Tripinary - Milestone 1 Report

John Camino, Renz Gabrinao, Beverly Yen, Lilian Pham

https://github.com/CMPT-276-SUMMER-2025/final-project-02-peak.git

Project planning report: (finalize 2 APIs/description of how its used)

I - OpenAI: GPT-4o-mini

The OpenAI API provides access to AI models like GPT-4 for tasks, and developers are able to integrate this API to allow conversational AI and content creation into applications. It allows developers to generate text based on user input.

Features:

1. Timestamped Itinerary List → Creates a personalized travel itinerary based on the user's location preferences and trip duration.
    a. Benefit: Users will save time and effort in planning as this allows them to input their destination and trip length, and this will create a schedule for them. This also allows them to maximize their leisure time since it saves them having to research about each place.
2. Activity Suggestions → Recommends specific activities and places (ex: restaurants, hikes, etc.) depending on user inputs and allows users to add this to their itinerary.
    a. Benefit: This allows users to have personalized recommendations during their trips, it helps users to discover unique activities that fit into their specific lifestyle, and makes sure that the trip is tailored to their preferences.
3. Itinerary Regeneration → Based on the itinerary that is created from user inputs, this feature would allow users to press a regeneration button that gives them a new itinerary.
    a. Benefit: This will allow users to have freedom over their travel plans, and users are not stuck with the itinerary that they are generated each time. This gives them more agency in planning their trip to their specific needs and creates a better overall travel experience

II - Google Maps & Places API

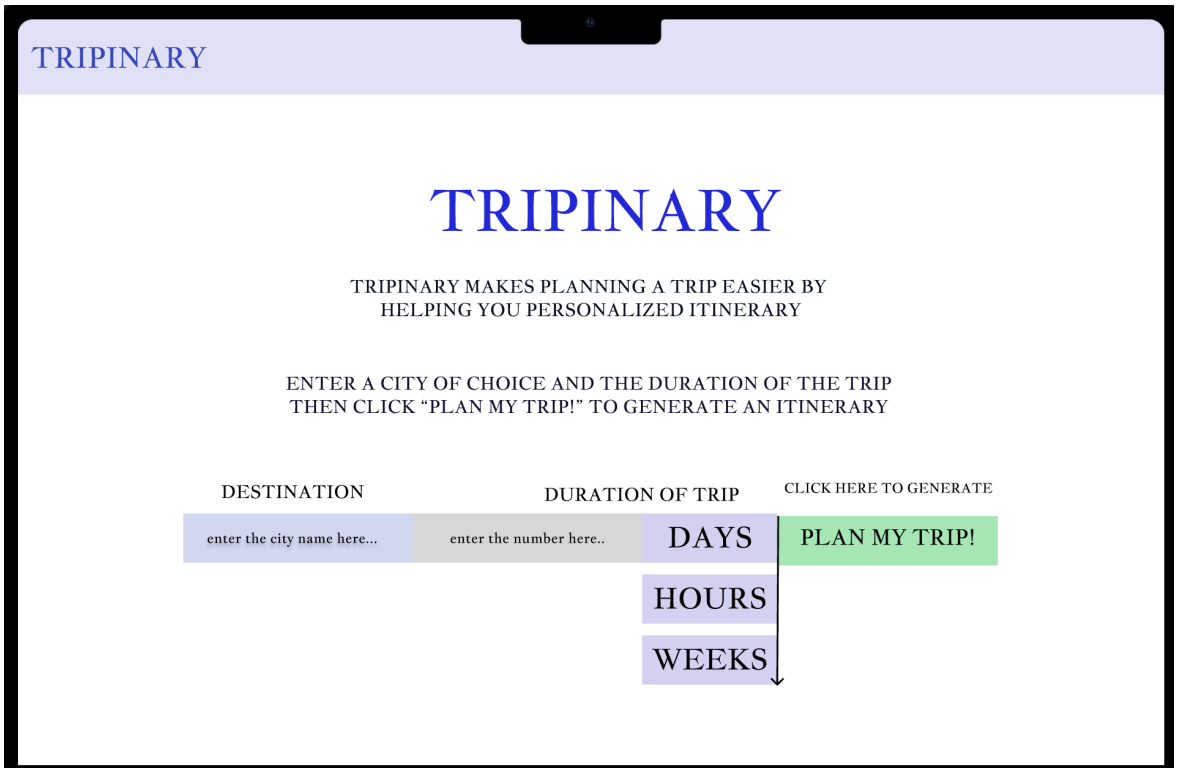
The Google Maps Places API allows the application to search and retrieve information about places using location data. It allows developers to embed Google Maps on web pages, offering geographical data and displaying user reviews.

Features:

1. Place Search → Based on the map of the itinerary that is generated, this feature allows users to locate and search for specific places (i.e. restaurants, hikes, etc.) and display a map view of its location.
    a. Benefit: This will allow users to quickly find specific activities that they are interested in based on their preferences. This feature will make it easier for them to discover specific information about their planned destinations instead of having to do external searches about each one separately.
2. Place Review → Shows the latest five reviews of specific activities on the user's personalized itinerary. The place review will include star rating (out of 5), description, and name of the reviewer.
    a. Benefit: This allows the user to see the most up-to-date ratings for that place. Maybe there are seasonal activities that a person can do while they visit the spot.

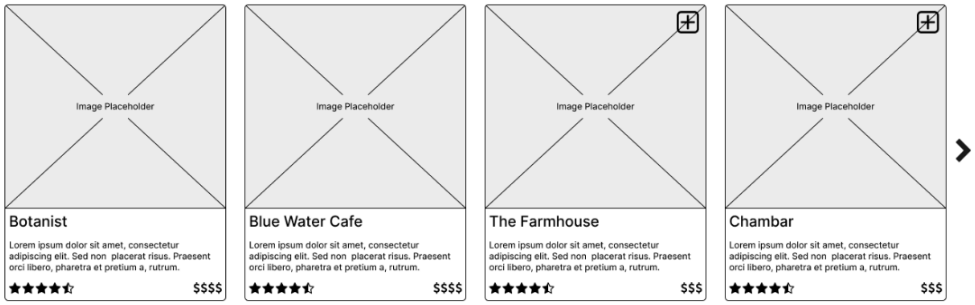
3. Map Display → This feature displays an interactive map of where each individual activity is located.
    a. Benefit: This will help users who prefer to have visual representations of their itinerary. This will allow users to also gauge distances between their activities and plan for efficient routes they should take.

Convert the low-fidelity storyboard to a mid-fidelity prototype

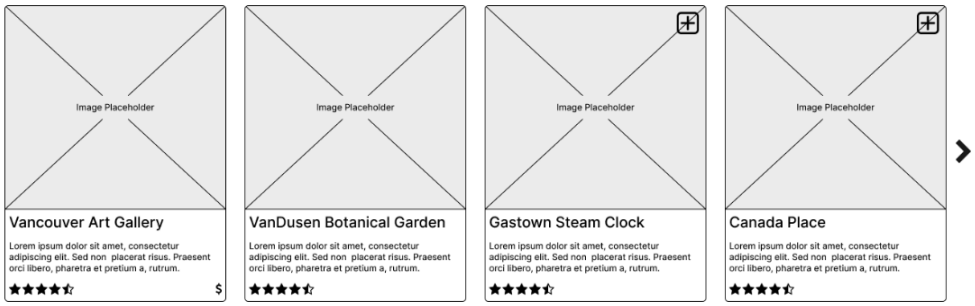


Plan a trip in **Vancouver**

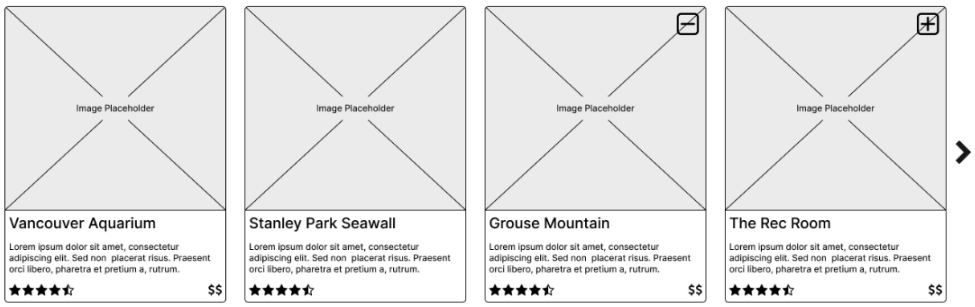
Food & Drinks:



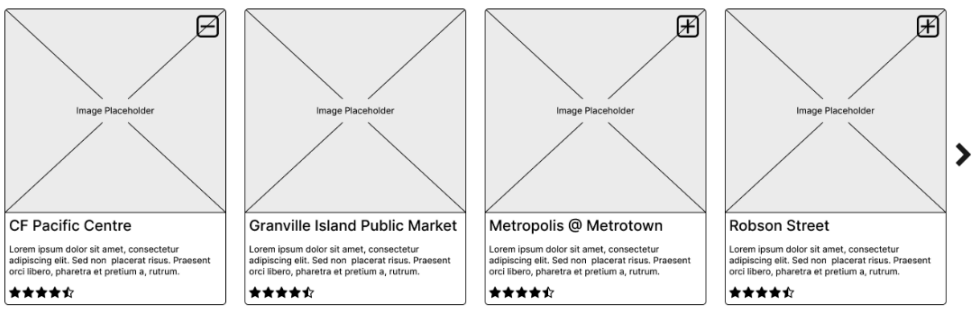
Attractions & Sightseeing:



Activities & Recreation:



Shopping:



6 Destinations Selected Generate Itinerary

ITINERARY

Trip to... VANCOUVER

DAY 1

MORNING
(06.00 - 07.00)
Breakfast at Jam Cafe

AFTERNOON
(15.00 - 17.00)
Explore Harbour front Centre

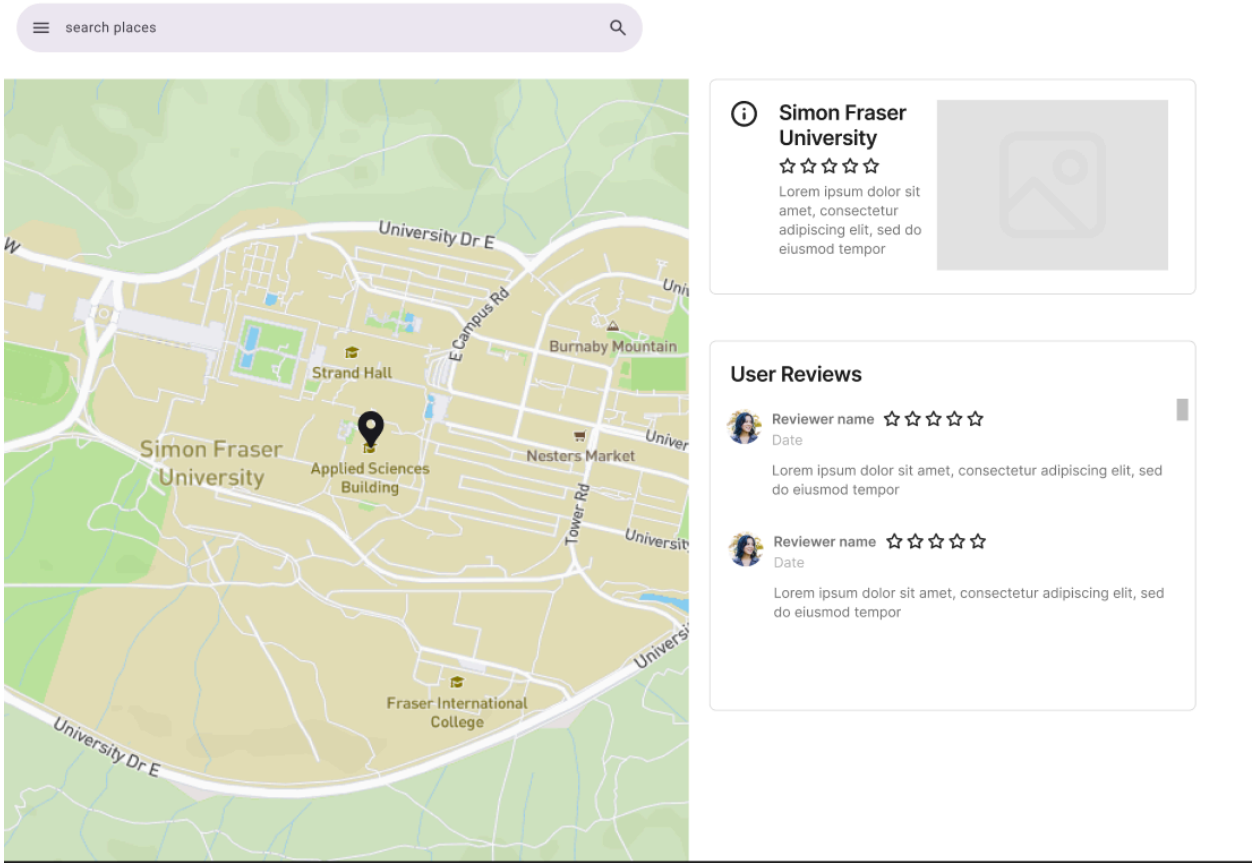
EVENING
(06.00 - 09.00)
Dinner at Cactus Club Cafe

DAY 2

MORNING
(06.00 - 09.00)
Breakfast at OEB Breakfast Co.

AFTERNOON
(15.00 - 17.00)
Simon Fraser University

REGENERATE



Choose a SDLC model

We chose to use the Agile SDLC framework because it allows for a more iterative and flexible project, which is ideal for our program. Agile would enable us to break our big project into more manageable sprints that would allow us to deliver features incrementally while modifying the project based on our needs and the group's discretion. This helps ensure that we stay aligned and can quickly adapt if any of the requirements change or circumstances change. As well, we can use Agile alongside Scrum and Kanban frameworks in order to keep track of our issues and to set dedicated periods for when each feature needs to be implemented by. Overall, Agile would help our team stay more organized, communicate effectively and deliver a product with improvements as we go.

WBS

Level 1 :

1.0 Project Planning and Requirements

1.1 Define Scope and Objective

1.1.1 Define Target User groups and use cases

- 1.1.2 Confirm Project deliverables and constraints
- 1.2 Requirements Gathering
- 1.3 Selecting Tech Stack

2.0 UI/UX Design
- 2.1 Wireframing
  - 2.1.1 Low-Fidelity Wireframes
  - 2.1.2 DFD diagrams
- 2.2 Visual Design
  - 2.2.1 High-Fidelity Wireframes
  - 2.2.2 Design Review

3.0 Frontend Development
- 3.1 Project Setup
  - 3.1.1 Setup Frontend Environment
- 3.2 Feature Implementation
  - 3.2.1 User Input Form (City, Duration, etc)
  - 3.2.2 Activity Carousel (restaurants, events to recommend)
  - 3.2.3 Itinerary Timeline View
  - 3.2.4 Prompt box with regenerate button
- 3.3 Map and Review Integration
  - 3.3.1 Slide-out map view for selected activity
  - 3.3.2 Display last 5 reviews from API
  - 3.3.3 Display place descriptions and image from API

4.0 API Setup
- 4.1 Planning and researching API
  - 4.1.1 Research Google Maps API capabilities (e.g., Places API)
  - 4.1.2 Research OpenAI API for itinerary generation
  - 4.1.3 Verify API compatibility with project
- 4.2 Environment set up
  - 4.2.1 Configure API keys, add required libraries
  - 4.2.2 Set up local Github repositories
  - 4.2.3 Configure secure storage for API keys
- 4.3 Implement required endpoints or data fetching
  - 4.3.1 Add error handling
- 4.4 Handle API responses (parse and store/process)
  - 4.4.1 Allow users to regenerate parts of the itinerary

5.0 Testing and QA
- 5.1 Peer review each group member’s code
  - 5.1.1 Provide feedback and offer assistance (if able)
- 5.2 Merge final changes to the main branch

5.3 Unit and Integration Tests
    5.3.1 Frontend component tests
    5.3.2 API logic tests
5.4 End-to-End Testing
    5.4.1 Test itinerary generation
    5.4.2 Test map and slide-outs
5.5 Bug Fixing and Optimization
    5.5.1 Performance Tweaks
    5.5.2 UI/UX polishing for any layout

6.0 Deployment
6.1 Vercel
    6.1.1 Configure Vercel to connect to repository
6.2 Deploy Backend and AI API
    6.2.1 Configure environment variables
    6.2.2 Ensure API limits

7.0 Documentation
7.1 Document Progress of the Project
    7.1.1 Review weekly tasks, future plans, and struggles
7.2 Technical Documentation
    7.2.1 README file
7.3 Final Representation
    7.3.1 Demo Video

Project Schedule and Deadlines

Milestone I : Homepage + Activity List
Milestone II : Itinerary + Maps Panel
Milestone III : API Integration
Milestone IV : Testing
Milestone V : Deployment

Phase	Dates	Goals	Details and Deliverables
Planning and Initial Setup	July 7 - July 10	API Project Setup	- setup developer environment - review API documents - Setup API keys, environment and usage
Homepage + Activity	July 11 - 17	Homepage UI +	- Build out

List		Activity Carousel	homepage layout and styling - Create placeholder static activities list **Internal Deadline : July 15**
Polish #1	July 18 - July 20	Cleanup	- Fix any issues with UI components **MILESTONE I DUE**
Check-In with TA + initial AI implementation	July 21 - July 25	Midpoint Review Basic AI integration	- Demo homepage and activities list - Gather feedback and fix quick bugs/features - Engineer AI prompts and connect them to API to take in user input
Itinerary + Maps Panel	July 26 - Aug 2	Itinerary Page + Maps Slide Out	- Build itinerary UI - Add draggable/resizable map panel **Internal Deadline : July 30** **MILESTONE II DUE**
AI Integration + Place API Integration + Optimization	July 28 - Aug 2	Connect AI API to core features Implement Google Places API to enhance map panel functionality Optimize layout and UX	- Hook up AI API to activities list, itinerary generator - Handle AI response parsing - Embed Places API into Maps Slide-Out Panel (display address, ratings and additional details)

			MILESTONE III DUE
Testing Phase Begins	July 28 - August 5	Functional Testing User Flow Testing Manual Testing	- Implement unit tests, integration tests and real user testing for each feature **Internal Deadline : August 3** **MILESTONE IV DUE**
Deployment	August 4 - August 6	Delivery Day	Final Cleanup, Deployment and Demo **Internal Deadline : August 5** **MILESTONE V DUE**

Risk Assessment

- Include at least 5 low-risk, 5 medium-risk, and 5 high-risk issues (15 total) with mitigation strategies for each

Low risk:

- Addition of optional features
    - Mitigation strategy: Once the core function and features have been implemented before the discussed deadline, then we can start implementing these optional features.

- UI/UX inconsistencies/layout issues on certain screen sizes
    - Mitigation strategy: All group members will be working on the same page at the same time so that we can align the formatting with each other. Ensure that the team is using the same screen size when making design changes.

- Learning curve of using API's
    - Mitigation strategy: Group members who are not experienced with API implementation should watch some tutorials before starting the development phase.

- Web Hosting issues
    - Mitigation strategy : Use reliable hosting sites, e.g Vercel
    - Mitigation strategy : Monitor build logs and error reports after each deployment

- Code style inconsistencies
    - Mitigation strategy : Agree on the same naming conventions and indentation
    - Mitigation strategy : Setup a formatter (e.g Prettier)

Medium risk:

- Core functions/features have bugs
    - Mitigation strategy: testing, code reviews, peer testing, fallback implementations, will be dedicate enough time to testing and debugging
- Inaccurate information from APIs
    - Mitigation strategy: Add disclaimers for the dynamic information, such as hours of operation
- AI generated itineraries lack personal feel
    - Mitigation strategy: Continuously fine-tune prompts
    - Mitigation strategy: Implement feedback loop where users can regenerate suggestions
- Team member is behind schedule
    - Mitigation strategy: We will be meeting weekly with our group members and checking in on what each member has completed and if they are following the schedule. If they are behind or notice that another member is behind, ask if anyone else can help with their task.
- Overcomplication of features that may harm application usability
    - Mitigation strategy: include details on navigation in README, try to simplify features and create minimalist interface. For more advanced features, include instructions and error messages if user inputs are concerning. We can also create a step-by-step tutorial in the README file.

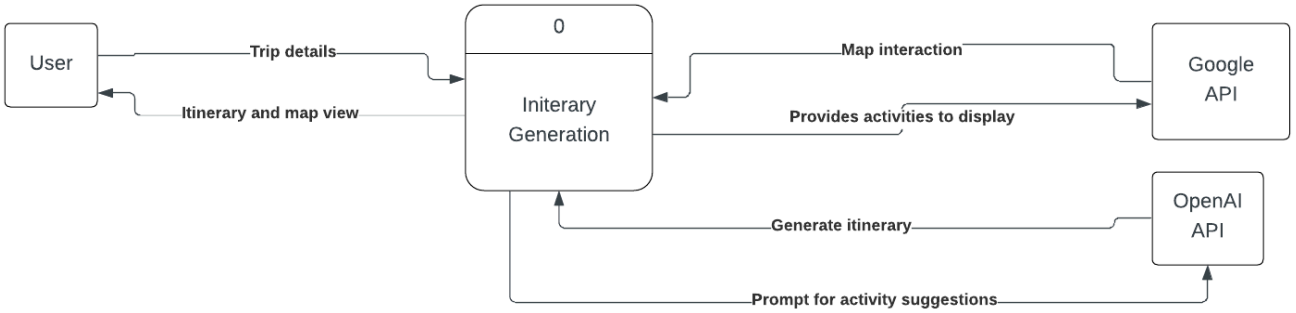
High Risk:

- API usage going over limit
    - Mitigation strategy: Monitor and set daily API usage and set up warnings before going over the limit
- Data/API Key leakage
    - Mitigation strategy: Store API keys in a .env file that’s not uploaded to GitHub
- Scope Creep
    - Mitigation strategy: Refer to the WBS and Project Schedule that was created in Milestone 1 to make sure we stay focused on our original goals. Worry about extra features later when core functions have been implemented.
- Suspension of API access if violation of any of the API terms of service
    - Mitigation strategy: Monitor API Terms of Services
    - Mitigation strategy: Keep logs for API calls to ensure compliance to terms of service
- AI Fails to Respond during Regeneration
    - Mitigation strategy: Add a retry and fallback (default suggestions)

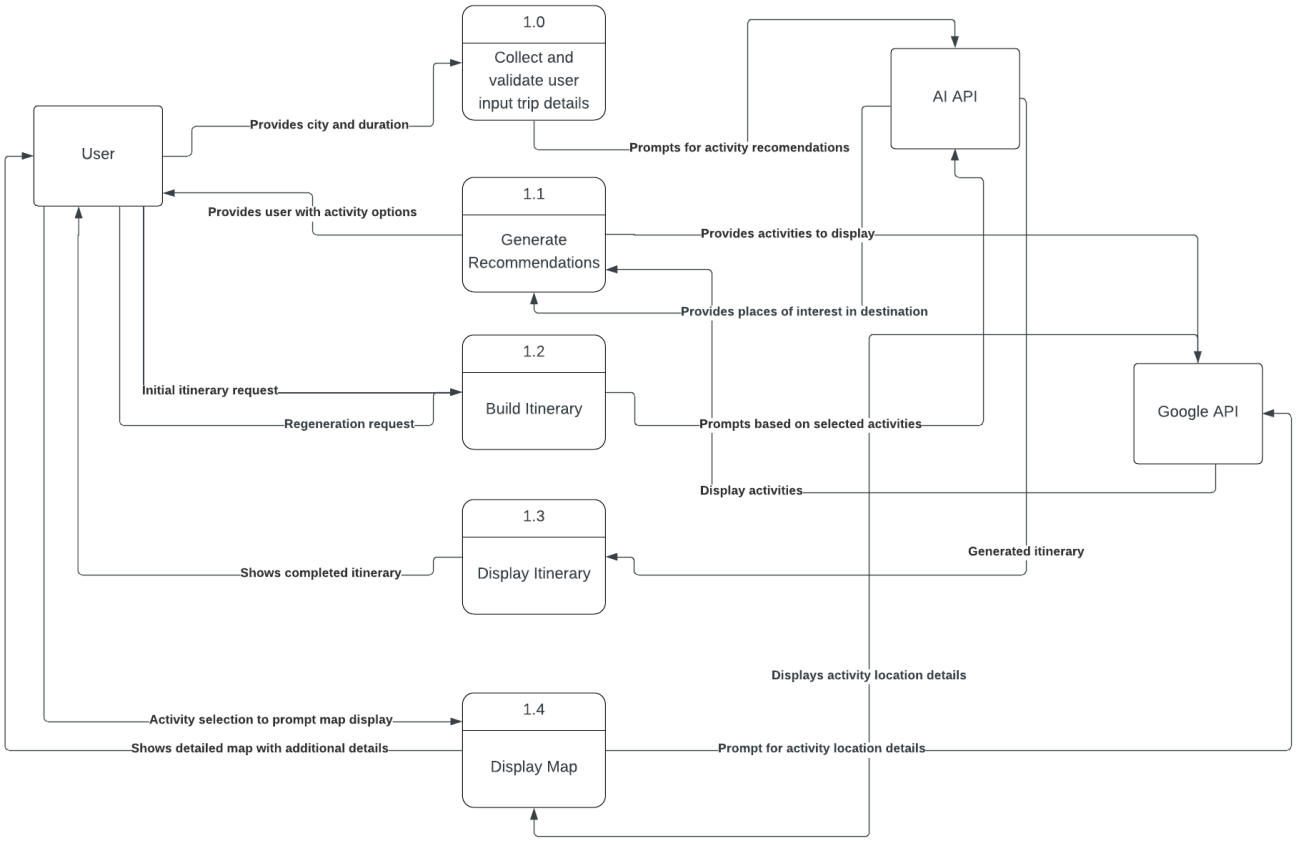
DFD Diagrams (Level 0 and 1)

Designed using LucidChart

Level 0:

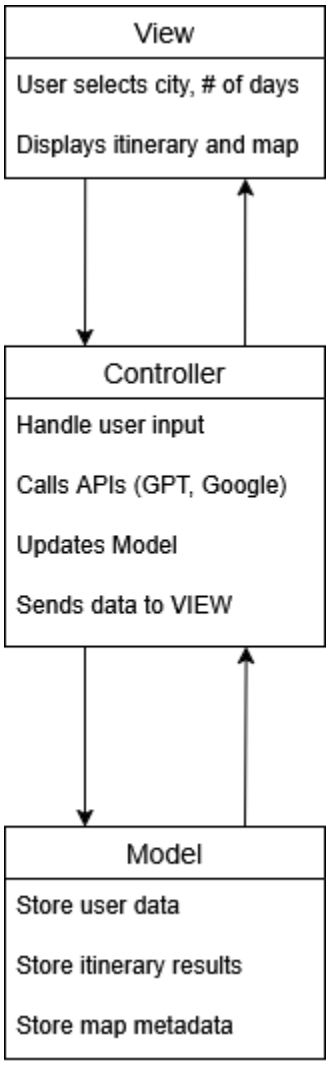


Level 1:



MVC Diagram

This should include a high-level overview of how the application will be structured and a description of each component



Appendix

Provide a list of all group members and their detailed contributions to the project

Group Members:	**Contributions to project:**
Entire Team:	● Reviewed and submitted the group contract

	• Researched different APIs, tech stacks, and brainstormed different ideas and features for our final project • Participated in many meetings to complete Milestone 0 and (currently) Milestone 1 • Created GitHub issues for each relevant task and a Kanban board to keep track of progress
Beverly Yen	• Completed detailed descriptions for all 12 features (primary and backup options). • Came up with and designed a hand-drawn project logo, and contributed to choosing the final logo. • Researched and decided on programs for web-hosting and front-end website design • Created low-fidelity wireframe for the map display on final project and later transformed it into a mid-fidelity wireframe using Figma • Created Project Schedule table, with deadlines assigned to specific tasks relating to the team's final project • Involved in covering and discussing all key WBS components (project planning, UI/UX design, frontend stack, API setup, testing, deployment, web hosting, documentation) • Created Level 0 & Level 1 Data Flow Diagrams
Renz Gabrinao	• Set up and organized the group's shared GitHub repository, creating specific directories and folders corresponding to the project starter template • Collaborated on writing a clear overview for the application's idea, purpose, problems addressed, and user group assistance, integrating it into the proposal document. • Created low-fidelity wireframe for the attraction/activities page on final project and later transformed it into a mid-fidelity wireframe using Figma • Contributed to identifying and discussing potential risks and issues for the final project in Milestone 1, as well as mitigation strategies • Involved in covering and discussing all WBS components (project planning, UI/UX design, frontend stack, API setup, testing, deployment, web hosting, documentation) • Created Level 0 & Level 1 Data Flow Diagrams • Created MVC diagram which is a high-level overview of how the application will be structured
John Camino	• Project Manager, responsible for keeping track of the team's GitHub issues and settling milestones with deadline • Started developing the low-fidelity wireframe and planned user interactions on the project website, detailing pages and feature placement. • Created low-fidelity wireframe for project's homepage and later transformed it into a mid-fidelity wireframe using Figma • Contributed to documenting the reasons for choosing Agile Scrum in the Milestone 1 working document • Brainstormed and documented items for the homepage footer • Involved in covering and discussing all WBS components (project planning, UI/UX design, frontend stack, API setup, testing, deployment, web hosting, documentation) • Created Level 0 & Level 1 Data Flow Diagrams

Lilian Pham	● Wrote all details for user stories (including demographics, background, characteristics, goals, challenges, behaviors, motivations) ● Worked on editing and finalizing the Milestone 0 proposal, reviewing sections like overview, user stories, user personas, feature descriptions, and tech stack to meet requirements. ● Created low-fidelity wireframe for project's generated itinerary page and later transformed it into a mid-fidelity wireframe using Figma ● Worked on structuring a Work Breakdown Structure (WBS), breaking the project into manageable tasks and defining deliverables/responsibilities ● Participated in the Risk Assessment and Project Schedule creation. ● Added Phase 1 (core) and Phase 2 (optional) features to the Milestone 0 proposal ● Involved in covering and discussing all WBS components (project planning, UI/UX design, frontend stack, API setup, testing, deployment, web hosting, documentation)

Provide a changelog table that includes any revisions since the proposal

Revision:	Date:
Low Fidelity wireframe → Mid-Fidelity using Figma	June 28th, 2025
Added Phase 1 & Phase 2 to Milestone 0, where Phase 2 contains optional features that will be implemented if time permits	June 27th, 2025