

Team Project

Team Design Review

“Student Assist app”

Course ID: CS487 Software Engineering I

Professor: Dennis Hood

Team Name: Team G8

Partner team - Team G7

Team Members:

- Shivdeep Bisurkar (A20525712)
- Prajakta Kumbhar(A20523710)
- Colin Brennan (A20418083)
- Hao-Chih Weng (A20480011)

Date: April 10, 2023

Section 1 - will summarize your assessment of your partner team's proposed design.

The application includes features such as a home page with login and registration options, collaboration tools, a job search feature, a career roadmap feature, user profiles, notifications, and settings. It has a user-friendly interface and can be accessed on multiple platforms. The collaboration tool allows students to connect and collaborate, while the job search feature allows them to search and apply for job openings. The career roadmap feature outlines the steps needed to achieve career goals. Notifications inform users of updates related to job search, career roadmap, and collaboration activities. The settings section allows users to customize their experience.

The student assistant application will interact with the Job Search API to display job postings, with the Career Roadmap API to provide personalized career guidance, and with the Student Database to access student information for personalization and collaboration purposes. The collaboration tool within the application will enable students to connect and work on projects together. The application will seamlessly integrate with external systems and provide a user-friendly experience for students.

The home page of the application provides an overview of its key features and benefits, along with login and registration options for new users. The job search feature allows students to search and apply for job openings related to their field of study, with filtering options for location and job type. The career roadmap feature outlines the steps needed to achieve career goals, including courses, internships, networking events, and other

resources. Workflow diagrams illustrate the user flow for each feature. The application interacts with external systems and data sources, such as job search API and career roadmap API, to provide personalized information and resources for students.

The design strategy for each of the non-functional requirements of the Student Assist app involves implementing different approaches. To meet the performance requirement, the app's code can be optimized, caching can be used, and scalable infrastructure can be installed. The security requirement can be met by implementing secure authentication, encryption, and security measures. To ensure scalability, the app can be designed to be modular and distributed, with load balancers and horizontal scaling. The usability requirement can be met by using a user-centered design approach, appropriate visual design, clear language, and simple navigation. Finally, the reliability requirement can be met by implementing appropriate error handling mechanisms, redundancy, and monitoring and alerting systems.

Section 2 - will detail specific recommended improvements to either the proposed design itself

The proposed design appears well-rounded and includes the essential elements that students would find beneficial. Here are some potential improvements that might be taken into account:

Section 1 part:

1. In the career roadmap, giving individualized recommendations that are based on the user's interests, abilities, and career aspirations would improve the career roadmap tool. Machine learning algorithms that examine the user's profile and platform activities could do this.
2. Integrate with external tools: Consider integrating the application with external tools and services that students might be using, such as LinkedIn or Google Drive. This would allow for a seamless experience and increase the value of the application for users.

Section 2 part:

This section depicts that the 'student assist app' is interacting with all the other components like 'Job search API', 'career roadmap API', 'student database', etc. But it is not clear through the system diagram. The 'student assist app' component should be in between and all other components should be shown as communicating with it.

Section 3 part:

1. In the 3.1 section diagram, the flow from 'register' to 'user' is not clear. No message shown for that flow, which leaves ambiguity.
2. In the 3.2 Job search diagram, again the external job search API interface can be shown.

3. Section 3.3: The description of 'career roadmap' and the corresponding diagrams is completely different. The career roadmap section says "this section would include information on relevant courses, internships, networking events, and other resources". But the diagram shows something else.
4. In section 3.4 naming for some components in the diagram is missing, which makes the diagram unclear.

Section 4 part:

1. In section 4.1: The algorithm can be improved by adding additional search criteria, such as job level, salary range, or company size, to help users refine their search results and find jobs that better match their preferences.
2. In section 4.2: algorithm can be improved by incorporating user feedback, providing personalized recommendations, integrating with job search engines, offering skill-building resources, and incorporating industry trends and forecasts. These improvements can help provide users with a more personalized and comprehensive career path.
3. In section 4.3: algorithm can be improved by implementing message encryption, adding message filtering, implementing message threading, adding message notifications, and implementing message archiving capabilities.

4. In section 4.4: The notification algorithm could be improved by adding notification preferences, implementing personalized notifications, adding notification filters, implementing notification reminders, and offering actionable notifications

Section 5 part:

This section can be improved by adding input validation, authentication methods, profile picture functionality, social media integration, and messaging functionality. These improvements can help create a more secure, engaging, and user-friendly experience for users of the collaboration tool.

And the content describes the data model and object model perspectives of the Student Assist app's collaboration tool, job search, and roadmap to career features. In the collaboration tool, there are four entities: User, Group, Message, and Attachment. The User object has properties like name, email, and password, and can interact with the Group object to create or join a group. The Group object has properties like name, description, and members, and can interact with the Message and Attachment objects to send and receive messages and share files. Similarly, the job search and roadmap to career features have their data and object models defined.

Overall, the content is informative and provides a clear understanding of the data and object models used in the Student Assist app. It would be helpful if the content included more details on how these models are implemented in the app's code and how they

interact with each other. Additionally, it would be useful to provide examples of how these features can benefit students in their academic and career goals.

Section 6 part:

While the overview mentions some important security mechanisms like encryption and SQL injection prevention, it does not mention other critical measures like input validation and output encoding.

And The design strategy utilized to satisfy each of the Student Assist app's non-functional requirements is an important aspect of developing a high-quality application. The approach described in the overview includes a combination of methods, such as optimizing code, implementing security mechanisms, designing for scalability, focusing on user-centered design, and ensuring reliability through appropriate error handling and monitoring. While this overview provides a good starting point, there are a few additional comments that could be made to enhance the design strategy and ensure the application meets the needs of its users.

In conclusion, while the design strategy outlined for the Student Assist app's non-functional requirements is a good starting point, there are several additional considerations that should be taken into account to ensure that the app meets the needs of its users. By considering factors such as performance consistency, emerging security threats, scalability, accessibility, and failover mechanisms, developers can create an app that is not only highly functional but also meets the needs of all users.