Travelopedia Project Document

Preface

Project Title: Travelopedia - An app for backpackers/travellers
Version Summary: Version 1.0 - Initial project document

Table of Contents

1. Introduction

Travelopedia is a comprehensive travel web app designed for backpackers and aspiring travellers to document their journeys and get personalised travel recommendations. The app allows users to track their travel history through an interactive timeline, showcase their top travel photos, and receive AI-driven suggestions for future destinations. It also provides valuable resources such as hotel, restaurant, and tourist spot recommendations.

Travelopedia aims to enhance the travel experience by incorporating travel statistics and user-generated content. The app will benefit travel enthusiasts by offering a unified platform where users can log their travel history, compare travel stats with global trends, share their top photos as memoirs, and receive personalised recommendations for future travel plans.

Travel-related APIs, such as getting data about hotels, restaurants, tourist spots, etc., will be scraped from third-party travel websites like Travel Advisor, Kayak, and so on. This will help us get and compare data and provide the user with the best available option while calculating budget and providing such other services.

2. Glossary

- Backpacker: A traveller who explores with minimal luggage and often on a budget
- Timeline: A chronological representation of travel events
- AI: Artificial Intelligence
- ML: Machine Learning
- Itinerary: A planned route or journey
- Wishlist: A collection of desired future travel destinations
- Full-stack: Encompassing both front-end and back-end development
- User session: A period of user interaction with the web application
- Microservices: An architectural style that structures an application as a collection of loosely coupled services

3. User Requirements

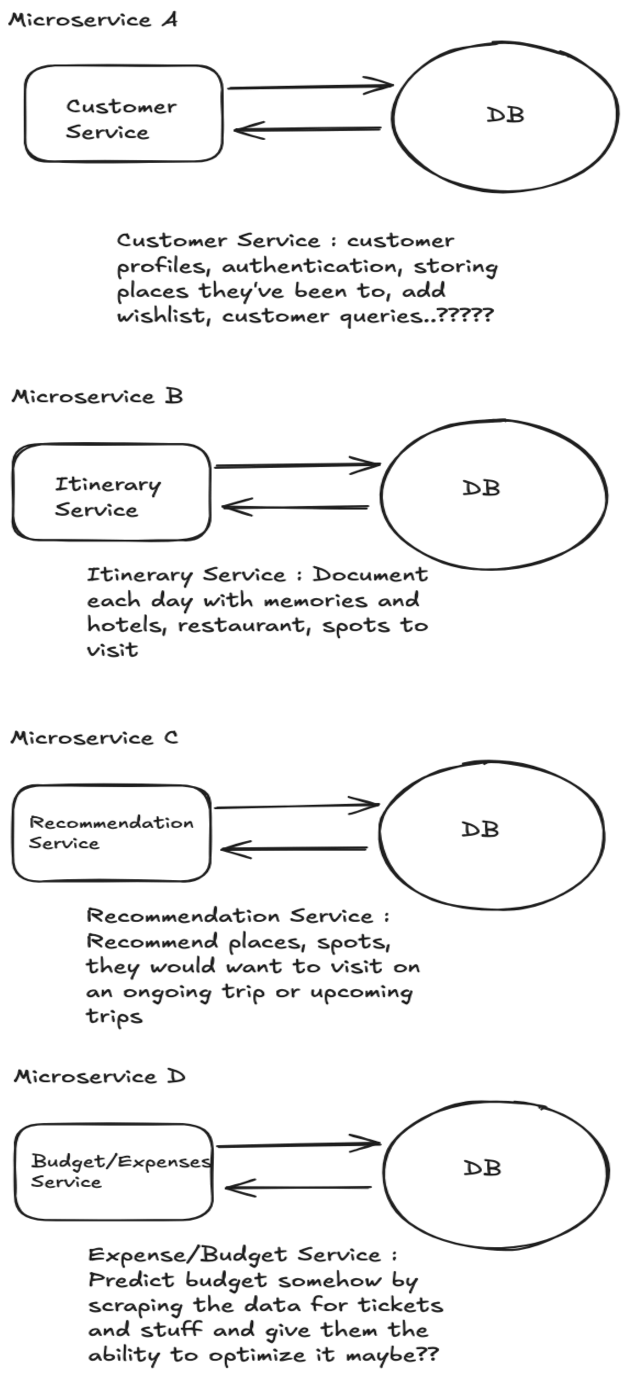
1. Users shall be able to create an account and log in
2. Users shall be able to input places they've visited
3. Users shall be able to view their travel history on an interactive timeline
4. Users shall be able to upload and showcase their top travel photos
5. Users shall be able to receive personalised travel recommendations
6. Users shall be able to create wish lists for future trips
7. Users shall be able to view their total traveled distance
8. Users shall be able to compare their travel stats with global trends
9. Users shall be able to access recommendations for hotels, restaurants, and tourist spots
10. Users shall be able to create and plan itineraries
11. Users shall be able to view their travel percentile compared to the global population
12. Users shall be able to share their travel experiences with other users
13. Users shall be able to create a rough estimate for their itineraries.

4. System Requirements

1. The system shall design a scalable and maintainable microservices architecture for different services
2. The system shall use a distributed caching mechanism to improve read performance and reduce database load (faster read access).
3. The system shall integrate with a machine learning model for personalised recommendations

4. The system shall have a common git-local-config-server for common configurations
5. The system shall implement auto-scaling to handle varying loads efficiently
6. The system shall utilise containerization technologies like Docker for service isolation and deployment
7. The system shall employ a service discovery and load-balancing mechanism to distribute traffic across multiple instances of microservices
8. The system shall support independent deployment and scaling of individual microservices

5. System Architecture



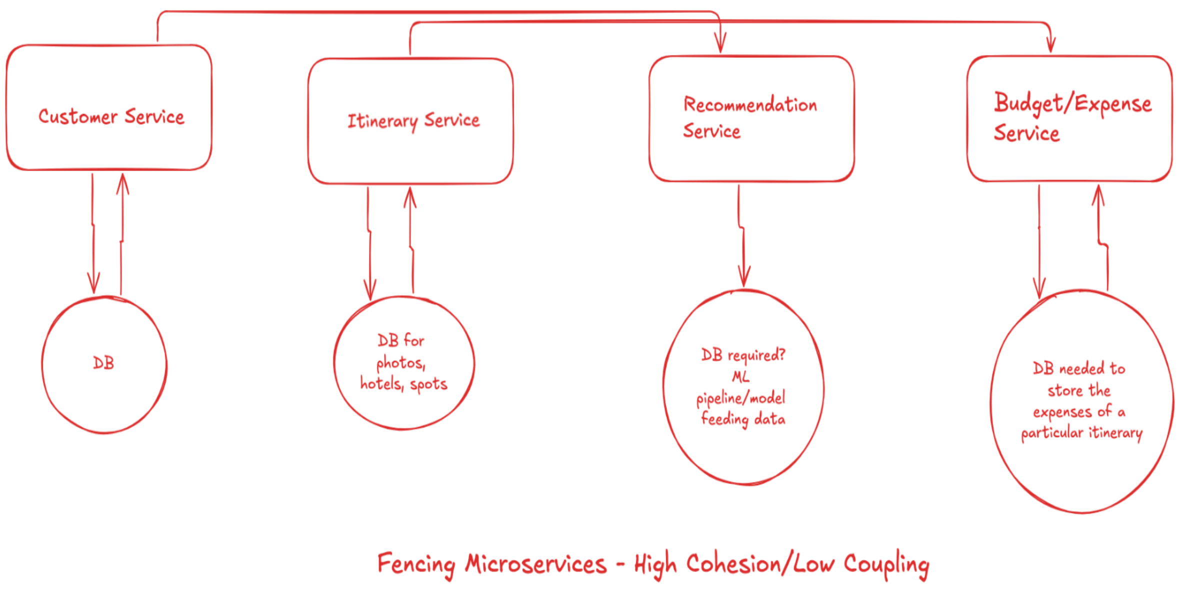
Customer Service : customer profiles, authentication, storing places they've been to, add wishlist, customer queries..?????

Itinerary Service : Document each day with memories and hotels, restaurant, spots to visit

Recommendation Service : Recommend places, spots, they would want to visit on an ongoing trip or upcoming trips

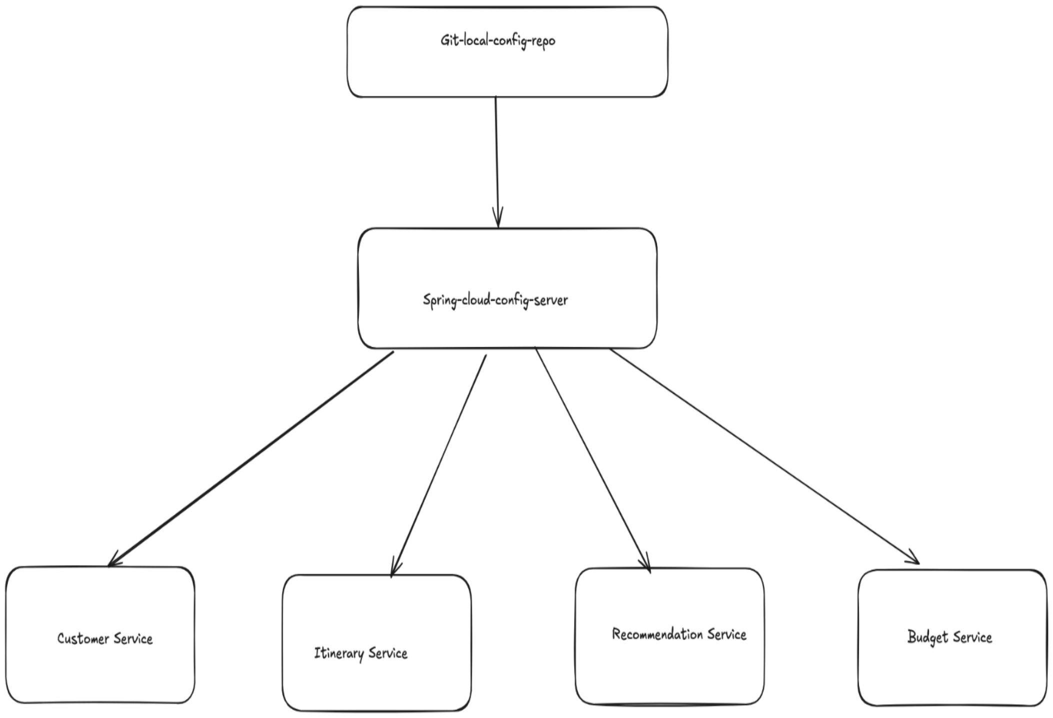
Expense/Budget Service : Predict budget somehow by scraping the data for tickets and stuff and give them the ability to optimize it maybe??

Fencing Microservices : Image 1

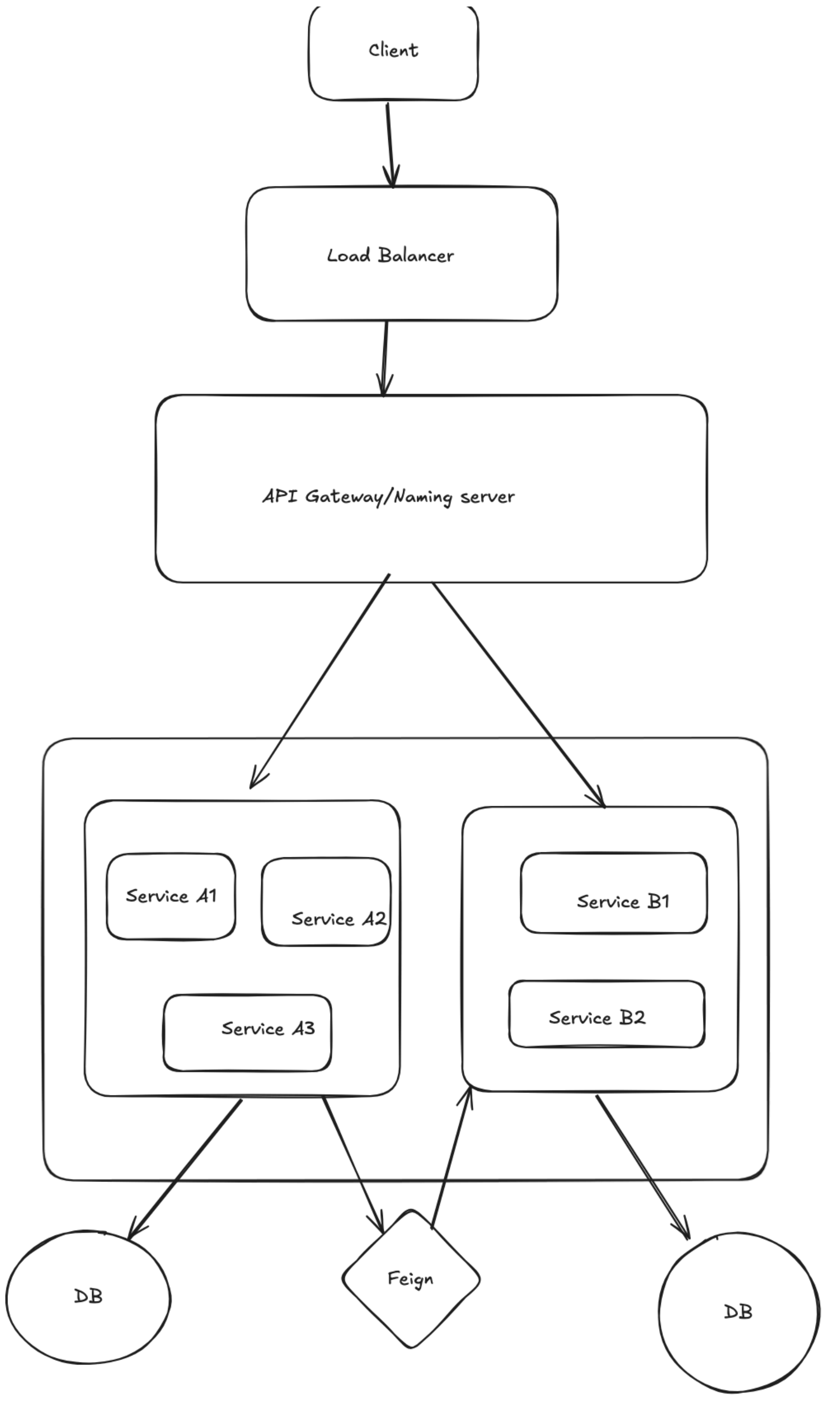

Customer Service
Itinerary Service
Recommendation Service
Budget/Expense Service
DB
DB for photos, hotels, spots
DB required? ML pipeline/model feeding data
DB needed to store the expenses of a particular itinerary
Fencing Microservices - High Cohesion/Low Coupling

Points to note :-
1. Recommendation Service will talk to the Customer Service to get the places for each user.
2. Budget/Expense Service will talk to the Itinerary Service to get the itinerary for a particular trip.

Fencing Microservices: Image 2

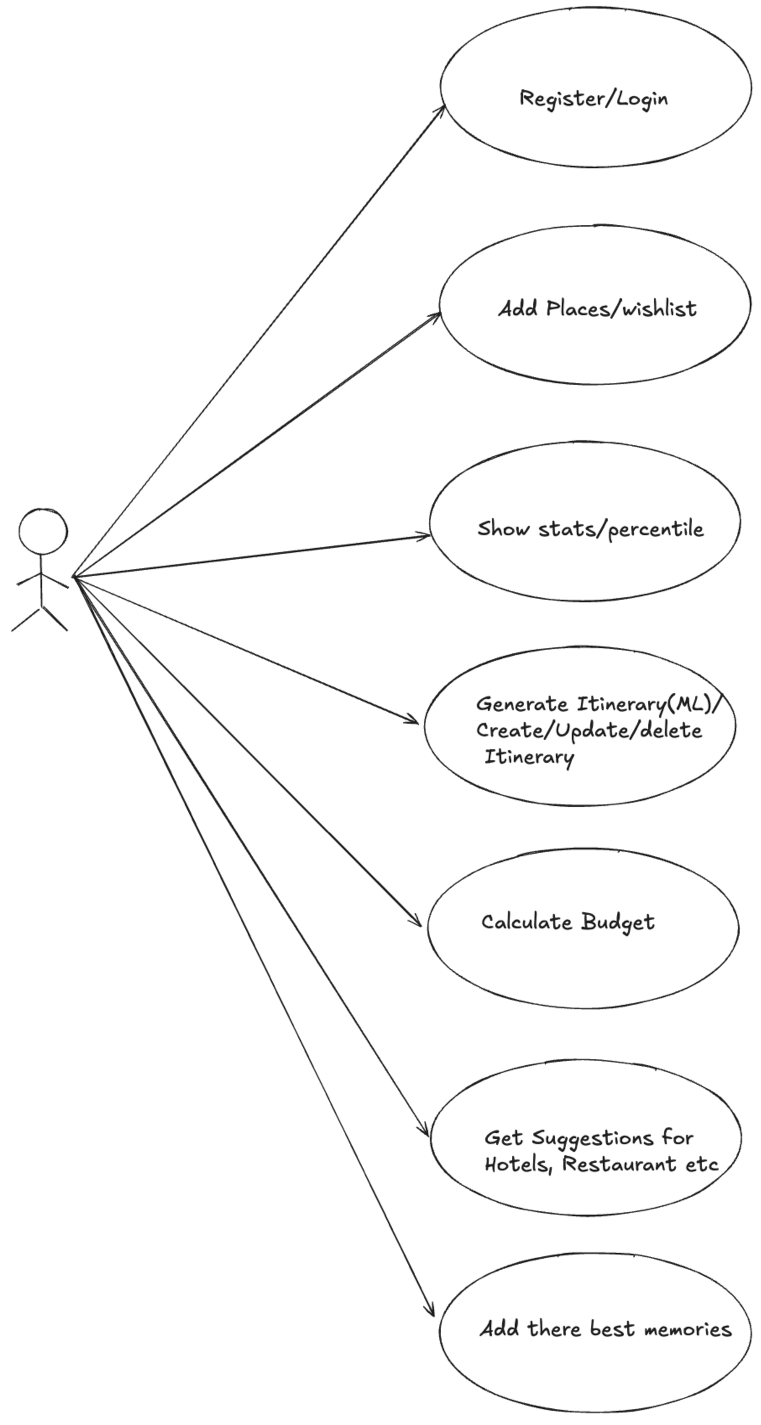


Common Config Repo architecture

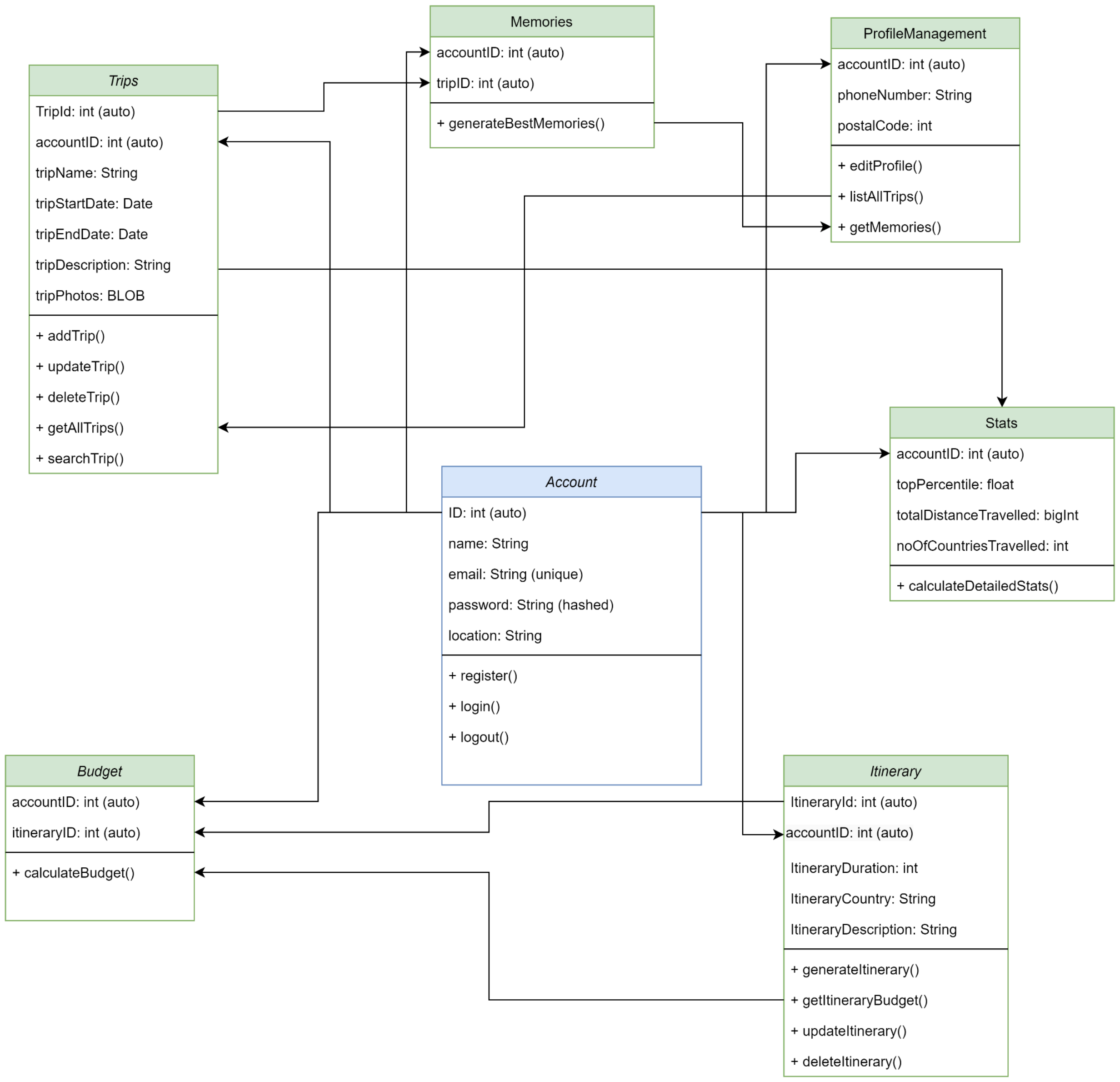


Communication and load balancing of the services

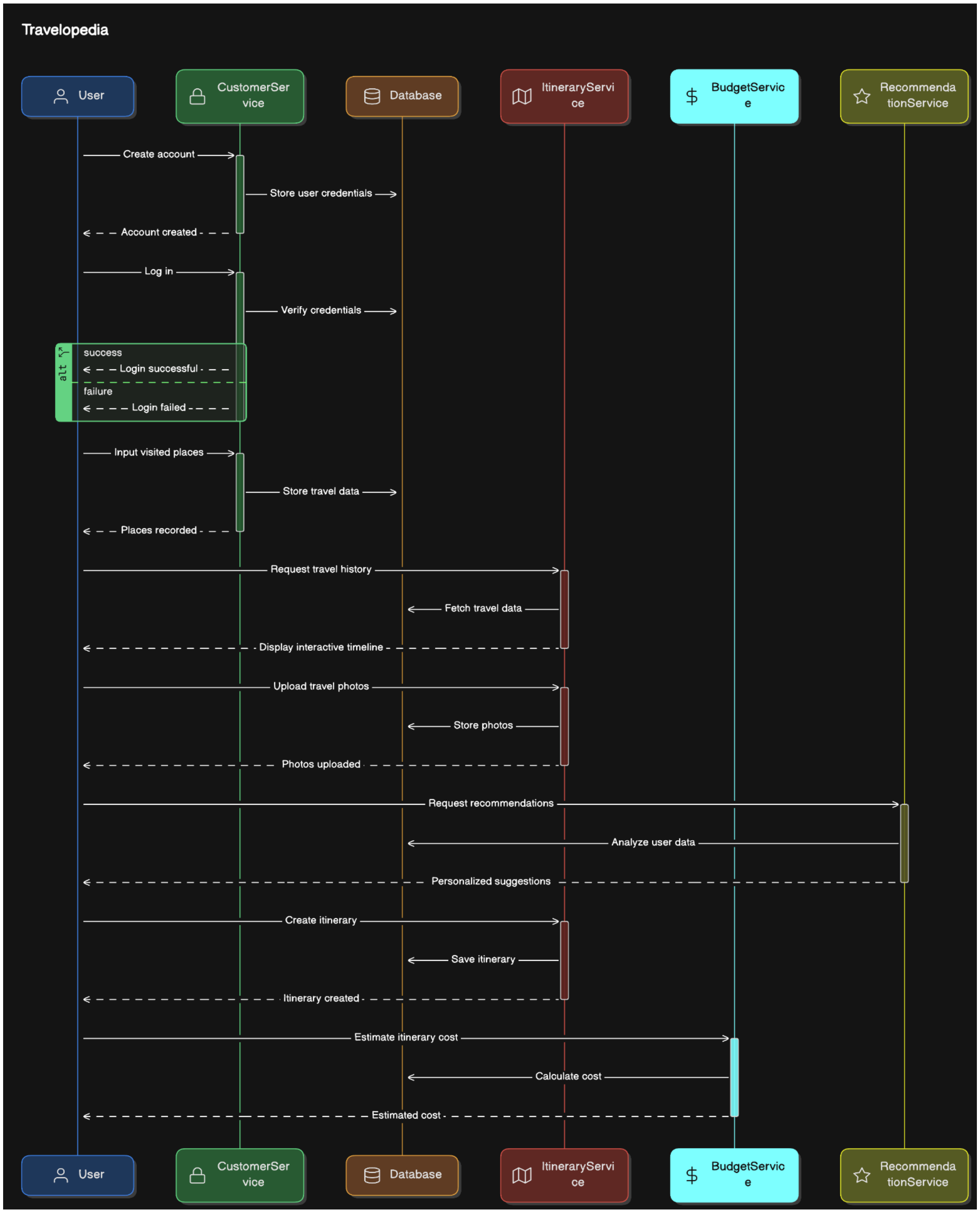
6. System Models



Use Case Diagram

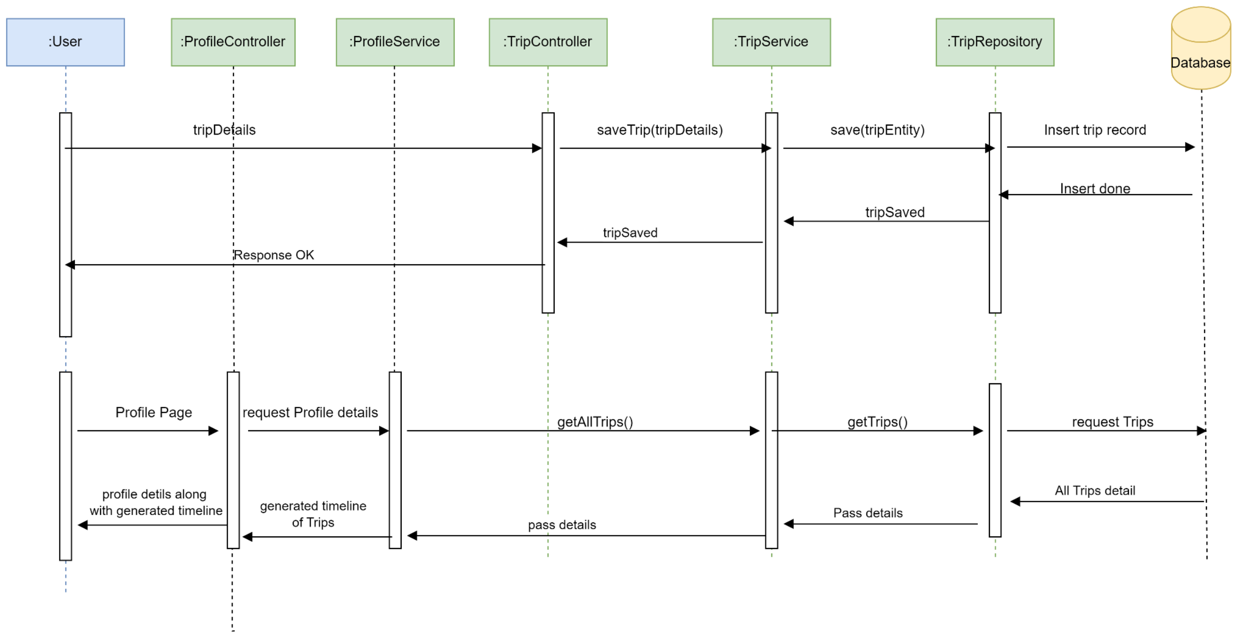


Class Diagram



1. Sequence Diagram

Below is Sequence diagram for saving trips and getting generated timeline of trips on profile page



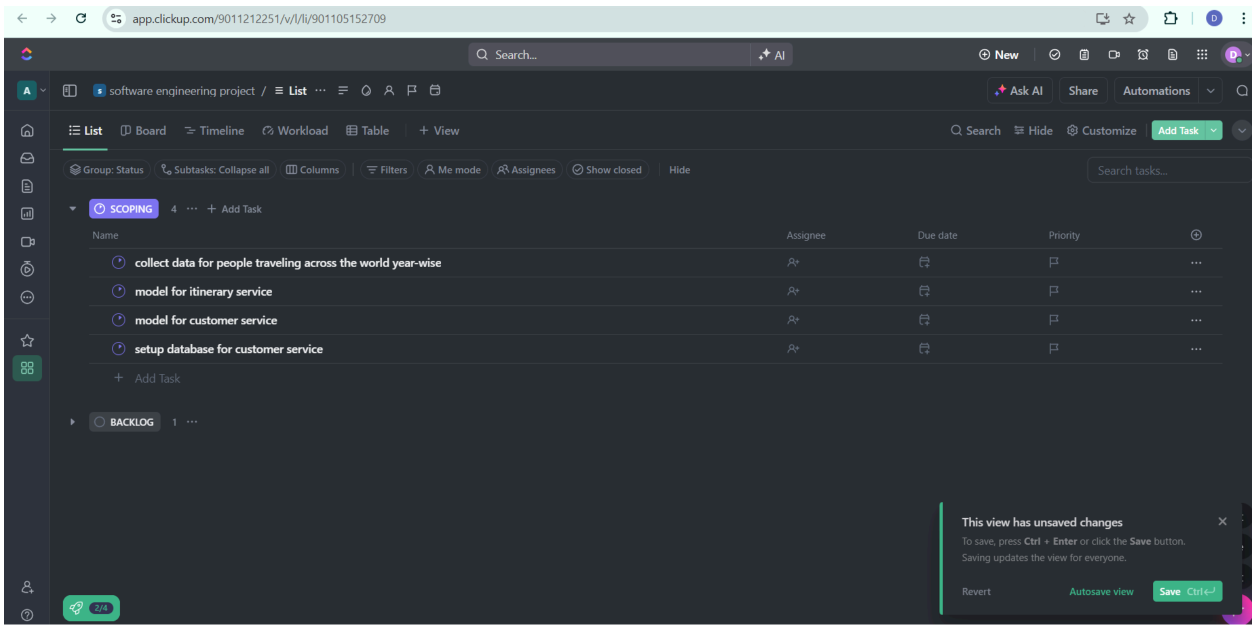
2. Sequence Diagram

7. Contribution And Other Links

Member	Name	Contribution
Member 1	Abhishek Jain	33.33%
Member 2	Alisha Walunj	33.33%
Member 3	Deep Shah	33.33%

WhiteBoarding and Brainstorming :-
https://excalidraw.com/#room=a572a50308d610ae4e86,PCczr8TE4Y9-VIDuXk0N4Q

Sprint Productivity Tool :-
https://app.clickup.com/9011212251/v/s/90111456411



SCOPING 4

Name	Assignee	Due date	Priority
collect data for people traveling across the world year-wise			
model for itinerary service			
model for customer service			
setup database for customer service			

This view has unsaved changes
To save, press Ctrl + Enter or click the Save button.
Saving updates the view for everyone.
Revert · Autosave view · Save Ctrl