**SP-100 — Campus AI Companion**

**Software Requirements Specification**

**CS 4850 - Section 02 – Fall 2024**

**September 01, 2024**

**Team Members:**

|  |  |  |
| --- | --- | --- |
| Name | Role | Cell Phone / Alt Email |
| Dorian (Team Lead) | Developer | 470.439.9907  doriantaponzing@gmail.com |
| Fredy | Project Owner and Developer | 470.836.8552  atakoutene@gmail.com |
| Yann | Developer | 404.649.4887  thisisyanndev@gmail.com |
| Leopold | Developer | 470.838.2345  leopoldgatsing@gmail.com |
| Manuella | Documentation | 404.599.6647  manuellakoodjo@gamil.com |
| Sharon Perry | Project Advisor | 770.329.3895  [Sperry46@kennesaw.edu](mailto:Sperry46@kennesaw.edu) |

Table of Content

[**I.** **Introduction** 3](#_Toc176110141)

[1. Overview 3](#_Toc176110142)

[2. Project Goals 3](#_Toc176110143)

[3. Definitions and Acronyms 3](#_Toc176110144)

[4. Assumptions 3](#_Toc176110145)

[**II.** **Design Constraints** 4](#_Toc176110146)

[1. Environment 4](#_Toc176110147)

[2. User Characteristics 4](#_Toc176110148)

[3. System Constraints 4](#_Toc176110149)

[**III.** **Functional Requirement** 5](#_Toc176110150)

[1. Login and Account Management 5](#_Toc176110151)

[2. Authentication 5](#_Toc176110152)

[3. Recommendation/Suggestion Management 5](#_Toc176110153)

[4. Schedule Management: 5](#_Toc176110154)

[**IV.** **Non-Functional Requirements** 6](#_Toc176110155)

[1. Usability: 6](#_Toc176110156)

[2. Scalability: 6](#_Toc176110157)

[3. Security: 6](#_Toc176110158)

[4. Maintainability: 6](#_Toc176110159)

[5. Portability: 6](#_Toc176110160)

[6. Compliance: 7](#_Toc176110161)

[**V.** **External Interface Requirements** 7](#_Toc176110162)

[1. User Interface Requirements 7](#_Toc176110163)

[2. Hardware Interface Requirements 7](#_Toc176110164)

[3. Software Interface Requirements 7](#_Toc176110165)

[4. Communication Interface Requirements 8](#_Toc176110166)

[**VI.** **Glossary** 9](#_Toc176110167)

[**VII.** **References** 10](#_Toc176110168)

# **Introduction**

## Overview

The Campus AI Companion is an AI-powered mobile platform built on top of the OpenAI API to help CS students manage their academic and career paths. This app will offer them personalized guidance, connect students to resources, and help them achieve their goals. Initially focused on CS students, the platform will gradually expand to include other disciplines, providing tailored support based on each major’s requirements.

## Project Goals

* Provide personalized recommendations and suggestion based on user information, courses, and activities being involved in. These recommendations include: course suggestions, club and event recommendations, career path and Job recommendations.
* Enable students to manage their schedule.

## Definitions and Acronyms

1. **AI:** Artificial intelligence
2. **RAG:** Retrieval Augmented Generation
3. **API:** Application Programming Interface
4. **SRS:** Software Requirements Specification.
5. **SDD:** Software Design Document.
6. **LLM:** Large Language Model

## Assumptions

* Users will have access to mobile devices and the internet.
* Users will have basic technical knowledge to navigate mobile applications.

# **Design Constraints**

## Environment

1. **Platform**: Cross platform i.e. iOS and Android.
2. **Development Environment**: React Native(Frontend), Node.js /Express.js and LangChain (backend).
3. **Styling Framework**: Tailwind CSS.
4. **Database**: MongoDB, Firebase Authentication and Storage.

## User Characteristics

* **Primary Users**: This will be computer science students, especially international students.
* **Secondary Users**: With time, students from other disciplines will use the app.

## System Constraints

1. ***Performance Constraints***: The application needs to operate efficiently across both iOS and Android platforms, requiring optimization to handle a potentially large user base without compromising on speed or responsiveness.
2. ***Security Constraints*:** user data must be securely stored and managed adhering to data protection regulations like FERPA.
3. ***API Usage Constraints***: The OpenAI and other third-party services have limitations on usage such as requests limit per hour and pricing tiers that must be considered when designing features.
4. ***Scalability Constraints***: The system should be scalable to handle an increasing number of users, but currently resources might limit the initial capacity.
5. ***Technological Constraints*:** The project is constrained by the need to use specific technologies like React Native, Node.js, Express.js, LangChain and TailwindCSS, which may limit certain design or architectural choices.

# **Functional Requirement**

## Login and Account Management

* **Create Account**: Users can register.
* **Login**: Users log in using their registered credentials.
* **Password Recovery**: Users can recover their passwords through email.

## Authentication

* **User Authentication**: Firebase Authentication to verify user identities.
* **Session Management**: Maintain user sessions with secure tokens.

## Recommendation/Suggestion Management

* Based on the user’s prompt and context provided to our LLM, the app will provide some answers, recommendation, and suggestions to guide the user.
* The app will also provide resources to assists students in their everyday activities and to grow into their academic life.

## Schedule Management:

* Based on the knowledge of users’ day to day activities, the app will generate a timetable for each user.
* The app will also provide reminders about the user’s schedule to make sure he misses no important tasks.

# **Non-Functional Requirements**

## Usability:

The platform should have an intuitive and user-friendly interface that allows students to easily navigate through features like personalized guidance and resource connections.

## Scalability:

The system should be designed to handle an increasing number of users as it expands from serving CS students to other disciplines.

It should support scaling both horizontally and vertically to accommodate growth in data, user interactions, and API calls.

## Security:

The platform must ensure secure authentication and authorization using Firebase Authentication, protecting user data with encryption both at rest and in transit.

## Maintainability:

The codebase should be modular, well-documented, and follow best practices in software development to facilitate easy updates, debugging, and feature enhancements.

The platform should use version control systems like Git, with clear branching strategies and code reviews to ensure code quality and manageability.

## Portability:

The application must be portable across different mobile operating systems (iOS and Android) with consistent performance and functionality.

It should also be easily deployable to different cloud environments, if needed, without significant reconfiguration.

## Compliance:

The platform must comply with relevant legal and regulatory requirements, including data protection laws like FERPA and COPPA ensuring that user data is handled responsibly.

It should also adhere to academic integrity policies and institutional guidelines where applicable.

# **External Interface Requirements**

## User Interface Requirements

* Consistent design using TailwindCSS.
* Mobile-responsive layout for both iOS and Android devices.
* Support for dark and light themes.

## Hardware Interface Requirements

Compatibility with standard smartphone features such as notifications, GPS (if needed for specific functions).

## Software Interface Requirements

* Integration with Firebase for user authentication and data storage.
* Use of OpenAI API for natural language processing and auto-completion.
* LangChain for external data integrations and context-aware responses.

## Communication Interface Requirements

* RESTful API communication between frontend and backend.
* Secure data transmission using HTTPS protocols.

# **Glossary**

1. **API (Application Programming Interface)**  
   A set of rules and definitions that allow different software applications to communicate with each other. In this project, RESTful APIs are used for communication between the frontend and backend services.
2. **Authentication**  
   The process of verifying the identity of a user or system. In the app, authentication is managed using JWT (JSON Web Tokens) to ensure that only authorized users can access certain features.
3. **Authorization**  
   The process of granting or denying access to specific resources or actions within an application, based on the user's identity and permissions. It determines what a user can or cannot do after being authenticated.
4. **Backend**  
   The server-side component of the application that handles business logic, database interactions, and external API calls. It includes the server (Node.js with Express.js), the database (MongoDB), and various services (e.g., user management, recommendation engine).
5. **Frontend**  
   The client-side component of the application that users interact with. It is built using React Native, allowing it to run on both iOS and Android platforms. The frontend handles UI rendering, user input, and communicates with the backend via APIs.
6. **MongoDB**  
   A NoSQL database used for storing and managing data in a flexible, JSON-like format. It is the primary database for this project, chosen for its scalability and ease of handling dynamic data.
7. **OpenAI API**  
   An API provided by OpenAI that allows developers to integrate AI models into their applications. In this project, the OpenAI API is used for generating personalized recommendations based on user data.
8. **React Native**  
   A JavaScript framework for building mobile applications using React. It allows developers to build apps for both iOS and Android using a single codebase, ensuring consistency and reducing development time and cost.

# **References**

1. **MongoDB Documentation**  
   MongoDB, Inc. "MongoDB Manual." [MongoDB Docs](https://www.mongodb.com/docs/)
2. **OpenAI API Documentation**  
   OpenAI. "OpenAI API." [OpenAI Docs](https://platform.openai.com/docs/)
3. **Node.js and Express.js Documentation**  
   OpenJS Foundation. "Node.js Documentation." Node.js Docs.
4. **React Native Documentation**  
   Facebook, Inc. "React Native – A framework for building native apps using React." React Native Docs.