



Daffodil International University

Department of Software Engineering SE-431

Project/thesis Project Documentation

Travel-ways

Submitted by

MD Naeem UL Islam

ID: 211-35-3170

Department of Software Engineering

Daffodil International University

Supervised by

MD Rajib Mia

Lecturer (Senior

Scale)

Department of Software Engineering

Daffodil International University

This Project report has been submitted in fulfilment of the requirements for the

Degree of Bachelor of Science in Software Engineering.

APPROVAL

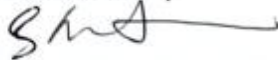
This thesis titled on "Travel-Way", submitted by **Md Naeem ul Islam (ID: 211-35-3170)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Chairman

Dr. Imran Mahmud
Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology Daffodil
International University



Internal Examiner 1

Md Shohel Arman
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 2

Md. Rajib Mia
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



External Examiner

Md Habibur Rahman
Associate Professor
Department of Computer Science and Engineering
Islamic University, Bangladesh

DECLARATION

I announce that I am rendering this study document under Md. Rajib Mia, Lecturer (Senior Scale) Department of Software Engineering, Daffodil International University. I therefore, state that this work or any portion of it was not proposed here therefore Bachelor's degree or any graduation.

Supervised By



Md. Rajib Mia

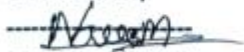
Lecturer (Senior Scale)

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

Submitted by



Md. Naeem Ul Islam

ID: 211-35-3170

Department of Software Engineering

Daffodil International University

ACKNOWLEDGEMENT

I am MD Naeem Ul Islam, and as I express my sincere thanks to my respected supervisor, MD Rajib Mia, Lecturer, Department of Software Engineering, I would like to highlight that he provided unwavering support and insight throughout this project.

His insights and timely feedback shaped the development of my conception “Travel Way”. It would have been much tougher without his guidance.

I want to thank my family as well for their unconditional patience, which allows me immense motivation throughout the process. Their unwavering belief in me provided me with unprecedented strength throughout every challenge.

ABSTRACT

For those who enjoy travelling but don't want to deal with the hassle of planning, Travel Way is a straightforward and practical website. Travel Way combines everything in one convenient location, saving you the trouble of browsing through dozens of websites to find the ideal tour, hotel, or guide.

Users can book their preferred destinations, explore exciting travel packages, and view all the crucial information—such as costs, descriptions, and reviews—with just a few clicks. It is intended to help people make hassle-free, confident travel decisions while also saving time and reducing confusion.

TABLE OF CONTENT

INDEX

Approval.....	i
Declaration	ii
Acknowledgement.....	iii
Abstract	iv
Chapter 1 Introduction	1-3
1.1 Project Overview	1
1.2 Project Purpose.....	2
1.2.1 Proposed System	3
Chapter 2 System Analysis	4-8
2.1 Feasibility Analysis	4
2.1.1 Technical Feasibility.....	4
2.1.2 Operational Feasibility	4
2.2 Functional Requirements	5
2.3 System Requirements	6
2.4 Non-Functional Requirements.....	7
2.5 Performance.....	8
Chapter 3 System Design	9-22
3.1 Development Model	9-10
3.1.1 Use Case Diagram	11
3.1.2 Use Case Descriptions	12-26
3.2 Activity Diagram.....	27-30
3.3 Sequence Diagram.....	31-32
3.4 ERD Diagram	33
3.5 Class Diagram	34
3.6 Block Diagram	35

Chapter 4 Development Tool and Technology	36
4.1 Integrated Development Environment (IDE).....	36
4.2 Programming Language	36
4.3 User Interface Design.....	36
4.4 Database	36
4.5 Deployment and Hosting.....	36
Chapter 5 System Testing	37-43
5.1 Testing Features	37
5.1.1 Feature to be tested.....	37-38
5.2 Testing Strategies	39
5.2.1 Test Approach	39
5.2.2 Pass/Fail Criteria	40
5.2.3 Testing Schedule	41
5.3 Test Cases.....	42-43
Chapter 6 User Manual	44-51
6.1 Landing Page.....	44-51
Chapter 7 Conclusion.....	52-54
7.1 Project Link	52
7.2 Limitations.....	53
7.3 Future Scope.....	54
Reference.....	55

CHAPTER 1: INTRODUCTION

1.1 Project Overview

Travel Way is a cutting-edge web-based travel booking tool made to make it easier and more enjoyable for users to research, arrange, and validate their trip experiences. This platform was developed as a final defence project and is thoughtfully designed to satisfy academic requirements while addressing a practical issue—making travel planning more accessible and easy.

Travellers can view destination details, browse carefully chosen tour packages, and make reservations in an easy-to-use environment. At the same time, administrators are given a straightforward yet effective dashboard to efficiently manage user information and tour data. A clever travel companion that connects people to new adventures with clarity, ease, and convenience, Travel Way is more than just a website.

1.2 Project Purpose

Making travel planning simpler, more intelligent, and more accessible for everyone is the main objective of the Travel Way platform. It provides a comprehensive solution for finding tour packages, organising travel options, and providing administrators and passengers with a seamless booking experience. The following are Travel Way's main goals:

Travel Planning Made Easy:

Give consumers a user-friendly platform so they can browse, choose, and reserve travel packages without difficulty or needless searching.

Tour Management in One Place:

Reduce operational effort and boost efficiency by giving administrators the ability to update packages, manage tour data, and keep an eye on user activity all from a single, secure dashboard.

A Stress-Free Reservation Process:

Provide a quick and easy booking process so that customers can gather all the information they need to plan their trip, make wise choices, and confidently confirm their reservations.

Effective Interaction Between Administrators and Users:

As administrators retain complete control over the platform's data, make sure users receive real-time updates and confirmations to promote transparency and trust.

Accessible and Dependable Travel Centre:

Serve as a trustworthy go-to travel website for people seeking well-organised, validated travel plans; this is especially helpful for families or first-time visitors seeking reliable information.

Travel Way was created to make travelling easier and more pleasurable. The system encourages users to explore the world with clarity and confidence by eliminating confusion, cutting down on planning time, and providing a clear, guided platform.

1.2.1 Proposed System

By providing an organised, user-friendly, and effective platform, the proposed Travel Way system seeks to streamline and enhance the travel booking process. By providing a smooth experience for finding, organising, and verifying tour packages, the system is made to satisfy the demands of both administrators and passengers. The following are important aspects of the suggested system:

- Admin Dashboard for Tour and User Management;
- Simplified Tour Discovery and Booking Confirmation;
- Intuitive and Responsible User Interface (UI);
- User Authentication and Role Management;
- Fast and Reliable Booking Confirmation Confirmation

Chapter 2: SYSTEM ANALYSIS

Feasibility Analysis

In terms of development, deployment, and use, a feasibility study helps ascertain whether the Travel Way web application is practical, reasonable, and successful. This covers a thorough investigation of its technical and operational viability to evaluate system performance in practical environments.

2.1.1 Technical Feasibility

Modern web technologies including React for the frontend and Node.js with MongoDB for backend and database operations shape the Travel Way platform. These technologies guarantee that desktop and mobile browsers can technically be used from the same standpoint. Its efficient deployment on several devices is made possible by responsive design and optimal performance, so preserving speed and usability without compromising any other aspect.

2.1.2 Operational Feasibility

Both users and administrators will find it easy to interact with the system because of its straightforward, user-friendly, and effective design. While administrators can easily manage data through a dedicated dashboard, travellers can complete bookings without any confusion. This user-friendliness encourages system adoption and user engagement. Its long-term sustainability and daily operational performance are further improved by the functional role-based access control, real-time feedback, and clear user interface.

2.2 Functional Requirements

- The system will allow users to create a personal account.
- Invalid login credentials will not be accepted by the system. Logging in with a valid email address and password is only possible for registered users.
- Users will need to log in in order to make a reservation, but they will be able to peruse the available tour packages without doing so.
- If a user forgets their password, the system will let them reset it.
- Each tour package's location, duration, and cost will be displayed to users.
- Users will be able to confirm and book a selected tour from the available packages.
- The system will save the information in the backend and show booking success messages.
- With secure login credentials, administrators will be able to access the system.
- Incorrect attempts to log in as an admin will be rejected by the system.
- Admins will be able to add, edit, and remove tour packages with a simple dashboard interface.
- Admins will be able to see a full list of users registered on the site.
- The system will allow admins to remove users from the site if they need to.
- Admins will be able to view and update their own account information.
- The system will restrict access to admin operations by user roles. There will be an interactive and responsive interface for all users with customizations according to their privileges (user/admin).

2.3 System Requirements

➤ **Hardware Requirements:**

Servers:

- Multi-core processors are required to handle user requests and efficiently manage the travel data.
- A minimum of 8GB of RAM is recommended for optimal performance of the backend and database services.
- Adequate storage capacity for Tour Packages, User Details, and Booking Records shall be ensured.

End-User Devices:

- Compatible with modern web-browsers like Chrome, Firefox, Edge, and Safari.
- Supports any device which can access the internet, i.e., desktops, laptops, tablets, and mobile phones.

➤ **Database Requirements:**

Tour- and Booking-Related Data:

Package title, description, location, price, and availability.

Booking information: user ID, tour ID, date of booking, and booking status.

User Profiles:

Full name, email address, password (encrypted), and role (user/admin).

Admin Controls:

Role-based access to manage tour data and user accounts.

Log system activity for monitoring and audit purposes.

➤ **Security Requirements:**

- User Authentication:
 - Secure login with password encryption.
 - Hash Algorithm is used for Security.

2.4 Non-Functional Requirements

1. The app should always represent the highest responsiveness to user interactions with great agility while being restricted by page loads, form submissions, and navigation.
2. The system should give real-time feedback in terms of visuals (like loading indicators, success messages, or error alerts) to ameliorate the user experience and notify the user on the outcome of his action.
3. A platform should be smooth enough to keep minimal response and processing times, especially during data fetching, booking, and administrative tasks.
4. There must be a login for users and administrators alike, wherein considerations such as password encryption and validation should be accounted for to prevent unauthorized entry.
5. The application should have advanced access control systems to ensure protection of sensitive features or administrative functions from all except authorized users.

2.5 Performance

Reaction Time:

Goal: We want the system to feel lightning-fast, responding immediately when users book things, browse around, or fill out forms.

Benchmark: For the most part, we're aiming to have the system react in less than 1-2 seconds.

Speed of Loading:

Goal: We want to make loading times as short as possible for tour packages, the admin dashboard, and user information.

Benchmark: Even when the network isn't perfect, pages and content should still load up within 1-3 seconds.

Error Resolution:

Goal: We want to handle any errors (whether from the system or the user) smoothly, with helpful messages that clearly explain what went wrong and how to fix it.

Benchmark: We need to keep errors to an absolute minimum and make sure the system is up and running nearly 100% of the time.

Experience of the User:

Goal: We want to create a user experience that's smooth, easy to use, and visually appealing, both for regular users and administrators.

Benchmark: We'll measure success by looking at user satisfaction scores from testing and usability studies to make sure people are really happy with the experience.

Chapter 3: System Design

3.1 Development Model:

Architecture of Three Levels For my project on a website that allows people to book tours, I used it. This approach divides the system into three parts: the user interface, where users interact, the middle section, which handles tasks like bookings, and the database, which contains all of the data. It keeps things organised and simplifies updating.

The system is designed in three main layers:

1. Frontend (user interface)-

Users and Admins interact through a clean UI to:

- Register/Login
- View tours and hotels
- Book tours and make payments
- Admins manage tours, bookings, and users

2. Backend (logic and control)-

- Processes user actions (like booking or adding to wishlist)
- Connects with the database.
- Controls access (user vs admin).

3. Database (storage layer):

- Stores user info, bookings, tours, payments, etc
- Admin actions and guide info.

Modules in the System :

Module	Description
Authentication	Register, Login, Logout, Session handling.
User Module	Profile management, Wishlist, Booking history.
Admin Module	Add/Update/Delete tours, Manage users/guides/bookings.
Tour Module	View tours, details, pricing, availability.
Booking Module	Assign guides to tours, manage guide info.
Payment Module	Secure payment.
Guide Module	Assign guides to tours, manage guide info.

Figure: System Modules and Their Descriptions

3.1.1 Use Case Diagram

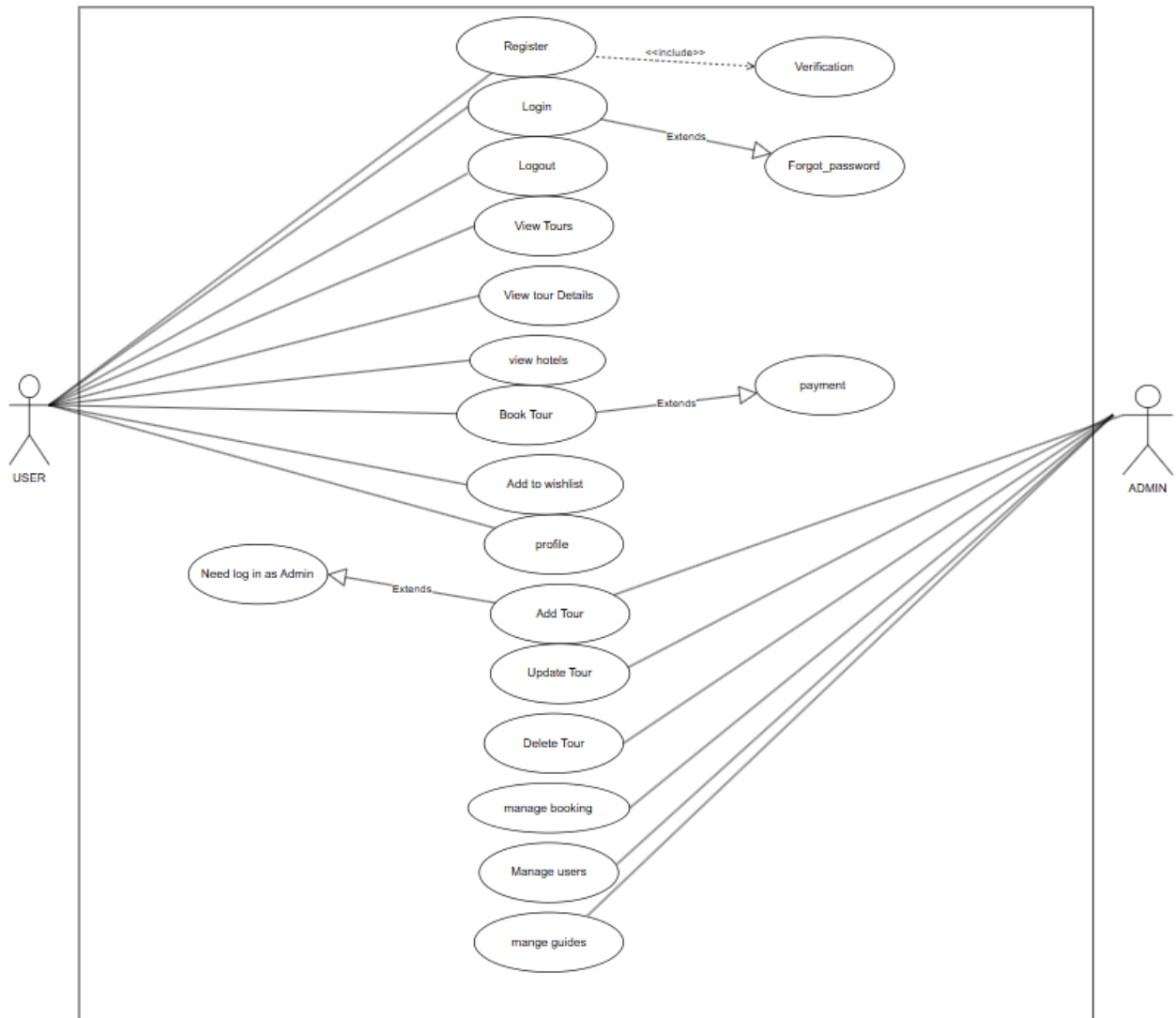


Figure: Use case diagram

3.1.2 Use Case Descriptions

1.

Use case Name	Register
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. The user is on the registration page.
Flow and Counts	<ol style="list-style-type: none">1. The user enters registration details (name, email, password, etc.).2. The system validates the inputs.3. The system sends a verification link or OTP.4. The user completes verification.5. The system creates a new user account.
Alternative Flow	<ol style="list-style-type: none">1. If the email is already registered, the system shows an error.2. If verification fails, the system asks to retry.

2. Use Case Descriptions

Use case Name	Log in
Actor	User / Admin
Pre-Condition	<ol style="list-style-type: none">1. The user has already registered.
Flow and Counts	<ol style="list-style-type: none">1. The user enters username and password.2. The system validates credentials.3. If correct, the user is redirected to the dashboard.
Alternative Flow	<ol style="list-style-type: none">1. If credentials are invalid, an error message is shown.2. User can select "Forgot Password" (extension).

3. Use Case Descriptions

Use case Name	View Tours
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. The user is logged into the system.
Flow and Counts	<ol style="list-style-type: none">1. The user selects "View Tours".2. The system fetches and displays all available tours.
Alternative Flow	<ol style="list-style-type: none">1. If no tours are available, a message is shown.

4. Use Case Descriptions

Use case Name	View Tour Details
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. Tours are visible.
Flow and Counts	<ol style="list-style-type: none">1. The user clicks on a tour from the list.2. The system fetches detailed info about that tour and displays it.3. The user will be able to easily find all the information and see the name of hotel and his tour guide.
Alternative Flow	<ol style="list-style-type: none">1. If tour details fail to load, an error message shown.

5. Use Case Descriptions

Use case Name	Book Tour
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. The user is logged into the system.2. The tour details are visible.
Flow and Counts	<ol style="list-style-type: none">1. The user selects a tour book.2. System prompts for payment.3. The user selects payment option.4. The system confirms payment and saves booking.5. Booking confirmation is shown.
Alternative Flow	<ol style="list-style-type: none">1. If the user isn't logged in, they're redirected to login.2. If payment fails, the user is prompted retry.3. If the tour is unavailable, an appropriate message is displayed.

6. Use Case Descriptions

Use case Name	View Hotels
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. The user is logged into the system.2. Hotels are associated with the selected tour.
Flow and Counts	<ol style="list-style-type: none">1. The system displays a list of hotels available for that tour.2. The user will be able to easily get an idea about the hotels that are included in their tour package.
Alternative Flow	<ol style="list-style-type: none">1. If no hotels are linked, a message is shown.2. If the system fails to load hotels, an error is displayed.

7. Use Case Descriptions

Use case Name	Add to Wishlist.
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. The user is logged in.
Flow and Counts	<ol style="list-style-type: none">1. The user views a tour.2. The user clicks "Add to Wishlist".3. A confirmation is shown.
Alternative Flow	<ol style="list-style-type: none">1. If the user is not logged in, the system redirects to login.2. If the tour is already in the wishlist, a message is shown.

8. Use Case Descriptions

Use case Name	View Bookings
Actor	User
Pre-Condition	<ol style="list-style-type: none">1. The user is logged in.
Flow and Counts	<ol style="list-style-type: none">1. The user navigates to the "My Bookings" section.2. The system retrieves and displays the user's past and upcoming bookings.
Alternative Flow	<ol style="list-style-type: none">1. If no bookings exist, a "No bookings found" message is shown.2. If there is a system error, an appropriate message is shown.

9. Use Case Descriptions

Use case Name	Logout
Actor	User / Admin
Pre-Condition	<ol style="list-style-type: none">1. The user or admin is logged in.
Flow and Counts	<ol style="list-style-type: none">1. The user clicks on the "Logout" button.2. The system ends the session and returns to the homepage or login page.
Alternative Flow	<ol style="list-style-type: none">1. If the session has already expired, the user is redirected to login.

10. Use Case Descriptions

Use case Name	Add Tour
Actor	Admin
Pre-Condition	<ol style="list-style-type: none">1. Admin is logged in.
Flow and Counts	<ol style="list-style-type: none">1. Admin enters tour details.2. The system validates and stores the new tour.3. A success message is shown.
Alternative Flow	<ol style="list-style-type: none">1. If input is invalid or incomplete, an error message is shown.2. If the system fails to store the tour, a retry option is provided.

11. Use Case Descriptions

Use case Name	Update tour details
Actor	Admin
Pre-Condition	<ol style="list-style-type: none">1. The admin is logged in.2. Tours exist in the system.
Flow and Counts	<ol style="list-style-type: none">1. Admin selects a tour to update.2. The system displays the current data.3. Admin edits fields and submits changes.4. The system saves the updated tour.
Alternative Flow	<ol style="list-style-type: none">1. If no tour is selected, an error message is shown.2. If update fails, an error is displayed.

12. Use Case Descriptions

Use case Name	Delete Tour
Actor	Admin
Pre-Condition	<ol style="list-style-type: none">1. Admin is logged in.2. Tours exist in the system.
Flow and Counts	<ol style="list-style-type: none">1. Admin selects a tour to delete.2. The system prompts for confirmation.3. Admin confirms.4. The tour is removed from the system.
Alternative Flow	<ol style="list-style-type: none">1. If deletion fails, an error is shown.2. If tour is already deleted, a message is shown.

13. Use Case Descriptions

Use case Name	Manage Bookings
Actor	Admin
Pre-Condition	<ol style="list-style-type: none">1. The admin is logged in.2. Booking exist in the system.
Flow and Counts	<ol style="list-style-type: none">1. Admin navigates to the booking section.2. The system displays all bookings.3. Admin can edit or cancel any booking.
Alternative Flow	<ol style="list-style-type: none">1. If no bookings exist, the system shows a message.2. If modification fails, an error is shown.

14. Use Case Descriptions

Use case Name	Manage Users
Actor	Admin
Pre-Condition	<ol style="list-style-type: none">1. Admin is logged in.
Flow and Counts	<ol style="list-style-type: none">1. Admin accesses user management.2. System displays a list of registered users.3. Admin can update or delete user accounts.4. System confirms the changes.
Alternative Flow	<ol style="list-style-type: none">1. If the user cannot be found, an error message is shown.2. If changes fail to save, the system shows an error.

15. Use Case Descriptions

Use case Name	Manage Guides
Actor	Admin
Pre-Condition	<ol style="list-style-type: none">1. Admin is logged in.
Flow and Counts	<ol style="list-style-type: none">1. Admin navigates to the guide management section.2. System shows the list of tour guides.3. Admin adds, edits, or deletes guide info.4. The system saves and confirms changes.
Alternative Flow	<ol style="list-style-type: none">1. If there are no guides, a message is displayed.2. If updates fail, an error is shown.

3.2 Activity Diagram

Register

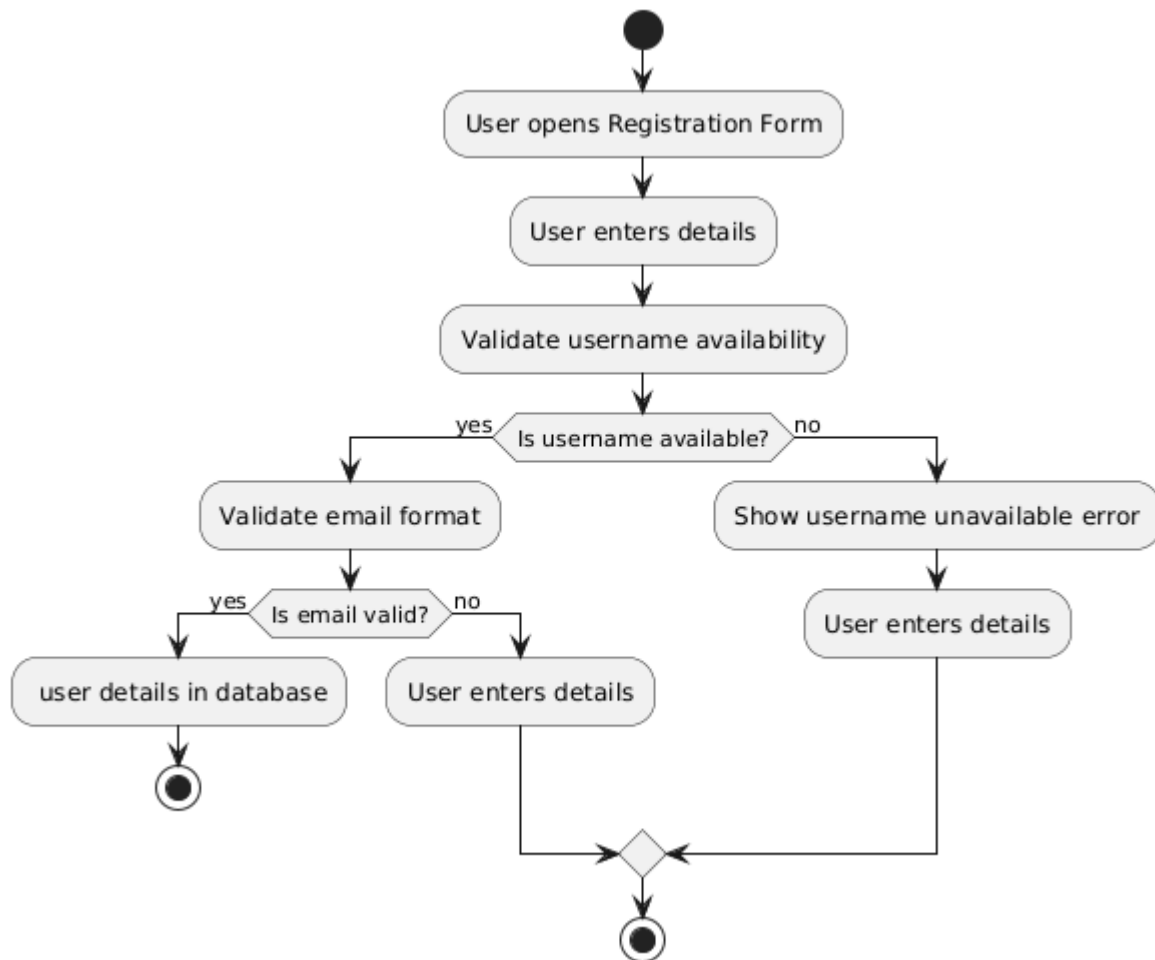


Figure: Activity Diagram for Register

Login

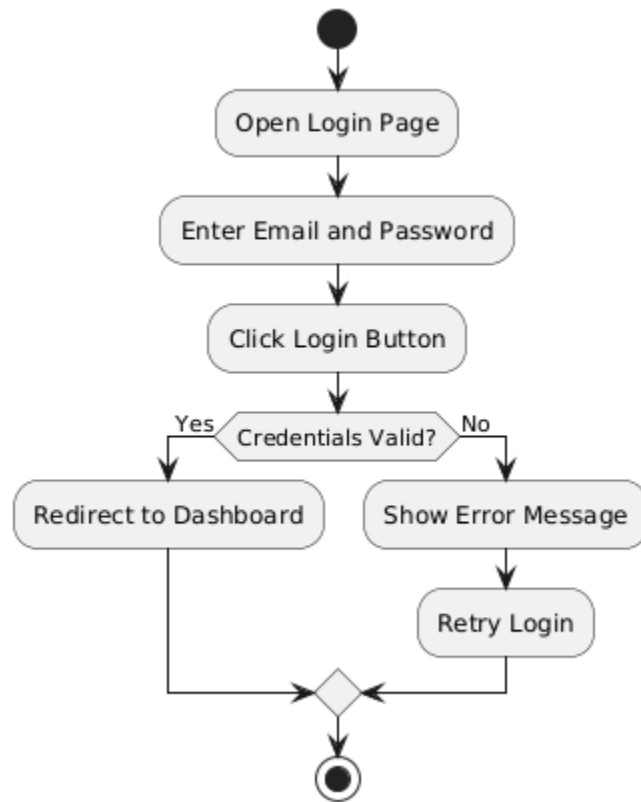


Figure: Activity Diagram for Log in

User

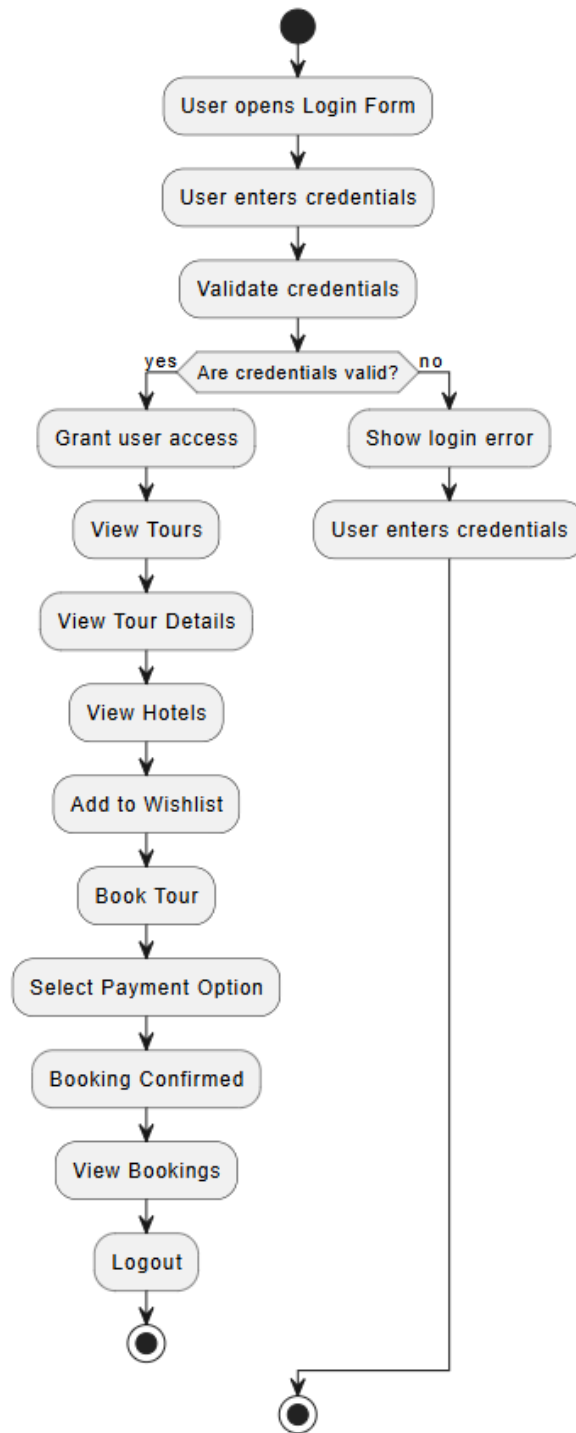


Figure: Activity Diagram for User

Admin

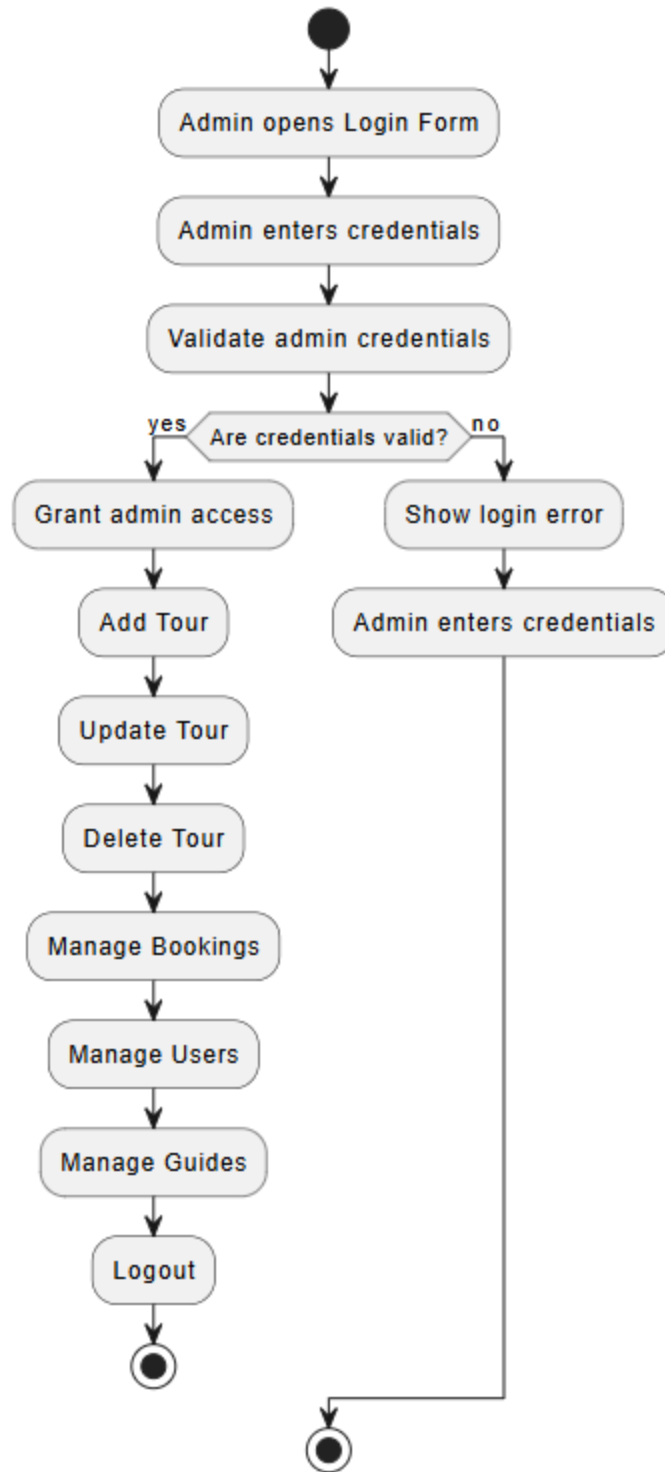


Figure: Activity Diagram for Admin

3.3 Sequence Diagram

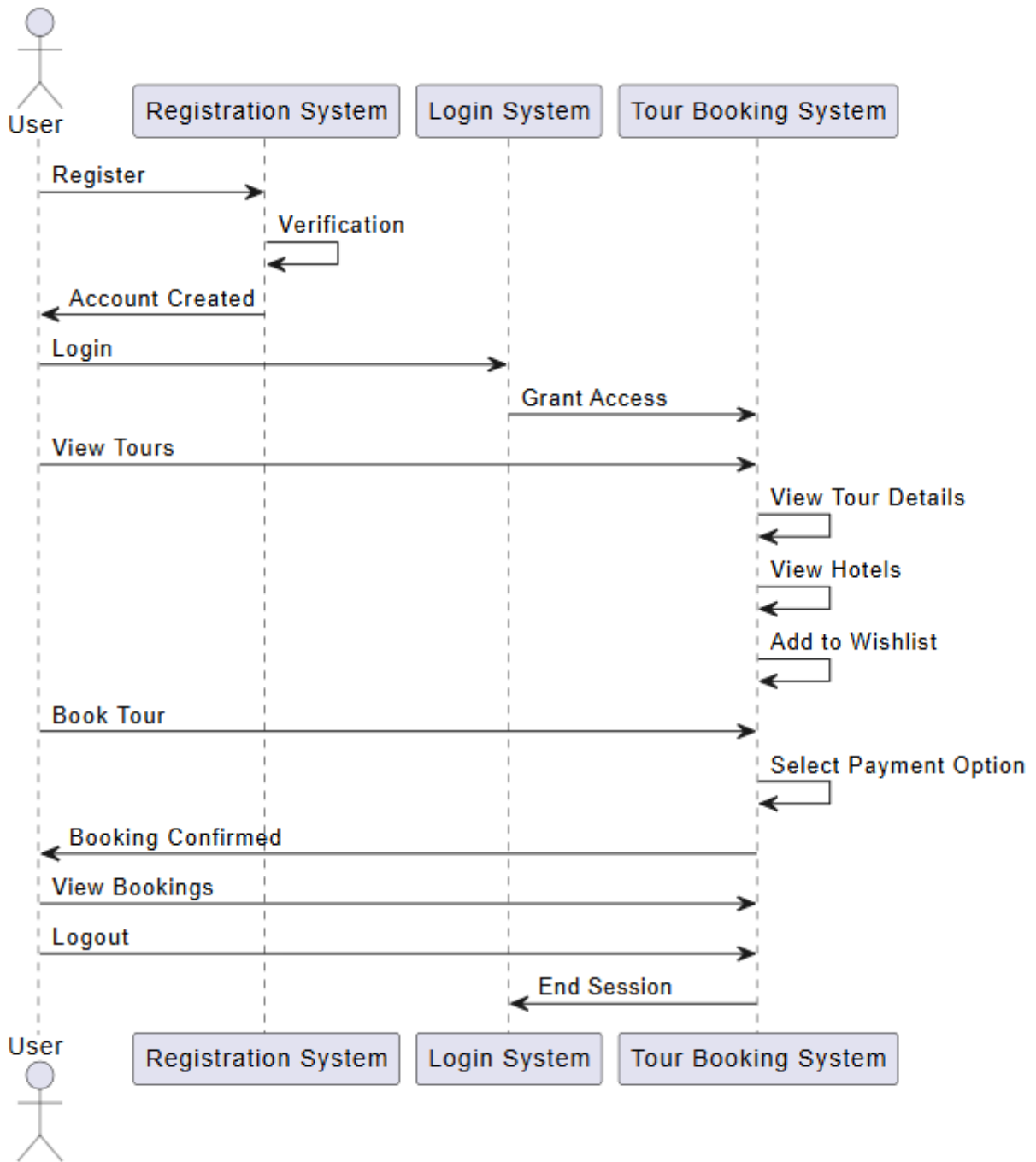


Figure: Sequence diagram for User

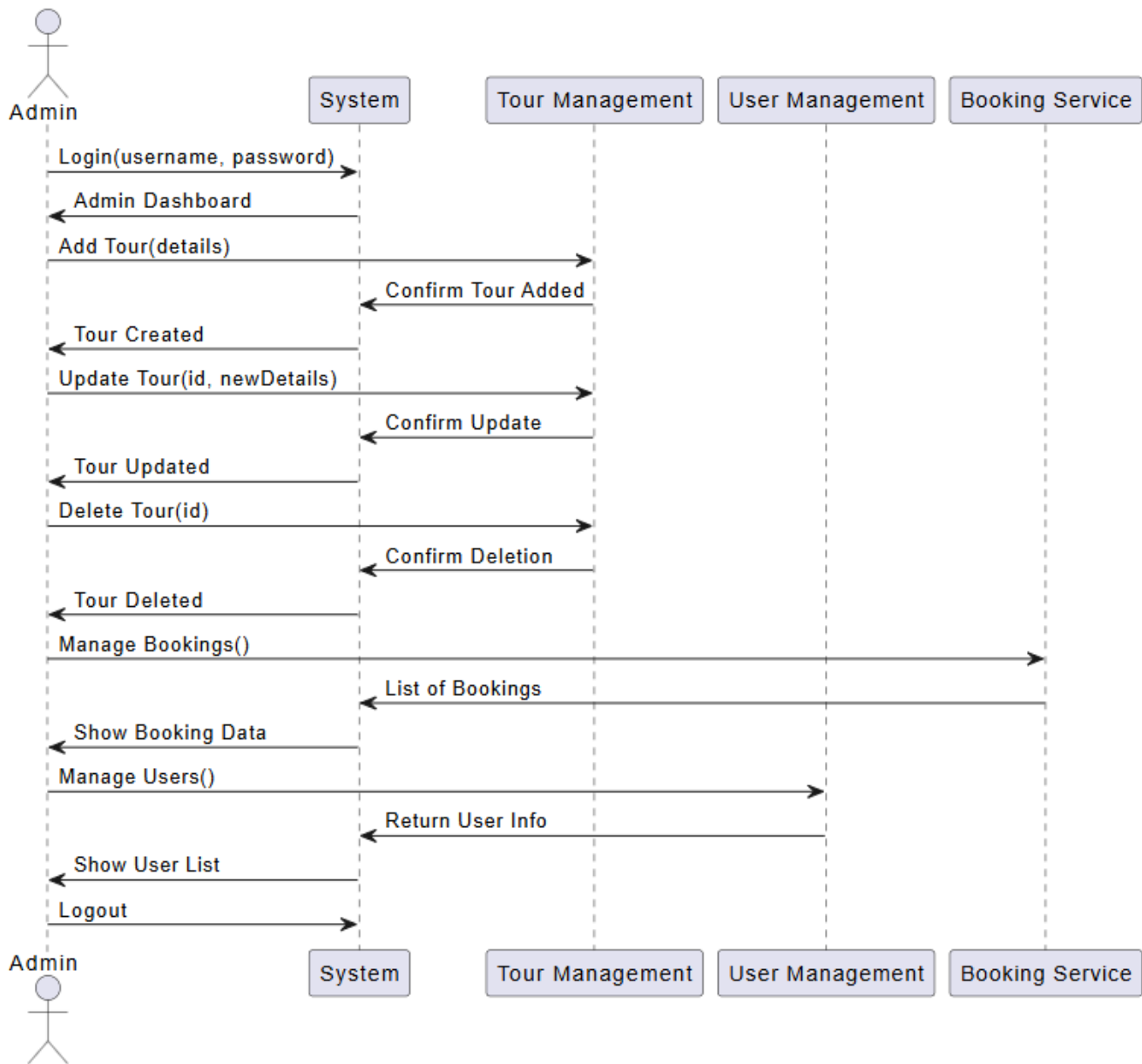


Figure: Sequence diagram for Admin

ERD Diagram

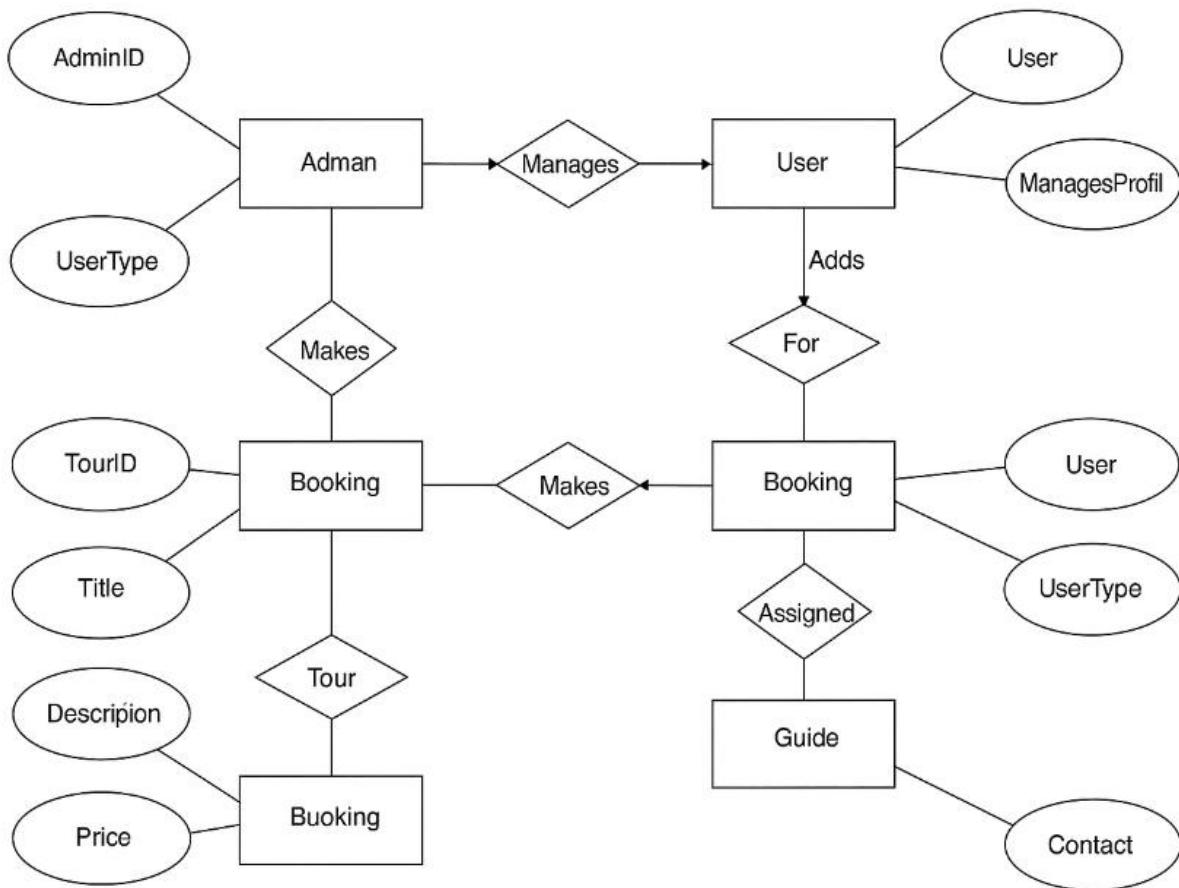


Figure: Entity-Relationship Diagram

3.4 Class Diagram

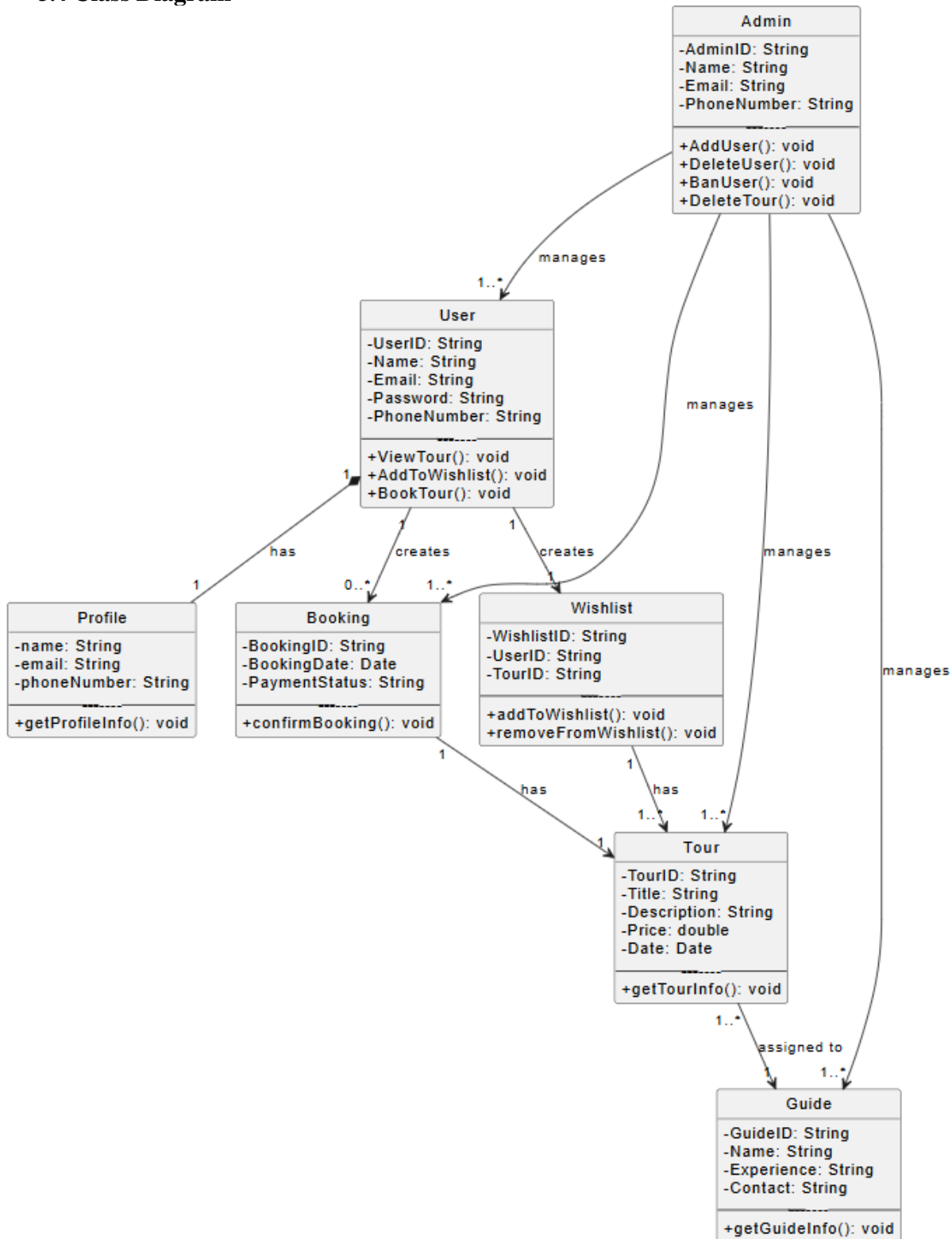
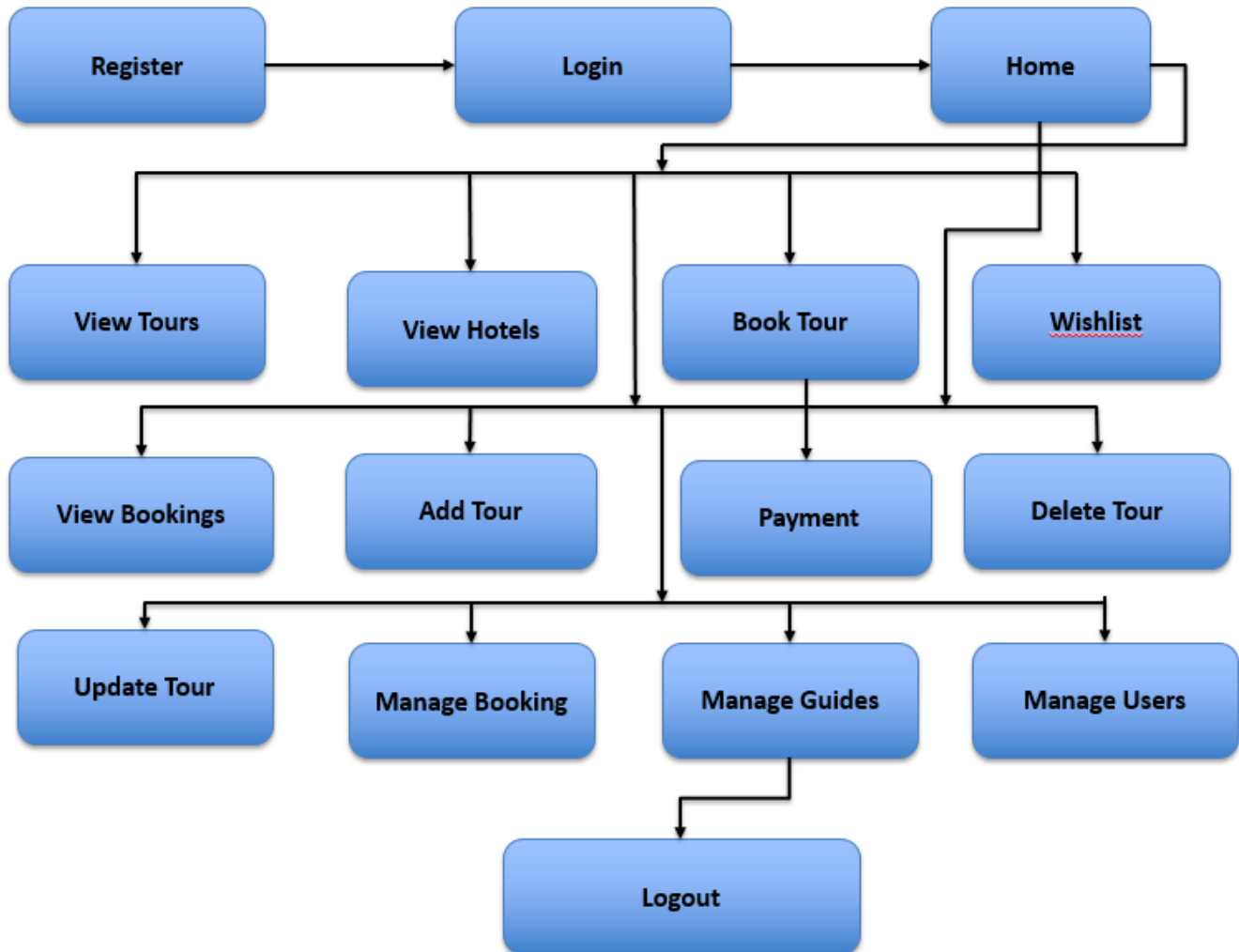


Figure: Class Diagram

3.5 Block Diagram



Chapter 4: Development Tool and Technology

4.1 Integrated Development Environment (IDE)	Visual Studio Code
4.2 Programming Language	Html 5, css, tailwind frameworks, react, javascript, node.js, express js
4.3 User interface Design	Figma
4.4 Database	mongodb
4.5 Deployment and hosting	vercel.

Chapter 5: System Testing

5.1 Testing Features

Testing new features is viewed as an addition to and modification of the existing feature system. New system functionality will be accomplished with the top system. These are meant to improve the efficacy, dependability, efficiency, and security of the online application.

5.1.1 Feature to be tested

Features	Priority Description
Log in	The user must be authenticated by email and password.
Registration	Researcher information must be saved properly.
View Tours	Users should see a list of available tours.
Book Tour	Users must be able to book selected tours.
Payment	Payment method must be selected and processed.
Wishlist	Users can save tours to wishlist.
View Bookings	Users can view their current/past bookings.

Figure: Features priority table for researcher

Features	Priority Description
Login account	The user must be authenticated by email and password.
Add Tour	Admin can add new tours.
Update Tour	Admin can edit existing tour info.
Delete Tour	Admin can remove a tour by ID.
Manage Users	Admin can view/edit/delete user accounts.
Manage Bookings	Admin can manage all bookings.
Manage Guides	Admin can assign or edit tour guide details.
Delete admin	Must delete an admin account.

Figure: Features priority table for administration.

5.2 Testing Strategies

5.2.1 Test Approach:

I used two distinct types of testing to ensure the excellent quality of my system. My primary focus was on functional and structural testing.

- Black box testing, sometimes referred to as functional testing, overlooks a system or component's internal operations in favour of concentrating solely on the outputs generated in response to specific inputs and execution conditions.
- White-box testing (also called structural testing and glass box testing) is testing that takes into account the internal operations of a system or component.

5.2.2 Pass/Fail Criteria:

User Satisfaction:

Pass: Positive comments from stakeholders, researchers, and administrators, among other end users. Fail: High user discontent or grievances with the functioning, performance, or usability of the system.

functionality:

Pass: Every important feature and function listed in the system requirements has been implemented and is operating as planned.

FAIL: Important functionalities are absent or operate improperly.

Data Accuracy:

Pass: Accurate and current information is consistently provided by the system. Fail: Data errors, inconsistent information, or out-of-date information.

Security:

Pass: To safeguard confidential research data and adhere to legal requirements, the system is equipped with strong security safeguards.

Fail: Security holes that can allow unwanted access or information leaks.

5.2.3 Testing Schedule

Test Phase Time
Testing plan create 1 Week.
Unit testing During development time.
Component test During development time.
Testing user interfaces 1 Week.
Performance testing 1 Week.
Accessibility testing 1 Week.

5.3 Test Cases:

User Authentication and Access Control:

Test Case: User Login

Steps:

- Open the app.
- Enter valid credentials (username and password).
- Click on the login button.

Expected Result:

- The user should be successfully logged into the system.

Test Case: Invalid Login Attempt

Steps:

- Open the app.
- Enter an invalid username or password.
- Click on the login button.

Expected Result:

- The system should display an appropriate error message, and access should not be granted.

Performance:

Test Case: Load Testing

Steps:

- Simulate a large number of concurrent users accessing the system.
- Monitor system response times and resource utilization.

Expected Result:

- The system should handle the load without significant performance degradation.

Test Case: Response Time**Steps:**

- Measure the response time for common operations

Expected Result:

- Response times should be within acceptable limits.

Usability:**Test Case: User Interface****Steps:**

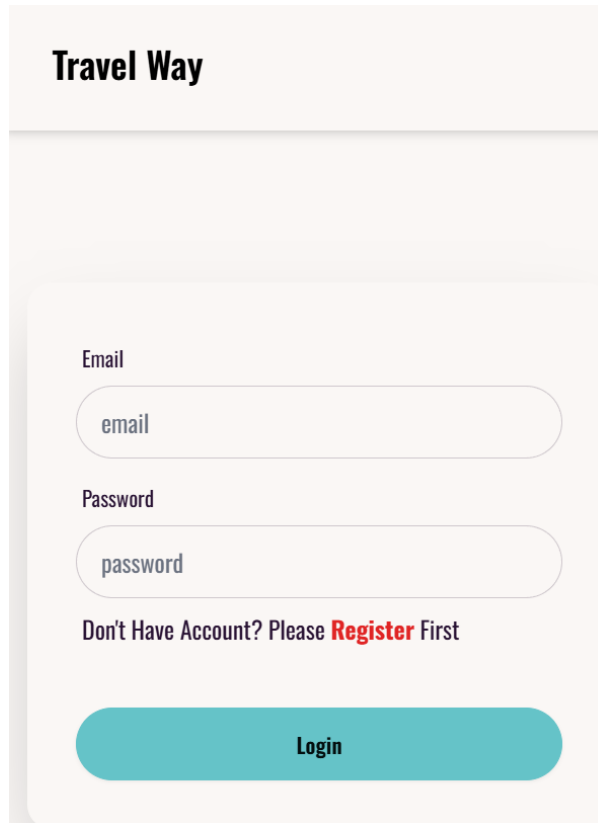
- Evaluate the clarity and organization of the user interface.

Expected Result:

- The interface should be intuitive, user-friendly, and consistent.

Chapter 6: User Manual

6.1 Login Page :



The image shows a mobile application login screen for 'Travel Way'. The page has a light beige background. At the top, the text 'Travel Way' is displayed in a bold, black font. Below this, there is a white rounded rectangle containing the login form. The form consists of two input fields: 'Email' with the placeholder text 'email' and 'Password' with the placeholder text 'password'. Below the password field, there is a link that says 'Don't Have Account? Please Register First', where 'Register' is in red. At the bottom of the form is a large, teal-colored button with the text 'Login' in white.

6.2 Sign Up Page:


Travel Way

Name

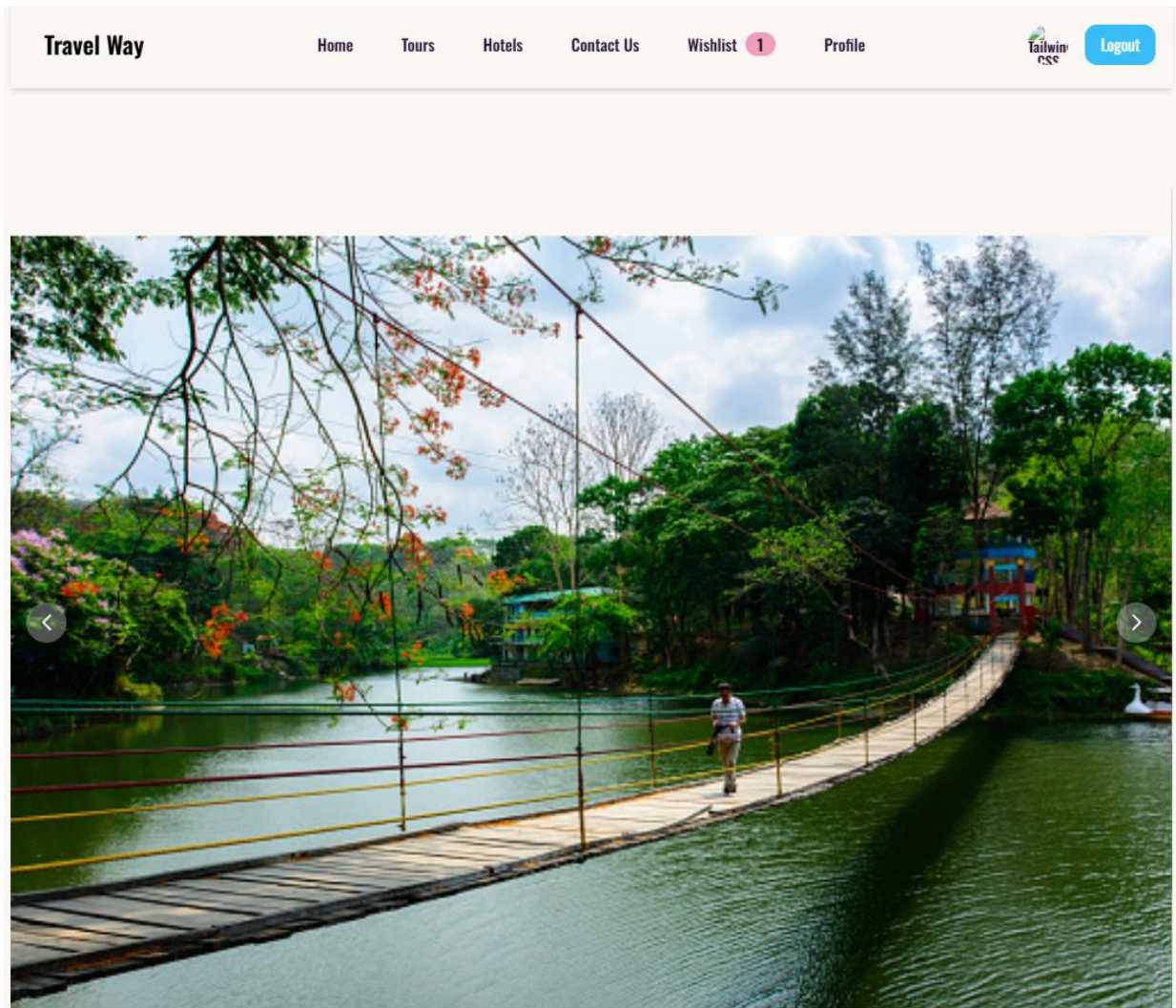
Email

Password


[Register](#)


 [Google](#)

6.3 Home Page:





6.4 View Details:

Travel Way [Home](#) [Tours](#) [Hotels](#) [Contact Us](#) [Wishlist](#) 1 [Profile](#)  [Logout](#)



Bandarban 0.1



Tour Location: Bandarban
Total Days: 3
Ratings: 4.7
Price: 6000 ₳

Tour Details

joss

Hotel Info

Name: Bandarban Inn
Description: vhal na

Bus Info

Name: Shamoli
Contact: 0198324445

Tour Manager

Name: MD.NAEEM
Phone: 0178235546

Tour Guide

Name: Mokbul
Phone: 01897634552

Places to Visit

Mirinja, Boga Lake

What to Carry

odomos, coil, blanket

[Book Now](#)

6.5 Booking system:


The screenshot shows a web application interface for a booking system. At the top, there is a navigation bar with the logo "Travel Way" on the left and several menu items: "Home", "Tours", "Hotels", "Contact Us", "Wishlist" (with a red circle containing the number "1"), and "Profile". On the right side of the navigation bar, there is a "Tailwin CSS" logo and a blue "Logout" button.

The main content area features a large heading "Book Desired Packages" centered at the top. Below this heading is a light blue rectangular box containing a form with the following fields:



- First name:** A text input field with the placeholder text "First name".
- Last name:** A text input field with the placeholder text "Last name".
- Phone:** A text input field with the placeholder text "Phone".
- Email:** A text input field containing the email address "kala@pani.com".
- Price:** A text input field containing the value "6000".

Below the form box, centered, is a yellow button with the text "Book".









6.6 Wishlist Page:

Travel Way Home Tours Hotels Contact Us **Wishlist 1** Profile Tailwin  [Logout](#)

My Wishlist


#	Image	Tour Title	Location	Days	Book	Details	Action
1		SreeMangal	Sylhet	2 Days	Book Now	Details	

6.7 User booking list:

Bookings List								
Tour Cover	Bus Name	Bus Contact	Tour Title	Hotel Name	Days	Tour Manager	Booking Status	Delete
	Huaf	01888234342	Svetlbergol	Hotel Tajmahal	2	MD.NAEEM	Accept	
	Unique	01229493432	Camilla	Hotel Victory	2	MD.NAEEM	Accept	
	Unique	01229493432	Camilla	Hotel Victory	2	MD.NAEEM	Accept	
	Unique	01229493432	Camilla	Hotel Victory	2	MD.NAEEM	Accept	


6.8 Tour guide information:

Our Professional Guides




Tanvir Ahmed
 Email: tanvi11r@gmail.com
 Phone: 017869249111
Speaks: english
Experience: 1 Year

Unavailable ★ 4.5 Part-time



Joy Ahmed
 Email: tanvir.79.ahmed@gmail.com
 Phone: 01786924911
Speaks: english, Bangla
Experience: 5 Year

Unavailable ★ 4.5 Full-time



MD: Masud Ali
 Email: tanvir.79.ahmed@gmail.com
 Phone: 01786924931
Speaks: English, Bangla, hindi
Experience: 2 Year

Available ★ 4.8 Full-time

6.9 User manage :

Manage Users					
#	Name	Email	Role	Create a Role	Delete
1	kala	kala4@pani.com	User	Create Admin	
2	kala	kala5@pani.com	User	Create Admin	
3	ridoy	ridoy@gmail.com	User	Create Admin	
4	ridoy	ridoy@gmail.com	User	Create Admin	
5	Tanvir Ahmed	tanvir@gmail.com	User	Create Admin	
6	Tanvir Ahmed	tanvir123@gmail.com	Admin	Create User	
7	Tanvir Ahmed	tanvir12@gmail.com	Admin	Create User	
8	Tanvir Ahmed	tanvir.79.ahmed@gmail.com	User	Create Admin	

6.10 Guide manage :

Manage All Guides									
#	Photo	Guide Name	Phone	Email	Status	Languages	Type	Update	Delete
1		Tanvir Ahmed	017869249111	tanvir11@gmail.com	Unavailable	english	Part-time		
2		Joy Ahmed	01786924911	tanvir.79.ahmed@gmail.com	Unavailable	english,Bangla	Full-time		
3		MD: Masud Ali	01786924931	tanvir.79.ahmed@gmail.com	Available	English, Bangla ,hindi	Full-time		

Chapter 7: Conclusion

7.1 Project Link

A complete digital solution for handling reservations, tours, and user interactions is offered by this tour booking system. The platform provides necessary features like user registration, tour management, booking processing, and admin controls, despite present drawbacks like SMS alerts and email verification. Future improvements like mobile accessibility, multilingual support, and payment integrations will put the system in a strong position to grow into a competitive and user-friendly platform for both travellers and tour operators.

<https://github.com/mrnaem34/Travel-ways-client.git>

<https://github.com/mrnaem34/Travel-ways-server.git>

7.2 Limitations

- SMS alert facility isn't available.
- Email verification is not available.
- Limited Payment Options.
- No Offline Functionality.

7.3 Future Scope:

Many things may be made easily understandable and simple. To improve functionality, several of the codings can be increased. The database cannot be backed up at this time. We will be able to integrate the backup feature in the upcoming development, which will let users take action based on previous data. The system can be modified to better suit the desired environment and new technologies as they become available. Depending on the possible security risks, emerging technologies can improve security.

Reference:

1. <https://www.deshghuri.com/>
2. <https://www.tourcms.com/>
3. https://www.getyourguide.com/en-gb/?visitor-id=7EH7BDTP3VMX9UY4SSLVC44CKVPE9TP0&locale_autoredirect_optout=true
4. <https://gozayaan.com/>