WeTravel - Software Requirements Specification

Group D3; April 16, 2024

Version: 1.0

Shuyang Song 1155173859, Department of Statistics
Yanze Yang 11552109886, Department of Computer Science and Engineering
Sriroth Poonyapat 1155205059, Department of Computer Science and Engineering
Ziqin Wei 1155173761, Department of Computer Science and Engineering
Thuwanontha Kittiphan 1155190410, Department of Computer Science and Engineering

Contents

1 Version History

Version	Modified by	Date	Comments
0.1	All	February 9, 2025	First Draft
0.2	All	March 29, 2025	Added License Management
0.3	All	April 10, 2025	Modified contents based on development
1.0	All	April 16, 2025	Final Version

2 Introduction

2.1 Document Purpose

This document serves as SRS for our software which is one social platform for matching travelers.

2.2 Product Scope

Our software is a web-based social platform that matches travelers with the same destination and creates a community for matched travelers to share their plans, preferences and tips. Users may suggest destinations, dates and lodging within a group. Besides, Users may post their traveling experiences, comment on destinations and receive recommendations.

2.3 Intended Audience and Document Overview

This document is intended for developers who participate in the development of the software. The rest of this SRS contains functionalities to be implemented in this software and specific requirements. Readers may start by reading section Product Overview, and then proceed in sequential order.

2.4 Definitions, Acronyms and Abbreviations

The software (software): refer to the web-based social platform our group develops for travelers.

Users: refer to the intended users, mainly people who travel or seek traveling ideas.

SRS: refer to Software Requirements Specification.

2.5 Document Conventions

This document follows the IEEE formatting requirements. Use Times New Roman size 11 throughout the document for text. Use italics for comments. Use bold letters for titles. The document text is single-spaced and maintains the 1” margins.

2.6 References and Acknowledgments

We would like to thank the Department of Computer Science, George Mason University for the SRS template this file is based on and the Department of Computer Science and Engineering, The Chinese University of Hong Kong for CSCI3100 course materials.

3 Overall Description

3.1 Product Overview

The system is a web-based social platform designed for travelers to discover, match with, and plan trips collaboratively. It integrates features such as social trip posting, AI-powered travel recommendations, and

group expense sharing. Each user can register an account, manage personal travel preferences, and interact with others through posts, comments, and likes.

Users can create trip posts with key details (destination, dates, budget, activities, and optional images), which are displayed in a public feed and can be filtered by user interest. To enhance discovery, the system allows users to input their preferences (e.g., desired location, budget, and trip duration) and retrieves personalized travel suggestions via the ChatGPT API. These recommendations are generated using OpenAI's large language models and stored for future reference.

In addition, the platform supports user-to-group interactions, group trip coordination, and detailed expense tracking. Each group can log trip-related expenses, assign contributions per member, and view calculated balances.

3.2 Product Functionality

- **User Profile Management :** The system manages user registration, login, and profile creation to ensure users' security via hashed password control.
- **Trip Posting:** Users can create, edit, and delete trip posts. Each post may include an image (stored in the backend), and fields like destination, budget, activities, and dates. Posts support comments and likes.
- **Expense Management:** Group members can add log expenses, categorize them (e.g., Food, Transport), and automatically calculate cost-sharing.
- **AI-Driven Recommendations:** Users input travel preferences which are formatted into prompts and submitted to the OpenAI API. Returned itineraries are parsed and stored as structured arrays.

3.3 Design and Implementation Constraints

- Image uploads must be handled securely via Multer and stored with a reference path.
- The ChatGPT API must respond to requests with usable content within 5 seconds.
- Posts and recommendations must be stored in normalized schemas referencing user or group documents.

3.4 Assumptions and Dependencies

Assumptions	Dependencies
Users have stable internet access while using the app.	Users may experience degraded functionality or delays when offline.
The ChatGPT API (OpenAI) is available and responsive.	AI-generated recommendations may be unavailable during API downtime.
Users access the app via updated browsers (Chrome, Firefox, Edge, Safari) released in the last 5 years.	Older or unsupported browsers may lead to layout or functionality issues.
Users operate on supported operating systems (Linux, Windows) from the last 5 years.	Legacy OS versions may result in degraded or incompatible experiences.
Backend upload directory is writable and accessible.	Post image uploads may fail if the file system is restricted.
APIs and database servers are reliably connected.	Backend or database downtime can interrupt critical functions.

4 Specific Requirements

4.1 External Interface Requirements

4.1.1 User Interfaces

Users can interact with the system and other users online via browser.

4.1.2 Hardware Interfaces

This website can be logged in on devices like PCs or laptops. And historical data in the website will be stored on a centralized database. Users only need keyboards to type and mouses to click on this website and no more devices are needed on the PC side.

4.1.3 Software Interfaces

The frontend communicates solely with our own backend over RESTful HTTP/HTTPS.

- Backend API: Endpoints under `/api/auth`, `/api/users`, `/api/posts`, `/api/recommendations`, `/api/expenses`, `/api/trips`, `/api/groups`, and `/api/licenses`, all exchanging JSON.
- OpenAI ChatGPT API: Used only for AI-driven recommendations; the backend sends user preferences and parses the JSON response into our `Recommendation` model.

4.2 Functional Requirements

4.2.1 Main Page

- The main page should serve as a public feed of trip posts, filterable by destination, budget range, keyword, and posts you've liked in order that users can match their will more swiftly.
- Users can manage the order of posts based on time, country, activity, etc. Filters will be appeared.
- Users can create a new post/trip via the '+' button in the navigation bar.
- In the menu, users' profile management should be included.

4.2.2 User Profile Management

- The system shall allow new users to register by providing their name and password.
- The system shall allow registered users to log in using their email and password.
- The system shall allow users to create and edit their profiles (e.g., age, phone, gender, language, email).

4.2.3 Trip Posting

- The system shall display a feed of trip posts.
- The system shall allow users to filter trip posts by destination, trip dates (both start_date and end_date), or budget range.
- The system shall allow users to upload picture optionally.
- The system shall allow users to comment regarding trip posts within their group.

4.2.4 AI-Driven Recommendations

- The system shall allow users to input their trip preferences, including desired destinations, travel dates, budget range, and preferred activities, then the system shall use a large language model API to generate personalized travel recommendations based on user preferences.

4.2.5 Expense Management

- The system shall allow users to create expense list for specific trips and invite other travelers to join the group.
- The system shall allow the group members to log expenses with expense categories (e.g., meals, housing, transportation), amount, time, and description (optional).

- The system shall automatically or custom calculate cost-sharing among group members based on logged expenses.

4.3 License Management

- The system shall restrict access to software functionalities to users with a valid license key.
- A valid license key is mandatory to activate the software and access all core features (e.g., trip posting, AI recommendations).

5 Other Non-functional Requirements

5.1 Performance Requirements

- **Response Time for Trip Search:** The system shall respond to user queries and display forum posts within 2 seconds after the user submits a query.
- **AI Recommendation Generation:** The AI recommendation system shall provide results within 5 seconds with over 90% of the time.
- **Concurrent Users:** The system shall support up to 10 concurrent users without significant performance degradation.

5.2 Safety and Security Requirements

- **Data Encryption:** The system shall encrypt all sensitive user data (e.g.,password).

5.3 Software Quality Attributes

5.3.1 Usability

- **Requirement:** The user interface must be intuitive and require no more than 7 interactions for common actions such as searching for a trip and posting a trip.
- **Approach:** Use familiar UI patterns for forms, search bars, and notifications.

5.3.2 Maintainability

- **Requirement:** The code must be easy to understand and adjustable by every developer and include comprehensive documentation for developers.
- **Approach:** Use a version control system (Git) for collaborative development, follow consistent coding standards, and conduct code reviews regularly.

5.3.3 Robustness

- **Requirement:** The app must handle invalid inputs (e.g., incorrect trip data) by showing error messages that are easy for users to understand instead of crashing.
- **Approach:** Implement input validation and error handling, and show error notifications for debugging without exposing sensitive information to users.