**Software Requirements Specification**

**Intelligent Travel Planer**

# 1. Introduction

## 1.1. Purpose

The purpose of this document is to specify the functional and non-functional requirements for the Intelligent Travel Planner, a web application designed to streamline and automate the trip-planning process. The system consolidates travel-related data from multiple external APIs and stores relevant information in a database, allowing users to efficiently plan trips by managing destinations, routes, weather information, budgets, and personalized itineraries.

This specification is intended for developers, project managers, and stakeholders involved in the design, implementation, and validation of the system.

## 1.2. Scope

The Intelligent Travel Planner aims to provide an integrated solution for travel panning by combining features traditionally spread across multiple platforms

Users will be able to:

- Manage destinations and personal trips (CRUD stored in database)
- Calculate travel distances and estimated times using mapping APIs
- Register and view weather information form weather APIs
- Manage budgets and travel expenses stored in a database
- Automatically generate and export personalized itineraries including API data

The system will leverage external APIs to obtain real-time data. Relevant information will be stored in a database for improved performance and availability.

The application will be web/based, accessible from modern browsers, and responsive across devices.

## 1.3. Definitions, Acronyms, and Abbreviations

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

## 1.4. Overview

This document contains a general description of the system (Section 2) and detailed functional requirements organized by features (Section 3). Each feature includes three core requirements aligned with the development backlog.

## 2. Overall Description

### 2.1. Product Perspective

The Intelligent Travel Planner is a standalone web application following a client-server architecture

A backend service will manage database operations and communication with external APIs (maps, routes, weather). The system offers a unified interface that centralizes automation of trip information and user decision-making.

### 2.2. Product Functions

The system will provide:

- Destination and trip management (CRUD) stored in database
- Route calculation and travel time estimation using mapping APIs
- Weather information retrieval and visualization using weather APIs
- Budget and expense tracking stored in database
- Automatic itinerary creation based on trip data and export options

### 2.3. User Characteristics

Target users are travelers seeking to simplify trip planning. Users are expected to have basic computer and web navigation skills and an interest in efficient, automated travel recommendations

### 2.4. Constraints

- Internet access is required for real-time API data
- External API usage is subject to provider limits and policies
- The system must support modern web browsers (Chrome, Edge, Firefox)
- The UI must be responsive for desktop and mobile devices
- The system must include a database for persistent data storage

### 2.5. Assumptions and Dependencies

- Route and distance calculations rely on mapping APIs
- Weather information will be obtained dynamically through weather APIs, with stored fallback data for offline availability
- Budget tracking depends on accurate user input and update currency exchange rates if applicable
- Service availability depends on external API uptime

## 3. Specific Requirements

### Feature 0: CRUD for Trips and Destinations

| ID | Requirement | Description | Priority | Source | Verification |
|---|---|---|---|---|---|
| F0-1 | Register Destination | Allows registering tourist destinations with name, description, country, image and stores them in the database; information may be complemented with API data where available | High | Backlog | Manual test |
| F0-2 | Manage Trips | Users can create, edit, and delete personal trips linking one or more destinations, stored in the database | High | Backlog | Manual test |
| F0-3 | Display Trips | Display a list of all trips with search and filtering by name or destination using DB data | Medium | Backlog | Manual test |

**Feature 1: Route and Time Management**

| ID | Requirement | Description | Priority | Source | Verification |
|---|---|---|---|---|---|
| F1-1 | Calculate Distance | Users can enter two destinations to calculate distance and travel time using mapping API; results may be stored to optimize future queries | High | Backlog | Manual test |
| F1-2 | Map Visualization | Display a map showing origin and destination points through mapping API integration | High | Backlog | Manual test |
| F1-3 | Save Favorite Routes | Users can save favorite routes in the database for easy access | Medium | Backlog | Manual test |

**Feature 2: Weather Information**

| ID | Requirement | Description | Priority | Source | Verification |
|---|---|---|---|---|---|
| F2-1 | Register Weather | Allows obtaining and displaying weather | Medium | Backlog | Manual test |

| | | | | | |
|---|---|---|---|---|---|
| | | data automatically from weather APIs, with option for manual entry if API unavailable; data may be stored for reuse | | | |
| F2-2 | View Weather | Users can view weather information from selected destinations retrieved from API or stored data | Medium | Backlog | Manual test |
| F2-3 | Weather for Itinerary | Display weather for all destinations included in the itinerary using API or stored data | Medium | Backlog | Manual test |

**Feature 3: Budget and Travel Expenses**

| ID | Requirement | Description | Priority | Source | Verification |
|---|---|---|---|---|---|
| F3-1 | Enter Budget | Users can enter and store the total available | High | Backlog | Manual test |

| | | | | | |
|---|---|---|---|---|---|
| | | budget for a trip in the database | | | |
| F3-2 | Add Expenses | Allow adding estimated expenses per destination stored in the database | High | Backlog | Manual test |
| F3-3 | Budget Check | Automatically calculate total expenses and indicate if within or over budget using stored data | High | Backlog | Manual test |

**Feature 4: Automatic Itinerary and Export**

| ID | Requirement | Description | Priority | Source | Verification |
|---|---|---|---|---|---|
| F4-1 | Generate Itinerary | Automatically generate travel itinerary organized by days and destinations, including API data where relevant | High | Backlog | Manual test |
| F4-2 | Save Itinerary | Users can save and | Medium | Backlog | Manual test |

| | | | | | |
|---|---|---|---|---|---|
| | | view generated itineraries in their profile stored in the database | | | |
| F4-3 | Export Itinerary | Allow downloading itinerary in PDF or sharing via link using stored itinerary data | Medium | Backlog | Manual test |