IST COLLEGE

Specialization: Computer Science

Interdisciplinary Semester Project

Topic: Applied Programming

Trainee: Andreas Kokkinos

Additional Instructor: Kapetanaki Lena

Spring Semester 2025

Prifysgol Glyndŵr Wrecsam | Wrexham Glyndŵr University

ASSIGNMENT BRIEF

Module title	Applied Programming
Module code	COM537
Module leader	Daren Morris
Assessment title	Assignment
Launch date	16th March 2025
Submission deadline	4th May 2025
Expected date for return of marks and feedback	
Module outcomes assessed	1, 2, 3, 4, 5
Assessment weighting	100
Word count (if relevant)	N/A
Assessment task details - provide a description of the task	

Table of Contents

Introduction 5
Application Overview 6
Use Case Diagram Explanation 7
Taxi Booking Process 8
Booking History 9
Backend System Implementation 10
Diagrams 11
UX/UI Analysis for TaxiEase 14
Ui & Ux design 17
Testing 22
TaxiEase – Windows Forms Taxi Booking System 27
Conclusion 30
Bibliography 31

List of Images

Figure 1. Flowchart 11
Figure 2. UML Use Case 12
Figure 3. DataBase Schema 13
Figure 4. Splash Screen 17
Figure 5. Sign In 17
Figure 6. Log In 18
Figure 7. Home Page 18
Figure 8. Booking System 19
Figure 9. Booking System 2 19
Figure 10. Booking System 3 20
Figure 11. Profile 20
Figure 12. History 21
Figure 13. Testing 1 22
Figure 14. Testing 2 23
Figure 15.Testing 3 23
Figure 16. Testing 4 24
Figure 17. Testing 5 25
Figure 18. Testing 6 25
Figure 19. Testing 7 26

Introduction

TaxiEase is a simple yet efficient taxi booking system created using C# and Windows Forms. It is designed to provide an easy-to-use system for passengers to book taxis, manage their profiles, and conveniently view their booking history. The application provides a smooth user experience by incorporating features such as user authentication, booking management, and profile customization.

The system has a feature for users to register, log in, and store their details securely. Once logged in, users can book taxis by selecting their destination and receiving an estimate of the fare. The system also has the feature to update personal details, such as contact information and password, through the profile management feature. The system also offers users a record of their past bookings, which is a convenient way to keep track of their rides.

Overall, TaxiEase combines ease and efficiency into a straightforward interface, allowing users to navigate the system for a hassle-free taxi booking experience.[5]

Application Overview

TaxiEase is designed with the passenger in mind and is made to simplify booking taxis with its modern, responsive design. The app is all about keeping things simple and providing the features necessary for easy taxi booking. Its main features include:

- **User Registration & Login:** Users can register using a unique email and password or log in using their credentials if they are returning customers. The login functionality offers a personalized experience by remembering user preferences and details.
- **Taxi Booking System:** The system allows users to book taxis by entering a destination. It then determines the fare based on the route and available taxis. The users can choose from a range of options such as low-fare or high-fare taxis, depending on availability.
- **Profile Management:** The users are allowed to edit their profile information, such as name, address, and phone number. This ensures the user's details are up to date, facilitating bookings and communication with drivers.
- **Booking History Tracking:** The history feature maintains a record of all bookings done, specifying the fare, driver information, and travel route. The users can view past bookings for future reference, which provides them with a clear view of their travel history.
- **Logout Functionality:** When the user has finished their activities, the system allows easy logouts, ensuring security and privacy for every session.

Use Case Diagram Explanation

The use case diagram establishes the key functionalities and the interaction flow between the system and the users. The flow begins with User Authentication:

- **User Authentication:** The new users are requested to register their details. The already registered users can simply log in, and the system will record their login details to avoid multiple logins. This feature is essential to enhance the convenience of the users.

Once a user is authenticated, they can access the main navigation menu. The choices are as follows:

- **Home Screen:** This is the starting point where users can begin by either making a booking or managing their profiles.
- **Profile Management:** Under this category, the users can see and edit their details. This includes changing their name, phone number, or password. In case a user wishes to alter their address or contact information, they can modify it under the profile management category.
- **History of Bookings:** Users have the option of viewing a list of all previous bookings made, along with travel dates and fare details. Users can keep an account of rides and a method of keeping costs in check easily.
- **Taxi Booking Option:** The customers can book a taxi by selecting their desired location. The application estimates the cost based on the route selected by the customer and available taxis. The system provides users with the best available taxis on their request.[3]

Taxi Booking Process

The heart of the application is the Taxi Booking feature:

- **Initiating a Booking:** Users begin by entering the destination they want to visit. The system will then show them available taxis by location, time of day, and chosen route.
- **Cost Estimation:** Based on the available options, the system will show two cost categories: Low Cost and High Cost. This offers flexibility to customers, allowing them to select the option that best suits their needs, whether it is the lowest or most expensive one.
- **Confirming the Booking:** After checking the details (i.e., destination, taxi availability, and fare), the users may confirm the booking. The process is triggered, and the system generates a booking confirmation with the required details.
- **Booking Information:** Once the booking is confirmed, the system shows the user the following information:
    - Date and time of booking
    - Route of travel (e.g., Syntagma – Neos Kosmos)
    - Total fare (Low-Cost or High-Cost options)
    - Taxi driver details (name and license plate number).

These details are stored in the application's database for later easy retrieval.

- **Canceling a Booking:** The user has the option to cancel the booking using the "Cancel Order" feature. This feature is important in cases where the user changes their mind or enters incorrect information. Importantly, canceled bookings are not retained in the database to prevent the storage of unnecessary data.

Booking History

The Booking History feature allows users to see their past rides:

- **Seeing Past Bookings:** The application keeps a history of all the bookings done in the past. Users can see a list of their past trips, along with full details, such as the date and price of each trip, the name of the driver, and the license plate number.
- **No Booking Record:** If there are no prior bookings, the system will notify the user with a message indicating that there are no records to display.
- **Ride Details Transfer:** Wherever users need to transfer their ride details (e.g., for reimbursement or friends/family), the system offers them the ability to export or transfer the details with ease.

Backend System Implementation

Although the application does not utilize a traditional server-based backend or a relational database, it implements core backend functionality through structured file-based data management. All essential data (such as user information, taxi orders, driver availability, and ride history) is stored and retrieved from local text files (user_data.txt, orders.txt, etc.).

Each of the application's functionality modules communicates with these files to execute backend logic operations like:

- **User authentication:** Checks login credentials against the stored user_data.txt.
- **Taxi booking management:** Saves new bookings in orders.txt with corresponding metadata (user ID, pickup/drop-off point, timestamp).
- **Driver dispatching:** The system looks for spare drivers based on a simple selection algorithm (for example, the first available driver).
- **Tracking order status:** Updates order files to indicate changes in trip status (e.g., assigned, ongoing, done).

The process of reading, writing, and modifying file data is encapsulated in independent methods with modulated logic, emulating standard backend services. This lean architecture permits the application to run locally and stand alone with no requirement for server setup, yet it still exemplifies fundamental backend concepts such as:

- **Data persistence**
- **Logic abstraction**
- **Disposition Logic**
- **Status/state management**

This strategy was selected to streamline the test and development process and to keep the architecture and flow of the system clear.

Diagrams

1. **Flowchart** diagram, how the app works

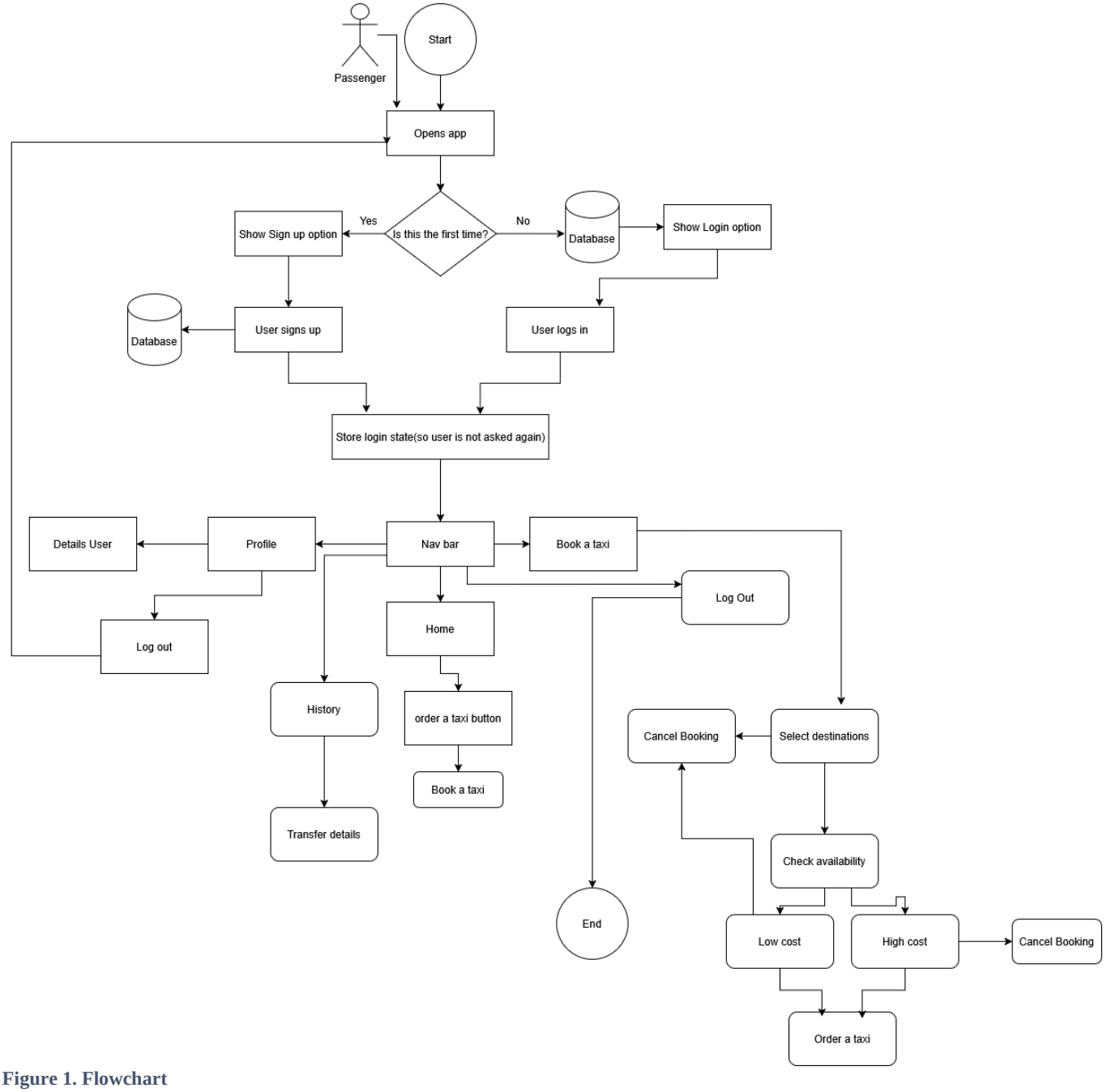

Figure 1. Flowchart

[2]

2. **UML use case** diagram

[3]

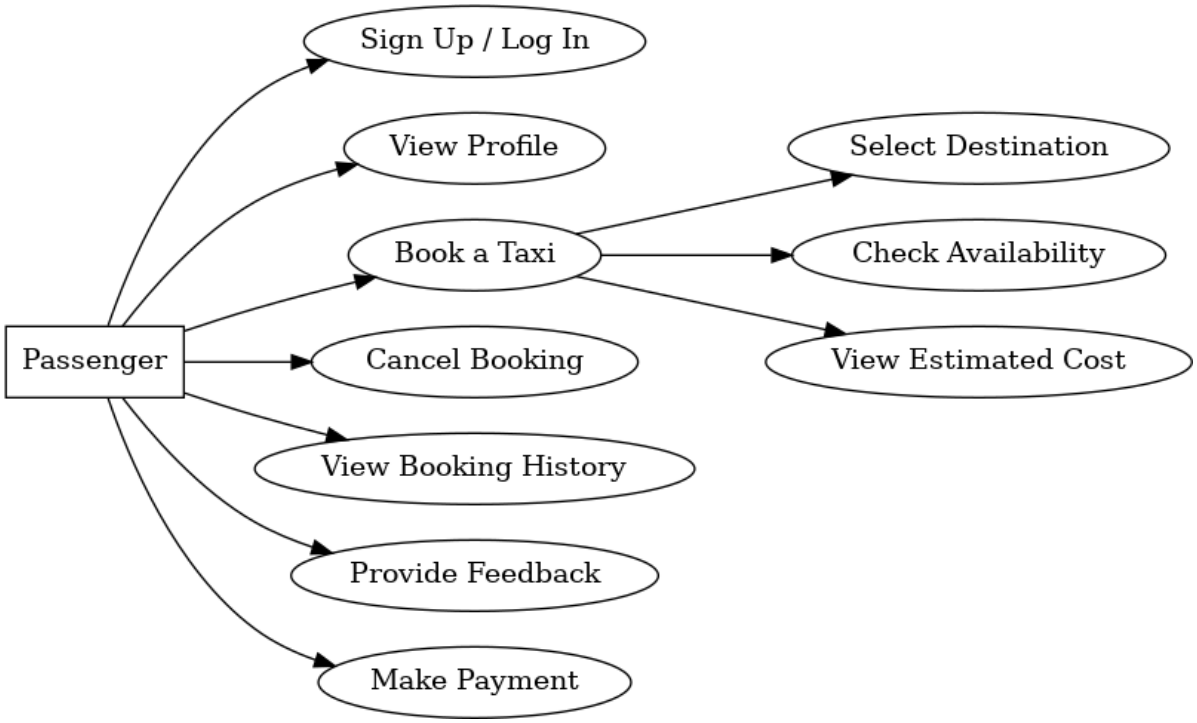

Figure 2. UML Use Case

3. **Database Schema** diagram

[4]

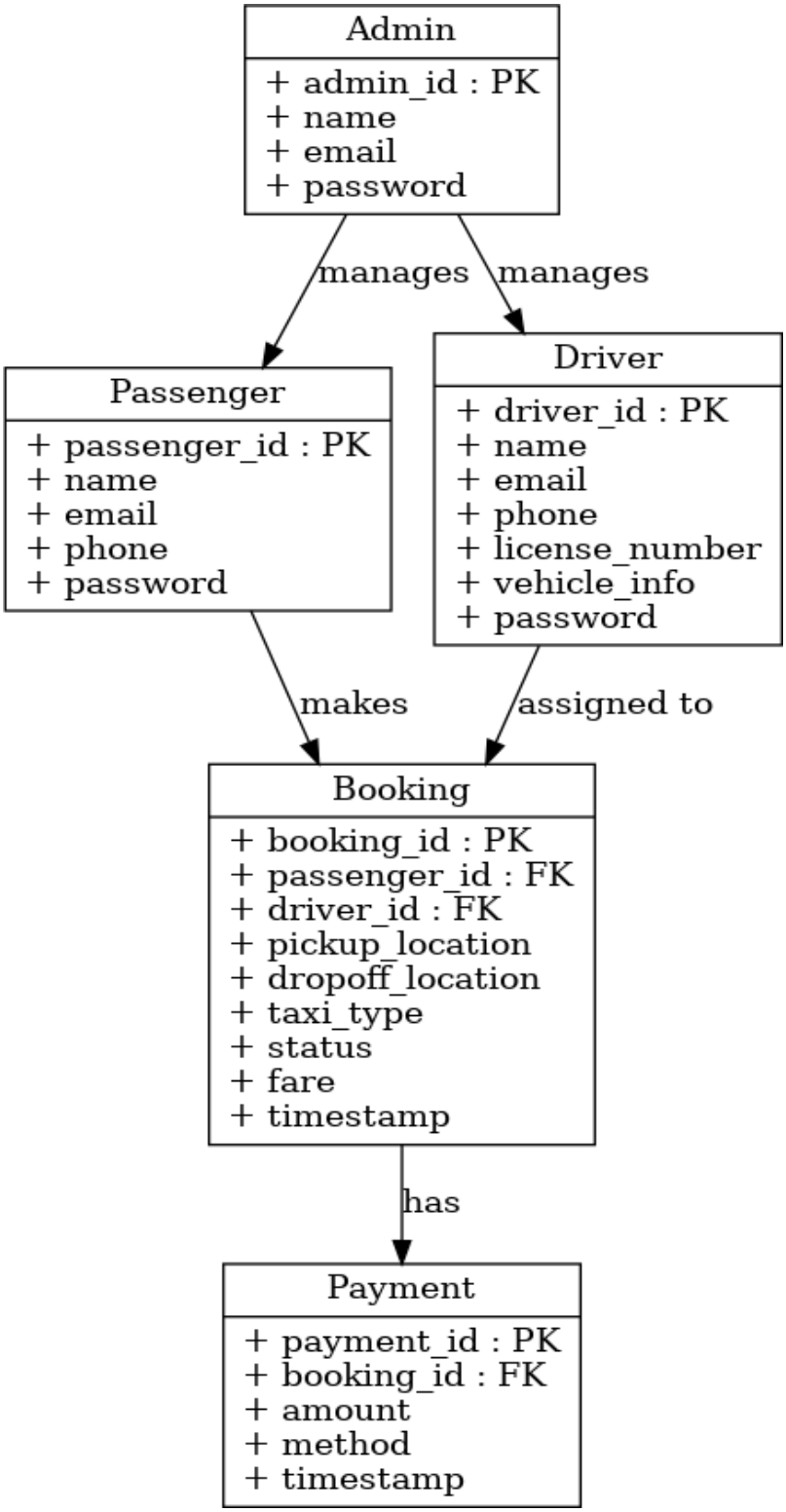

Figure 3. DataBase Schema

UX/UI Analysis for TaxiEase

The user interface (UI) of TaxiEase has been designed with attention to detail to provide a smooth and efficient user experience (UX). The design principles are built around simplicity, clarity, and navigability to allow users to interact with the system effortlessly when booking a taxi, editing their profiles, or viewing their booking history.

1. Layout and Navigation Design

The design follows a logical, modern approach, with strong emphasis placed on usability and accessibility.

- **Navigation Bar Positioning**
  - The navigation bar is situated at the top of the application window and allows users to move quickly between the Home, Booking, Profile, and History sections.
  - This positioning is a common convention employed by most desktop applications and, hence, is natural for users.
- **Screen Transitions and Visual Hierarchy**
  - Each section (i.e., Booking, Profile, History) follows the same format to prevent confusion.
  - Most frequent actions, such as "Book Taxi", are given prominence by having buttons of contrasting colors and bigger font sizes.

2. Color Scheme and Visual Aesthetics

The chosen color scheme enhances usability without compromising modernity and sophistication.

- **Primary Colors:**
  - **Gold** (#FFD700) – Used for buttons and key highlights, giving a premium and luxurious look.
  - **Dark Blue** (#1E3A8A) – Used for headers and major UI elements, providing contrast and readability.
  - **Charcoal Gray** (#37474F) – Utilized for text and secondary elements, it offers clarity without straining the eyes.
  - **White** (#FAFAFA) – The background color, which gives a minimalistic and clean look.
- Why this color scheme?

The dark blue and gold theme provides a feeling of professionalism and dependability, and it makes the application appear modern and reliable.

  - The gray and white components provide good contrast and improve readability, especially for visually challenged users.

3. Button & Interaction Design

All buttons and interactive elements are designed to be easily recognizable and interactive.

- The primary buttons ("Book Now", "Confirm Booking") are given gold backgrounds and bold text, making them stand out visually.
- Secondary actions (like “Cancel Order”, “Logout”) are shown in a muted gray or red, so they're visible but not distracting.
- Buttons have hover effects, providing users with distinct feedback when clicking on them.

4. Booking System UX Enhancements

The booking process has been simplified as much as possible:

1. Step-by-Step Flow – The user is taken through choosing a destination, taxi type, and payment confirmation in a logical order.
2. Real-time Availability Checks – The system ensures that users are displayed only available taxis, reducing errors.
3. Fare Estimation Before Booking – The user can view both Low-Cost and High-Cost options, making it easier to select based on budget.
4. Booking Confirmation Summary – Users can view the details (route, driver, price) before confirming the ride to prevent mistakes.

5. User Experience & Profile Management

The profile section ensures that users can update personal details effortlessly with minimal effort.

- Editable fields for name, phone, address, and password.
- Saved preferences and auto-fill reduce the need for repetitive manual entry.
- The logout option is strategically placed in the profile area for easy access.

6. Booking History & Data Accessibility

Users can easily retrieve past booking details in a well-organized manner.

- List View with Date, Fare, and Driver Details – Users can scroll through past rides easily.
- •Search and Filter Options – Allows sorting by date or fare range, helping users find specific rides faster.
- No Data Notification – If a user does not have any past bookings, the system provides clear feedback instead of displaying an empty page

Ui & Ux design

[1]

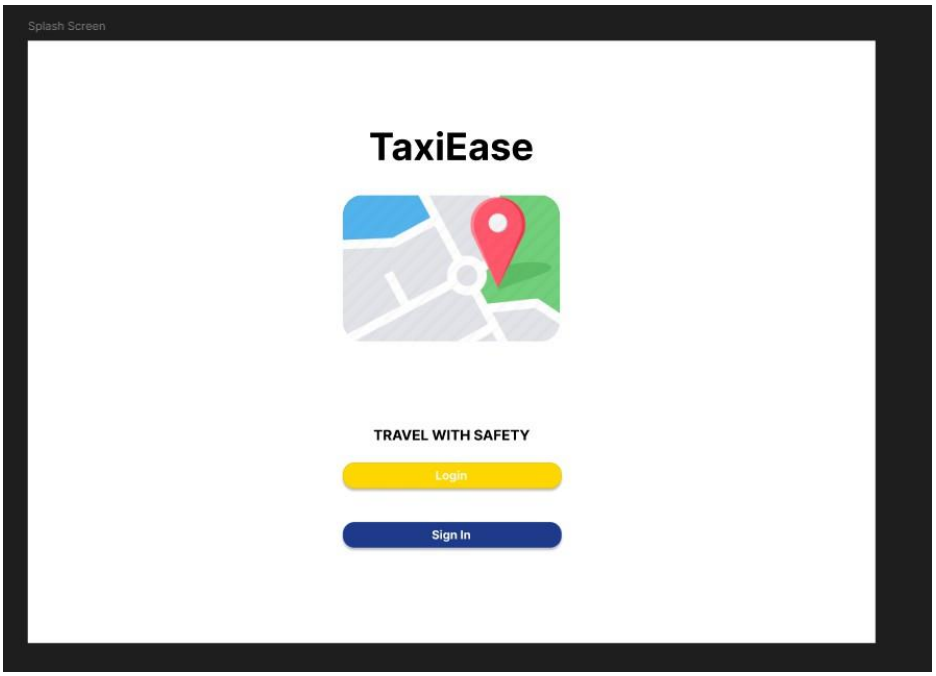

Figure 4. Splash Screen

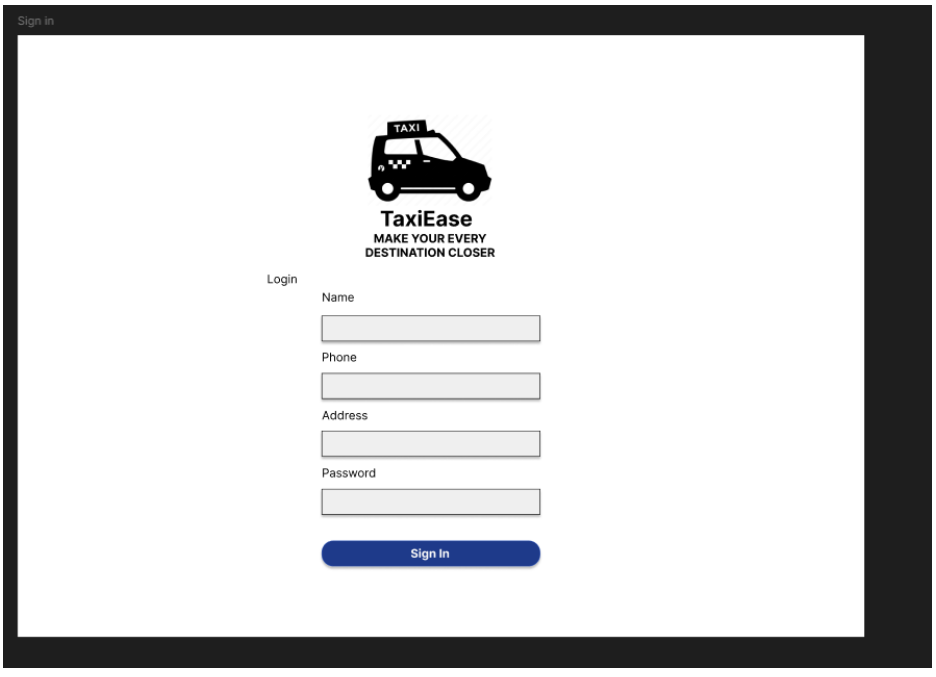

Figure 5. Sign In

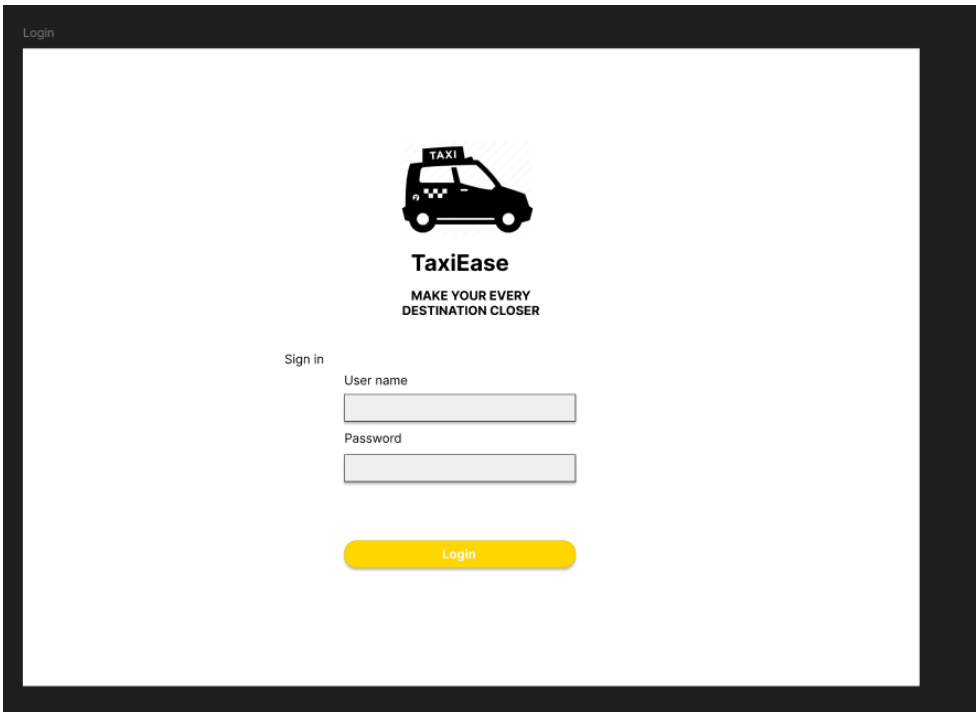

Figure 6. Log In

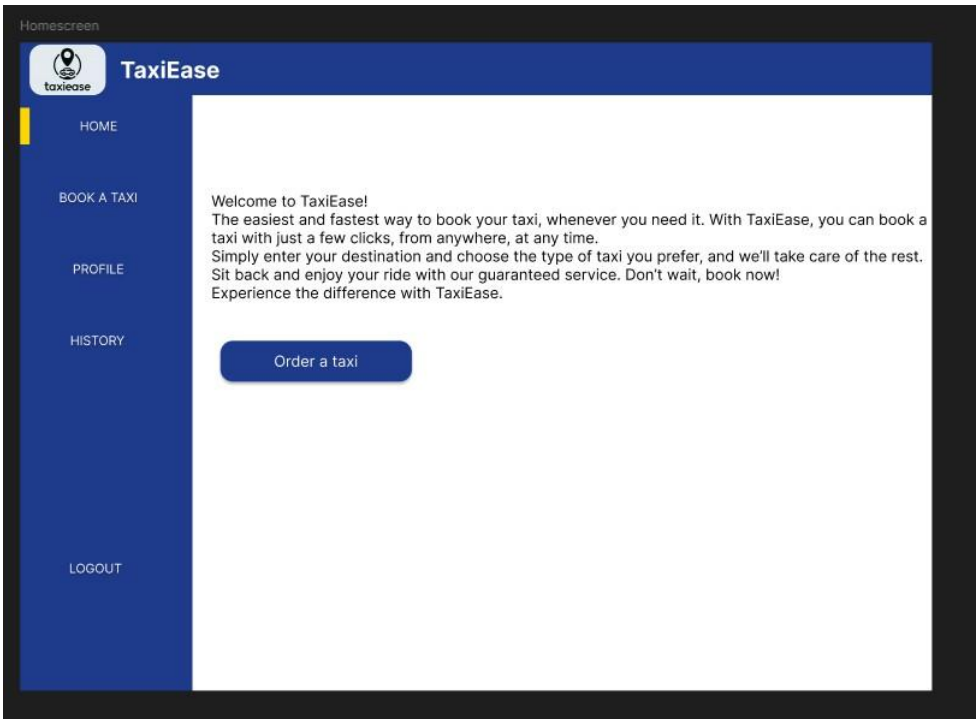

Figure 7. Home Page

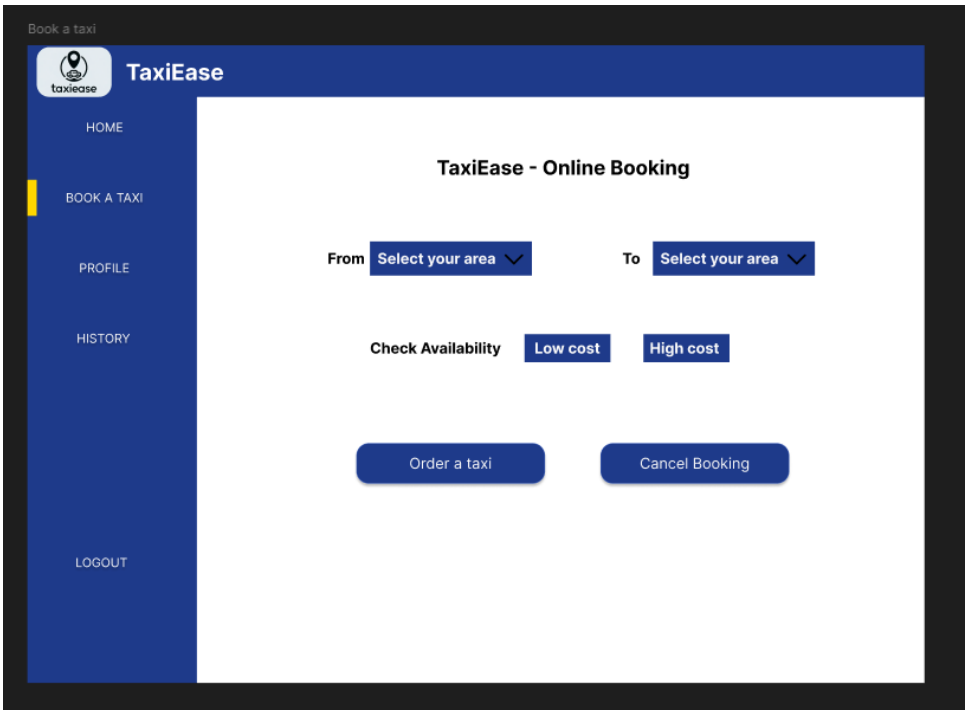

Figure 8. Booking System

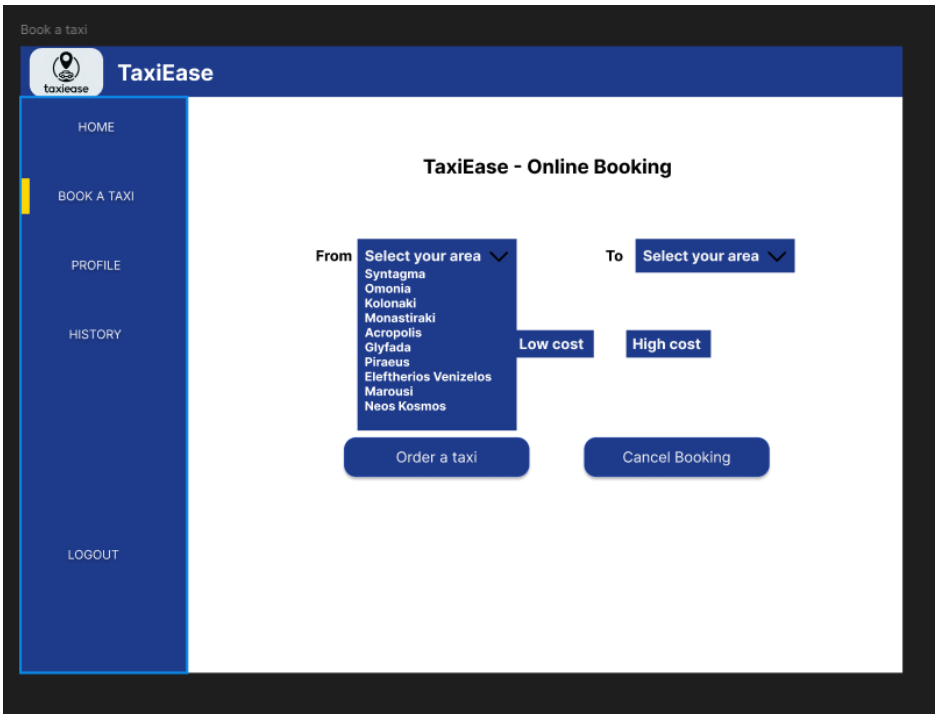

Figure 9. Booking System 2

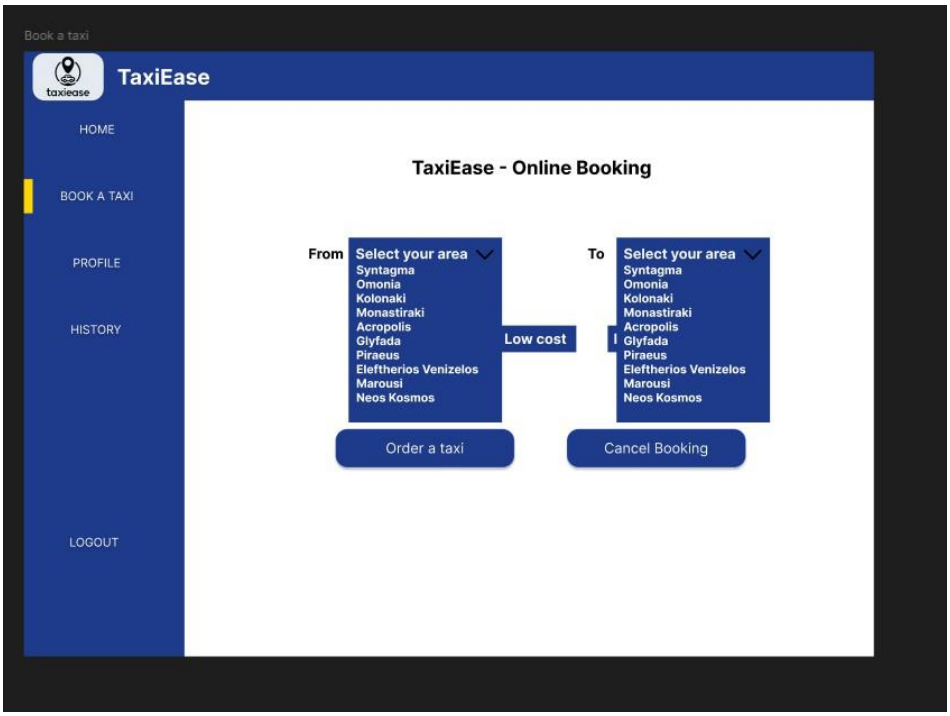

Figure 10. Booking System 3

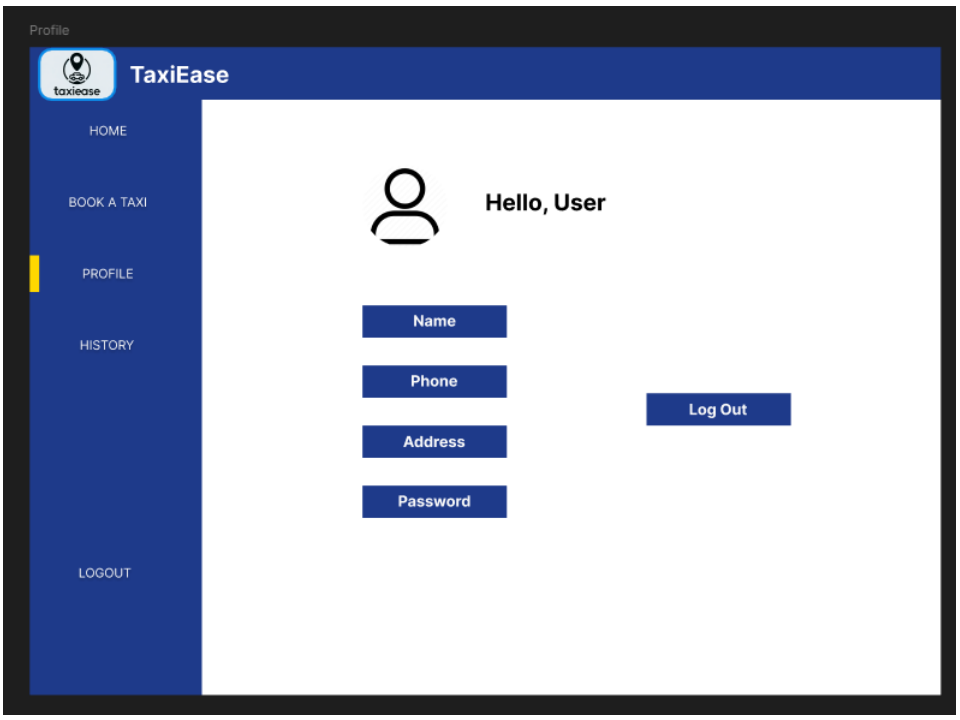

Figure 11. Profile

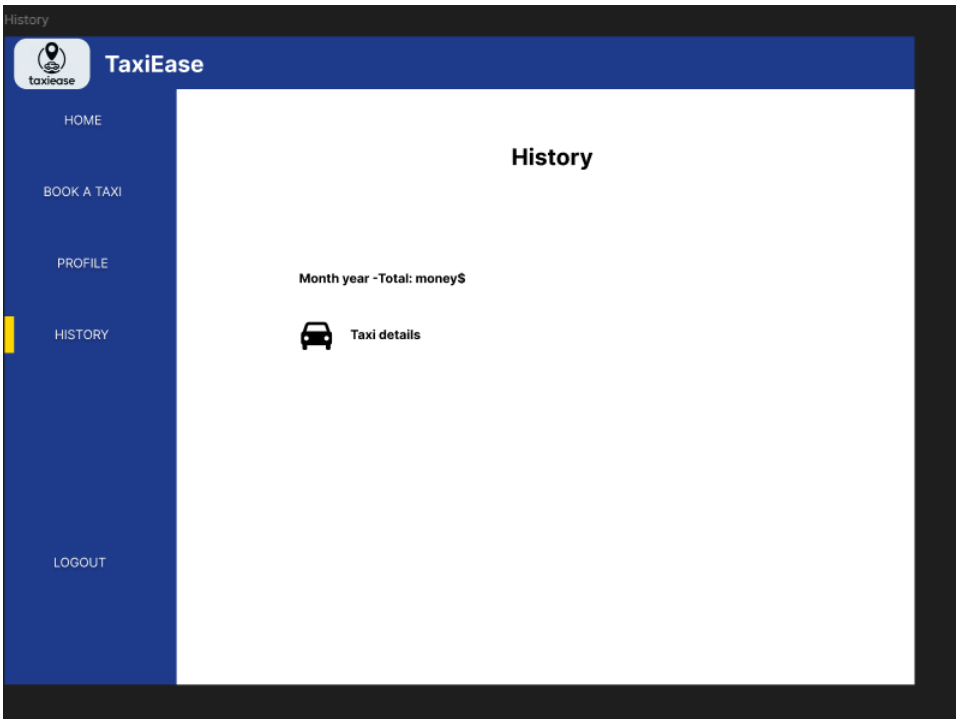

Figure 12. History

Testing

The process of testing was done in a sequence of organized scenarios that attempted to test the functionality, input validation, and navigation flow through all the major components of the application. The tests helped to ensure the system acts as expected in both valid and invalid input situations.

Firstly, in the splash screen, the user is offered a choice between Sign in and Log in. When choosing the Log In option and entering credentials that do not match any registered user, the system shows an error message that the user does not exist and redirects automatically to the Sign In screen. On the other hand, if the user selects the Sign In option and properly fills in all the details required, then it redirects successfully to the Home Page.

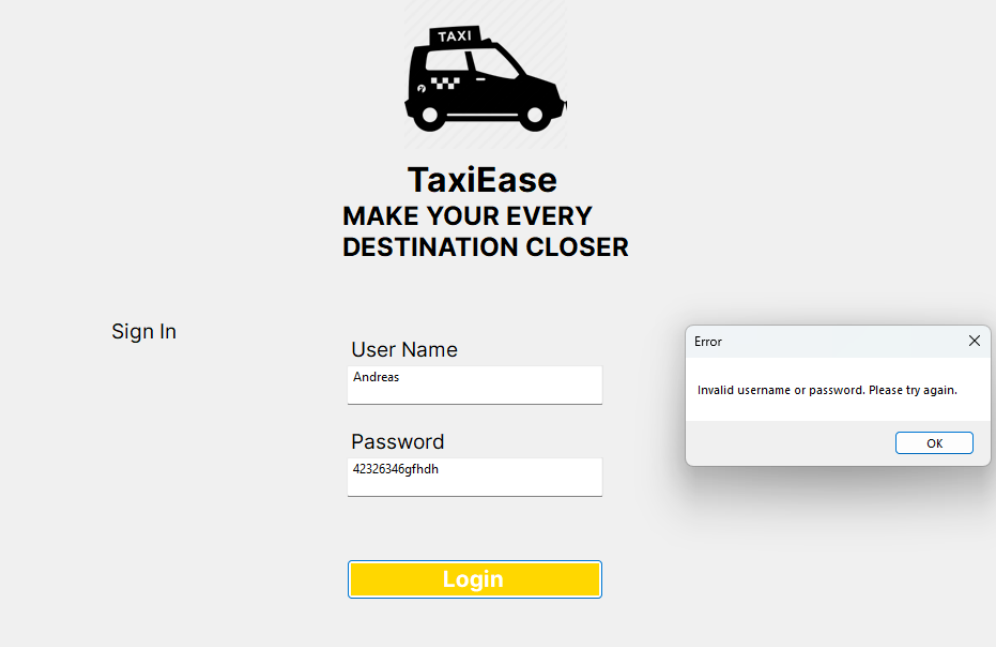

Figure 13. Testing 1

Some input validation rules are implemented during the registration process. For instance, the Name entry should consist only of alphabetic characters. Upon entry of numbers or alphanumeric data, like Andreas331 or 321Andreas, an error message is presented asking the user to input only alphabets. The same validation is applicable to the Phone entry that should accept only numeric data. Inputting characters in this entry triggers a corresponding error message. For the Password entry, any mix of characters and numbers can be input as long as the cumulative length is greater than 8 characters. When an entry less than 8 characters in length is input, a corresponding warning is triggered by the system.

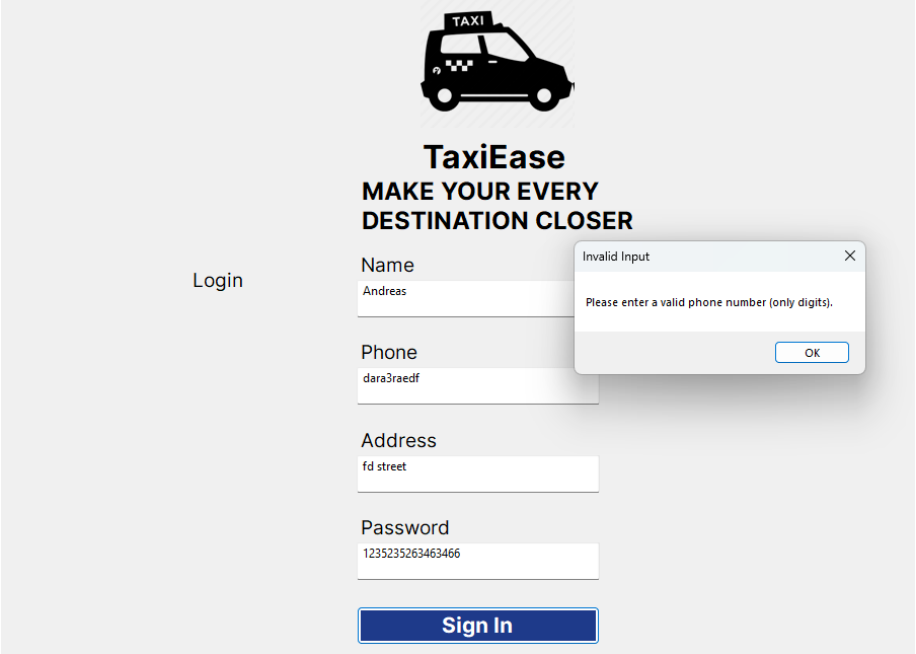

Figure 14. Testing 2

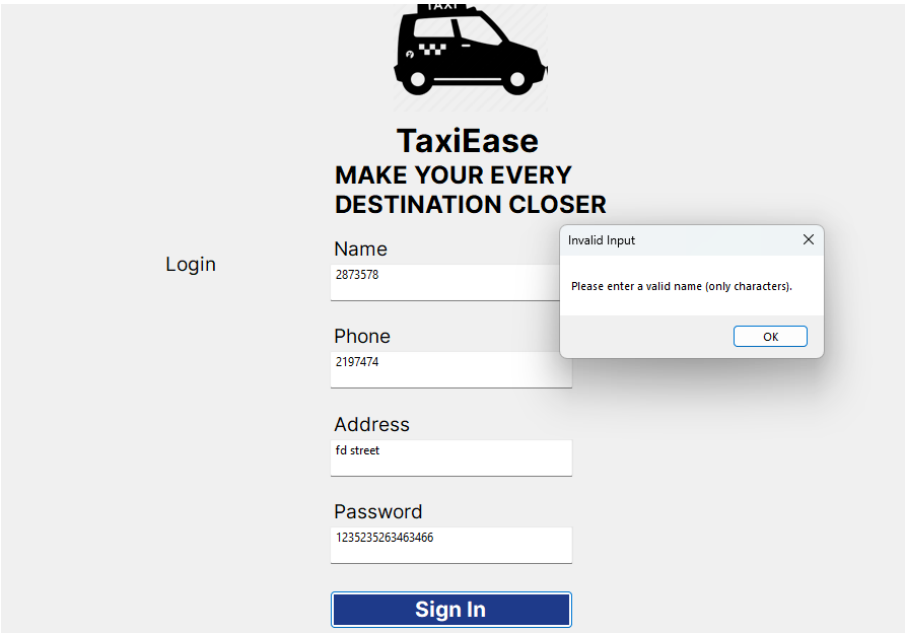

Figure 15.Testing 3

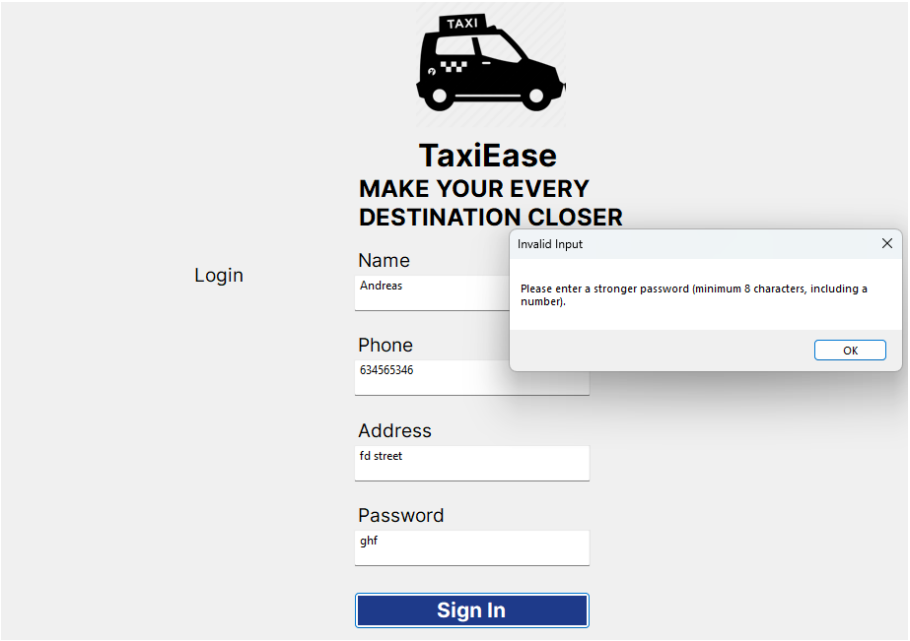

Figure 16. Testing 4

After a successful login or registration, the user is redirected to the Home Page featuring four primary options: Book a Taxi, Profile, History, and Log Out. Clicking on Log Out returns the user to the original splash screen. The option to book a Taxi takes the user to a booking form in which the user has to choose a pickup place, destination, and availability level. When the user clicks on Cancel, if any fields are left blank or if fields are only partly filled in, the system shows a message with the missing fields. For instance, if the user has chosen only the “From: Omonia” option and has set “Availability: High Cost” but left the “To” field blank, an alert message pops up to say that some fields are not selected.

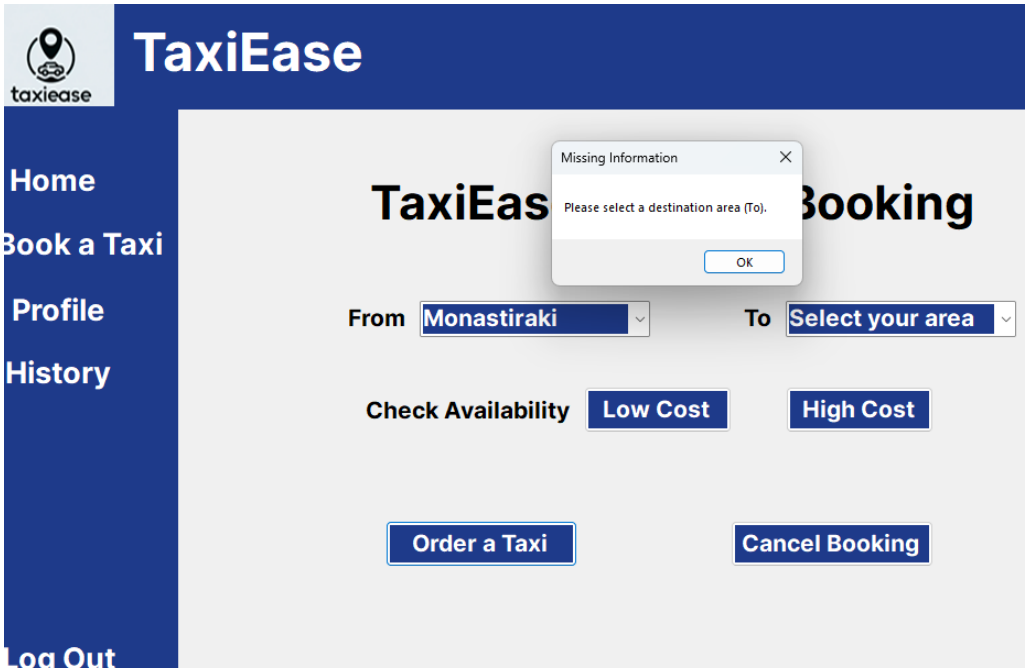

Figure 17. Testing 5

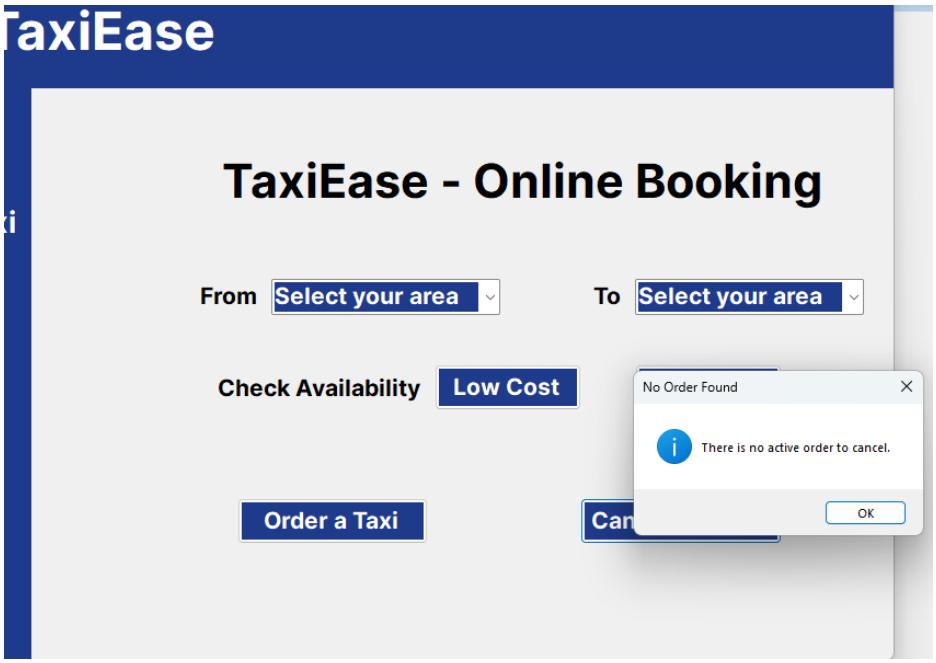

Figure 18. Testing 6

In the History section, the user can view previous bookings along with their respective costs. If bookings were never made before, the system shows that no history is present. Otherwise, a list of past rides is shown with all associated details.

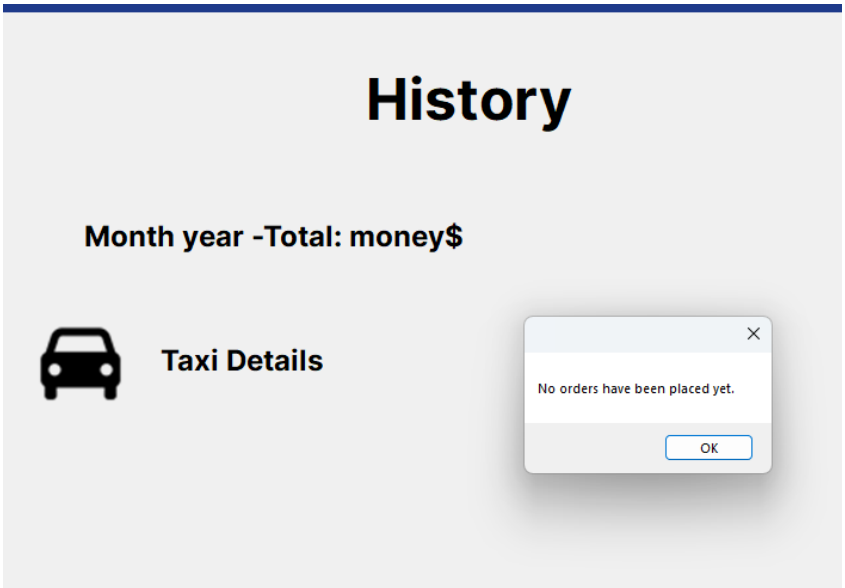

Figure 19. Testing 7

The application reacted appropriately to all the test situations mentioned above with precise error messages and a smooth and rational user experience. All the key points in the process are supported by validation rationale to maintain both usability and the avoidance of incorrect activities.

TaxiEase – Windows Forms Taxi Booking System

TaxiEase is a desktop application developed using Windows Forms (C#), designed to simulate a simple taxi booking platform.

Users can register, log in, book rides, view their profile, and review their ride history.

1. Features

- Splash screen with navigation to Login and Sign Up

- User registration with validation (name, phone, address, password)

- Secure login functionality

- Home page with access to:

  • **Taxi booking**

  • **Profile information**

  • **Ride history**

  • **Logout**

- Booking system with route and availability selection

- Local file-based backend (no database)

- Error handling and user feedback for all inputs

2. Project Structure

```
TaxiEase1
│
├── Forms
│   ├── SplashScreen.cs
│   ├── LoginForm.cs
│   ├── SignUpForm.cs
│   ├── HomePage.cs
│   ├── BookTaxi.cs
│   ├── History.cs
│   └── Profile.cs
│
├── Resources
│   └── Icons, Images, etc.
│
├── Data
│   ├── users.txt
│   └── orders.txt
│
├── Program.cs
└── README.txt
```

Technologies Used

- .NET Framework

- Windows Forms (WinForms)

- C# Programming Language

- File I/O for data handling (users and orders stored in .txt files)

How to Run

1. Open the project in Visual Studio.

2. Build the solution.

3. Run the project (F5 or Start).

4. Navigate using the splash screen to sign up or log in.

5. Explore all features from the home page.

Note: No external database is used. All user and booking data are saved in local .txt files in the Data folder.

Input Validation

- Name: Only letters allowed.

- Phone: Only digits allowed.

- Password: Minimum 8 characters.

Testing

Various test cases were applied to all forms (Login, Sign Up, Booking, History).

Detailed descriptions are available in the project documentation (see Taxi_Report_Andreas_Kokkinos.pdf).

License

This project was created as part of a university coursework assignment. Free to use for educational purposes.

Conclusion

TaxiEase provides an easier taxi booking experience through a simple and intuitive interface. The well-designed features of the app make it convenient for users to navigate through the app and perform activities like booking a ride or editing profiles without much hassle. The system is designed to be functional and user-friendly, offering several conveniences like cost estimate, booking history, and editing personal details. The use case diagram involves the significant workflows of the system so that the user's requirements are fulfilled effectively.

With TaxiEase, the users have a system that is working for them, be it in the form of a convenient way of booking a taxi or simply keeping track of their travel history. The application offers a simple, enjoyable experience for passengers, and that makes it an essential application for travel in the modern age.

Bibliography

[1] W. S. L. Nasution and P. Nusa, “UI/UX Design Web-Based Learning Application Using Design Thinking Method,” *ARRUS Journal of Engineering and Technology*, vol. 1, no. 1, pp. 18–27, Aug. 2021, doi: https://doi.org/10.35877/jetech532

[2] N. Ensmenger, “The Multiple Meanings of a Flowchart,” *Information & Culture*, vol. 51, no. 3, pp. 321–351, 2016, doi: https://doi.org/10.2307/44667617. Available: https://www.jstor.org/stable/44667617

[3] M. N. Arifin and D. Siahaan, “Structural and Semantic Similarity Measurement of UML Use Case Diagram,” *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 11, no. 2, p. 88, Jul. 2020, doi: https://doi.org/10.24843/lkjiti.2020.v11.i02.p03

[4] N. Roy-Hubara, L. Rokach, B. Shapira, and P. Shoval, “Modeling Graph Database Schema,” *IT Professional*, vol. 19, no. 6, pp. 34–43, Nov. 2017, doi: https://doi.org/10.1109/mitp.2017.4241458

[5] “Windows Forms Programming in Visual Basic .NET,” *Google Books*, 2025. Available: https://books.google.gr/books?hl=el&lr=&id=ee_dZN3y6fAC&oi=fnd&pg=PR33&dq=windows+forms+application&ots=5Qe2sCF9i0&sig=wEoPp_ey5hBhPVRh6u87PjuxP0I&redir_esc=y#v=onepage&q=windows%20forms%20application&f=false[Accessed: Mar. 29, 2025]