Requirements Specification Document

Globetrotter

Trip Planning App

Table of Contents

1 Scope

1.1 Identification

This Software Requirements Specification (SRS) documents the requirements for the individual trip planning app, called Globetrotter.

1.2 System Overview

Globetrotter will be a mobile application built with React Native and Python. This application will be used for trip planning, and will draw from certain APIs such as the Google Maps API and the OpenAI API, and will draw from certain databases including MySQL for user authentication and storing user information. Globetrotter will feature many different components, consisting of a Login Page, a profile page displaying the user's planned trips and trip wishlist, a home/main page where the user will generate trips, and a settings page.

2 CSCI Component Breakdown

CSCI Globetrotter is composed of the following CSCs:

2.1 GUI Client CSC*:*

2.1.1 Login Page CSU: Page that allows users to login/create a new account with username and password.
2.1.2 Home Page CSU: Page that allows user to generate tip itineraries based off of certain constraints, and displays results
2.1.3 Setting Page CSU: Page that allows users to edit profile, log out, change aesthetic features, etc
2.1.4 Profile Page CSU: Page that displays user's profile picture, saved itineraries, and wish list consisting of destination the user wants to go

2.2 Database CSC*:* Stores user's profile information and trip information

2.2.1 User data CSU: User information shall include username, password, profile photo, travel wishlist
2.2.2 Trip Wishlist CSU: The database will store the destinations that the user wants to travel to in a wishlist (up to 5 destinations).
2.2.3 Saved Itineraries CSU: The database will store trips that the user saves. These trips will include the destination, day-to-day itineraries (including specific activities, accommodations, etc), and a map of all of the stops during the trip.

2.3 Backend CSC*:*

2.3.1 Authentication CSU: Backend will handle authentication with MySQL.
2.3.2 Trip Generation CSU: The app will process user input and utilize the openAI API to generate relevant results.
2.3.3 Maps Integration CSU: Interfaces with the Google Maps API to generate trip visualizations with each itinerary.

3 Functional Requirements

3.1 GUI Client CSC

3.1.1 Login Page CSU

3.1.1.1 The login page shall log users into the app.
3.1.1.2 The login page shall allow users to create an account through creating a username and password.
3.1.1.3 The login page shall bring users to the main page.

3.1.2 Main Page CSU

3.1.2.1 The main page shall contain a form/modal with questions for the user to fill out about their trip preferences.
3.1.2.1.1 The form shall include fields for: trip length (days), the date that the user wants to depart on, temperature preference, environment type (beach, city, mountains, cruise), and vacation style.
3.1.2.2 The main page shall contain a button at the end of the form that submits their preferences.
3.1.2.2.1 This button shall generate 3+ itinerary results with activity suggestions, and day-by-day plans.
3.1.2.2.2 When the button is pressed, the form shall disappear.
3.1.2.2.3 An “x” button shall appear when the form is submitted, resetting the form and results when pressed.
3.1.2.2.4 Each result shall have a 'Save to Profile' button
3.1.2.3 The main page shall contain a menu bar with 2 buttons.
3.1.2.3.2 The menu bar shall feature a home button for the home/main page.
3.1.2.3.3 The menu bar shall feature a profile icon, bringing users to the profile page.

3.1.3 Profile Page CSU

3.1.4.1 The profile page shall display the user’s name at the top
3.1.4.2 The profile page shall display the user’s profile picture below.
3.1.4.3 The profile page shall display the user’s username below the picture.
3.1.4.4 The profile page shall feature two tabs.
3.1.4.4.1 The first tab shall be the user’s wishlist, with desired trip destinations.
3.1.4.4.2 The second tab shall be the user’s saved itineraries, which they can scroll through and/or create new itineraries with.

3.1.4 Settings Page CSU

3.1.3.1 The settings page shall have an “edit profile” button.
3.1.3.2 The settings page shall have a “log out” button.
3.1.3.3 The settings page shall have an option to change profile pictures.

3.2 Database CSC

3.2.1 User Data

3.2.1.1 The database shall store the user's username.
3.2.1.2 The database shall store the user’s password.
3.2.1.3 The database shall store the user’s current profile picture.
3.2.1.4 The database shall store the user’s full name.
3.2.1.5 The database shall store the user’s email.
3.2.1.6 The database shall store the time that the user created an account.
3.2.1.5 The database shall store an id for each user.

3.2.2 Trip Wishlist

3.2.2.1 The database shall store the wishlist destinations that the user adds.
3.2.2.2 The database shall remember what user is logged in by grabbing the id from the user data table.

3.2.2 Saved Itineraries

3.2.2.1 The database shall store the generated trip itineraries with the following information:
3.2.2.1.1 Unique trip identifier
3.2.2.1.2 Trip creation date
3.2.2.1.3 Trip constraints/preferences (length, trip departure date, environment, temperature preference)
3.2.2.1.4 Map visualization data
3.2.2.2 The database shall allow trips to be updated/deleted by the user.
3.2.2.3 The database shall be able to save trips generated from the wishlist.
3.2.2.4 The database shall remember what user is logged in by grabbing the id from the user data table.

3.3 Back-end CSC

3.3.1 Authentication CSU

3.3.1.1 The service shall handle user registration through MySQL.

3.3.1.2 The service shall handle user login through MySQL.
3.3.1.3 The service shall handle password reset requests.
3.3.1.4 The service shall handle grabbing user data from the database and sending it to the front-end.

3.3.2 Trip Generation CSU

3.3.2.1 The service shall process user constraints from the form input.
3.3.2.2 The service shall communicate with OpenAI API to generate trip itineraries.
3.3.2.3 The service shall format API responses into structured trip data.
3.3.2.4 The service shall optimize itineraries based on travel time and budget.

3.3.3 Maps Integration CSU

3.3.3.1 The service shall communicate with Google Maps API
3.3.3.2 The service shall generate map visualizations for each itinerary.
3.3.3.3 The service shall generate location pins for each destination.