

SkyFare

Group-19 Leafs members:

Pal Rohit Tilva

Kaiwen Tao

Navjot Singh

Karnpreet Cheema

GitHub repository:

<https://github.com/CMPT-276-SPRING-2025/final-project-19-leafs>

APIs

1. Amadeus API Features (Free)

- Flight Offers Search: Search flights from over 400 airlines based on departure, destination, date, and budget.
- Flight Offers Price: Confirm final pricing (after taxes).
- Flight Status API: Provide real-time flight schedule updates, including departure/arrival times, terminal, and gate information.

Feature	API Used	Endpoint
Flight Search	Flight Offers Search	/v2/shopping/flight-offers
Final Pricing	Flight Offers Price	/v1/shopping/flight-offers/pricing
Live Flight Updates	Flight Status API	/v2/schedule/flights

2. Google Gemini API Features(Cost: \$0.10 per 1M input tokens, \$0.40 per 1M output tokens; free tier available)

- Chat Completion: AI chatbot answers flight-related queries.
- Context Memory: Helps users by remembering previous responses.
- Summarization: Provides clear explanations of ticket policies and booking details.

Feature	API Used	Endpoint
Chat Completion	Gemini Pro Model API	/v1/models/gemini-pro:generateContent
Context Memory	Gemini Pro Model API	/v1/models/gemini-pro:generateContent
Summarization	Gemini Pro Model API	/v1/models/gemini-pro:generateContent

Mid-fidelity Prototype

Figma link:

<https://www.figma.com/design/uGPJdRA1FtkMU2p4QRSxws/1?node-id=0-1&t=ejFvqw9nIpOoYxXA-1>

SDLC Model:

Chosen Model: Agile (Scrum)

Justification:

- Our project involves multiple integrations (APIs, chatbot, UI), which require flexibility.
- Agile allows iterative development, making it easy to adjust based on testing feedback.
- Small, manageable sprints will help track progress efficiently.

Work Breakdown Structure:

Phase 1: Project Setup

- Create & set up a GitHub repository.
- Set up a project board on GitHub (Kanban for task tracking).
- Define coding standards & branching strategy.

Phase 2: Chatbot & Frontend Development

Frontend Development (React.js + Tailwind CSS)

- Create SkyFare's Landing Page (user input for flight search).
- Develop Flight Results Page (display flight options).
- Build Flight Details Page (pricing, seat selection).
- Implement Saved Flights Page (user bookmarking).

AI Chatbot Development (Gemini API)

- Integrate Gemini API for flight queries.
- Implement SkyFare's basic chatbot UI.
- Enable chatbot context memory for past responses.
- Implement policy summarization feature (Gemini).

Phase 3: Flight Features Development

- Integrate Amadeus API (flight search, pricing, status).
- Develop Flight Offers Search feature.
- Build Flight Offers Pricing feature.
- Implement Flight Status Updates feature.

Phase 4: Testing & Deployment

- Conduct manual testing (test SkyFare's APIs and UI).
- Deploy SkyFare on Vercel.

Tasks are converted to GitHub Issues for real-time tracking.

Task Prioritization

High Priority (MVP Features, must be completed first):

1. GitHub setup & repo organization (repo, board, standards).
2. Gemini API integration (flight queries for chatbot).
3. Frontend - Basic UI for chatbot & flight search (Landing Page, basic chatbot UI).
4. Chatbot - Basic responses using Gemini API.

Medium Priority (Enhancements & API Backups):

5. Amadeus API integration (flight search, pricing, status).
6. Flight Features - Search, Pricing, Status (core flight functionality).
7. Frontend - Flight Details & Saved Flights UI (Details Page, Saved Flights Page).

Low Priority (Refinement & Final Touches):

8. Chatbot memory & summarization (context memory, summarization feature).
9. Manual testing & bug fixes (full app testing).
10. Deployment to Vercel (final deployment)

Project Schedule

Date	Task
<i>Milestone 1: due 7th March</i>	
28th Feb - 1st March	<ul style="list-style-type: none"> Mid-fidelity Prototype and Project Schedule: Create mid-fidelity prototype on figma and plan a project schedule
2nd March	<ul style="list-style-type: none"> SDLC and WBS: Decide the SDLC method and the WBS
	<ul style="list-style-type: none"> Create DFD and MVC Diagrams
3rd March	<ul style="list-style-type: none"> Risk Assessment and GitHub Issues: Analyse the risks and assign Github Issues
4th March	<ul style="list-style-type: none"> Prepare slides for video Prepare report
5th March	<ul style="list-style-type: none"> Record the video. Finalise the report
6th March	Buffer
<i>Milestone 1.5: due 17th March-21st March</i>	
8th March - 9th March	<ul style="list-style-type: none"> Setup React Project Initialize project, install dependencies, and structure components.
	Chatbot: <ul style="list-style-type: none"> Set Up GeminiPro API in Project. Implement Flight Query Chatbot. Initialize API calls and configure authentication. Use Chat Completion API for basic queries.
	<ul style="list-style-type: none"> Create Landing Page (User input for flight search). Implement the UI using ReactJs and HTML-CSS.
9th March - 11th March	Chatbot: <ul style="list-style-type: none"> Add Context Memory feature: Store conversation history using the same API for context retention Context Memory Feature testing. Debugging.

	FlightSearch: <ul style="list-style-type: none"> Create Flight Results Page (Display flight options): Implement the UI using ReactJs and HTML-CSS.
12th March - 13th March	FlightSearch: <ul style="list-style-type: none"> Create Flight Details Page (Pricing, seat selection): Implement the UI using ReactJs and HTML-CSS.
	Chatbot: <ul style="list-style-type: none"> Implement Summarization. Enable chatbot to summarize interactions using GeminiPro's API. Summarization testing. Debugging.
14th March - 15th March	FlightSearch: <ul style="list-style-type: none"> Create Saved Flights Page (User bookmarking): Implement the UI using ReactJs and HTML-CSS.
	Chatbot: <ul style="list-style-type: none"> Debug & Optimize Chatbot. Chatbot Testing. Fix bugs, handle errors, and refine responses.
	Start working on Milestone 1.5 Report
16th March - 18th March	Chatbot: <ul style="list-style-type: none"> Integrate chatbot with main UI. Chatbot Testing. Debugging.
	FlightSearch: <ul style="list-style-type: none"> Setup Amadeus API in the project Initialize API calls and configure authentication.
	Work on the Milestone 1.5 Report
18th March - 19th March	Chatbot: <ul style="list-style-type: none"> Buffer for Milestone 1.5
	Work on the Milestone 1.5 Report
20th March : Milestone 1.5 Check-in	
<i>Milestone 2: Delivery</i>	

21st March - 23rd March	Chatbot: <ul style="list-style-type: none"> ● Improve to the chatbot based on feedback
	FlightSearch: <ul style="list-style-type: none"> ● Work on Flight Offer Search features. ● Connect relevant API endpoints and process the API data as per the project requirements. ● Testing the Flight Offer Search Feature. ● Testing. ● Debugging.
24th March - 25th March	FlightSearch: <ul style="list-style-type: none"> ● Work on Flight Pricing feature. ● Testing. ● Debugging.
26th March - 28th March	FlightSearch: <ul style="list-style-type: none"> ● Work on Flight Status Check feature. ● Testing. ● Debugging.
29th March - 31st March	<ul style="list-style-type: none"> ● Overall Flow testing. ● Debugging.
1st April - 3rd April	Prepare a report for Milestone 2.
4th April - 6th April	Prepare a video presentation.
7th April	Buffer
8th April	Submission for Milestone 2

Risk Assessment

Low:

1. UI/UX Inconsistencies
 - a. Risk: Some components may not align with the intended design, leading to a poor user experience.
 - b. Mitigation: Regular UI testing, consistency checks using a design system, and user feedback collection.
2. Minor API Downtime
 - a. Risk: APIs may experience temporary outages, causing short-term disruptions.
 - b. Mitigation: Implement error handling, retry mechanisms, and display user-friendly error messages.
3. Delayed Feature Implementation
 - a. Risk: Some features may take longer to develop than expected.
 - b. Mitigation: Follow agile development, set realistic timelines, and prioritize essential features.
4. Cross-Browser Compatibility Issues
 - a. Risk: The app may not function properly on all browsers.
 - b. Mitigation: Test on multiple browsers (Chrome, Firefox, Edge, Safari) and use responsive design practices.
5. Minor coding bugs
 - a. Risk: Encounter bugs while implementation.
 - b. Mitigation: Test after each sprint during development.

Mid:

6. API Rate Limits Reached
 - a. Risk: Exceeding API rate limits could temporarily block API calls.
 - b. Mitigation: Implement caching, optimize API requests, and monitor API usage.
7. Unclear or Inaccurate API Responses
 - a. Risk: API data may be missing, outdated, or incorrect.
 - b. Mitigation: Validate responses before displaying them and have a fallback mechanism.
8. Slow Performance on Low-End Devices
 - a. Risk: The app may run slowly on older devices due to high resource usage.
 - b. Mitigation: Optimize rendering, use lazy loading, and reduce unnecessary re-renders.
9. Difficulty Integrating Multiple APIs
 - a. Risk: APIs may have different data formats, making integration complex.
 - b. Mitigation: Standardize API responses, write utility functions for data transformation, and document integration steps.
10. Limited Testing Before Deployment
 - a. Risk: Not enough testing could lead to unnoticed bugs in production.
 - b. Mitigation: Implement unit, integration, and end-to-end testing before release.

High:

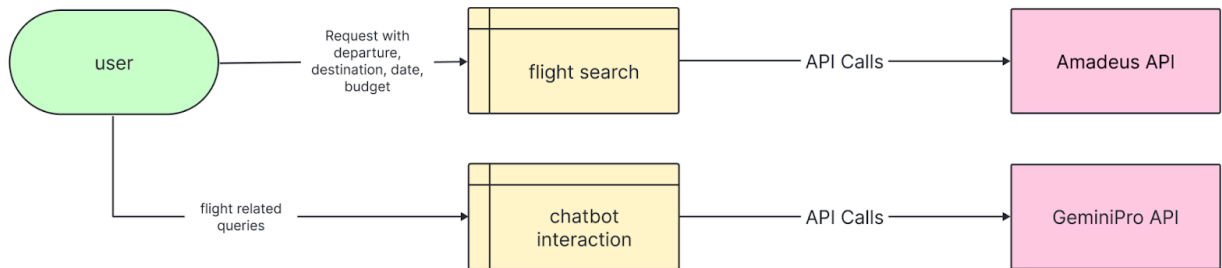
11. Major API Downtime or Deprecation
 - a. Risk: If a key API (Amadeus, Gemini) goes down, core features will stop working.
 - b. Mitigation: Have backup APIs ready (e.g., OpenAI for Gemini, AviationSateck for Amadeus).
12. Missing M2 deadline
 - a. Risk: Progress delayed or falling behind on project schedule.
 - b. Mitigation: Focus on MVP (chatbot + search) by Apr 8.
13. Unforeseen Technical Challenges
 - a. Risk: Some features may be more difficult to implement than expected.
 - b. Mitigation: Allocate extra time for complex features, seek help from forums or mentors, and iterate solutions.
14. Team member unavailability
 - a. Risk: Cross-train tasks among Pal, Kaiwen, Navjot, Karnpreet.
 - b. Mitigation: Address the issue in the group and contact the teaching team.
15. Loss of Project Data or Codebase Issues
 - a. Risk: Accidental deletion, repository corruption, or conflicts may lead to data loss.
 - b. Mitigation: Use Git version control, regularly push code to GitHub, and maintain backups.

Data Flow Diagrams

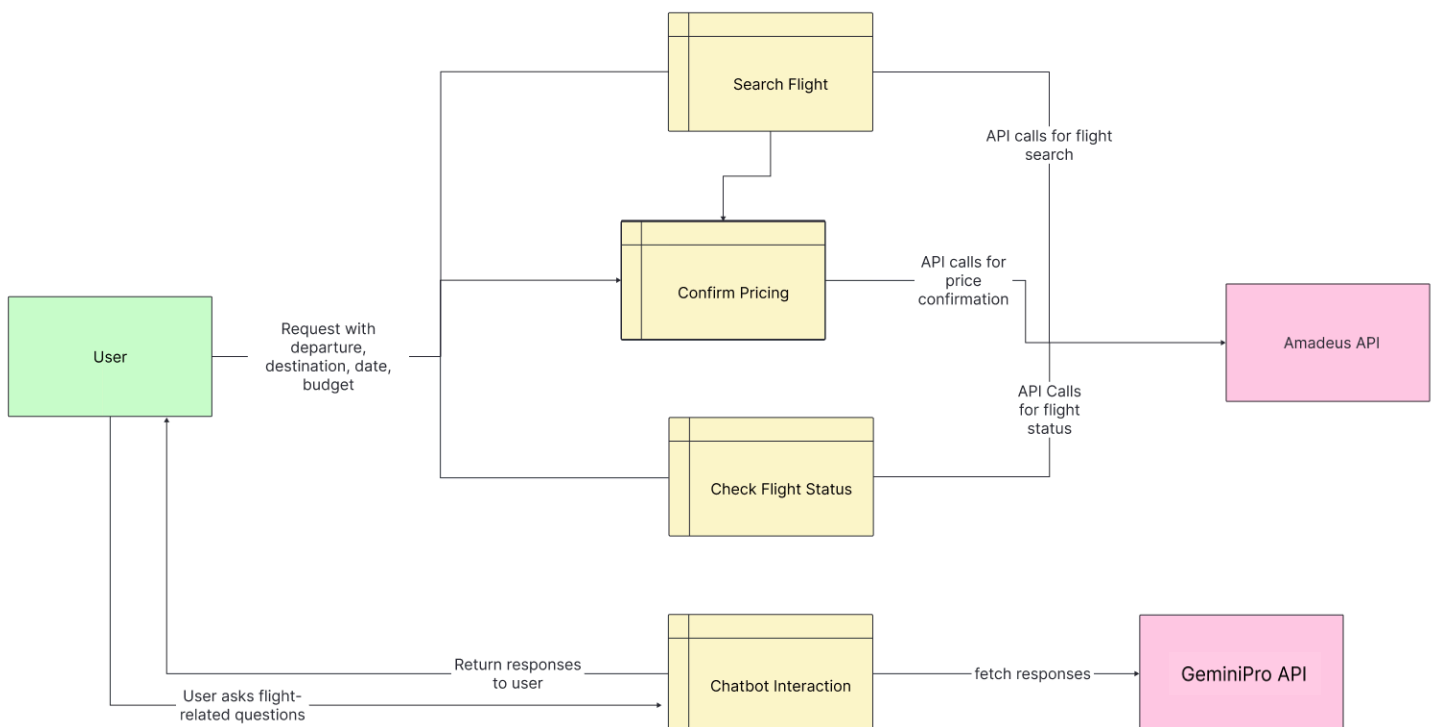
Figma link:

<https://www.figma.com/design/C2AVvgilTGwTGpThOJTUv5/Untitled?node-id=0-1&t=GRK1WN2SHj8jHFdS-1>

Level 0

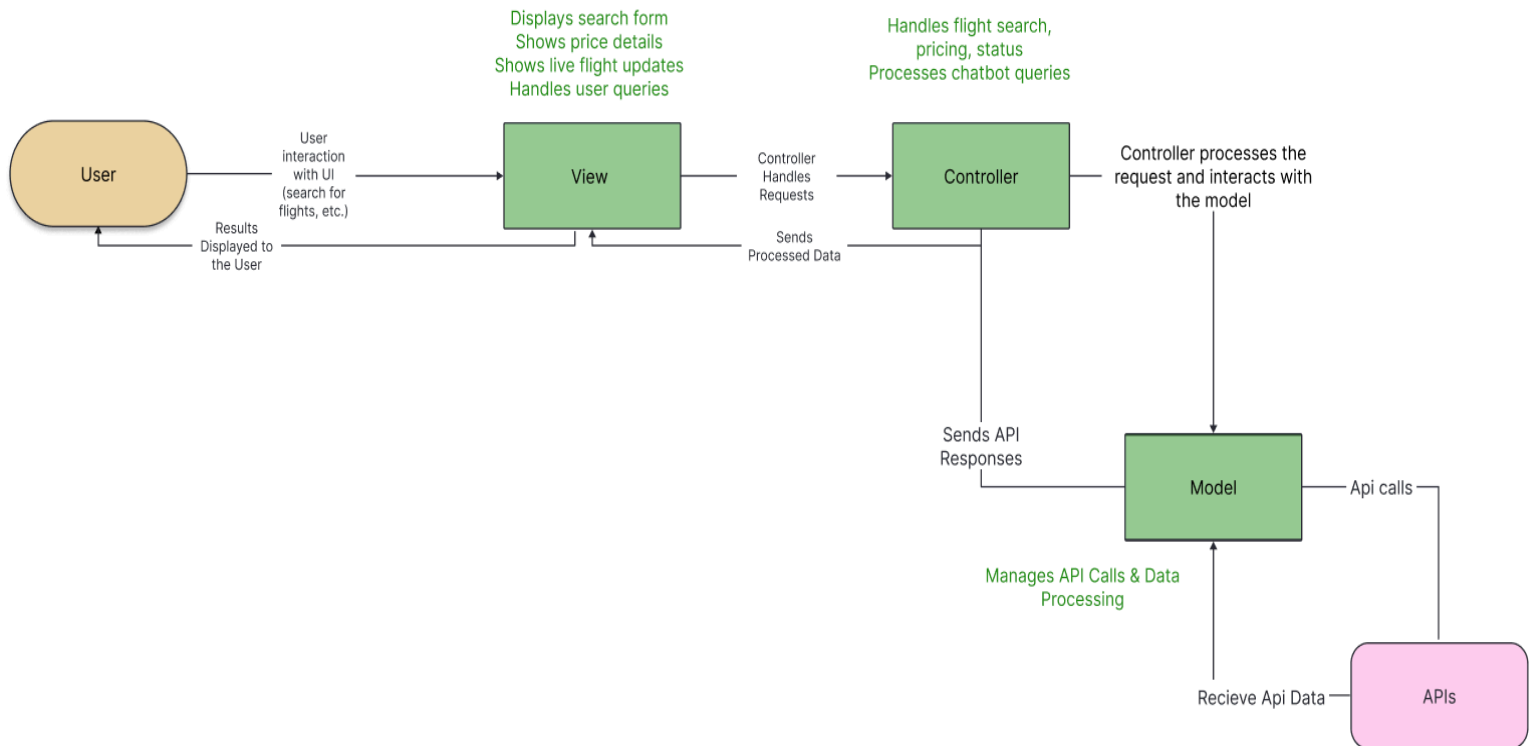


Level 1



Model View Controller Diagram

Figma: <https://www.figma.com/design/C2AVvgilTGwTGpThOJTUv5/Untitled?node-id=0-1&t=kDnrVT90fDyJigqh-1>



Appendix:

Contributions:

Name	Contribution
Karnpreet Cheema	<ul style="list-style-type: none"> ● Run performance tests and fix any issues to enhance overall functionality of the web application. ● Optimize API requests to ensure quick and efficient chatbot responses. ● Manage chatbot state to maintain smooth and natural conversations. ● Test the chatbot thoroughly to make sure it works reliably. ● Create backup strategies to handle API failures gracefully. ● Improve how the chatbot processes and understands API responses for better accuracy.
Navjot Singh	<ul style="list-style-type: none"> ● Merge pull requests (PRs) and manage repository structure. ● Optimize API requests to ensure quick and efficient chatbot responses. ● Manage chatbot state to maintain smooth and natural conversations. ● Test the chatbot thoroughly to make sure it works reliably. ● Create backup strategies to handle API failures gracefully. ● Improve how the chatbot processes and understands API responses for better accuracy. ● Run performance tests and fix any issues to enhance overall functionality of the chatbot.
Kaiwen Tao	<ul style="list-style-type: none"> ● Lead the design and user experience (UX) for the project. ● Work on integrating the Amadeus API for flight search related functionality. ● Optimize API request handling to improve search efficiency. ● Collaborate on implementing and refining the flight search UI.

	<ul style="list-style-type: none"> • Debug and test API responses to ensure accuracy and completeness.
Pal Rohit Tilva	<ul style="list-style-type: none"> • Oversee project progress and ensure timely completion of tasks. • Integrate the Amadeus API for flight search related functionalities. • Design and implement the user interface for a seamless search experience. • Ensure API requests and responses work correctly within the UI. • Run performance tests and fix any issues to enhance overall functionality of the flight search related functionalities.

Changelog:

Date	Change	Description	Modified By
2nd March	Switched from OpenAI API to Google Gemini API for chatbot.	Our team decided to use the Google Gemini API for our chatbot feature instead of the OpenAI API due to cost considerations. Google Gemini offers a more affordable alternative while still providing robust AI capabilities suitable for our project.	Team Decision

Video Presentation Slides:

https://docs.google.com/presentation/d/1Ekx0oRtTB9ak_OF2EUdWyHlk-Y3IO0jfLHYrVfp0NXQ/edit?usp=sharing