The shipping price will be adjusted accordingly.

The following figure shows Send Request Alert:

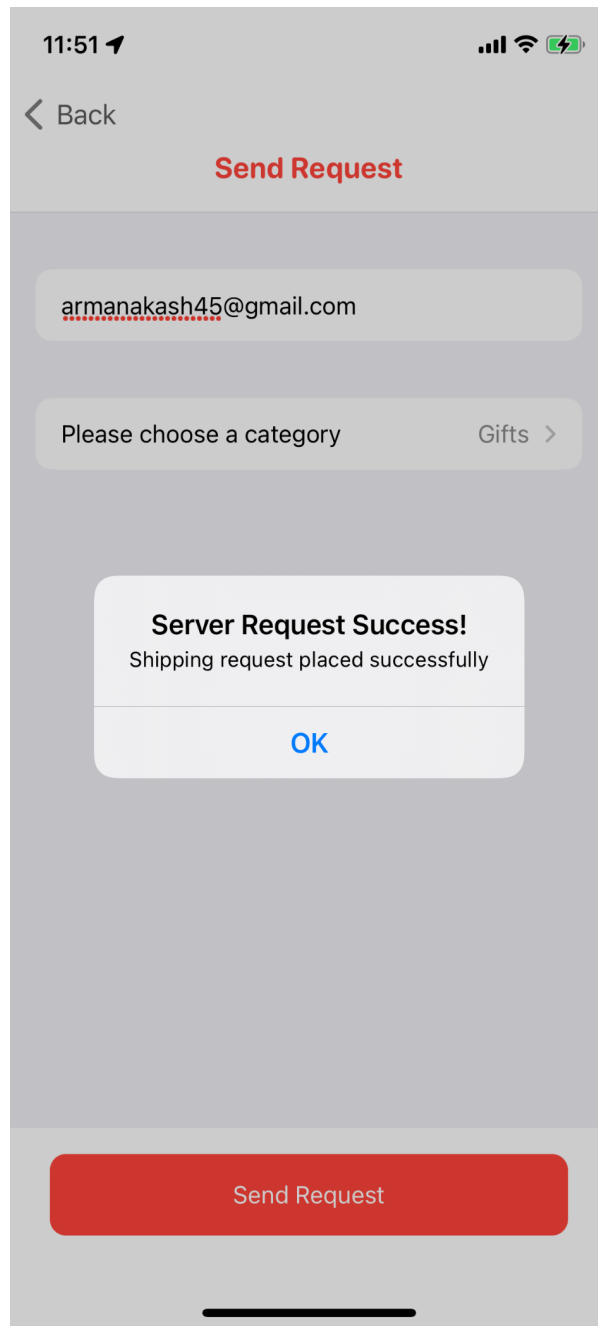


Figure 6.9.5: Send Request Alert

The sender gets an alert after hitting the Send Request button, which shows whether the request is a success or fails.

6.10 Shipping Details

The following figure shows Shipping Details View:

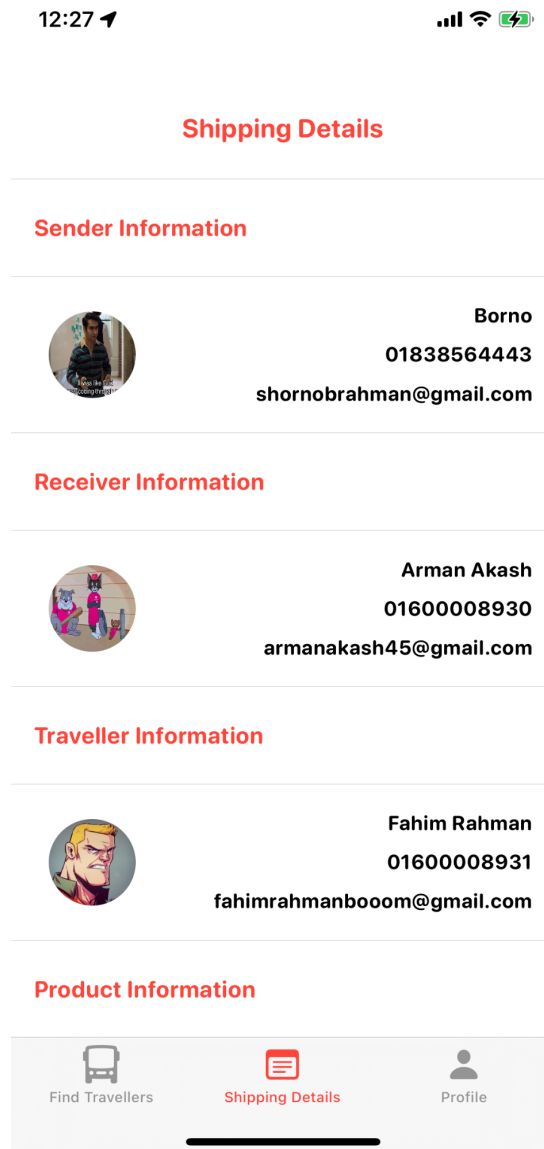


Figure 6.10: Shipping Details View

The following figure shows Shipping Details View:

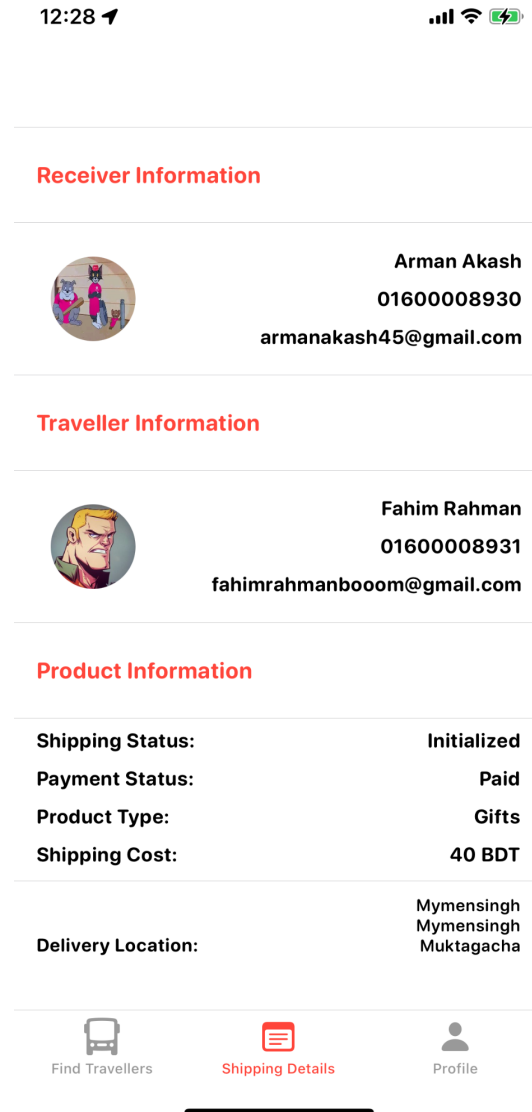


Figure 6.10.1: Shipping Details View

This View Figure 6.10 and Figure 6.10.1 will appear when the sender taps on the Shipping Details tab, Here the sender can see the Traveller, Receiver, and product information.

6.11 Others:

When a user selects his or her role as a receiver, they get Figure 6.10.1 and Figure 6.10.2 in a new tab. On the other tab, they get Figure 6.8.

CHAPTER 7

Working Procedure

7.1 Common user:

To use this app, users need to create an account by registering. To register, tap on the Register button from the Login Register Selection view. It will navigate the user to the Register view, where the user has to fill in the form with their information and credentials.

Then tap on the Get OTP button. After tapping on that button, the user will get an OTP to their email address. So they have to type or copy and paste the OTP into the Enter OTP placeholder. Then tap on the "Register" button. In that way, the user will create their account.

Users have to log in after creating an account. The Login view appears automatically in front of the user's screen. Their users have to put their email and password in the Email Password placeholder TextField and SecureField. After doing that, users have to select a role from the available three roles. Then you have to tap on the "Login" button.

7.2 Traveller:

If the user is logged in as a traveller, then the user can see a tab view that has four tabs. The first tab is the "Travel" tab, where the user has to fill out the form, and the user has to select the travel time and date, then tap on the "Post" button to make a post.

When the user (traveller) taps on the "Request" button, the user can see the incoming request with the information of the sender and receiver (sender, receiver email addresses, and locations).

At this point, the traveller can either accept or reject the request thanks to the "Accept" and "Reject" buttons.

To open the details view, the traveller needs to tap on the "Details" tab, and here the accepted request details will appear. From here, the traveller gets the necessary information about the sender and receiver to contact them. When a traveller reaches a receiver, the traveller has to open it and press the "Get OTP" button. This OTP will be sent to the receiver's email. The receiver has to provide the OTP to the traveller.

The traveller has to put OTP on the OTP placeholder and has to press the "Delivered" button to complete the process.

To open someone's own profile, tap on the Profile tab. Here the user's information is available. To change roles, press the "Switch Role" button and select the role. Users can log out by pressing the "Logout" button.

7.3 Sender:

When a user selects a role as a sender, the Find Traveller form will appear on the user's screen. Here, the sender has to fill out the form and has to tap on the Find Travellers button to find travellers. Once the sender does that, the view will appear called Available Travellers, the sender can look for the available travellers list and can select the traveller by tapping on the list.

The sender will be navigated to another view called "Send Request," where they can get the email of the traveller and have to select the category of the product, then press the "Send Request" button to make a request. After hitting the Send Request button, the sender gets an alert message that shows whether the request was successful or not.

If the user taps on the Shipping Details tap, the Receiver, Sender, Traveller, and Product information will show on the screen.

7.4 Receiver:

When a user selects his or her role as a receiver, two tabs will appear on their screen. One of them is the Shipping Details, where Receiver, Sender, Traveller, and Product information will be shown, and the other tab is the Profile.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

We tried to make this whole system as good as possible. And it kind of met our expectations. We wanted to make a robust system that could potentially replace our traditional delivery system. Although it was hard to develop the project because the framework for the iOS app was a new framework and the backend framework was new to us, we tried our best and hopefully delivered a project.

Another obstacle in our path was a lack of time. Making a project that has a frontend and a backend requires a lot of time and research. We had to work until midnight to finish our work. Some functions in this project were quite easy to make, while many functions took a lot of work to implement.

It took us two to three months of research and three months of hard work to finish ShipX.

We chose the iOS app for the client part of our project because it is very rare in our country, and we also wanted to further our career as an iOS developer. We chose Express as a backend service because the learning process is faster. Furthermore, we have used the MongoDB NoSQL database because we think it's the future of databases. This is an achievement for us as CSE students to add a new delivery system concept to our country's delivery system. This project was a dream for us. We are really happy that while studying at Daffodil International University, we made it possible. This project also developed our software development skills, which will help us make more quality software in the future.

After a long time working on this project, we accomplished many things. Some of them are as follows:

We made the project using all open source libraries. Making a polished app takes time, but making the proof of concept doesn't take much time, but it needs hard work and a lot of research. Our goal was not to build a project to release it to the market as soon as possible, but to see if it's possible.

8.2 Future Scope

We tried our best to make a unique delivery system, but because of our limited resources, we couldn't do everything we hoped for. There are many things we will try to do in the future for this project.

We will implement maps so that users don't have to manually type in their location. We also intend to add push notifications, OTP sent to the phone number instead of email, and a payment gateway. A user's NID or BID checking API requires a business license as per

Bangladeshi government rules. So if we run the project as a business, then we would implement this API to ensure maximum security.

We will update the backend and user app every two months and add new exciting features and visual changes so that we can get more users. There are a lot of possibilities for ShipX - An express P2P delivery system. We believe we can do a lot better in the near future for the P2P delivery system.

REFERENCES

- [1] The Evolution of Mobile Apps, Available at: <<<https://arkenea.com/blog/evolution-of-mobile-apps/>>> , last accessed date: 03-01-2022 at 08.55 pm.
- [2] Mobile Operating System Market Share Worldwide, Available at: <<<https://gs.statcounter.com/os-market-share/mobile/worldwide>>> , last accessed date: 03-01-2022 at 09.02 pm.
- [3] Backpack, Available at: <<<https://backpackbang.com/>>>, last accessed date: 03-01-2022 at 10.40 pm.
- [4] 101 Delivery Statistics for 2020, Available at: <<<https://elogii.com/blog/delivery-statistics-2020/>>>, last accessed date: 03-01-2022 at 11.00 pm.
- [5] macOS, Available at: <<<https://en.wikipedia.org/wiki/MacOS>>>, last accessed date: 04-01-2022 at 06.10 pm.
- [6] Linux, Available at: <<<https://en.wikipedia.org/wiki/Linux>>>, last accessed date: 04-01-2022 at 06.16 pm.
- [7] Web server, Available at: <<https://en.wikipedia.org/wiki/Web_server>>, last accessed date: 04-01-2022 at 06.31 pm.
- [8] Xcode, Available at: <<<https://en.wikipedia.org/wiki/Xcode>>>, last accessed date: 15-01-2022 at 10.20 pm.
- [9] Visual Studio Code, Available at: <<https://en.wikipedia.org/wiki/Visual_Studio_Code>>, last accessed date: 15-01-2022 at 10.30 pm.
- [10] JavaScript, Available at: <<<https://en.wikipedia.org/wiki/JavaScript>>>, last accessed date: 16-01-2022 at 2.46 pm.
- [11] Swift (programming language), Available at: <<[https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))>>, last accessed date: 17-01-2022 at 2.10 am.
- [12] Express.js, Available at: <<<https://en.wikipedia.org/wiki/Express.js>>>, last accessed date: 17-01-2022 at 2.50 am.
- [13] SwiftUI, Available at: <<<https://developer.apple.com/documentation/swiftui/>>>, last accessed date: 18-01-2022 at 9.39 pm.
- [14] MongoDB, Available at: <<<https://www.mongodb.com/why-use-mongodb>>>, last accessed date: 23-01-2022 at 2.38 am.
- [15] Cloudinary, Available at: <<<https://en.wikipedia.org/wiki/Cloudinary>>> last accessed date: 24-01-2022 at 3.00 am.

[16] Heroku, Available at: <<<https://en.wikipedia.org/wiki/Heroku>>>, last accessed date: 25-01-2022 at 3.12 pm.

[17] Git, Available at: <<<https://en.wikipedia.org/wiki/Git>>>, last accessed date: 26-01-2022 at 4.50 pm.

[18] GitHub, Available at: <<[GitHub - Wikipedia](#)>>, last accessed date: 26-01-2022 at 8.05 pm.

[19] Docker, Available at: <<[Docker \(software\) - Wikipedia](#)>>, last accessed date: 01-02-2022 at 2.51 pm.

[20] Node.js, Available at: <<<https://en.wikipedia.org/wiki/Node.js>>>, last accessed date: 01-02-2022 at 3.15 pm.

[21] ShipX RestAPI Documentation, Available at: <<<https://hackmd.io/s/r1n1Fr0dY>>>, last accessed date: 22-07-2022 at 4.00 am.

[22] Heroku Server Status, Available at: <<<https://shipxbackend.herokuapp.com/>>>, last accessed date: 22-07-2022 at 3.00 am.

Plagiarism Report

SHIPX - AN EXPRESS DELIVERY SYSTEM

ORIGINALITY REPORT

11 %	%	%	11 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	7 %
2	Submitted to University of Bolton Student Paper	1 %
3	Submitted to Manchester Metropolitan University Student Paper	1 %
4	Submitted to Stanmore College Student Paper	<1 %
5	Submitted to University of Greenwich Student Paper	<1 %
6	Submitted to Fareham College Student Paper	<1 %
7	Submitted to University of Hertfordshire Student Paper	<1 %
8	Submitted to Buckinghamshire Chilterns University College Student Paper	<1 %
9	Submitted to NCC Education	

Student Paper

<1 %

10

Submitted to United Colleges Group - UCG

Student Paper

<1 %

11

Submitted to National School of Business
Management NSBM, Sri Lanka

Student Paper

<1 %

12

Submitted to Barking College

Student Paper

<1 %

13

Submitted to University of Ulster

Student Paper

<1 %

14

Submitted to University of Wales, Lampeter

Student Paper

<1 %

15

Submitted to Imperial College of Science,
Technology and Medicine

Student Paper

<1 %

16

Submitted to University of Brighton

Student Paper

<1 %

17

Submitted to Liberty University

Student Paper

<1 %

18

Submitted to Middlesex University

Student Paper

<1 %