**Travel and Tour Management System**

Prepared By ©

Shrabon Das

ID: 2230827

Web Application & Internet

Supervised By

Redwanullah Yousuf

Adjunct Lecturer

Department of Computer Science

A logo on a black background

Description automatically generated

**Independent University Bangladesh**

**ABSTRACT**

Travel and Tour Management System is designed to simplify the process of planning and managing trips. The system allows users to search for destinations, book services, and manage travel arrangements seamlessly. Key features include integration with booking systems for hotels and flights, itinerary generation, and the ability to book local tour guides. The platform aims to enhance user experience and operational efficiency for both travelers and administrators.

**ACKNOWLEDGMENTS**

We extend our gratitude to Redwanullah Yousuf , Adjunct Lecturer, Department of Computer Science ,for their guidance and constant supervision. We also thank Sadia Binte Alam, for their motivation and support throughout this project. Finally, we thank our families, friends, and institution members for their cooperation and encouragement.

**……………………………………**

**Shrabon Das**

ID: 2230827

Department of Computer Science & Engineering

Independent University Bangladesh

Certified by:

**……………………………**

Redwanullah Yousuf   
Adjunct Lecturer

Department of Computer Science

### DECLARATION

We hereby declare that we have taken this project under the supervision of Redwanullah Yousuf, Department of Computer Science. We also declare that this report or any part of it has not been submitted elsewhere for the award of any degree. All sources used have been duly acknowledged, and the design and development are our personal effort.

## Table of Content

Title Page

Declaration..............................................................................................(i)

Acknowledgement...................................................................................(ii)

Abstract................................................................................................(iv)

1. Introduction........................................................................................1

1.1 Overview.......................................................................................1

1.2 Purpose........................................................................................1

1.3 Beneficiaries and Benefits................................................................1

1.4 Objectives.....................................................................................1

2. Background Study...............................................................................2

2.1 Reasons for TTMS..........................................................................2

2.2 Features.......................................................................................2

3. Inception..........................................................................................3

3.1 Overview......................................................................................3

3.2 Thinking.......................................................................................3

3.3 Stakeholders..................................................................................3

3.4 Gathering Requirement.....................................................................3

3.4.1 Techniques..............................................................................4

3.5 SRS Assumption Dependencies...........................................................4

3.6 Functional Requirements...................................................................4

3.7 Non-Functional Requirements...........................................................5

3.8 Software Requirement Prioritizing......................................................6

4. System Design.................................................................................9

4.1 System Design Before developing a system..........................................9

4.2 Use Case Diagrams and Scenario......................................................9

5. Development and installation tool..........................................................16

5.1 Development tools.........................................................................16

5.2 Installation process........................................................................16

6. Software Testing...............................................................................17

6.1 Analysis about testing....................................................................17

6.2 Implement White Box Testing...........................................................17

6.2.1 Finding Some Errors....................................................................17

6.2.2 Solve error for White box.............................................................18

7.overview......................................................................................19

**CHAPTER 1: INTRODUCTION**

**1.1 Overview**

Travel and Tour Management System (TTMS) is a web-based platform designed to facilitate travel arrangements. It enables users to explore destinations, book services, and manage itineraries efficiently. This system reduces manual processes and ensures a seamless travel experience.

**1.2 Purpose**

* Simplify travel planning and booking.
* Provide a centralized platform for managing accommodations, transportation, and guided tours.
* Enable users to discover and explore destinations with ease.

**1.3 Beneficiaries and Benefits**

* **Users:** Simplified travel arrangements and personalized itineraries.
* **Tour Guides:** Direct connection with travelers.
* **Administrators:** Efficient system management and analytics.

**1.4 Objectives**

* Develop a user-friendly and efficient travel management system.
* Ensure secure and seamless booking processes.
* Provide real-time updates and notifications for users.

**CHAPTER 2: BACKGROUND STUDY**

**2.1Reasons for TTMS**

* Reduce the complexity of travel planning.
* Provide a one-stop platform for all travel needs.
* Enhance user experience through personalized services.

**2.2 Features**

**User:**

1. Destination Exploration
2. Booking Integration (Hotels, Flights, Bus, Train)
3. Tour Guide Scheduling
4. User Feedback and Ratings
5. Personalized Itineraries
6. Print booking invoice

Admin:

1. Add bus,train,flight,hotel
2. Check booking history
3. Send notification to customer
4. Analyse

**CHAPTER 3:**INCEPTION

3.1 Overview

Inception is the beginning step of needs engineering. This step helps to orient and create a primary draft for project design. After collecting the needs of the Travel and Tour Management System (TTMS), the scope and nature of the problem to be solved are identified. The goal of this section is to identify the concurrent wants and conflicting needs among the stakeholders of this project. For this, we have worked on the following steps:

• Thinking About the Topic.  
• Identifying Stakeholders.  
• Gathering Project Requirements.

3.2 Thinking

At the beginning, we needed to think about the topic. Where do we find those requirements? Who are the users of this system? At this step, it may take some time. We have chosen stakeholders based on the nature and quality of the project and its product deliverables.

3.3 Stakeholders

A Stakeholder could be a person or organization that has rights, shares, claims, or interests regarding the system or its properties to meet their needs and expectations. Simply put, stakeholders influence the project, so their opinions must always be taken into consideration. If key stakeholders are overlooked, it may ruin the project and increase costs. Stakeholders provide opportunities and limitations for the system and are the source of requirements.

In this TTMS, there are Four (4) Stakeholder types:

• Users  
• Administrators  
• Tour Guides  
• Transport and Accommodation Providers

3.4 Gathering Requirement

Requirements collection (also known as Requirements Gathering) is the process of maintaining a list of requirements (functional, system, technical, etc.) from various stakeholders (users, administrators, tour guides, vendors, etc.). The process is not as straightforward as just asking stakeholders what they want the system to do, as in many cases, they are unaware of what is feasible or may be limited by their immersion in the current state.

3.4.1 Techniques

• **Interviews**: These are essential for understanding business problems and current perspectives. Diverse stakeholders should be interviewed to avoid biased requirements.

• **Questionnaires**: Probing questions based on initial interviews help uncover hidden requirements and refine design details.

• **User Observation**: Observing users during their daily tasks provides insight into real-world processes and areas for improvement.

• **Workshops**: Collaborative sessions validate initial requirements, generate additional details, and reconcile conflicting views.

• **Brainstorming**: Encourages innovative ideas by considering "what-if" scenarios. Tools like whiteboards and mind-mapping are useful.

• **Role Playing**: Helps understand the system’s workflow from different user perspectives.

• **Use Cases & Scenarios**: Validates functionality in different situations and identifies edge cases.

• **Prototyping**: Stakeholders can visualize potential systems, providing feedback to refine requirements.

3.5 SRS Assumption Dependencies

Requirements analysis is the first phase of large-scale software development projects. It identifies and documents precise system requirements after a feasibility study. At this stage, requirements are end-user focused.

3.6 Functional Requirements

Functional requirements define specific functionality or tasks that the system must accomplish. For our project, essential functional requirements include:

3.6.1 Destination Exploration

|  |  |
| --- | --- |
| FR-1 | Destination Exploration |
| Description: | Users can explore destinations based on state, district, or specific location. |
| Stakeholders: | Users, Administrators |

3.6.2 Booking Integration

|  |  |
| --- | --- |
| FR-2 | Booking Integration |
| Description: | Users can book hotels, flights, buses, and trains through the platform. |
| Stakeholders: | Users, Administrators, Providers |

3.6.3 Tour Guide Scheduling

|  |  |
| --- | --- |
| FR-3 | Tour Guide Scheduling |
| Description: | Users can schedule and book local tour guides based on availability and ratings. |
| Stakeholders: | Users, Tour Guides |

3.6.4 Personalized Itineraries

|  |  |
| --- | --- |
| FR-4 | Personalized Itineraries |
| Description: | Users can generate custom itineraries based on selected destinations and activities. |
| Stakeholders: | Users |

3.6.5 Booking History

|  |  |
| --- | --- |
| FR-5 | Booking History |
| Description: | Administrators can view and analyze booking history. |
| Stakeholders: | Administrators |

3.6.6 Notifications

|  |  |
| --- | --- |
| FR-6 | Notifications |
| Description: | Users receive notifications for booking confirmations, changes, or special offers. |
| Stakeholders: | Users, Administrators |

3.6.7 Adding Transport and Accommodation Options

|  |  |
| --- | --- |
| FR-7 | Adding Options |
| Description: | Administrators can add bus, train, flight, and hotel details to the system. |
| Stakeholders: | :Administrators, Providers |

3.6.8 Printing Invoice

|  |  |
| --- | --- |
| FR-8 | Printing Invoice |
| Description: | Users can print booking details and invoices. |
| Stakeholders: | Users |

3.7 Performance Requirements

3.7.1 Speed and Latency Requirement

|  |  |
| --- | --- |
| SLR-1 | The system must provide fast responses to user actions. |
| Description: | Browsing and booking should be seamless, depending on the user’s internet connection. |
| Stakeholders: | Users, Administrators |

3.7.2 Supportability Requirements

|  |  |
| --- | --- |
| SRS-1 | The system must have support mechanisms to assist operators during technical issues |

3.8 Non-Functional Requirements

• **Dependability**: The system must ensure 99% uptime for users.

• **Maintainability**: User-friendly interface with secure login for easy maintenance.

• **Security**: Robust mechanisms to prevent unauthorized access and protect sensitive data.

• **Reliability**: Content is sourced from verified providers to ensure quality.

• **Usability**: Intuitive and step-by-step navigation ensures a smooth user experience.

3.9 Software Requirement Prioritizing

The prioritization of requirements ensures critical functions are developed first. The prioritization levels are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SRS No.** | **Requirements** | **Functional/Non-Functional** | **Priority** | **Description** |
| 01 | Registration | Functional | H | Users register with valid details. |
| 02 | Login | Functional | H | Users log in using credentials |
| 03 | Add Transport & Hotels | Functional | H | Admins can add transport and accommodation. |
| 04 | Booking Integration | Functional | H | Book hotels, flights, buses, and trains. |
| 05 | Tour Guide Scheduling | Functional | M | Users book guides based on availability. |
| 06 | Notifications | Functional | M | Notifications for users. |
| 07 | Printing Invoice | Functional | M | Users can print booking invoices. |
| 08 | Security | Non-Functional | H | Ensure secure login and payment mechanisms. |

**CHAPTER 4:** SYSTEM DESIGN

* 1. System Design Before developing a system:

we have to design our system to understand its use cases. Data Flow Diagram (DFD) provides a view of how the system or business flows, helping to increase efficiency and effectiveness to achieve system objectives. For end-users, we have a Use Case Diagram to ensure they can easily understand the system. We will design it with state transition diagrams showing the entities that interact with the system, and lastly, Gantt charts illustrating the start and finish dates of various project elements.

4.2 Use Case Diagrams and Scenario

UML Use Case Diagrams can be used to describe the functionality of a system in a horizontal way. Rather than representing the details of individual features, UCDs show all of the system's available functionalities. It is important to note that they provide a high-level understanding of system interactions.

|  |  |
| --- | --- |
| **Use Case Scenario** | |
| **Use Case Name** | **Actor** |
| Destination Exploration | User |
| Booking Services | User, Administrator |
| View Booking History | Administrator |
| Add Transport Options | Administrator |
| Add Accommodation Options | Administrator |
| Schedule Tour Guide | User, Tour Guide |
| Sign In | User, Administrator |
| Sign Out | User, Administrator |
| View Notifications | User, Administrator |
| Print Booking Invoice | User |
| Change Password | User, Administrator |
| Send Notifications to Customers | Administrator |
| Analyze Bookings | Administrator |

Table 3: Use Case Scenario

A diagram of a program

Description automatically generatedUse Case Diagram:

Pic:uae case diagram

Flow Chart:

A diagram of a software company

Description automatically generated

Pic: flowchart

A diagram of a computer

Description automatically generatedERD:

Pic: ERD

Sequence diagram:

A white sheet of paper with yellow labels

Description automatically generatedA diagram of a diagram

Description automatically generated with medium confidence

Pic:Sequence diagram

Pic:activity diagram

Chapter 5: development and instrallation tool

Devlopmrnt tools:

Application Server:

* Operating System: Windows,Linux,Mac
* Language: php
* Framwork: Laravel
* Fontend:Html,Tailwind,js
* Database:Mysql

Instrllation process:

* 1. Downlodw journyly :<https://github.com/shrabondas5544/Journyly>
  2. Opening Journyly folder in Your Code editor
  3. Open rename env example to env .
  4. Open terminaland type: “composer install”
  5. Then, “composer dump-autoload”
  6. Run “composer update”
  7. Create database “Journyly” on your database
  8. Then in terminal run “php arisan migrate”
  9. Run “php artisan key:generate”
  10. Run “npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p”

11. npm run dev

**CHAPTER 6:** SOFTWARE TESTING

6.1 Analysis about testing:

Testing can be defined as a process of analyzing a software item to find the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item. Testing is crucial to ensure the quality and reliability of the Tour and Travel Management System. Various types and methods of testing are considered. For our system, we have decided to perform two primary testing methods: ‘Black Box Testing’ and ‘White Box Testing.’

6.2 Implement White Box Testing:

Contrary to Black Box testing, White Box testing (also called Glass Box testing) involves examining the structure and flow of the software, where the internal workings of the system are visible to the tester. Testing plans are created based on the details of the software implementation, such as programming logic and structure. Test cases are derived from the program's structure to ensure every line of code is executed at least once (statement coverage) and every branch is traversed (branch coverage).

White Box testing is essential for verifying logical paths, loops, and error-handling mechanisms. Below are some techniques applied:

• Statement Coverage: Execute every statement in the code. • Branch Coverage: Test every branch within control structures. • Path Coverage: Test all possible paths in the program logic.

6.2.1 Finding Some Errors:

During White Box testing of the Tour and Travel Management System, we identified several issues:

a. Incomplete functionality for hotel registration and management.

b. Missing train booking features.

c. Errors in uploading operator logos (e.g., bus and airline logos).

6.2.2 Solve error for White box:

These issues were addressed by reviewing the implementation logic, fixing missing endpoints, and validating data flow across modules. Continuous testing will ensure that the resolved errors do not reoccur

**CHAPTER 7:** CONCLUSION

7.1 Conclusion

The global travel and tourism industry is evolving rapidly with the increasing adoption of online platforms. This project aims to simplify travel planning and booking by integrating essential services such as accommodation, transportation, and tour guides into one platform. The Tour and Travel Management System provides users with seamless functionality for destination exploration, booking, and itinerary management. This system enhances communication between users and service providers, saving time and effort while ensuring a better travel experience.

7.2 Further Suggested Work

In the future, the system can incorporate advanced features to improve user experience and operational efficiency:

• Integration of automated payment gateways for secure and faster transactions.

• AI-driven recommendation systems to provide personalized itineraries based on user preferences and past behavior.

• Implementation of data mining techniques to analyze booking trends and suggest popular destinations.

• Integration of real-time transportation tracking to enhance reliability and convenience for users. • Virtual Reality (VR) features to provide virtual previews of destinations.

These enhancements will make the system more robust, user-friendly, and adaptable to future technological trends.

A login screen shot

Description automatically generated

User login page :

A screenshot of a login form

Description automatically generated

Signup page :

User Dashboard:

A screenshot of a computer

Description automatically generated

Profile page:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

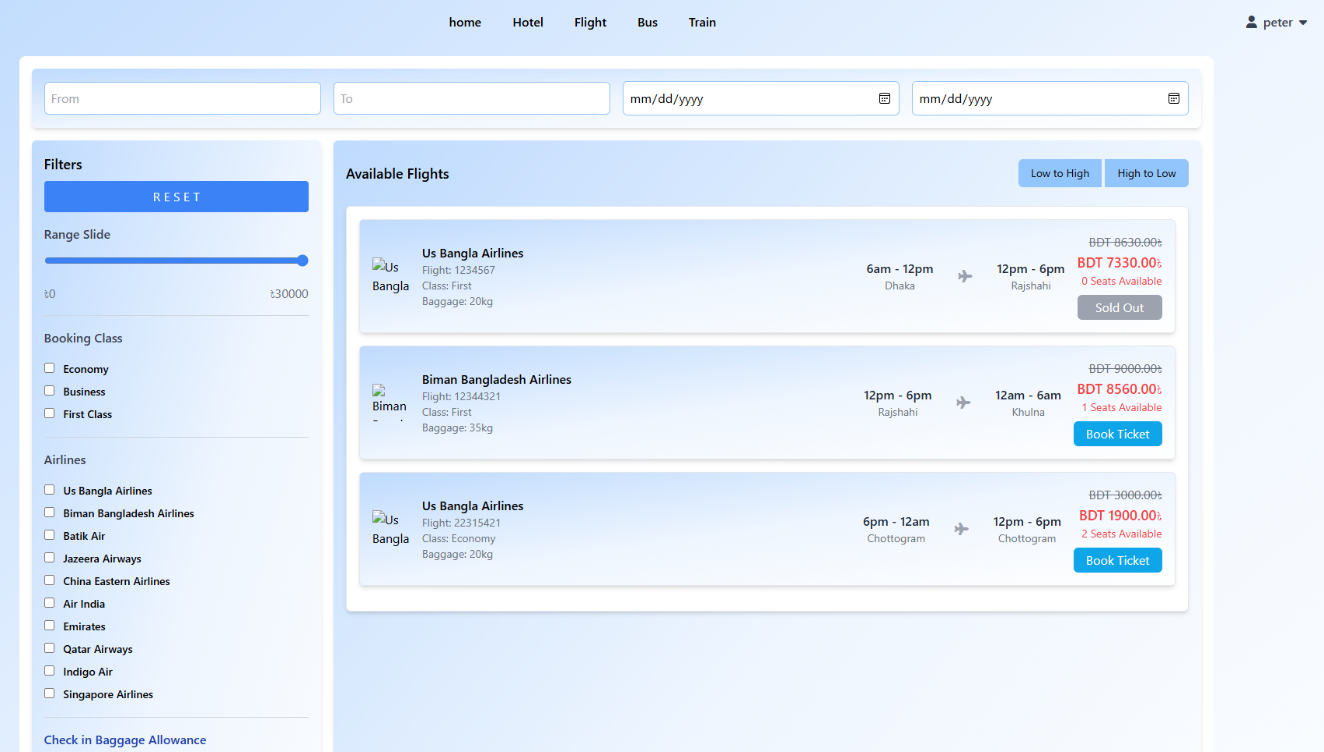
Description automatically generated

Inbox:

A screen shot of a computer

Description automatically generated

Flight history :



Flight search:

A screenshot of a computer

Description automatically generated

Payment page:

A screenshot of a map

Description automatically generated

A screenshot of a computer

Description automatically generated

Admin dashboard:

Feedback section:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Feedback check:

Flight list:

A screenshot of a flight registration

Description automatically generated

Add flight:

Flight history:

A screenshot of a computer

Description automatically generated

Show statistics:

A screenshot of a computer

Description automatically generated