UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

Department of Computer Science

Software Engineering Degree

Subject: Advanced Web Development – NRC27819

Software Requirements Specifications (SRS)

Team: 2

Members:

Báez, Gabriel
Blacio, Julio
Cáceres, Germán

Teacher: Ing. Edison Lascano

Sangolquí – Ecuador

Índice

1. Introduction

1.1. Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements of the **Intelligent Travel Planner**, a web-based application designed to optimize and automate the trip-planning process. The system integrates external APIs to obtain real-time data and stores relevant information in a database. This document is intended for developers, designers, project managers, testers, and stakeholders involved in the project.

1.2. Scope

The Intelligent Travel Planner provides an integrated solution to assist users in creating personalized trips. It allows users to:

- Manage destinations and personal trips (CRUD).
- Calculate distances and travel times using mapping APIs.
- Retrieve weather information from weather APIs.
- Register and track budgets and expenses.
- Automatically generate and export personalized itineraries.

The system will be accessible from modern web browsers and compatible with desktop and mobile devices.

1.3. Definitions, Acronyms, and Abbreviations

- **CRUD**: Create, Read, Update, Delete
- **API**: Application Programming Interface
- **PDF**: Portable Document Format
- **UI**: User Interface
- **DB**: Database

1.4. Overview

This document presents a general description of the system (Section 2) and detailed functional requirements categorized by features (Section 3). Non-functional requirements are included in Section 4.

2. Overall Description

2.1. Product Perspective

The Intelligent Travel Planner is a standalone web system based on a client-server architecture. The backend manages:

- database operations,
- integration with mapping and weather APIs,
- business logic for itinerary generation and budget analysis.

The frontend provides users with a unified interface to centralize planning activities.

2.2. Product Functions

The system provides the following functionalities:

- CRUD management of destinations and trips.
- Calculation and visualization of routes and travel times.
- Weather data retrieval and storage.
- Budget creation and expense tracking.
- Automated itinerary creation and export.

2.3. User Characteristics

Target users include travelers seeking an automated and simplified trip-planning experience. Basic web navigation skills are required.

2.4. Constraints

- Internet connection required for API operations.
- External API usage limitations may apply.
- System must support modern web browsers (Chrome, Edge, Firefox).
- Fully responsive UI design is required.
- Persistent storage must be implemented through a database.

2.5. Assumptions and Dependencies

- Mapping APIs provide accurate distances and travel times.
- Weather APIs provide real-time weather data.
- Budget calculations depend on correct user input.
- System availability depends on external API uptime.

3. Personnel Involved

Name	Role	Professional Category	Responsibilities
Julio Blacio	Project Leader / Developer	Software Engineering	Leads planning, validates requirements, manages workflow, develops modules, and supports API integration.
Gabriel Báez	Analyst / Developer	Software Engineering	Requirement analysis, system design, feature development, API/DB integration.
Germán Cáceres	Developer / Tester	Software Engineering	Module implementation, testing, requirement validation, UI/DB support.

4. Specific Requirements

Feature 0: CRUD for Trips and Destinations

ID	Requirement	Description	Priority	Source	Verification
F0-1	Register Destination	Allows registering destinations with name, description, country, image, and stores them in the DB.	High	Backlog	Manual Test
F0-2	Manage Trips	Users can create, edit, and delete trips linked to destinations. Data is stored in the DB.	High	Backlog	Manual Test
F0-3	Display Trips	Displays all trips with search and filtering options using DB data.	Medium	Backlog	Manual Test

Feature 1: Route and Time Management

ID	Requirement	Description	Priority	Source	Verification
F1-1	Calculate Distance	Calculates distance and travel time using a mapping API.	High	Backlog	Manual Test

F1-2	Map Visualization	Displays map with origin and destination using API integration.	High	Backlog	Manual Test
F1-3	Save Favorite Routes	Allows saving user-selected routes into the DB.	Medium	Backlog	Manual Test

Feature 2: Weather Information

ID	Requirement	Description	Priority	Source	Verification
F2-1	Register Weather	Retrieves weather data from API with optional manual entry.	Medium	Backlog	Manual Test
F2-2	View Weather	Displays weather of selected destinations using API or stored data.	Medium	Backlog	Manual Test
F2-3	Weather for Itinerary	Displays weather for destinations included in the itinerary.	Medium	Backlog	Manual Test

Feature 3: Budget and Travel Expenses

ID	Requirement	Description	Priority	Source	Verification
F3-1	Enter Budget	Stores the total available budget in the DB.	High	Backlog	Manual Test
F3-2	Add Expenses	Adds estimated expenses per destination stored in the DB.	High	Backlog	Manual Test
F3-3	Budget Check	Calculates if the total expenses exceed the budget.	High	Backlog	Manual Test

Feature 4: Automatic Itinerary and Export

ID	Requirement	Description	Priority	Source	Verification
F4-1	Generate Itinerary	Automatically generates itinerary organized by days and destinations including API data.	High	Backlog	Manual Test

F4-2	Save Itinerary	Saves generated itineraries in the user profile.	Medium	Backlog	Manual Test
F4-3	Export Itinerary	Exports itinerary as PDF or shareable link.	Medium	Backlog	Manual Test

5. Non-Functional Requirements

5.1. Performance Requirements

- API responses must be processed in under 3 seconds.
- CRUD operations must complete in less than 1 second.

5.2. Security Requirements

- User authentication and role-based access are required.
- API keys must be stored securely and never exposed client-side.
- System must enforce HTTPS communication.

5.3. Usability Requirements

- UI must be intuitive and responsive.
- Users must complete trip creation in under five steps.

5.4. Availability Requirements

The system must maintain 99 % uptime excluding maintenance.

5.5. Maintainability Requirements

The codebase must use modular architecture to support extensibility.