Software Requirements Specifications
For
Multi-Purpose Planner Web Application Development

February 9, 2025

Version 2

Instructor: Lam Tak Kei

Course: CSCI3100

Group ID: B3

Group Members:

Lee Chun Yin	1155192574
Chow Sui Ki	1155193739
Fung Pak Hei	1155193590

1 Introduction

1.1 Document Purpose

This Software Requirements Specification (SRS) document serves as a guideline for the project development process for a web-based multi-purpose planner application. This document includes all requirements needed for the system, and its design and implementation will follow the requirements in this document.

1.3 Definitions, Acronyms and Abbreviations

(nil)

1.4 Reference

(nil)

2 Overall Description

2.1 Product Overview

The Multi-Purpose Planner is a web-based application designed to help users organize their travel plans and keep track of their daily expenses. This application aims to provide a seamless experience for travelers by simplifying itinerary management.

2.2 Product Functionality

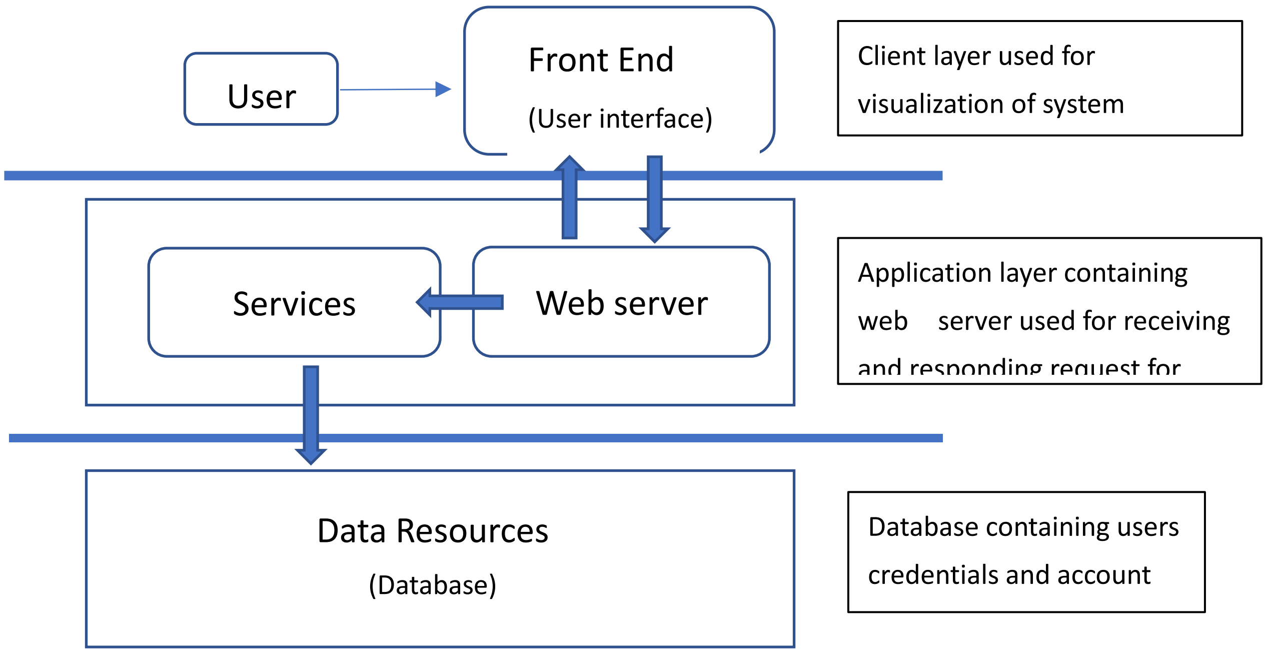
The system allows users to create and manage travel itineraries, including transportation, accommodation, activities, and schedules. Besides, it tracks and categorizes expenses to help users stay within budget and generate reports and summaries of expenses for better fiscal management.

2.3 Assumptions and Dependencies

Assumption No.	Title	Description
1	User connectivity	Users will have access to the internet and a compatible device
2	Browser Compatibility	The application will be accessed using the latest versions of popular web browsers (e.g., Chrome, Firefox, Safari, Edge)
3	Third-party Services	It is assumed that any third-party services (e.g. mapping services) will be operational and available for integration during the application's lifecycle
4	Web Server	The application depends on a reliable web server to host the application and serve requests from users
5	Database System	The application relies on a database management system (e.g., MySQL) for data storage and retrieval
6	Third-party APIs	The application depends on third-party APIs for features such as google map
7	Development Framework	The application is developed using specific frameworks (React), which must be maintained and updated regularly

3 High-Level System Architecture

The system adopts a three-tier architecture: Client tier, Application tier, and Data tier.


User
Front End
(User interface)
Client layer used for visualization of system
Services
Web server
Application layer containing web server used for receiving and responding request for
Data Resources
(Database)
Database containing users credentials and account

4 Specific Requirements

4.1 Functional Requirements

Req. No.	Title	Description
FR1	User Management	
FR1.1	User Registration	Create new account by providing username and password
FR1.2	User Authentication	
FR1.2.1	Login Authentication	Allow user to log in using their username and password
FR1.2.2	Logout Authentication	Save all user changes after user logout their account
FR1.3	Password Management	Allow user to change password after logging in, and reset password with email account when forget password

FR1.4	Account Deletion	Allow user to remove their account and related data
FR2	Budget Planner	
FR2.1	Manage Expense/Income	
FR2.1.1	Add Expense/Income Records	Add new expense/income record with details (amount, date, categories)
FR2.1.2	Edit Records	User can change the information of records
FR2.1.3	Delete Records	
FR2.2	Overview of Expense/Income	Review Expense/ Income/ Balance by months or by categories
FR3	Travel planner	
FR3.1	Create and manage travels	
FR3.1.1	Add travels	
FR3.1.2	Edit travels	
FR3.1.3	Delete travels	
FR3.2	Add itineraries	Add details for flights, accommodation, activities, and transportation to their itineraries.
FR3.3	Daily View	Provide users with a daily view of their itineraries.
FR3.4	Share Travel Plan	Allow users to share their Share Travel Plan with other users via email.
FR3.5	Collaborative Editing	Allow collaborative editing of travel plans by shared users.

4.2 Actors

An Actor is an entity that interacts with the system by exchanging data. These entities can be users, other systems, or devices that perform actions with the system.

4.2.1 System Administrator

The System Administrator will have full access to the whole application.

4.2.2 End Users

End users can access planner functions, personal profile, account management in the web application.

4.3 Use Case

4.3.1 User Registration

Initiator: End User

Description: This use case allows new users to create accounts.

Basic Flow of Events:

The system presents a registration form to the user, requesting information such as username, email address, and a chosen password.
The user fills out the form and submits it[E1][E2].
The system validates the information, creates a new account, and sends a confirmation email to activate the account[E3].

Exceptional Flow of Events:

[E1] If username or email address existed, the system display an error message.
[E2] If the password is too weak, the system displays a “use a stronger password” message.
[E3] If confirmation email is not received, the account is not activated.

4.3.2 Login Authentication

Initiator: End User

Description: This use case allows registered users log into the web application, verifying their identity to access authorized resources.

Basic Flow of Events:

The system presents a login form, prompting for a username/email and password.
The user enters their credentials and submits them.
The system verifies the credentials against the stored data[E1].
If valid, the system grants access to the user's account and the web application's secured areas.

Exceptional Flow of Events:

[E1] If the username/email and password does not match against the stored data, the system displays a “wrong password” message.

4.3.3 Logout Authentication

Initiator: End User

Description: This use case allows users to log out of the web application, terminating their active session.

Basic Flow of Events:

The user selects the "Logout" option.
The system terminates the user's session, clears any session-related data, and redirects the user to the login page of the web application.

Exceptional Flow of Events:

There are typically no exceptional flows for logout authentication.

4.3.4 Forgot Password and Change Password Using Email

Initiator: End user

Description: This use case allows users to reset their password by requesting a password reset link sent to their registered email address.

Basic Flow of Events:

The user selects “forget password” while login failed.
The system asks the user to enter their email address.
The actor enters their email address and submits the request[E1].
The system then sends a password reset link to the provided email address.
The user clicks the link, is directed to a password reset form, and sets a new password.

Exceptional Flow of Events:

[E1] If the email is not found in database, the system displays a “account not found” message.

4.3.5 Account Deletion

Initiator: End User

Description: This use case allows users to delete their account from the web application.

Basic Flow of Events:

The user navigates to the account settings and selects the "Delete Account" option after logging in.
The system asks for confirmation and requires the user to enter their password[E1].
Upon confirmation, the system permanently deletes the user's account and associated data.

Exceptional Flow of Events:

[E1] If the user enters a wrong password, the system will display an error message.

4.3.6 Add Expense/Income Records

Initiator: End User

Description: This use case allows the user to add a new expense or income record to their personal planner with details such as amount, date, and categories.

Basic Flow of Events:

The system presents the user with an "Add Record" form
The user enters information: Amount, Date, Category, Type (Expense or Income)
Description (optional) [E1]. The user submits the form. The system validates the input data.
The system saves the new record to the database, associating it with the user's account.
The system displays a success message and updates the expense/income overview.

Exceptional Flow of Events:

[E1] If any required field is missing or invalid, the system displays an appropriate error message (e.g., "Amount must be a number," "Date must be in the format YYYY-MM-DD," "Category is required"). The user is prompted to correct the invalid entries.

4.3.7 Edit Expense/Income Records

Initiator: End User

Description: This use case allows the user to modify the information of an existing expense or income record.

Basic Flow of Events:

The system presents the user with a list of existing expense/income records. The user selects a record to edit. The system displays an "Edit Record" form with the selected record's current data. The user modifies the desired fields [E1]. The user submits the form.
The system validates the input data. The system updates the record in the database with the modified information.

Exceptional Flow of Events:

[E1] If any required field is missing or invalid, the system displays an appropriate error message. The user is prompted to correct the invalid entries.

4.3.8 Delete Expense/Income Records

Initiator: End User

Description: This use case allows the user to remove an existing expense or income record from their personal planner.

Basic Flow of Events:

The system presents the user with a list of existing expense/income records.
The user selects a record to delete.
The system displays a confirmation dialog asking the user to confirm the deletion.
The user confirms the deletion. The system deletes the record from the database.
The system displays a success message and updates the expense/income overview.

Exceptional Flow of Events:

[E1] If the user cancels the deletion in the confirmation dialog, the system returns to the record listing or view without deleting the record.

4.3.9 Review Expense/Income graphically

Initiator: End User

Description: This use case allows users to view a graphical representation of their expense and income records

Basic Flow of Events:

The user navigates to the "Expense/Income Overview" section.
The system retrieves the user's expense and income data from the database. The user selects the period of data to display. The system generates a graph representing the data [E1] [E2]. The system displays the graph to the user. The graph should be interactive, allowing the user to hover over elements (data points) to see more detailed information. The system should also display the total expense and income for the period.

Exceptional Flow of Events:

[E1] If the user has no expense/income data for the selected period, the system displays a

message indicating "No data available for the selected period" along with a placeholder graph or an empty chart.
[E2] If the database is unavailable or an error occurs during the retrieval of expense/income data, the system displays a general error message and informs the user that the overview could not be displayed.

4.3.10 Add travel plan

Initiator: End User

Description: This use case allows users to create a new travel plan in the system.

Basic Flow of Events:

The system presents the user with a "Create New travel plan " option.
The system prompts the user to enter the basic details of the itinerary: travel plan name. The system validates the input data. The system creates a new travel plan record in the database, associated with the user's account. The system displays the newly created travel plan with options to add travel details.

Exceptional Flow of Events:

[E1] If any required field is missing or invalid, the system displays an appropriate error message. The user is prompted to correct the invalid entries.

4.3.11 Edit travel plan

Initiator: End User

Description: This use case allows users to modify existing travel plan details.

Basic Flow of Events:

The system presents the user with the selected travel plan. The system displays an "Edit travel plan" with the travel plan current details. The user enters the details [E1]. The system validates the input data. The system updates the travel plan record in the database. The system displays a successful message.

Exceptional Flow of Events:

[E1] If any required field is missing or invalid, the system displays an appropriate error message. The user is prompted to correct the invalid entries.

4.3.12 Remove travel plan

Initiator: End User

Description: This use case allows users to delete a travel plan.

Basic Flow of Events:

The system presents the user with an option to remove the current travel plan. The system displays a confirmation dialog asking the user to confirm the deletion [E1]. The user confirms the deletion. The system deletes the travel plan record from the database. The system removes the travel plan from the list.

Exceptional Flow of Events:

[E1] If the user cancels the deletion in the confirmation dialog, the system returns to the itinerary list without deleting the itinerary.

4.3.13 Add Itinerary

Initiator: End User

Description: This case allows users to add an itinerary to their travel plan.

Basic Flow of Events:

The user selects an travel plan. The user navigates to the "Add Itinerary" section.
The system prompts the user to enter travel details: flights, accommodation, activities, and transportation to their travel plan [E1]. The system validates the input data. The system adds the itinerary into the travel plan in the database.
The system displays the updated travel plan.

Exceptional Flow of Events:

[E1] If any required field is missing or invalid, the system displays an error message.

4.3.14 Daily View

Initiator: End User

Description: This use case allows users to view their itineraries in a calendar format.

Basic Flow of Events:

The user navigates to the Daily View. The system displays itineraries at a certain date within the travel.
The system displays itinerary (flights, accommodations, activities) on the UI component on the appropriate dates [E1] [E2].

Exceptional Flow of Events:

[E1] If the travel plan has no itineraries, the component displays a message indicating "No itineraries available."
[E2] If the user has not specified a start and end date for the travel plan, the component displays a message indicating “Enable Daily View by specifying start and end date.”

4.3.15 Share travel plan

Initiator: End User

Description: This use allows users to share their travel plan with others via email.

Basic Flow of Events:

Description: The user selects a travel plan. The user chooses to share the travel plan. The URL of the current travel plan is copied onto clipboard.

4.3.16 Collaborative Editing

Initiator: End User

Description: This use case allows a user to invite another user to collaboratively edit their travel plan

Basic Flow of Events:

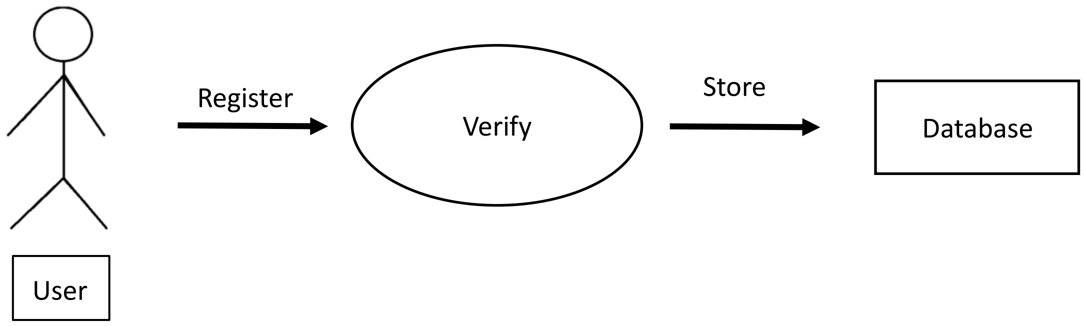
The user selects a travel plan. The user chooses to invite collaborators.
The system prompts the user to enter the email address of the user to invite.
The user enters the email address. The system sends an invitation email to the specified user, granting them editing permissions on the itinerary.
The system adds the invited user to the list of collaborators for the itinerary.

Exceptional Flow of Events:

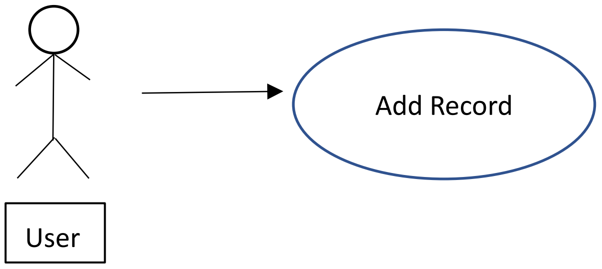
[E1] If the email address is invalid, the system displays an error message.
[E2] If the email service is unavailable, the system displays an error message.
[E3] If the user being invited already has editing rights or is the owner, the system displays a proper warning.

4.4 Use Case Diagram

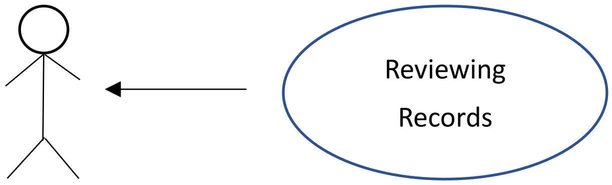
4.4.1 User Management



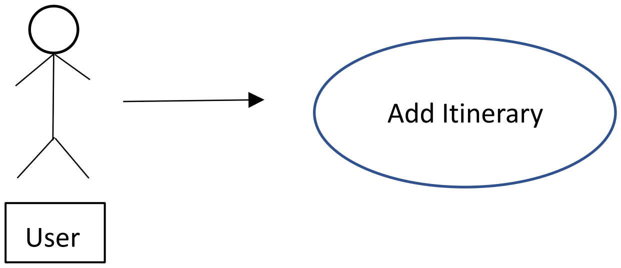
4.4.2 Add Records



4.4.3 Viewing Records



4.4.4 Add itinerary


Add Itinerary
User