Tour Management System

Software Design

CSCI-P465/565 (Software Engineering I)

Project Team
Hanna Adams
Ava Greene
Owen Harris
Avishmita Mandal
Aditya Padgal
Yashaswini Sampath Kumar

1. Introduction

1.1 System Description

The EzTravel Tour Management System is designed to streamline tour planning by offering users a comprehensive platform with features such as place searching, destination exploration, accommodation and travel booking, customizable itinerary creation, and collaborative tools for travelers. This system aims to simplify the complex process of tour planning, providing a one-stop solution for users.

1.2 Design Evolution

1.2.1 Design Issues

In addressing design considerations, the team faced several challenges. These challenges encompassed the creation of an intuitive user interface, integration of diverse travel-related features, ensuring robust user authentication, and optimizing the planning process. One significant constraint was determining how to acquire and structure data for the system. Decisions were made regarding real versus sample data usage. Integrating place search, accommodation, and travel bookings seamlessly, and designing an efficient day-wise itinerary planning system, were additional challenges. A secure and user-friendly login/signup process, including features like password recovery and OTP integration, was also crucial.

1.2.2 Candidate Design Solutions

Multiple design solutions were explored, spanning architectural patterns, database structures, and third-party API integrations. Decisions involved system setup, global versus local focus, transportation modes, and user authentication methods. The idea of utilizing real data for accurate cost calculations was considered, requiring a careful balance between authenticity and system performance.

1.2.3 Design Solution Rationale

The team opted for a simple and scalable model using the MERN stack, emphasizing flexibility and efficiency. This architecture facilitates a modular approach, allowing independent evolution of components like place search, accommodation, and bookings.

Our team decided to create a design that allows users to open our system to a search bar where they can type in their desired location. When a location is searched, the system will provide information about the location desired with tabs including flights, hotels, and things to do, allowing users to easily plan their trip. Our system will have an itinerary displayed like a shopping cart to easily allow users to add desired activities. When a user adds something to their itinerary cart, they will either sign up or log into the system to ensure their desired trip is saved. The chosen design prioritizes a clean and intuitive user interface, implementing OTP authentication for secure and convenient user authentication.

1.3 Design Approach

1.3.1 Methods

Object-oriented design principles and design patterns are integral to the design, with prototypes employed for interface testing to ensure a user-friendly experience. The recommendation algorithm incorporates machine learning techniques.

1.3.2 Standards

The design adheres to web development standards, including security protocols for user data protection. The MERN stack follows industry best practices, and user authentication conforms to OAuth standards.

1.3.3 Tools

A diverse set of front-end and back-end technologies are employed, focusing on responsiveness. Version control systems and collaboration tools facilitate efficient team development. Prototyping tools such as Figma assist in visualizing the user interface, while MongoDB, Express, React, and Node.js form the core development stack. Additional tools for interface prototyping and testing contribute to a robust design process.

2. System Architecture

2.1 System Design

The high-level design adopts a user-friendly front-end connecting to a MERN stack-based backend. Major components, including user authentication, search, recommendation, itinerary planning, and bookings, are illustrated in the system diagram. The design ensures seamless interaction between these components to provide a unified and intuitive user experience.

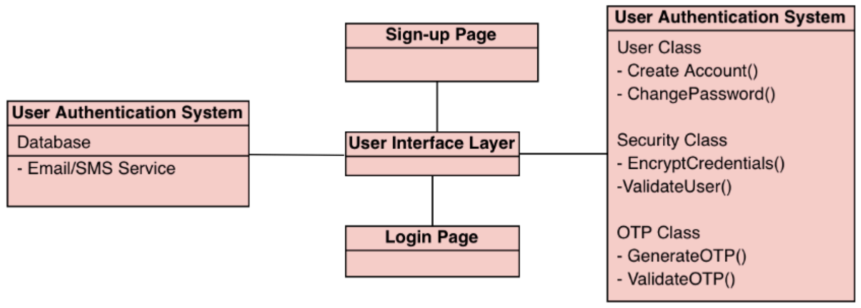
2.2 External Interfaces

External interfaces incorporate third-party APIs for place information, travel booking, and currency conversion. Integration with OAuth for social media login and SMS gateways for password recovery enhances both user experience and security. These interfaces contribute to a robust and feature-rich system architecture.

3. Component Design

Below are several system components and a proposed design of each system component.

Component Name: User Authentication System
Component Description: The User Authentication System is responsible for managing user sign-in, sign-up processes, and maintaining session integrity. It allows users to create new accounts or log into existing ones using either their email/password or through OAuth providers. The system is designed to handle password recovery functions, OTP (One-Time Password) validation, and enforce security measures to protect user credentials.
Responsible Development Team Member: Yashaswini Sampath Kumar & Aditya Padgal
Component Diagram:
The component interfaces with the User Interface layer to receive user input, the Database to store and retrieve user credentials, and the Email/SMS services for OTP and password recovery functionalities. Internally, it contains classes for user management, session handling, and security.



Component User Interface:

- **Sign-up Page:** Users can create a new account by providing required details.
- **Login Page:** Users can sign in using their credentials or via OAuth.
- **Password Recovery Page:** Users can reset their passwords using their email and OTP.

Component Objects:

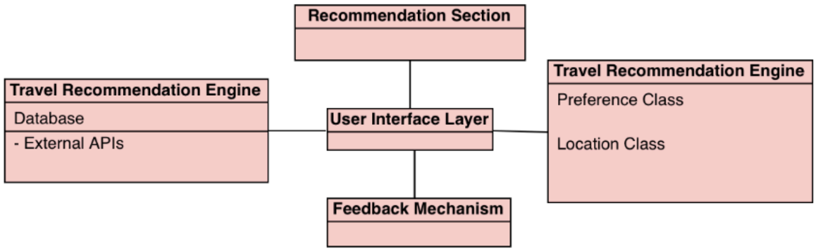
- **User Class:** Contains data members such as email, password, and profile information. It has methods for account creation, password change, etc.
- **Security Class:** Handles encryption and validation of user credentials.
- **OTP Class:** Manages the generation and validation of OTPs sent to users.

Component Interfaces (internal and external): Communicates with the Email/SMS service provider for sending OTP/password recovery emails or messages. Interacts with the database to store and verify user credentials. Uses OAuth for allowing users to sign in via social media accounts.
Component Error Handling:

- **Error Case 1:** Incorrect Credentials: The system prompts the user to retry or reset their password.
- **Error Case 2:** Session Timeout: Automatically logs the user out and requests re-authentication.

Component Name: Travel Recommendation Engine

Component Description: The Travel Recommendation Engine provides personalized destination and activity suggestions to users based on their location, preferences, and ratings. It utilizes a machine learning algorithm to refine recommendations over time.
Responsible Development Team Member: Yashaswini Sampath Kumar & Aditya Padgal
Component Diagram:
Interacts with the User Interface for capturing user preferences, the Database for storing and retrieving preferences and history, and external APIs to fetch real-time data on destinations and activities.

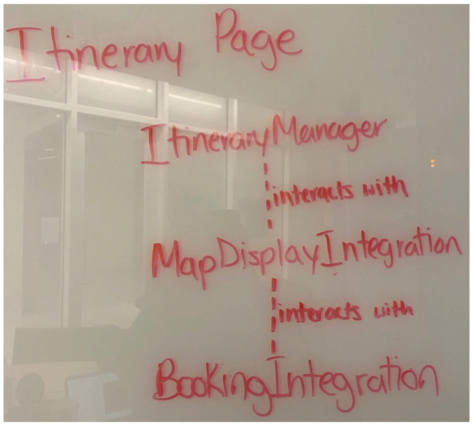


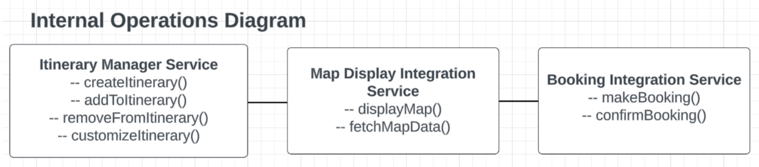
Component User Interface:
- **Recommendation Section:** A part of the homepage where personalized recommendations are displayed.
- **Feedback Mechanism:** Allows users to rate and review destinations and activities.

Component Objects:
- **Preference Class:** Holds user preferences and has methods to update and retrieve them.
- **Location Class:** Tracks user location within the app to inform recommended places that are near to the user.
- **Algorithm Class:** Contains the machine learning model and methods for generating recommendations.

Component Interfaces (internal and external): Uses APIs to fetch up-to-date information on destinations and activities. Interfaces with the User Profile component to obtain user preferences.
Component Error Handling:
- **Error Case 1:** Unavailable Data: Falls back to more general recommendations if specific data is not available.

Component Name: Itinerary Builder
Component Description: The Itinerary Builder allows users to create, customize, and manage travel itineraries. It provides an interface for adding flights, hotels, and activities, and helps users organize their travel schedule efficiently. Allows users to customize day-wise plans, view selected places on a map, and make bookings directly from the itinerary page. Also allows users to delete added places, flights, or activities if itinerary plans change. The user also has the ability to delete entire itineraries. Users can make comments on specific itineraries and share an itinerary with other people.
Responsible Development Team Member: Yashaswini Sampath Kumar & Aditya Padgal
Component Diagram:




Internal Operations Diagram

Itinerary Manager Service
-- createItinerary()
-- addToItinerary()
-- removeFromItinerary()
-- customizeItinerary()

Map Display Integration Service
-- displayMap()
-- fetchMapData()

Booking Integration Service
-- makeBooking()
-- confirmBooking()

This component communicates with the Flights, Hotels, and Activities components for bookings, and with the User Interface for itinerary management and display.

Component User Interface:

- **Itinerary Creation Page:** Where users can start building their itinerary.
- **Itinerary Overview Page:** A summary view of the entire itinerary.

Component Objects:

The Itinerary Page component uses several classes such as ItineraryManager, MapDisplayIntegration, and BookingIntegration. The ItineraryManager class manages user itineraries, including adding, editing, and deleting selected places. The MapDisplayIntegration class integrates with mapping services to display selected places on a map. Lastly, the BookingIntegration class facilitates booking options for accommodations and travel directly from the itinerary.

- **Itinerary Class:** Stores all details of a user's itinerary, with methods to add, remove, or modify items.
- **Calendar Class:** Manages the scheduling aspect of the itinerary.
- **Item Class:** Represents an individual item (flight, hotel, activity) within the itinerary.
- **Map Class:** Displays a visual representation to show where all items (flight, hotel, activity) are located on a map.

Component Interfaces (internal and external): Interacts with the Flights, Hotels, and Activities components for real-time availability and booking. Interfaces with the User Authentication System to save and retrieve user-specific itineraries.

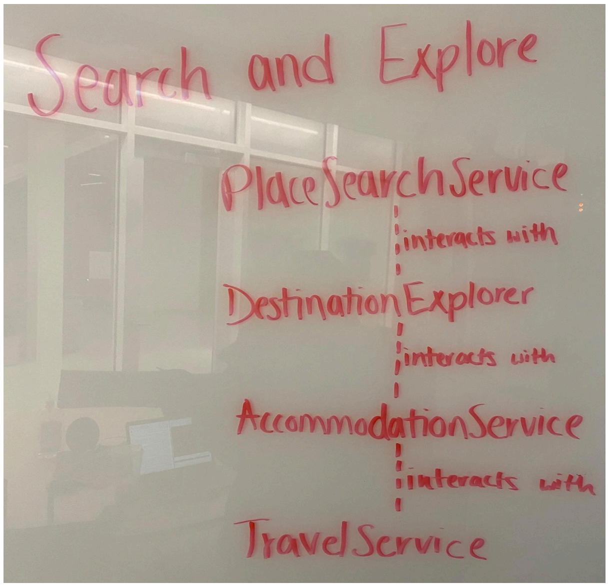
Component Error Handling:

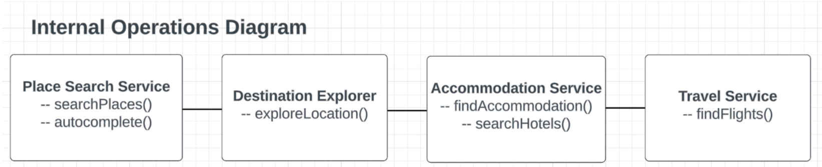
- **Error Case 1:** Scheduling Conflict: Notifies users of any scheduling overlaps and offers alternative solutions.

Component Name: Search and Explore

Component Description: This component handles the search functionality and exploration of places. It includes subcomponents for searching places, exploring destination details, and managing accommodations and travel options. Utilizes external APIs for place information,

images, and travel-related data. Users are able to explore the site without logging in, allowing them to find places to go and things to do without the requirement of creating an account.
Responsible Development Team Member: Ava Greene
Component Diagram:





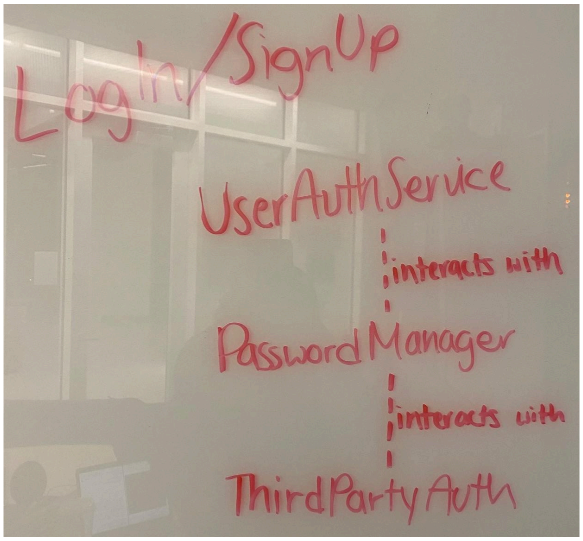
Component User Interface: Allows users to type in their desired destination. When they click the ‘Search’ button, information about the location searched is displayed. If the ‘Search’ button is not clicked, all locations will be displayed. Destination Exploration includes tabs for flights, hotels, restaurants, and attractions. Users can add selected places to their itinerary plan. In their itinerary plan, the user can plan out days using specific times, restaurants, flights, hotels, and activities. The user can edit their itinerary at any time by logging into the app.
Component Objects: The Search and Explore component uses many classes such as PlaceSearchService, DestinationExplorer, AccommodationService, and TravelService. PlaceSearchService manages place search functionality, interacting with external APIs. The DestinationExplorer class retrieves and displays detailed information about selected destinations. AccommodationService handles accommodation search and filtering. Filters for this search will allow users to search only ‘Flight’ data entries if they do not desire ‘Hotel’ or ‘Things to Do’. Lastly, TravelService manages travel options with flights including ‘From’ and ‘To’ and how many tickets are available.
This component searches through data to provide the best recommended searches based on the users preferences. The search bar provided to the user expects a location. The search bar has been updated to allow the user to enter a partial location and the app will complete the intended location’s spelling. The search bar also allows the user to type in a country, perhaps ‘United States’, and all locations will be displayed within the country that is typed in. The ‘Flight’, ‘Hotel’, and ‘Things to Do’ classes are then displayed to the user.

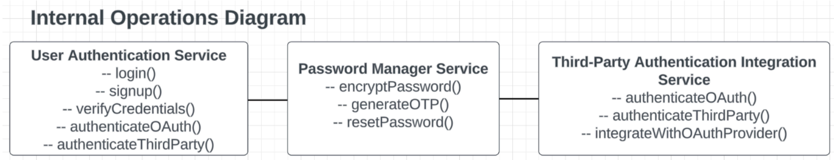
Component Interfaces (internal and external):

> **Internal Interfaces:** PlaceSearchService communicates with DestinationExplorer, AccommodationService, and TravelService using RESTful endpoints. DestinationExplorer interacts with AccommodationService and TravelService.
> **External Interfaces:** Utilizes third-party APIs for place information, images, and travel-related data.

Component Error Handling: Implements input validation to prevent malicious queries. Handles API errors gracefully and caches search results to minimize external dependencies.

Component Name: LogIn/SignUp
Component Description: Manages user authentication, login, and sign up processes. Ensures secure user accounts, password recovery, and integration with third-party authentication services.
Responsible Development Team Member: Yashaswini Sampath Kumar & Aditya Padgal
Component Diagram:

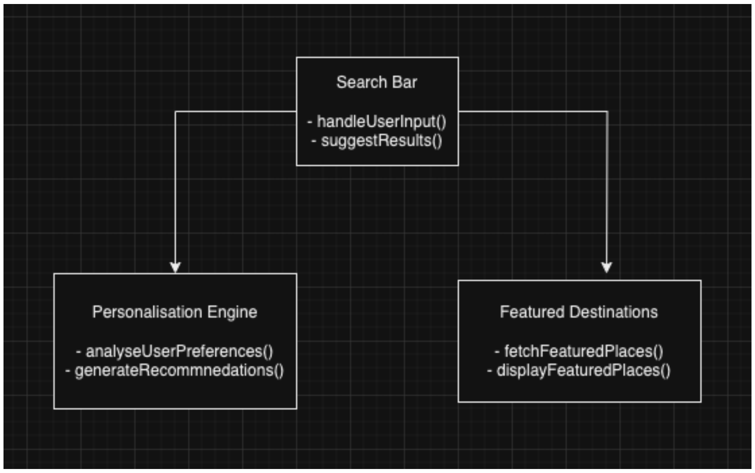



Internal Operations Diagram
User Authentication Service
-- login()
-- signup()
-- verifyCredentials()
-- authenticateOAuth()
-- authenticateThirdParty()
Password Manager Service
-- encryptPassword()
-- generateOTP()
-- resetPassword()
Third-Party Authentication Integration Service
-- authenticateOAuth()
-- authenticateThirdParty()
-- integrateWithOAuthProvider()

Component User Interface: The LogIn/SignUp component provides login and signup pages with secure authentication mechanisms. This component will also include password recovery options that utilize email, security questions, and OTP.
Component Objects: The LogIn/SignUp component uses several classes such as UserAuthService, PasswordManager, and ThirdPartyAuth. The UserAuthService class manages user authentication processes, including login and signup. PasswordManager will handle password encryption, reset, and recovery. The ThirdPartyAuth class integrates OAuth and third-party authentication services.
Component Interfaces (internal and external): Internal Interfaces: UserAuthService interacts with PasswordManager and ThirdPartyAuth. External Interfaces: This section of this component

will utilize OAuth for third-party authentication, allowing users to login using Google/Facebook. Utilize SMS gateways for OTP-based authentication.
Component Error Handling: Implements input validation for secure user authentication. Provides secure password recovery options. Handles errors during third-party authentication integration.

Component Name: Home Screen Page
Component Description: This page is the landing page of the web application, the user first comes across this page and has various recommendations to choose from. The user enters their trip details in the search bar to get a list of options of Hotels, Flights, Restaurants, Activities, Recommendations and Itinerary.
Responsible Development Team Member: Owen Harris & Hanna Adams
Component Diagram:



Component User Interface: The HomeScreen component presents a visually appealing and informative interface. It includes sections for featured destinations, displaying captivating images and brief descriptions. Personalized recommendations are based on user preferences and past activities. A prominent search bar allows users to initiate the planning process.
Component Objects: The HomeScreen component employs classes such as FeaturedDestinations, PersonalizationEngine, SearchBar. The FeaturedDestinations class manages the content and presentation of featured travel spots. PersonalizationEngine tailors recommendations based on user data. SearchBar handles user input for destination searches.
Component Interfaces (internal and external):

> **Internal Interfaces**: PersonalizationEngine and FeaturedDestinations interact with SearchBar.
> **External Interfaces:** Utilizes external APIs for fetching featured destinations, data, and personalized recommendations.

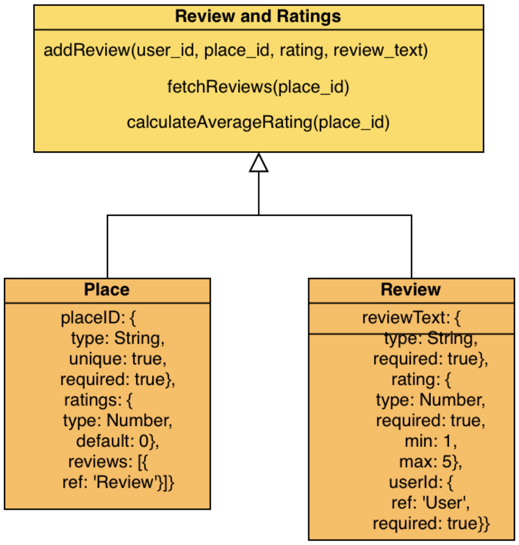
Component Error Handling: Implements input validation for the search bar to prevent invalid queries. Gracefully handles errors related to fetching featured destinations, trend analysis, and personalized recommendations. Provides informative error messages for a seamless user experience.

Component Name: Review and Rating

Component Description: The Review and Rating Component is responsible for collecting, managing, and displaying user-generated feedback on various services offered within the system such as places, accommodations, and travel services. It enables users to provide a numerical rating and write textual reviews about their experiences. The component also aggregates ratings to produce an average score for each service, which aids other users in making informed decisions.

Responsible Development Team Member: Hanna Adams

Component Diagram:



Component User Interface: Allows users to type in their review of a place, activity, or restaurant. Also allows users to rate their experience with a particular place, activity, or restaurant by giving the rating a numerical value 1-5, 1 being a terrible experience and 5 being an excellent experience. Ratings will then be displayed for the particular place, activity, or restaurant. For the numerical values of a particular rating, we will display the average rating to users. The average rating will then be displayed to the user using stars.

Component Objects: The Review and Rating component uses a review model wilh specific fields that are labeled if they are required or not and have a default text. We will create APIs for users to create, edit, delete, and display reviews for a particular place, hotel, or activity.

Component Interfaces (internal and external):

Internal Interfaces: ReviewManager interacts directly with the database to store and retrieve review data. ReviewManager calls on RatingCalculator to update average ratings when new reviews are added.

External Interfaces: Provides an API endpoint for frontend services to submit and fetch reviews. Connects with third-party services if external reviews need to be integrated.

Component Error Handling: Implements input validation to prevent malicious queries. Handles API errors gracefully and caches search results to minimize external dependencies.

Revision History

Revision	**Date**	**Change Description**
1	**3.3.2024**	Minute changes in Itinerary & Added the HomeScreen component.
2	**3/24/2024**	Search and Explore component & the Itinerary page component was updated to reflect changes
3	**4/7/2024**	Added the Review and Rating component to our Component Design section
4	**4/16/2024**	Added the User Authentication System & Travel Recommendation Engine components to our Component Design Section
5	**4/21/2024**	Made huge updates to the Itinerary component