

Proposal for Orbital 25

Team Name: TravelGo!

Proposed Level of Achievement: Apollo 11

Motivation

Traveling is an exciting and enriching experience, but the process of planning a trip can often be overwhelming and stressful. As young Singaporeans who love to travel frequently, we understand the challenges of juggling multiple tasks. Amidst researching destinations, creating itineraries, tracking budgets, and coordinating with travel companions, we must ensure everything is organised and accessible. The current tools available are either too fragmented, requiring users to switch between multiple platforms, or lack the visualisation and real-time updates to provide a seamless planning experience.

This drives across a problem, which develops into our problem statement: How might we empower travellers to plan their trips confidently with ease, so that their excitement for a trip will not be overwhelmed by a complex and inefficient trip-planning process?

Aim

To address these pain points, we aim to develop a comprehensive web application that simplifies and centralises the trip-planning process. We seek to not only save time but also provide convenience to users when they plan for their trips together.

User Stories

1. As a frequent traveller who wants to keep my travel plans secure and personalized, I want to be able to create an account and log in to access my saved itineraries and preferences.
2. As an excited traveller who wants to plan my activities efficiently, I want to be able to add destinations, dates, activities and notes to build a detailed itinerary.
3. As a careful traveller who wishes to prepare for my trip based on the destination's climate, I want to be able to view official weather forecasts for my travel dates.
4. As a budget-conscious traveller who wishes to not overspend on my trip, I want to be able to input my budget and track my spending throughout the trip.
5. As a visual planner who wants to visualise the proximity of attractions at my destination, I want to be able to view and drop pins of the locations of my planned visits on an interactive map.
6. As a group traveller travelling with my friends, I want to be able to share my itinerary and collaborate with my travel mates in real-time.

Features

1. User Authentication (core): Users can create accounts and sign up/ log in to access their personalised travel plans and preferences.
2. Itinerary Builder (core): Users can add destinations, dates, activities, notes to create and manage detailed travel itineraries.
3. Weather Forecast (core): Users can see an accurate and official weather forecast for their destination during their travel dates.
4. Budget Planner (extension): Users can input their budget and track their expenses to ensure they stay within their financial limits.

5. Interactive Map (extension): Users can save the locations of places they planned to visit and visualise the locations' proximity geographically.
6. Collaboration Planning (extension): Users can share their itinerary with their travel mates which allows for collaboration in real-time.

Timeline

1. Milestone 1 - Technical proof of concept (i.e., a minimal working system with both the frontend and the backend integrated for a very simple feature)
 - a. User Authentication done to level 2 - Users can login and save their progress.
 - b. Itinerary Builder done to level 1 - Users can create a simple travel itinerary.
 - c. Weather Forecast done to level 1 - Users can view weather information presented on the application.
2. Milestone 2 - Prototype (i.e., a working system with the core features)
 - a. Itinerary Builder done to level 2 - Users can create a travel itinerary with destinations, dates, activities and notes.
 - b. Weather forecast done to level 2 - Users can view weather information processed to be of relevance to their trip.
 - c. Budget Planner done to level 1 - Users can input their budget split by trip or daily.
 - d. Commence initial round of usability testing.
3. Milestone 3 - Extended system (i.e., a working system with both the core + extension features)
 - a. Budget Planner done to level 2 - Users get to track expenses and view their remaining budget live.
 - b. Interactive Map done to level 2 - Users can save destination's locations and view them on a visualisation map.
 - c. Collaboration Planning done to level 1 - Users can access the same instance of itinerary using a unique access code, progress is saved.
 - d. Commence final round of usability testing.

Tech Stack

Our team intends to use a widely used tech stack: MERN stack.

Frontend

1. ReactJS
2. Tailwind CSS
3. Axios

Backend

4. NodeJS
5. ExpressJS
6. MongoDB

API & Third Party Integrations

7. OpenWeatherMap API
8. Google Maps API

Authentications

9. PassportJS

Version Control

10. GitHub

Qualifications

Our team has a strong foundation in Python, Java, and JavaScript, with hands-on experience in developing both backend and frontend solutions. We are proficient in Python to optimize backend algorithms for performance, and we've applied Java to build scalable applications with OOP principles. Our proficiency in Javascript enables us to develop dynamic and interactive

web applications, and we have experience working with ReactJS to design dashboards and landing pages.

Beyond coding, we have experience with SQL and NoSQL databases, using them to structure and query data efficiently. We also follow collaborative workflows using Git for version control, ensuring effective team coordination. While we have not yet worked on formal, enterprise-level software projects, we see this project as an opportunity to apply our knowledge in a structured, real-world setting. Through iterative usability testing, we aim to refine our development practices, implement scalable solutions, and contribute to an impactful application.

Software Engineering

Keep it simple, stupid (KISS) Principle will be used to minimise complexity as much as possible to avoid confusion and reduce likelihood of errors, making maintenance and further implementations easier. This prioritises simplicity which creates a clean and intuitive software.

Don't Repeat Yourself (DRY) Principle will be used to improve maintainability through designing reusable components and functions. This avoids code duplication, enabling more straightforward future updates.

AGILE Software methodology will be used when developing the application. By breaking down the project into smaller manageable tasks, we could work to deliver features one at a time. This iterative approach allows for more adaptability, so changes and errors that may arise when we combine different features together can be resolved effectively.

Github will be used for effective code management to maintain version control. This includes utilising branches for feature development, conducting collaborative code reviews and managing changes through push/pull requests.

Team Members

1. Teoh Hao Yi (SoC): Year 1 Computer Science
2. Lee Yi Heng (CHS): Year 1 Mathematics & Computer Science (Double Major)