Lab 2 – WiseTraveler Prototype Product Specification

Team Yellow

Old Dominion University

CS 411W, Spring 2025

Professor S. Sanober

April 18, 2025

Version: 2

Table of Contents

[*This space is left intentionally blank*]

3. Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.1.1

The system shall provide an interactive user interface on:

- desktops
- tablets
- mobile devices

(O: Montgomery)

3.1.2 Hardware interfaces

3.1.2.1

The system shall be accessible via web browsers on the following:

- desktop
- tablet
- mobile devices.

(O: Montgomery)

3.1.2.2

The system shall not require any specialized hardware beyond standard consumer computing devices. (O: Montgomery)

3.1.3 Software interfaces

3.1.3.1

The system shall integrate with Supabase for user authentication and database management. (O: Corley, M1: Montgomery)

3.1.3.2

The system shall interact with the OpenAI API for AI-based travel assistance. (O: Montgomery)

3.1.3.3

The system shall integrate with the Google Maps API for the following:

- interactive mapping
- navigation features.

(O: Montgomery)

3.1.4 Communication interfaces

3.1.4.1

The system shall support RESTful API calls for account management (O: Montgomery)

3.1.4.2

The system shall support RESTful API calls for authentication. (O: Montgomery)

3.2 System Features

3.2.1 Traveler Registration

3.2.1.1 Introduction/Purpose of feature

The system must provide travelers with account creation and management capabilities to support trip personalization based on the traveler's preferences.

3.2.1.2 Stimulus/Response Sequence

Stimulus: The user navigates to the WiseTraveler site and selects the Sign Up link.

Response: The system prompts the user to provide the required details

Stimulus: The user enters their details and submits the form.

Response: The system verifies the uniqueness of the email address and stores the user data securely.

3.2.1.3 Associated Functional Requirements

- The system shall provide a sign-up page where new travelers can create an account. (O: Novak, M1: Montgomery)
- The system shall require users to provide a valid email address during registration. (O: Novak)
- The system shall enforce password security requirements, including: minimum of 10 characters; at least one uppercase letter; at least one

lowercase letter; at least one number; and at least one special character (O: Novak, M1: Montgomery)

- The system shall verify that the entered email address is unique such that no other accounts exist in the database with the same email before creating a new account. (O: Novak)
- The system shall store user account information using Supabase's internal security features. (O: Novak, M1: Montgomery)

3.2.2 Traveler Authentication

3.2.2.1 Introduction/Purpose of Feature

WiseTraveler requires travelers to log in securely using their registered credentials to prevent unauthorized access to user data.

3.2.2.2 Stimulus/Response Sequence

Stimulus: A traveler navigates to the login page and inputs their user credentials.
Response: The system verifies the credentials, and upon successful authentication, redirects the user to their profile page.
Response: If authentication fails, the system displays an error message

3.2.2.3 Associated Functional Requirements

- The system shall provide a login page where registered users can authenticate using their email and password. (O: Novak, M1: Corley)
- The system shall redirect authenticated users to their profile page upon successful login. (O: Novak)
- The system shall provide error messages for failed login attempts. (O: Novak, M1: Montgomery)

3.2.3 Password Management

3.2.3.1 Introduction/Purpose of Feature

This feature allows travelers to reset or update their password.

3.2.3.2 Stimulus/Response Sequence

Stimulus: Traveler selects "Forgot password" and enters the email address associated with their account.

Response: System sends a password reset link to the registered email

3.2.3.3 Associated Functional Requirements

- The system shall provide a "Forgot Password" feature that allows users to reset their password. (O: Novak)
- The system shall send a password reset link to the user's registered email address when requested. (O: Corley)
- The system shall enforce the same password security requirements for password resets as for initial creation. (O: Novak)
- The system shall allow users to update their password from their profile page. (O: Corley, M1: Novak)

3.2.4 Traveler Profile Management

3.2.4.1 Introduction/Purpose of Feature

This feature allows users to manage their personal information and travel preferences.

3.2.4.2 Stimulus/Response Sequence

Stimulus: A traveler navigates to their profile page and updates personal details or travel preferences.

Response: The system saves the changes to the database in real-time.

Response: The system provides a confirmation alert.

3.2.4.3 Associated Functional Requirements

- The system shall provide a profile page where users can view and update their personal information. (O: Hill, M1: Novak)
- The system shall allow users to update their information including: name; date of birth; security word (O: Novak, M1: Corley, M2: Montgomery)
- The system shall allow users to update their travel preferences. (O: Novak)
- The system shall save changes to user profiles to the database. (O: Novak, M1: Corley)

- The system shall provide alerts when profile updates are successfully saved. (O: Novak, M1: Corley)

3.2.2 Landing Page

3.2.2.1 Introduction/Purpose of Feature

This feature provides a ‘home page’ that introduces WiseTraveler and provides navigation to major features.

3.2.2.2 Stimulus/Response Sequence

Stimulus: A user visits the root URL.

Response: The system displays the landing page.

3.2.2.3 Associated Functional Requirements

- The system shall provide a landing page that displays accordingly for mobile and desktop devices. (O: Novak)
- The landing page shall provide navigation links to key features of the application. (O: Novak)
- The landing page shall showcase the main benefits and features of WiseTraveler. (O: Novak)

3.2.3 Terms and Conditions

3.2.3.1 Introduction/Purpose of Feature

This feature ensures that users agree to the application’s legal terms.

3.2.3.2 Stimulus/Response Sequence

Stimulus: A user accesses registration.

Response: The system requires agreement to the Terms before proceeding.

3.2.3.3 Associated Functional Requirements

- The system shall provide a dedicated page for Terms and Conditions. (O: Novak)
- The system shall require users to accept the Terms and Conditions during the registration process. (O: Novak)

3.2.4 Interactive Destination Quiz

3.2.4.1 Introduction/Purpose of Feature

This feature helps users find travel destinations based on their preferences.

3.2.4.2 Stimulus/Response Sequence

Stimulus: A user navigates to the quiz and submits their responses to the questions.

Response: The system processes the response and provides recommendations.

3.2.4.3 Associated Functional Requirements

- The system shall provide an interactive quiz that recommends travel destinations based on user preferences. (O: Novak)
- The quiz shall collect user preferences including: budget; preferred climate; travel interests (e.g., culture, adventure, relaxation); trip duration; preferred activities (O: Novak, M1: Montgomery, M2: Montgomery)
- The system shall analyze user responses to generate personalized destination recommendations. (O: Novak)
- The system shall display at least one destination recommendation. (O: Novak)
- The system shall display a default location if no recommendation is available. (O: Montgomery)
- The system shall provide the option to retake the quiz with different preferences. (O: Novak)

3.2.5 RESTful API Integration

3.2.5.1 Introduction/Purpose of Feature

This features supports frontend-backend communication for user and profile management

3.2.5.2 Stimulus/Response Sequence

Stimulus: An API request is sent from the frontend.

Response: The backend processes the request and returns a JSON response

3.2.5.3 Associated Functional Requirements

- The system shall implement RESTful APIs to handle user authentication (O: Baker, M1: Corley, M2: Novak)
- The system shall implement RESTful APIs to manage user profiles, including the ability to retrieve, update, and delete user information (O: Baker, M1: Corley, M2: Novak)
- The system shall provide API endpoints for user registration. (O: Baker, M2: Corley, M1: Novak)
- The system shall provide API endpoints for user login and authentication. (O: Baker, M2: Corley, M1: Novak)
- The system shall provide API endpoints for password reset requests. (O: Baker, M2: Corley, M1: Novak)
- The system shall provide API endpoints for updating user profile information. (O: Baker, M2: Corley, M1: Novak)
- All API requests containing PII shall use secure HTTPS protocol. (O: Baker, M2: Corley)

3.2.6 AI Travel Assistant

3.2.6.1 Introduction/Purpose of Feature

This feature provides an AI-powered assistant to help users by answering travel-related questions.

3.2.6.2 Stimulus/Response Sequence

Stimulus: A traveler interacts with the chatbot

Response: The AI responds contextually using the OpenAI API.

3.2.6.3 Associated Functional Requirements

- The system shall integrate the OpenAI API to provide an AI chatbot for travel assistance. (O: Baker, M1: Edwards)
- The AI assistant shall respond to user queries about travel destinations, local customs, safety information, and travel advice. (O: Baker, M1: Edwards)

- The AI assistant shall provide tailored recommendations for each registered user (O: Baker, M1: Edwards, M2: Montgomery)
- The AI assistant shall maintain session-based memory of user inputs and preferences to enable follow-up questions and contextually relevant responses during an active session. (O: Baker, M1: Edwards)
- The AI assistant shall provide factually accurate and location-relevant information about travel destinations, local customs, and safety advisories. This information shall be sourced from APIs or datasets that are updated at least monthly for general content, and daily for time sensitive data such as government travel advisories. (O: Baker, M1: Edwards)
- The AI assistant interface shall be accessible from all pages within the application. (O: Montgomery)

3.2.7 Interactive Map

3.2.7.1 Introduction/Purpose of Feature

This feature allows users to explore destinations visually on a map, including destinations from a traveler's itinerary.

3.2.7.2 Stimulus/Response Sequence

Stimulus: A user navigates to the map interface and inputs a destination.
Response: The map loads and displays the retrieved travel data from the Google Maps API for the input destination

3.2.7.3 Associated Functional Requirements

- The system shall integrate with Google Maps API to provide an interactive map interface. (O: Martinez)
- The interactive map shall allow users to explore and discover locations within their travel destination. (O: Martinez)
- The map shall display points of interest, attractions, accommodations, and other relevant travel information front the Google Maps and Places APIs. (O: Martinez)

- The map shall allow users to filter displayed information based on categories (e.g., attractions, restaurants, accommodations). (O:Martinez)
- The map shall adapt its layout and functionality based on the user’s device screen size, ensuring usability on both mobile and desktop platforms. It shall provide intuitive controls for zooming, panning, and interacting with map markers on all supported devices. (O: Martinez)

3.2.8 Health & Safety Alerts

3.2.8.1 Introduction/Purpose of Feature

Travelers want to be safe when they are travelling. This feature provides real-time health and safety updates for destinations.

3.2.8.2 Stimulus/Response Sequence

Stimulus: A traveler selects a destination

Response: The system displays relevant alerts

3.2.8.3 Associated Functional Requirements

- The system shall provide real-time health and safety alerts specific to the user's selected destination. (O: Martinez)
- The system shall display information about local health risks and recommended precautions. (O: Martinez)
- The system shall provide safety advisories related to crime, natural disasters, and political instability. (O: Martinez)

3.2.9 Cultural Information

3.2.9.1 Introduction/Purpose of Feature

This feature educates users about local cultures and laws.

3.2.9.2 Stimulus/Response Sequence

Stimulus: A user navigates to the cultural content area for a destination.

Response: The system displays the relevant information

3.2.9.3 Additional Functional Requirements

- The system shall provide information about: local customs; traditions; etiquette for various destinations (O: Martinez)
- The system shall include information about local laws that travelers should be aware of. (O: Martinez)
- The system shall provide cultural dos and don'ts to help travelers avoid cultural faux pas. (O: Martinez)
- The system shall include information about local holidays and special events. (O: Martinez)

3.2.10 User Reviews

3.2.10.1 Introduction/Purpose of Feature

This feature allows users to review aspects of their trip.

3.2.10.2 Stimulus/Response Sequence

Stimulus: A user submits a review.

Response: The system stores the review to the database

Response: The system displays the review under the destination

3.2.10.3 Additional Functional Requirements

- The system shall allow users to write and submit reviews for: attractions; accommodations; activities/events (O: Baker, M1: Montgomery)
- The system shall allow users to rate experiences on a scale from 1 to 5 (O: Baker, M1: Montgomery)
- The system shall display the average rating for each location and attraction. (O: Baker, M1: Montgomery)
- The system shall display reviews for each location and attraction. (O: Baker, M1: Montgomery)

3.2.11 Itinerary & Calendar

3.2.11.1 Introduction/Purpose of Feature

This feature helps users to plan and view their trip itineraries using a calendar interface.

3.2.11.2 Stimulus/Response Sequence

Stimulus: A user creates/modifies trip events.

Response: The system stores the data in the database.

Response: The system displays the event in the calendar interface.

3.2.11.3 Additional Functional Requirements

- The system shall allow a traveler to create a new itinerary for a trip consisting of: trip name; location; start and end dates; specific events (e.g., dinner plans, showtimes, sightseeing tours, etc.) (O: Montgomery)
- The minimum information required to create an itinerary is a trip name, the location(s), and the start and end dates. (O: Montgomery)
- The system shall store the created itinerary locally to allow viewing when no internet connection is available. (O: Montgomery)
- The system shall make the itinerary accessible to and modifiable by the user. (O: Montgomery)
- The system shall provide a calendar feature for displaying itinerary information. (O: Hill)
- The calendar shall allow users to create and save events for specific dates and times. (O: Hill)

3.3 Performance requirements

3.3.1

The system shall load the landing page within 3 seconds. (O: Novak)

3.3.2

The system shall process login requests within 5 seconds. (O: Novak)

3.3.3

The system shall process destination quiz results within 2 seconds. (O: Novak)

3.3.4

The system shall support at least 1000 concurrent users without degradation in performance. (O: Martinez)

3.3.5

The AI assistant shall respond to user queries within 10 seconds. (O: Edwards)

3.3.6

The interactive map shall load and render within 5 seconds. (O: Martinez)

3.3.7

Database queries shall execute within 10 seconds. (O: Corley)

3.4 Database requirements

3.4.1

The system shall use PostgreSQL via Supabase as the primary database. (O: Montgomery)

3.4.2

The system shall implement row-level security (RLS) to prevent unauthorized access. (O: Montgomery)

3.4.3

Sensitive data shall be encrypted at rest and in transit. (O: Montgomery)

3.4.4

The system shall provide role-based access control (RBAC) for database users. (O: Montgomery)

3.5 Design Constraints

3.5.1

The frontend shall be developed using Next.js framework. (O: Novak)

3.5.2

The backend shall be implemented using Node.js. (O: Novak)

3.5.3

The database management shall use PostgreSQL via Supabase. (O: Corley)

3.5.4

The application shall follow responsive design principles to ensure compatibility with various device types. (O: Novak)

3.5.5

The application shall implement regular expressions for password validation. (O: Novak)

3.5.6

The application shall adhere to modern web security standards and best practices. (O: Novak)

3.5.7

The application shall integrate with the OpenAI API for AI-powered travel assistance. (O: Edwards)

3.5.8

The application shall integrate with Google Maps API for interactive map functionality. (O: Martinez)

3.5.9

The system architecture shall have separate tiers for the presentation layer, the application layer, and the data layer. (O: Novak, M1: Montgomery)

3.5.10

The web application shall be compatible with all modern web browsers, including the latest versions of Chrome, Firefox, Safari, and Edge. (O: Novak, M1: Montgomery)

3.5.11

The system shall maintain consistent design elements and navigation patterns throughout the application. (O: Novak)

3.6 Software system attributes

3.6.1 Reliability

3.6.1.1

The system shall implement data validation to prevent corruption of user profile information. (O: Corley)

3.6.1.2

The system shall maintain data consistency across all interfaces (O: Corley, M1: Montgomery)

3.6.1.3

The system shall maintain data consistency across all database operations. (O: Corley, M1: Montgomery)

3.6.1.4

The system shall implement error logging for all failed operations. (O: Novak)

3.6.1.5

The system shall gracefully handle failures from third-party APIs (e.g., Google Maps, OpenAI) by providing fallback responses or user-friendly error messaging without disrupting the user experience. (O: Baker, M1: Montgomery)

3.6.2 Availability

3.6.2.1

The system shall maintain a service availability of at least 99.9%, aligned with Supabase’s service level agreement. Minimal scheduled downtime may occur during planned maintenance windows. (O: Montgomery, M1: Montgomery)

3.6.3 Security

3.6.3.1

The system shall store all user data using Supabase’s secure authentication mechanisms. (O: Corley, M1: Montgomery)

3.6.3.2

The system shall enforce HTTPS for all client-server communications. (O: Baker)

3.6.3.3

The system shall implement protection against common web vulnerabilities including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). (O: Baker)

3.6.3.4

The system shall implement rate limiting on authentication attempts to prevent brute force attacks. (O: Corley)

3.6.3.5

The system shall validate all user inputs. (O: Corley, M1: Montgomery)

3.6.4 Maintainability

3.6.4.1

The codebase shall follow modular architecture with separation of functions for the frontend, backend, and data layers. (O: Montgomery)

3.6.4.2

The system shall be documented using JSDoc and README files. (O: Montgomery)

3.6.4.3

All endpoints and frontend components shall include tests. (O: Montgomery)

3.6.5 Portability

3.6.5.1

The frontend should be portable across modern web browsers. (O: Montgomery)

3.7 Other Requirements

3.7.1

The system shall comply with the defined data privacy policy. (O: Corley, M1: Montgomery)

3.7.2

The system shall support keyboard navigation for accessibility. (O: Novak)

3.7.3

The system shall conform to the Web Content Accessibility Guidelines (WCAG) for accessibility. (O: Novak)