

Software Architecture Model Report

Software Architecture

Frontend (Presentation) Layer will be implemented using React, JavaScript, HTML, CSS, etc. This layer is responsible for the user interface and graphics. This layer is also responsible for handling the logins and authentication, with features such as calendar view, task bar, syllabus uploads, hyperlinks to the UF map, etc.

Backend (Application) Layer will be built with Python, Flask, etc. This layer manages the background application logic including the authentication through Google OAuth. This will also handle the guest vs student account types and the integration with the Canvas API. It also runs the AI Parser that extracts events and deadlines from uploaded syllabuses. This layer handles the response exchange between the frontend and backend.

Database (Data) Layer will be implemented with MongoDB which stores key information such as user profiles, scheduled events (both manually added and AI added), tasks, notification rules, etc.

System Decomposition

Frontend (React JS/HTML/CSS/etc)

- Google OAuth Login and user authentication UI
- Calendar views (month, week, day, etc)
- Task Sidebar for assignments and smart to do list
- File upload for syllabus uploads
- Notification settings and rules
- Campus map hyperlink

Backend (Python/Flask/etc)

- Authentication services (Google OAuth tokens)
- User services: profile management, account types (guest, student), permissions
- Events: handles create, read, update, delete operations (including recurring notifications and events)
- Canvas API: Fetches and syncs from Canvas
- AI Parse/Notification: Extracts key deadlines and information from uploaded syllabuses and automatically generates events and to do lists. Also schedules notification reminders.

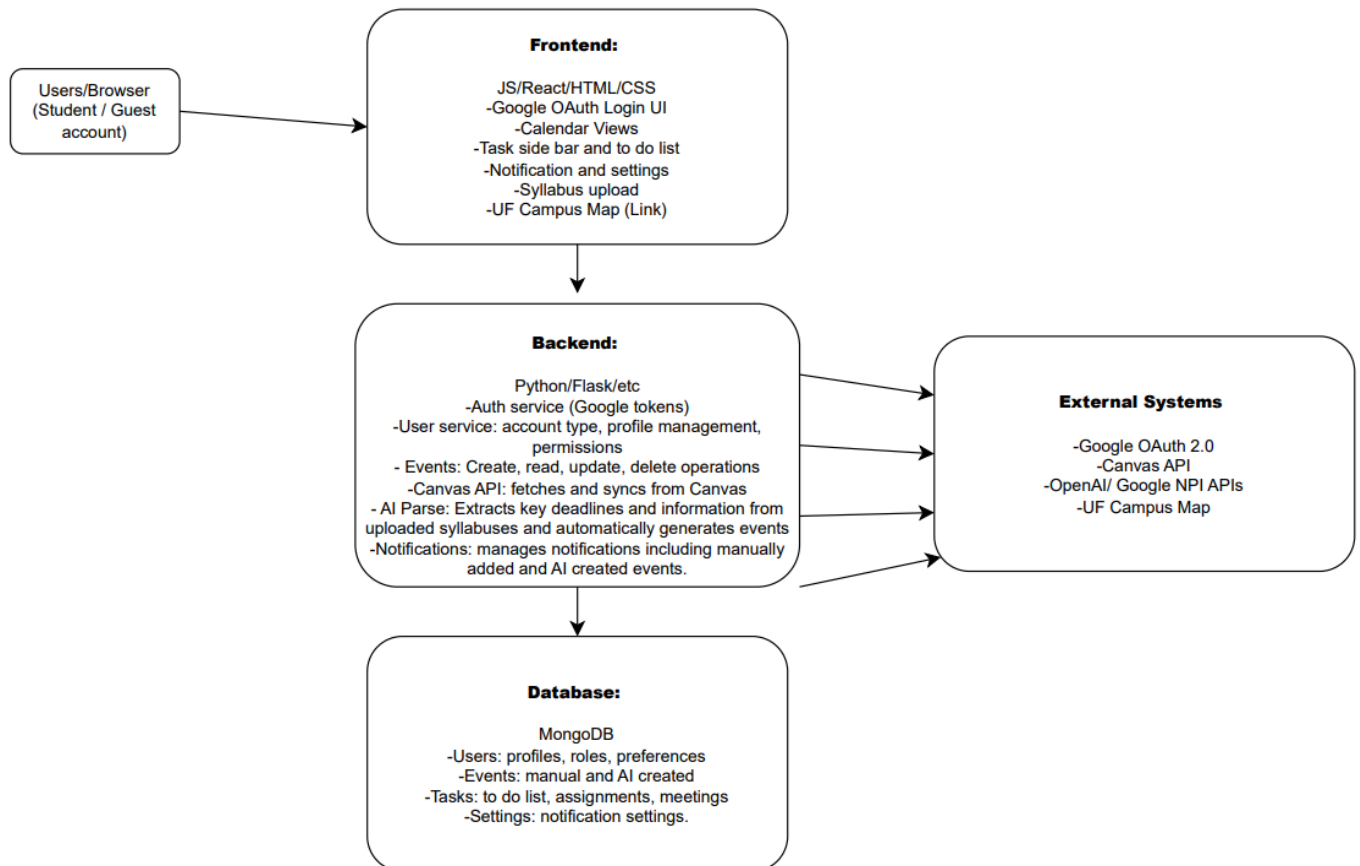
Database (MongoDB)

- User Data: Stores profiles, account types, preferences, settings
- Events: Stores manual and AI added events with their respective data
- Tasks: Stores user assignments, tasks, to do list, etc.
- Settings: Manages permissions and customized preferences.

External

- Google OAuth 2.0: for secure authentication and login
- Canvas API: for syncing between Canvas and Canvas Compass
- UF Campus Map: direct hyperlink
- AI API: OpenAI, Google NLP, etc, for advanced syllabus data extraction.

System Model



Technologies and Tools

Languages: Python, JavaScript, HTML, CSS

Frameworks: React + Tailwind CSS, Flask

Database: MongoDB

Libraries/Tools: Google OAuth 2.0, UF Campus Map, OpenAI

React was chosen for its modular UI and Tailwind CSS was chosen for its fast and consistent styling.

Flask was chosen for its lightweight API, and MongoDB was chosen for its flexible storage.

Google OAuth ensures secure login with minimal overhead.