
Software Requirements Specification

for

Travel Itinerary Management App

Version 1.0 approved

Prepared by Sachin Sharma

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

26th August 2025

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope	2
1.5 References.....	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	3
2.3 User Classes and Characteristics	3
2.4 Operating Environment	4
2.5 Design and Implementation Constraints.....	4
2.6 User Documentation.....	4
2.7 Assumptions and Dependencies	4
3. External Interface Requirements	5
3.1 User Interfaces.....	5
3.2 Hardware Interfaces.....	5
3.3 Software Interfaces	6
3.4 Communications Interfaces	6
4. System Features	7
4.1 Automated Itinerary Generation	7
4.2 Booking Consolidation and Management	7
5. Other Nonfunctional Requirements.....	8
5.1 Performance Requirements.....	8
5.2 Safety Requirements.....	8
5.3 Security Requirements.....	9
5.4 Software Quality Attributes.....	9
5.5 Business Rules.....	9
6. Other Requirements	9
Appendix A: Glossary.....	10
Appendix B: Analysis Models	11
Appendix C: To Be Determined List.....	11

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements for the Travel Itinerary Management App, version 1.0. This SRS outlines the features, design constraints, user classes, and functional/nonfunctional needs for a smart, mobile-based application that automates trip planning and manages personalized travel itineraries.

This document serves as the foundation for development, testing, and validation of the application, and is intended for all stakeholders including developers, testers, project managers, and clients. The entire system described herein is self-contained and addresses the needs outlined in the project problem statement, focusing on overcoming the fragmentation and stress of modern travel planning.

1.2 Document Conventions

- *Requirements are uniquely tagged (e.g., REQ-1, REQ-2) and classified as either “shall” (mandatory) or “should” (recommended).*
- *Placeholders identified with angled brackets (< >) are to be replaced with finalized project details as available.*
- *System features are described in subsections with a three-digit hierarchical number for traceability (e.g., 4.1.1).*
- *Unless otherwise specified, requirements at a higher level are inherited by subsidiary items.*
- *Tables, figures, and diagrams referenced are numbered sequentially within each section.*
- *Technical terms and acronyms are defined in Appendix A (Glossary).*

1.3 Intended Audience and Reading Suggestions

This SRS is intended for:

- **Developers & Designers:** *For implementation and architecture planning, with focus on Sections 2, 3, and 4.*
- **Project Managers & Clients:** *For scoping, progress tracking, and change management; start with the Introduction and Overall Description.*
- **Testers & QA:** *For validation of system features and nonfunctional constraints; focus on System Features and Nonfunctional Requirements.*

- **Documentation Writers:** To prepare user manuals and help resources based on the user interface and system features.

It is recommended to review Section 1 for an overview, Section 2 for context and main functions, and Section 4 for detailed system features.

1.4 Product Scope

The Travel Itinerary Management App aims to serve as a centralized travel planning solution that eliminates scattered information and manual scheduling. It will:

- *Automatically generate day-wise itineraries optimized for time, budget, and user preferences.*
- *Consolidate bookings, activities, routes, and notes into one platform.*
- *Adapt plans in real-time to account for delays, cancellations, or changes.*
- *Provide offline access for convenience during travel.*
- *Support collaboration by allowing users to share itineraries with family, friends, or colleagues.*

By providing structure, personalization, and adaptability, the system will save time, reduce stress, and enhance the overall travel experience. In future iterations, integration with booking APIs, real-time weather/event updates, and expense tracking will further expand the app's capabilities.

1.5 References

- *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.*
- *Karl E. Wiegers, Software Requirements Specification Template (1999).*
- *Travel behavior research studies on itinerary planning and decision-making.*
- *Industry examples of travel-related apps (e.g., Google Trips, TripIt, Kayak) used for comparative analysis.*
- *User experience design principles for mobile travel applications.*

2. Overall Description

2.1 Product Perspective

The Travel Itinerary Management App is a, standalone mobile application designed to simplify and automate travel planning. It is not a component of a larger system but may integrate with external

APIs and services such as flight, hotel, and local activity booking platforms in future updates. The app addresses fragmentation issues inherent in typical travel planning by acting as a centralized repository for all travel-related information—bookings, schedules, routes, preferences, and notes—thus streamlining the trip management process. It ensures adaptability through AI-driven itinerary generation and supports real-time changes like delays or cancellations. The app will coexist with device native functionalities such as calendar, push notifications, and offline storage to enhance the user experience.

2.2 Product Functions

Key functions of the product include:

- *Automated creation of optimized day-wise travel itineraries based on minimal user inputs (destination, dates, preferences, budget).*
- *Consolidation and management of all travel bookings (flights, accommodations, transport) and activity scheduling within a single app.*
- *AI-enhanced personalized recommendations for attractions, dining, and activities tailored to individual user interests and travel context.*
- *Real-time notifications and alerts for itinerary changes, travel delays, and reminders related to bookings or activities.*
- *Offline availability of the itinerary and essential travel data to ensure uninterrupted access during connectivity issues.*
- *Collaborative features enabling sharing and joint editing of itineraries among travel companions.*
- *Interactive, user-friendly interface including drag-and-drop itinerary editing and search/filter tools for trip customization.*

2.3 User Classes and Characteristics

The application targets diverse user classes including:

- **Solo Travelers:** *Seeking simplicity, customization, and stress-free planning with quick itinerary generation.*
- **Families and Groups:** *Needing collaborative planning features, shared itineraries, and easy coordination across multiple travelers.*
- **Business Travelers:** *Prioritizing efficiency, timely notifications, and adaptability to last-minute changes.*
- **Travel Planners and Tour Operators:** *Utilizing the app for managing multiple clients' travel plans, requiring robust data handling and sharing features.*

Users will range from tech-savvy millennials to less experienced travelers, so the app must feature an intuitive UI with minimal learning curve.

2.4 Operating Environment

The app will operate on:

- *Smartphones running Android (version 10 and above) and iOS (version 13 and above).*
- *Devices with a minimum of 2GB RAM recommended for smooth performance.*
- *Offline mode capabilities to provide users access to saved itineraries and travel information without internet connectivity.*

Network environments including 4G, 5G, WiFi, to sync data, receive real-time updates, and push notifications.

2.5 Design and Implementation Constraints

- *Compliance with platform-specific UI/UX guidelines such as Google's Material Design and Apple's Human Interface Guidelines to ensure consistent and native-feeling user experiences.*
- *Data privacy adherence mandated by regulations such as GDPR and India's IT Act, including encrypted data storage and secure communication protocols.*
- *Device storage limitations for offline data caching must be considered.*
- *The app architecture must allow for scalability and integration with third-party APIs for bookings and mapping in future versions.*
- *Network reliability variations require robust offline functionality and sync mechanisms.*
- *Support for multiple screen sizes and device orientations.*

2.6 User Documentation

- *Embedded onboarding tutorials guiding new users through itinerary setup and application features.*
- *In-app help sections covering FAQs, troubleshooting, and feature explanations.*
- *User manuals available in digital format (PDF/HTML) for advanced usage scenarios.*
- *Future plans include video tutorials and contextual tips to enhance user engagement.*

2.7 Assumptions and Dependencies

- *Users possess basic smartphone usage proficiency.*
- *Initial data input is manual with future versions planned to integrate automatic data import from emails and booking confirmations.*

- *Availability and accessibility of third-party travel data APIs for bookings and local activity information.*
- *Reliable device storage and permission access granted by the user for notifications, location services, and offline caching.*
- *Stable internet connectivity for sync and updates when available, but offline mode caters to intermittent connectivity*

3. External Interface Requirements

3.1 User Interfaces

The Travel Itinerary Management App will provide a clean, intuitive, and responsive user interface designed for mobile devices, consistent with platform-specific UI guidelines (Material Design for Android, Apple Human Interface Guidelines for iOS). Key UI components include:

- **Home/Dashboard:** *Overview of upcoming trips, recent activity, and quick action buttons.*
- **Trip Creation Wizard:** *Step-by-step guided input for destinations, dates, budgets, and preferences with form validation and contextual help.*
- **Itinerary View:** *A day-wise scrollable schedule presenting activities, bookings, routes, and notes with drag-and-drop reorder functionality.*
- **Booking Management:** *Sections to add, edit, and view flight, hotel, and transport bookings alongside reminders.*
- **Personalized Recommendations:** *Interactive lists/cards of attractions, dining, and experiences based on user interests.*
- **Notification Center:** *Accessible interface for all alerts related to itinerary changes and reminders.*
- **Offline Mode Indicators:** *Clear UI hints ensuring users know when data is cached versus live.*

UI will support gesture-based interactions (swipes, taps, and drag/drop) and accessibility features such as scalable fonts and voiceover support. Navigation will be optimized for single-handed use with a bottom navigation bar and contextual menus for efficient ease of use.

3.2 Hardware Interfaces

- **Device Storage:** *The app will utilize local storage to cache trip details and assets for offline accessibility. Storage management includes efficient data compression and cleanup routines.*
- **Location Services:** *For location-aware recommendations and mapping, the app will access GPS data with user permission.*

- **Camera Access:** *Optional use to capture travel documents and photos for itinerary attachment.*
- **Push Notifications:** *Leverages OS-specific push services (Google Firebase Cloud Messaging for Android, Apple Push Notification Service for iOS) for timely alerts and reminders.*
- **Sensors:** *Basic interaction with device accelerometers and gyroscope may be employed for enhanced gesture controls but is not mandatory.*

3.3 Software Interfaces

The app will rely on the following software interfaces:

- **Operating Systems:** *Android 10+ and iOS 13+.*
- **Third-Party APIs (Planned):**
 - *Booking APIs for flights, hotels, and tours (e.g., Amadeus, Skyscanner) for direct data import and booking management.*
 - *Weather APIs to provide real-time updates influencing itinerary changes.*
 - *Map and navigation SDKs (e.g., Google Maps, Mapbox) for route visualization and travel time estimation.*
- **Backend Services:**
 - *REST/GraphQL APIs for itinerary generation, storage, and user authentication.*
 - *AI Recommendation Engine (hosted on the backend) for activity suggestions.*
- **Database Integration:** *Cloud-based database (e.g., MongoDB/PostgreSQL) for storing user data, preferences, and itinerary details.*
- **Authentication Systems:** *OAuth 2.0 / Firebase Authentication for secure login and account management.*

3.4 Communications Interfaces

- **Network Protocols:** *HTTPS will be mandated for all server communications to ensure data encryption in transit.*
- **Push Notifications:** *Utilizes platform messaging protocols (FCM/APNs) to deliver timely, relevant notifications.*
- **Data Synchronization:** *Background sync services operating via RESTful APIs for keeping itinerary data consistent across devices and cloud backups. Sync includes delta-update optimization to preserve bandwidth.*

- **Offline Functionality:** The app will gracefully degrade by allowing editing and viewing of previously cached itineraries during loss of connectivity, with changes queued for synchronization upon reconnection.
- **Sharing Mechanisms:** Itineraries can be exported as shareable files (PDF/JSON) or via secure links, supporting email, messaging apps, or social media platforms with proper data privacy safeguards.

4. System Features

4.1 Automated Itinerary Generation

4.1.1 Description and Priority

This feature automatically creates optimized day-wise travel itineraries from minimal user inputs such as destination, travel dates, interests, and budget. It dynamically accounts for factors like opening hours, travel times, and user pace. This is a high priority feature as it forms the core value proposition of the app.

4.1.2 Stimulus/Response Sequences

User inputs basic trip details → System processes data and preferences → System generates a complete daily itinerary → User reviews and modifies if needed.

4.1.3 Functional Requirements

REQ-4.1.1: The system shall accept destination, dates, budget, and preferences as inputs.

REQ-4.1.2: The system shall generate a day-wise schedule including activities, routes, meals, and rest periods.

REQ-4.1.3: The system shall allow manual overrides or adjustments by the user post-generation.

REQ-4.1.4: The system shall re-optimize the itinerary upon changes in input or disruptions.

4.2 Booking Consolidation and Management

4.2.1 Description and Priority

This feature consolidates all travel-related bookings (flights, hotels, transportation, and activities) in one unified location within the app, providing users with holistic trip management and reducing information fragmentation. It delivers timely notifications and reminders about all bookings. This is a high priority feature as all major travel elements must be accessible and organized for user convenience.

4.2.2 Stimulus/Response Sequences

User adds or imports booking details (e.g., uploads a confirmation or forwards an email) → System stores and categorizes the booking information → System syncs

bookings with the itinerary timeline → System triggers timely reminders and alerts for check-ins or departures.

4.2.3 Functional Requirements

REQ-4.3.1: The system shall allow manual entry and import of booking details (flights, hotels, transport, activities).

REQ-4.3.2: The system shall associate each booking with the correct itinerary day and display key booking details within the trip schedule.

REQ-4.3.3: The system shall generate automated reminders for upcoming check-ins, departures, or reservation expiration.

REQ-4.3.4: The system shall permit users to update or delete bookings and reflect those changes in the itinerary.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system shall generate an optimized itinerary within 5 seconds of receiving user input under normal network conditions.*
- The app shall handle at least 10,000 concurrent users with minimal latency (< 2 seconds response time for API calls).*
- Offline access shall allow retrieval of saved itineraries in under 2 seconds.*
- Notifications shall be delivered within 30 seconds of scheduled trigger time.*
- Maps and location-based queries shall load within 3 seconds on devices with a stable internet connection.*

5.2 Safety Requirements

- The app shall prevent data loss during unexpected shutdowns by maintaining auto-save states.*
- Sensitive user data (personal info, trip details, payment history) shall be backed up in encrypted form on secure servers.*
- In offline mode, all unsynchronized changes shall be stored locally and synchronized upon reconnection without data corruption.*
- The app shall ensure safe handling of external links (e.g., bookings, ticketing) to avoid phishing or malicious redirects.*

5.3 Security Requirements

- *All personal and travel data shall be stored encrypted at rest (e.g., AES-256) and transmitted using secure protocols (e.g., HTTPS/TLS).*
- *The app shall not access device data (e.g., contacts, calendar) or share itinerary details externally without explicit user permission.*
- *User authentication processes (e.g., email/password, OAuth, or biometrics) shall be used before granting access to personal itineraries and trip data.*
- *The system shall comply with relevant data protection laws including GDPR and the Indian IT Act.*

5.4 Software Quality Attributes

- *The app shall be intuitive and simple for first-time users to complete their first itinerary in under 10 minutes, even without prior experience.*
- *The app codebase shall enable critical bug fixes to be deployed within 24 hours of identification.*
- *The system shall operate reliably with no more than one crash per 1000 sessions under normal usage.*
- *Consistency of experience shall be maintained across all supported Android and iOS devices*

5.5 Business Rules

- *Only registered users can create, edit, or save more than one itinerary; guest users are limited to a single session itinerary with no sharing or booking storage.*
- *Only the itinerary owner(s) or explicitly invited collaborators may alter a shared trip.*
- *The app will not display promotional offers or sponsored content unless explicitly enabled by the user in preferences.*

6. Other Requirements

- **Database Requirements:**
The app shall use a secure, lightweight local mobile database (e.g., SQLite, Core Data) for offline access, with encrypted cloud synchronization (e.g., Firebase, AWS) as an enhancement for backup and sharing. Data structures must support nested itineraries, multi-user access, and version history for edits.

- **Internationalization & Localization:** The system must support multiple languages (at least English and Hindi at launch), easily extensible to additional languages. All display strings, messages, and date/time/currency formatting shall be externalized from code for easy localization.
- **Legal & Regulatory Compliance:** The app must explicitly obtain user consent for any data collection, storage, or sharing. Privacy policies and terms of service must be accessible from within the app and comply with GDPR, Indian IT Act, and other applicable regulations.
- **Reuse Objectives:** Modular architecture design should enable code and component reuse (e.g., authentication, notification, itinerary builder widgets) in future travel or event-planning projects. Documentation and API endpoints should follow standards to promote reuse.
- **Reporting & Export:** Users shall be able to export their itineraries as PDF or CSV for external sharing and backup.
- **Accessibility:** The app shall adhere to accessibility guidelines (e.g., WCAG 2.1 AA), supporting screen readers, high-contrast themes, and scalable UI elements for users with disabilities.

Appendix A: Glossary

- **Itinerary:** A detailed, date-wise plan of travel containing all bookings, activities, and related notes.
- **Activity:** Any task, event, or visit scheduled within the itinerary (e.g., sightseeing, dining, meetings).
- **Booking:** A reservation for travel (flight, hotel, transport, event) integrated into the itinerary.
- **Offline Mode:** App functionality that allows access, editing, and storage without internet connectivity.
- **Push Notification:** An automated alert delivered to the user's device regarding bookings, changes, or reminders.
- **AI Recommendation:** Personalized activity, route, or venue suggestions generated based on user data and trip context.
- **Collaborator:** A person with shared edit/view access to the itinerary, usually a travel companion.
- **GDPR:** General Data Protection Regulation, European privacy protection law.
- **Localization:** The process of adapting software for different regions, languages, or cultural expectations.

Appendix B: Analysis Models

Analysis models such as data flow diagrams, class diagrams, and state-transition diagrams will be added here once available. This section will be updated in future versions of the SRS to include all relevant system design models.

Appendix C: To Be Determined List

- *Specific third-party APIs to be used for real-time data (e.g., flights, weather, maps).*
- *Final list of supported languages and locales for the initial release.*
- *Cloud storage provider to be chosen for synchronization and sharing.*
- *UI/UX design mockups and accessibility test coverage.*
- *Detailed logging and analytics requirements.*
- *Regulatory certification needs for additional geographies (if expansion planned).*
- *Data retention policy and disaster recovery procedures.*