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

1 Introduction **3**

2 System Architecture **4**

2.1 System Architecture Diagram 4

2.2 System Architecture Overview 5

2.2.1 Client-Side (Frontend) 5

2.2.2 Server-Side (Backend) 5

2.2.3 Database 5

2.2.4 Integration with External Services 5

2.2.5 Admin Dashboard 5

2.2.6 Security and Reliability 5

Index of Diagrams

- **Figure-1.1:** Flow Chart for Travel Agency Software 04
- **Figure-2.1:** System Architecture Diagram 06
- **Figure-3.1:** Activity Diagram For Travel Agency Software 10
- **Figure-3.1:** Use Case 01 11
- **Figure-3.2:** Use Case 02 12
- **Figure-3.3:** Use Case 03 13
- **Figure-3.4:** Use Case 04 14
- **Figure-3.5:** Use Case 05 15
- **Figure-3.6:** Use Case 06 17
- **Figure-3.7:** Use Case 07 18
- **Figure-3.8:** Use Case 08 19
- **Figure-3.3.1:** Use Case Diagram For Travel Agency Software 20
- **Figure-3.4.1:** Sequence Diagram For Travel Agency Software 21
- **Figure-3.6:** Level 0 DFD of Travel Agency Software 22
- **Figure-3.7:** Level 1 DFD of Travel Agency Software 22
- **Figure-3.8:** Entity Relationship(ER) Diagram For Travel Agency Software 23

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

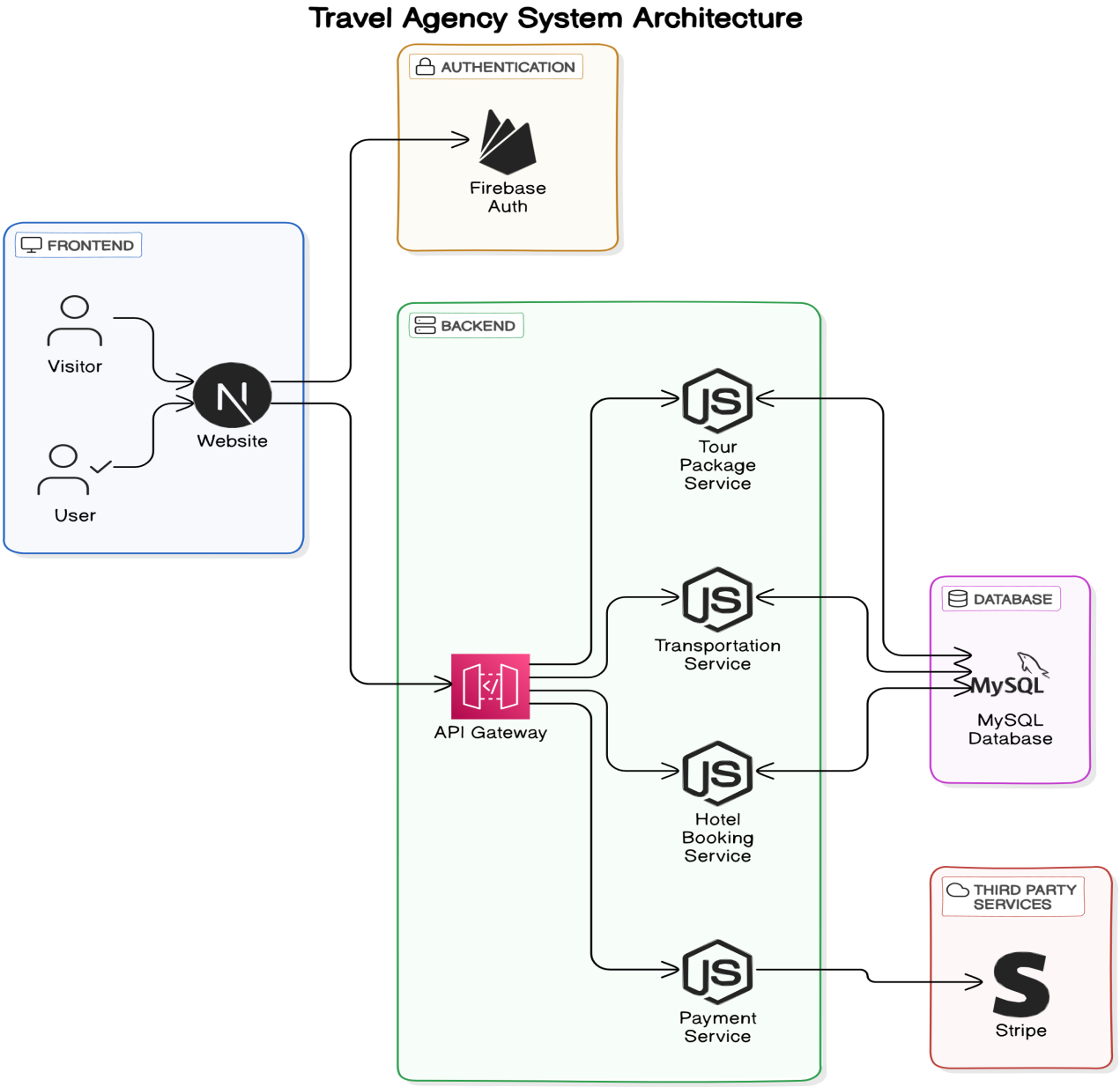

Figure - 2.1 : System Architecture Diagram For Travel Agency Software

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.

Test Case Design

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.

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.

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.

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.

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 is ready to cancel the booking.
4. Click on "Cancel".	N/A	The cancellation confirmation is displayed with a summary of the booking details.	The user is ready to confirm the cancellation.
5. Confirm the cancellation.	N/A	The booking is successfully canceled, and the user receives a confirmation of the cancellation.	The booking is removed from the user's active bookings, and the cancellation is recorded in the system.

Test Case ID:	TC006		
Test Scenario:	Verify that a logged-in user can view their profile details		
Test Case:	View user profile information		
Pre-Condition:	The user is logged in and on the home page.		
Test Steps	**Test Data**	**Expected Result**	**Post Condition**
1. Navigate to the user profile page.	URL: /profile	The profile page loads successfully.	The user is presented with their profile information.
2. View profile details.	N/A	The profile details are displayed correctly, including name, email, and contact information.	The user can see their updated profile details.

Test Case ID:	TC007		
Test Scenario:	Verify that a user receives an email confirmation after booking a travel package		
Test Case:	Check email confirmation for a successful booking		
Pre-Condition:	The user has completed a successful booking and provided a valid email address.		
Test Steps	**Test Data**	**Expected Result**	**Post Condition**
1. Complete the booking process.	Travel Dates: 2024-10-01 to 2024-10-07	Booking is successfully processed.	The booking is recorded, and an email confirmation is triggered.
2. Check the registered email account.	Email: user@example.com	The user receives an email confirmation with booking details.	The user can view the confirmation email containing the booking details.

Test Case ID:	TC008		
Test Scenario:	Verify that a user can update their profile information		
Test Case:	Update user profile details		
Pre-Condition:	The user is logged in and on the profile page.		
Test Steps	**Test Data**	**Expected Result**	**Post Condition**
1. Navigate to the profile update section.	URL: /profile/edit	The profile edit page loads successfully.	The user is presented with editable profile fields.
2. Update profile information.	1. New Name: Jane Doe 2. New Email: jane.doe@example.com	The new profile information is accepted by the system.	The profile details are updated.
3. Save changes.	N/A	The system confirms that the profile information has been updated successfully.	The updated profile information is displayed on the user's profile page.

Test Case ID:	TC009		
Test Scenario:	Verify that a user can search for travel packages using keywords		
Test Case:	Search for travel packages		
Pre-Condition:	The user is logged in and on the travel packages page.		
Test Steps	**Test Data**	**Expected Result**	**Post Condition**
1. Enter a keyword in the search field.	1. Keyword: "beach"	The search query is processed by the system.	The user is shown search results related to the keyword.
2. Click "Search".	N/A	The search results display travel packages related to the keyword, including destination and price.	The user can view and select from the search results.