
**Department of Computer Science and Engineering**
Premier University

CSE306: Software Engineering & Information System Design Laboratory

# Software Design Document

## Odyssey Travel Agency Software

**Submitted by**



| Name | ID |
|---|---|
| Mohammad Hafizur Rahman Sakib | 0222210005101118 |
| Arnab Shikder | 0222210005101098 |
| Mohammad Ohidul Alam | 0222210005101123 |
| Sayed Hossain | 0222210005101102 |
| Mohammad Asmual Hoque Yousha | 0222210005101121 |



**Submitted to :**
Jannatul Maowa Hasi
Lecturer,Department of CSE
Premier University
Chittagong

**Remarks**

# Contents

## Index of Diagrams

# 1 Introduction

The purpose of this document is to provide a comprehensive description of the webbased project named "Odyssey Travels," developed using Next.js. It aims to outline the system's objectives, functionalities, user interfaces, operational constraints, and how it handles external interactions. This document serves as a detailed guide for stakeholders and developers involved in the project, ensuring a clear understanding of the system's scope and requirements. It will define how "Odyssey Travels" enhances the online travel booking experience through innovative features and responsive web interfaces. This web-based application, "Odyssey Travels," is designed to enhance user efficiency and streamline travel planning processes. It aims to empower users by providing intuitive tools to manage and prioritize travel itineraries and bookings seamlessly. Users will no longer need to rely on traditional methods like spreadsheets or multiple websites to organize their travel plans. "Odyssey Travels" will enable users to categorize and prioritize travel activities effortlessly, ensuring optimal utilization of their time and resources. By offering clear insights into itinerary management and suggesting efficient scheduling of activities, the application enhances user productivity while maintaining ease of use. Specifically, the system will guide users in making informed decisions about their travel plans, suggesting ideal times for activities to maximize efficiency throughout their journey. It will help users save valuable time by centralizing booking processes and providing real-time updates on travel arrangements. Additionally, the application will assist users in optimizing their itineraries by recommending adjustments or identifying unnecessary tasks, ensuring that their travel experiences are both productive and fulfilling. Overall, "Odyssey Travels" aims to revolutionize the travel planning experience by offering a user-friendly interface that simplifies task management and enhances productivity, thereby meeting the diverse needs of modern travelers.

# 2 System Architecture

The system architecture of the Odyssey Travel website is designed to provide a robust and scalable platform for managing travel bookings and related services. It comprises several key components:

## 2.1 System Architecture Diagram

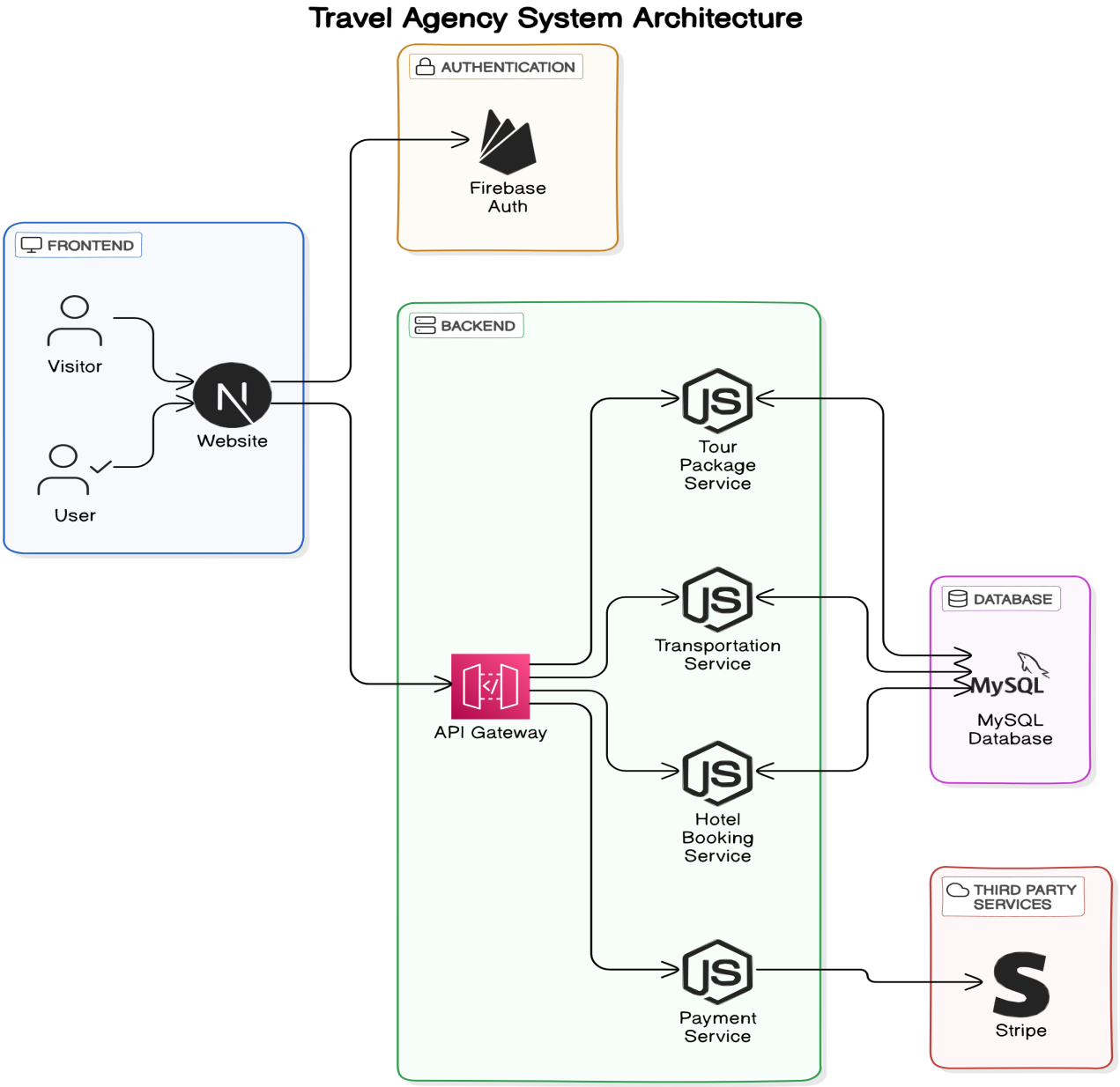


**Figure - 2.1 : System Architecture Diagram For Travel Agency Software**

## 2.2 System Architecture Overview

The system architecture of **Odyssey Travels** is designed to efficiently manage travel planning and booking processes. It is structured around a web-based software application that interacts with various user types—visitors, registered users, and administrators.

### 2.2.1 Client-Side (Frontend)

The user interface is built using **Next.js**, a React framework that supports server-side rendering for fast and efficient page loads. The application is optimized for both desktop and mobile devices, ensuring a responsive and user-friendly experience. Key features include browsing travel packages, booking accommodations, and managing user profiles. The interface components are designed to be intuitive, allowing users to navigate the site easily.

### 2.2.2 Server-Side (Backend)

The backend handles all data processing and business logic. It uses RESTful APIs to manage communication between the client-side and server-side components. The server processes user requests, interacts with databases to retrieve or store information, and ensures that all transactions are securely handled.

### 2.2.3 Database

The application utilizes a relational database to store user information, booking details, travel packages, and other related data. The database is designed for quick access and secure data storage, ensuring that user information is protected and easily retrievable.

### 2.2.4 Integration with External Services

Odyssey Travels integrates with external services such as payment gateways (e.g., PayPal, Stripe) for secure transactions and mapping services (e.g., Google Maps) to provide location-based functionalities. Communication interfaces like SMTP or Email APIs are used to manage booking confirmations and notifications, ensuring users are kept informed throughout their travel planning process.

### 2.2.5 Admin Dashboard

Administrators have access to a dedicated dashboard where they can manage travel packages, oversee user bookings, and perform system maintenance. This component allows for efficient system management and ensures that the application remains up-to-date and functional.

### 2.2.6 Security and Reliability

The architecture includes robust security measures such as HTTPS for encrypted data transmission, secure authentication protocols, and regular security audits to protect user data and maintain system integrity. The system is designed to be highly reliable, with mechanisms in place to handle errors gracefully and ensure continuous operation even in the event of unexpected issues.

# 3 Data Flow Diagram

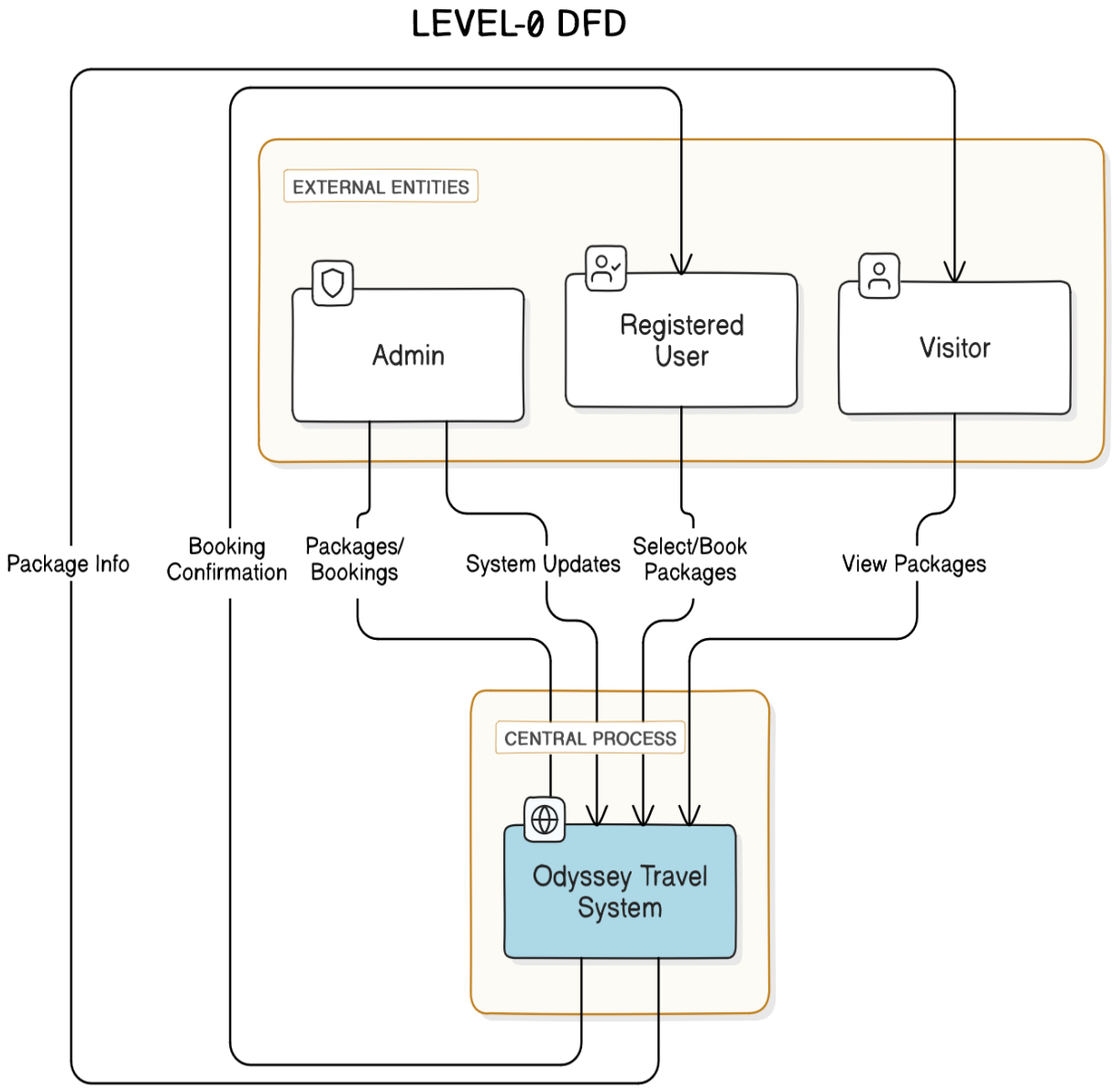


**Figure - 3.1 : Level 0 Data Flow Diagram**

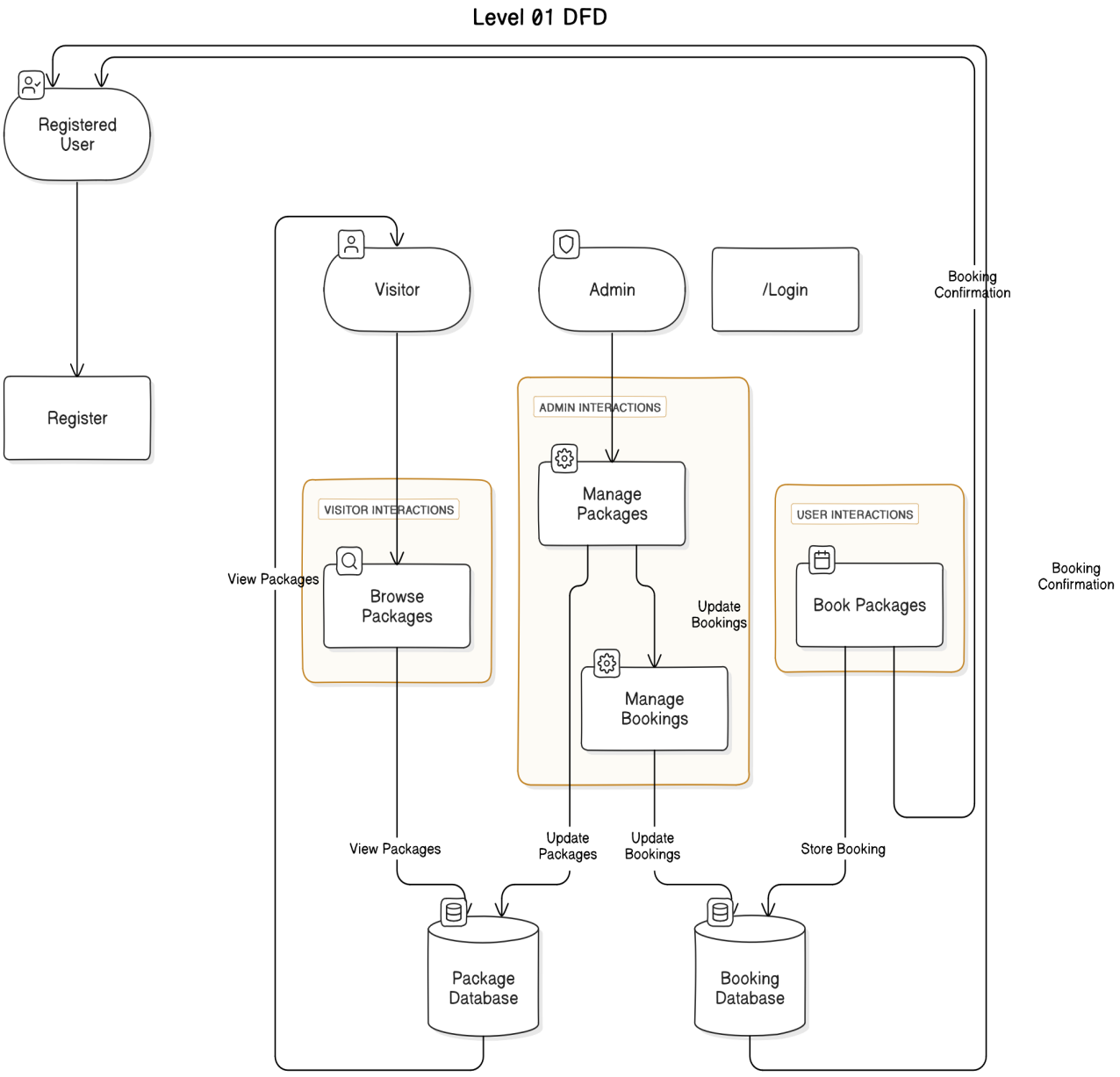


**Figure - 3.1 : Level 1 Data Flow Diagram**

# 4 Test Case Design

Table 4.1: Test Case for User Registration (TC001)

| **Test Case ID:** | TC001 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful user registration | | |
| **Test Case:** | Register a new user on the platform | | |
| **Pre-Condition:** | The user is on the registration page and has not registered before. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Navigate to the registration page. | URL: /register | The registration page loads successfully. | The user is presented with a registration form. |
| 2. Enter a valid username. | Username: user123 | The username is accepted by the system. | The system accepts the username. |
| 3. Enter a valid email. | Email: user@example.com | The email is accepted by the system. | The system accepts the email address. |
| 4. Enter a secure password. | Password: Password123! | The password is accepted by the system. | The system accepts the password. |
| 5. Click "Register". | N/A | The user is successfully registered and redirected to the login page with a confirmation message. | User account is created, and the system is updated with the new user details. |

Table 4.2: Test Case for User Login (TC002)

| **Test Case ID:** | TC002 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful user login | | |
| **Test Case:** | Log in with valid credentials | | |
| **Pre-Condition:** | The user must be registered and on the login page. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Navigate to the login page. | URL: /login | The login page loads successfully. | The user is presented with a login form. |
| 2. Enter the registered email. | Email: user@example.com | The email is recognized by the system. | The system accepts the email address. |
| 3. Enter the correct password. | Password: Password123! | The password is recognized by the system. | The system accepts the password. |
| 4. Click "Login". | N/A | User successfully logs in and is redirected to the home page. | User session is initiated, and the user is logged into the system. |

Table 4.3: Test Case for Viewing Travel Packages by Visitors (TC003)

| **Test Case ID:** | TC003 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful travel package viewing by visitors | | |
| **Test Case:** | View travel packages as a visitor | | |
| **Pre-Condition:** | User is not logged in and on the website's home page. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |

| | | | |
|---|---|---|---|
| 1. Navigate to the home page. | URL: /home | The home page loads successfully with the "Packages" menu option visible. | The visitor can see the available travel packages. |
| 2. Click on the "Packages" menu option. | N/A | Visitor can view a list of available travel packages, including destination, duration, and price details. | System displays travel packages without requiring a user login. |

Table 4.4: Test Case for Booking a Travel Package (TC004)

| **Test Case ID:** | TC004 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful booking of a travel package | | |
| **Test Case:** | Book a travel package as a logged-in user | | |
| **Pre-Condition:** | The user must be logged in and have access to the list of travel packages. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Log in to the platform. | Email: user@example.com Password: Password123! | The user successfully logs in and accesses their account. | The user session is initiated. |
| 2. Navigate to "Packages". | URL: /packages | The list of travel packages is displayed. | The user can view available travel packages. |
| 3. Select a travel package. | Travel Package: Package1 | The selected package details are displayed, including booking options. | The user can proceed with the booking. |
| 4. Click "Book Now". | N/A | The booking form is displayed with options to select dates and confirm the booking. | The user is ready to enter booking details. |
| 5. Enter booking details. | Travel Dates: 2024-10-01 to 2024-10-07 | The booking details are accepted by the system. | The user is ready to confirm the booking. |

| | | | |
|---|---|---|---|
| 6. Confirm the booking. | N/A | Booking is successfully processed, and the user receives a confirmation with booking details. | The booking is recorded in the system, and the booking confirmation is sent to the user. |

Table 4.5: Test Case for Canceling a Travel Package Booking (TC005)

| **Test Case ID:** | TC005 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful cancellation of a booking | | |
| **Test Case:** | Cancel a booked travel package | | |
| **Pre-Condition:** | The user has an active booking and is logged in. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Log in to the platform. | Email: user@example.com Password: Password123! | The user successfully logs in and accesses their account. | The user session is initiated. |
| 2. Navigate to "My Bookings". | URL: /my-bookings | The user's active bookings are displayed. | The user can view their current bookings. |
| 3. Select a booking. | Booking ID: BK12345 | The selected booking details are displayed with options to cancel the booking. | The user can proceed with cancellation. |
| 4. Click "Cancel Booking". | N/A | The system prompts the user to confirm the cancellation. | The user can confirm or decline the cancellation request. |
| 5. Confirm cancellation. | N/A | The booking is successfully canceled, and a confirmation is sent to the user. | The booking is removed from the user's active bookings, and the cancellation is recorded in the system. |

Table 4.6: Test Case for Adding a New Travel Package by Admin (TC006)

| **Test Case ID:** | TC006 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful addition of a new travel package by admin | | |
| **Test Case:** | Admin adds a new travel package to the system | | |
| **Pre-Condition:** | Admin is logged into the system and has the required privileges. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Log in as admin. | Admin credentials | Admin successfully logs into the system. | Admin session is initiated. |
| 2. Navigate to the "Add Package" section. | URL: /admin/add-package | The "Add Package" form is displayed. | Admin can add a new package. |
| 3. Enter package details. | Package Name: Tropical Adventure<br>Price: $2000<br>Duration: 7 days | The package details are accepted by the system. | The system is ready to save the new package. |
| 4. Click "Save". | N/A | The package is successfully added and appears in the available travel packages list. | The new package is saved in the system. |

Table 4.7: Test Case for Modifying a Travel Package by Admin (TC007)

| **Test Case ID:** | TC007 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful modification of an existing travel package by admin | | |
| **Test Case:** | Admin modifies a travel package details | | |
| **Pre-Condition:** | Admin is logged into the system and has the required privileges. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Log in as admin. | Admin credentials | Admin successfully logs into the system. | Admin session is initiated. |

| | | | |
|---|---|---|---|
| 2. Navigate to the "Packages" section. | URL: /admin/packages | The list of available packages is displayed. | Admin can select a package to modify. |
| 3. Select a package to modify. | Package Name: Tropical Adventure | The selected package details are displayed for modification. | Admin can make changes to the package. |
| 4. Update package details. | New Price: $1800 | The updated package details are accepted by the system. | The system is ready to save the modified package. |
| 5. Click "Save". | N/A | The package is successfully updated in the system and reflects the modified details. | The changes are saved and reflected in the available travel packages. |

Table 4.8: Test Case for Deleting a Travel Package by Admin (TC008)

| **Test Case ID:** | TC008 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful deletion of a travel package by admin | | |
| **Test Case:** | Admin deletes a travel package from the system | | |
| **Pre-Condition:** | Admin is logged into the system and has the required privileges. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Log in as admin. | Admin credentials | Admin successfully logs into the system. | Admin session is initiated. |
| 2. Navigate to the "Packages" section. | URL: /admin/packages | The list of available packages is displayed. | Admin can select a package to delete. |
| 3. Select a package to delete. | Package Name: Tropical Adventure | The system prompts the admin for confirmation before deleting the package. | Admin can proceed with deletion. |

| | | | |
|---|---|---|---|
| 4. Confirm deletion. | N/A | The package is successfully deleted from the system. | The package is removed from the list of available travel packages. |

Table 4.9: Test Case for Viewing Booking History by User (TC009)

| **Test Case ID:** | TC009 | | |
|---|---|---|---|
| **Test Scenario:** | Verify successful viewing of booking history by the user | | |
| **Test Case:** | View booking history of a logged-in user | | |
| **Pre-Condition:** | User is logged in and has previous bookings. | | |
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Log in to the platform. | Email: user@example.com Password: Password123! | User successfully logs in and accesses their account. | User session is initiated. |
| 2. Navigate to "My Bookings". | URL: /my-bookings | The user's booking history is displayed. | The user can view the list of past and current bookings. |
| 3. Select a booking. | Booking ID: BK12345 | The selected booking details are displayed, including travel dates and payment information. | User can view the complete booking details. |

Table 4.10: Test Case for Searching Travel Packages (TC010)

| **Test Case ID:** | TC010 |
|---|---|
| **Test Scenario:** | Verify successful search of travel packages |
| **Test Case:** | Search for a travel package based on destination and price |

| **Pre-Condition:** | User is on the home page or package listing page. | | |
|---|---|---|---|
| **Test Steps** | **Test Data** | **Expected Result** | **Post Condition** |
| 1. Enter search criteria in the search bar. | Destination: Maldives<br>Price Range: $1000-$3000 | The system filters and displays packages matching the criteria. | User can view the filtered package list. |
| 2. Select a package from the search results. | Travel Package: Maldives Luxury | The selected package details are displayed. | User can proceed with the booking or view more details. |

# 5 Website Snapshots

This chapter provides snapshots of the travel agency website, showcasing key functionalities such as viewing travel packages and user interface features.

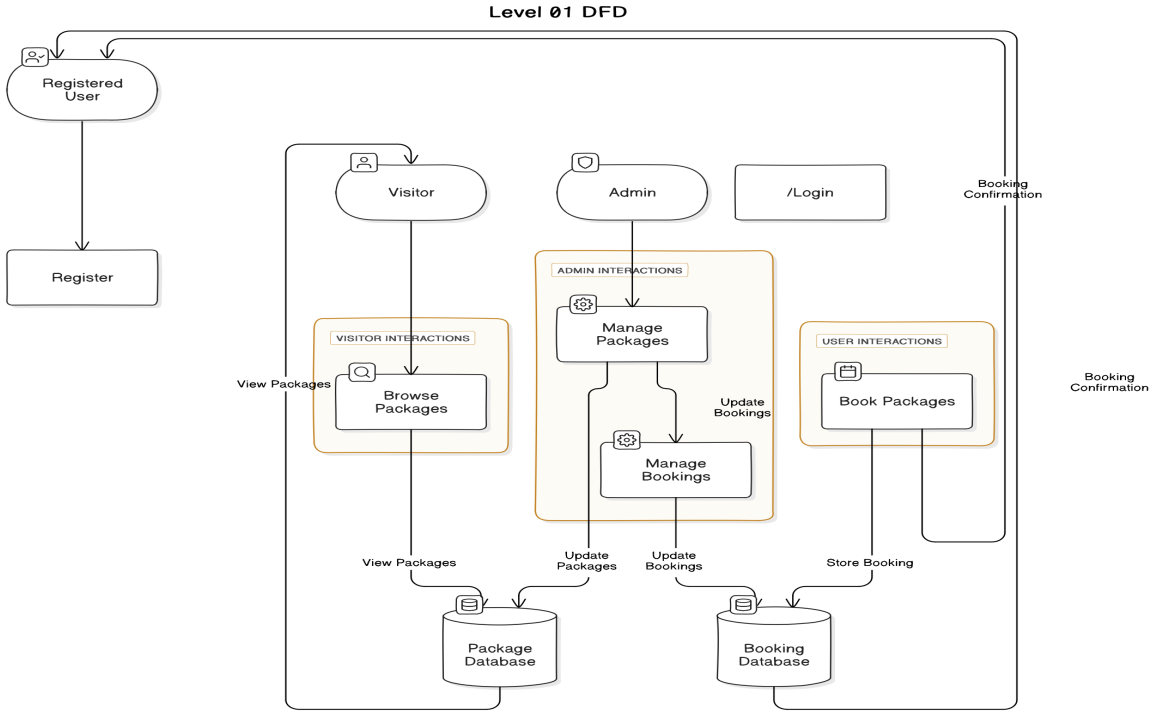


Figure 5.1: Homepage displaying available travel packages

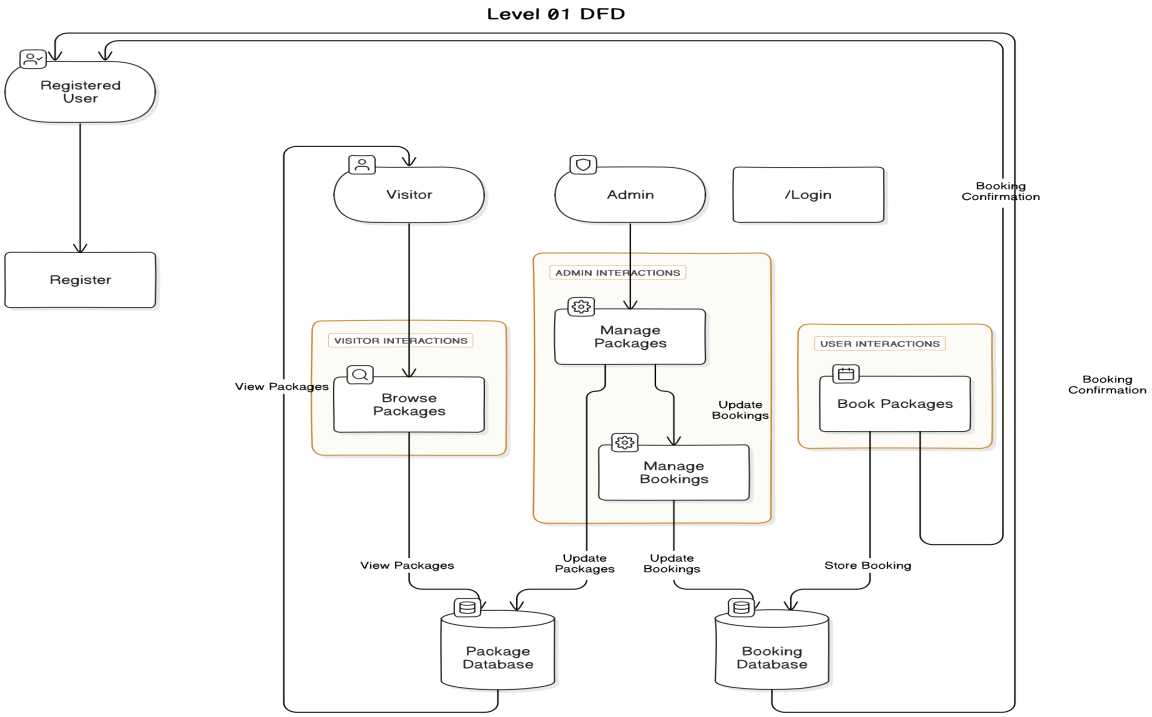


Figure 5.2: Travel package details page

# 6 Future Work

In the future, the travel agency software can be improved with several key features to enhance user experience and system functionality. Below are some potential areas for future work:

- **Recommendation System:** Implement a machine learning-based recommendation engine that suggests travel packages, accommodations, and transportation options based on user preferences and past behavior, improving customer satisfaction through personalized suggestions.
- **Real-time Data Integration:** Integrate third-party APIs to provide real-time data on flight and hotel availability, allowing users to make seamless bookings without delays in information.
- **Review and Rating System:** Develop a user review and rating system where users can share their experiences and feedback, helping future customers make informed decisions based on peer reviews.
- **Security Enhancements:** Strengthen the security of the platform by implementing advanced encryption methods, two-factor authentication, and improved data privacy protocols to safeguard user information and transaction data.
- **Multi-language and Currency Support:** Expand the platform to support multiple languages and currencies, making it accessible and user-friendly for a global audience.
- **Mobile Application Development:** Create a mobile application that mirrors the website's functionality, allowing users to manage bookings, view travel details, and receive notifications on the go, offering greater convenience.