**Software Requirements Specification**

**for**

CareerWise

**Version 1.0 approved**

**Prepared by <authors>**

**<organization>**

**<date created>**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

This document is the requirements specification for CareerWise. CareerWise is a mentor matching service for college students and recent grads, to help receive mentorship regarding skill development, resume revising, and interview practice. CareerWise will use a career questionnaire to help match students and mentors based on needs. CareerWises objective is to better help students transition from University into the job market and increase career readiness. This document describes the requirements and specifications for the associated systems.

## **Document Conventions**

ThisDocument was created based on the IEEE template for System Requirement

Specification Documents.

The Following conventions are used as follows:

| **Convention** | **Description** |
| --- | --- |
|  |  |

## **Intended Audience and Reading Suggestions**

The intended audience for this document includes developers, project managers, and end users. It provides detailed descriptions of the system, its functional and non-functional requirements, and the associated software components. End users should focus on Sections 1 and 2 to gain a general understanding of the system’s purpose and functionality, while developers and project managers should review the remaining sections for technical details and implementation considerations.

## **Product Scope**

CareerWise is a react based mentorship platform designed to help students connect with experienced mentors to support their academic and professional growth. The system enables mentees to complete career interest questionnaires, receive personalized mentor matches, and track their progress through career goals and milestones. Mentors can monitor mentee progress, provide feedback, and guide goal development. The primary objective of CareerWise is to enhance student employability by offering structured mentorship, progress tracking, and personalized career development tools.

## **References**

CareerWise GitHub: <https://github.com/COS420-F25/TeamA>

# **Overall Description**

## **Product Perspective**

Careerwise is a web application that’s inspired from already existing student mentor relationship applications. The key purpose of Careerwise is to allow graduate students to select mentors and talk to them to get further experienced advice after graduation. The application will allow for video calls between mentors and students and AI will be integrated and trained on videos to further help graduate students. The application will be open sourced and open to anyone with a google or email account.

## **Product Functions**

* Google/Sign up feature: Allows the user to sign in with their gmail or email address to make a profile on the site.

* Payment methods: Allows the user several options to pay for mentor time. Incentivising the mentors to sign up for the site.

* Reviews: Allows the users to rate mentors on their effectiveness and displays the ratings in a 5 star rating over every mentor.

* Video calls: Uses Zoom API to allow users and mentors to speak with each other via video conferencing.

* Progress Tracker: Allows users to set goals on a dashboard and update their progress every time users advance in their knowledge. Progress will be displayed via a line chart.

* AI summarizer: Users will be able to upload transcripts from zoom video calls to a generative AI. The user will be able to ask follow up questions to the AI and the AI will respond using the knowledge from the uploaded transcripts.

## **User Classes and Characteristics**

The data collected by Careerwise is sensitive and needs to be handled with strong privacy policies. It must also comply with the CPRA (California Privacy Rights acts). Users must be given options to opt in or out of certain data collection methods.

## **Operating Environment**

Careerwise is a web application made with the react app, it uses HTML, CSS, and Javascript. It is designed to work on web browsers and phones. It will support any hardware that can run a modern browser like chrome, firefox, microsoft edge, that runs Javascript. If a user decides to disable Javascript on their web browser the website will not function.

## **Design and Implementation Constraints**

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## **User Documentation**

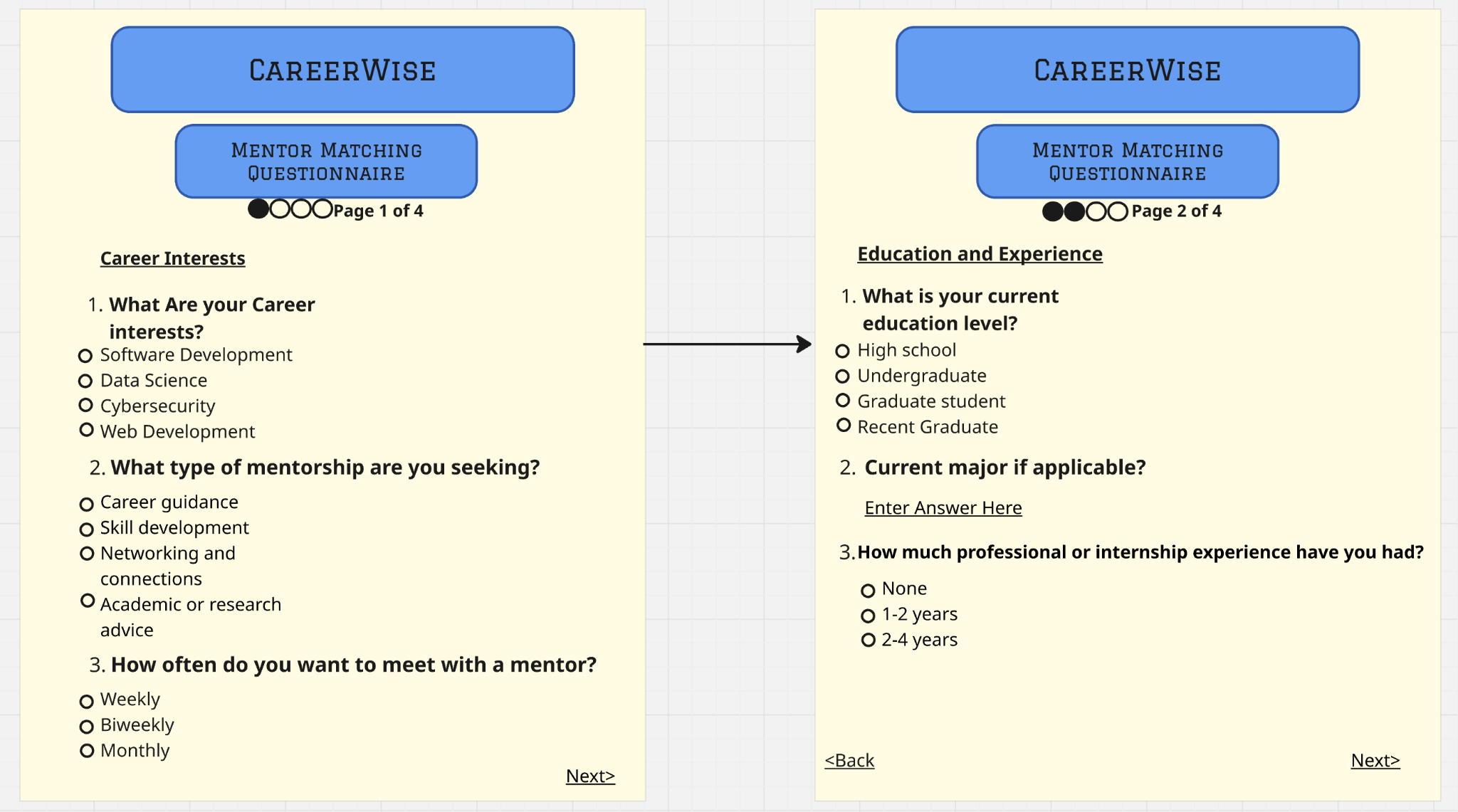
*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# **External Interface Requirements**

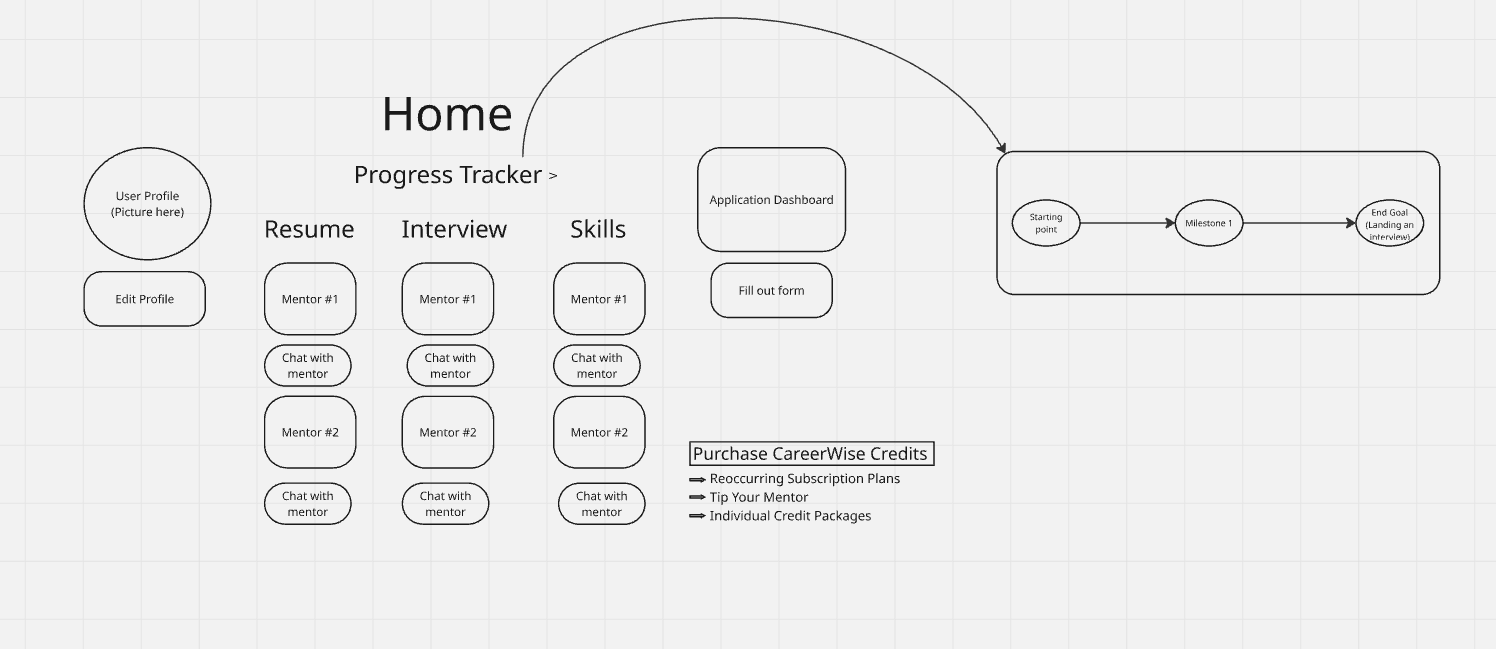
## **User Interfaces**

**

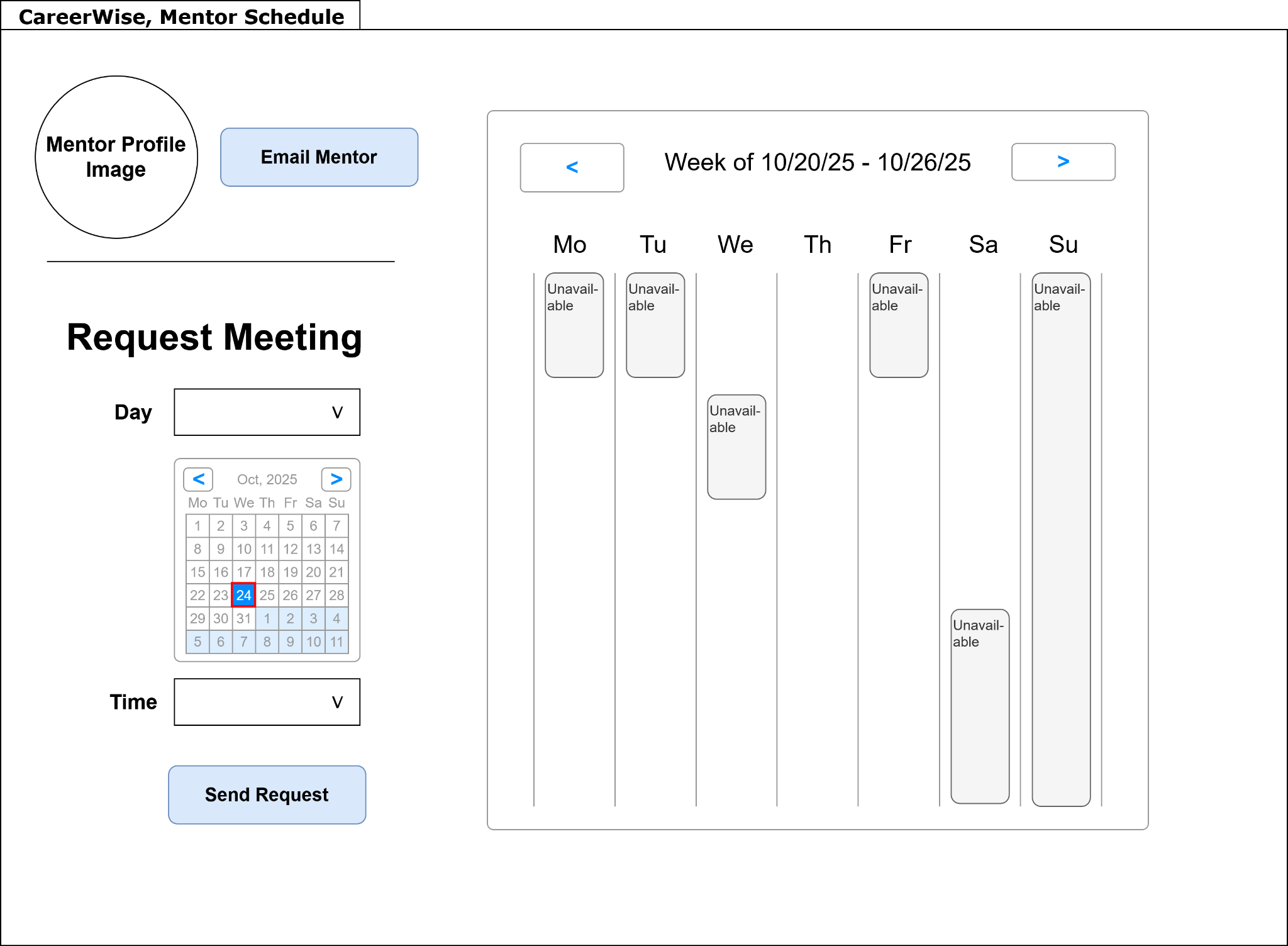
[*https://miro.com/app/board/uXjVJ6TK2u4=/*](https://miro.com/app/board/uXjVJ6TK2u4=/)

This is a mockup of the mentor mentee matching questionnaire with some example questions. The questionnaire will be used to help match mentors with mentees based on career interests, type of guidance, and how often you want to meet.

<https://miro.com/welcomeonboard/TmVzQlF1cVZ5a0VidEswcXN3MUoxZlB2d0NraWViUW5Gb25jNUhFQlUxVXF0bTVKV09qdmlHSTZqR3Z0YUNNbXc2OVdqZTVma2VoUmxOaG9rNDZtS3AvMlIya2VKeVpQeU51dW96Wm1JY1g0S3dMelc5NVNFTHo5OW55TnFGd2tBd044SHFHaVlWYWk0d3NxeHNmeG9BPT0hdjE=?share_link_id=493900405600>

The following wireframe is a mockup of the home screen after you log in. The display shows functions of profile setup, dashboards with surveys (elaborated in wireframe 1), then a display of forms and applications.

The following wireframe is a mockup of the view of a mentor’s schedule, from a user’s perspective. The application displays a schedule that informs the user what days and times they may request for a virtual meeting. To the left of the schedule are input fields for choosing a date and time, as well as a button to send the meeting request.



## **Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# **System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## Account Management

4.1.1 Description and Priority

\*High Priority

This section defines the functionalities that allow users to create, access, and manage their personal accounts within the system. It includes account registration, authentication, and profile management to ensure secure and personalized user experiences.

4.1.2 Stimulus/Response Sequences

-User clicks create account -> System prompts user for email and password and stores in database

-User clicks sign in -> System prompts user for email and password

-User clicks sign out -> System signs out user and returns to the default screen

-User edits profile ->System allows user to enter information and stores it in the database

4.1.3 Functional Requirements

FUNC\_REQ-1: The system shall allow the user to create an account using an email and password.

FUNC\_REQ-2: The system shall allow the user to log in and out of the account.

FUNC\_REQ-3: The system shall allow users to create and edit a personal profile including name, contact information, education, skills, experience, and resume upload

## Mentorship Matching

4.2.1 Description and Priority

\*High Priority

This section describes functionalities related to the personalized mentor matching. system. This allows users to take a questionnaire and receive mentors based on their career needs.

4.2.2 Stimulus/Response Sequences

-User creates an account -> User is prompted with the option to take CareerWise questionnaire.

-User takes a career questionnaire -> User is presented with a list of personalized mentors they can choose from.

-User views mentor recommendations -> The system displays each mentor’s profile, including bio, experience, areas of expertise, and availability.

-User cancels or archives an active mentorship -> The system notifies the mentor, archives communication and progress records, and updates the user’s status to allow rematching.

-User confirms a mentor relationship -> The system finalizes the pairing, enabling access to the mentor’s schedule, communication tools, and progress tracking features.

4.2.3 Functional Requirements

FUNC\_REQ-4: The system shall allow users to meet with more than one mentor.

FUNC\_REQ-5: The system shall allow users to archive an active mentorship and notify the mentor automatically, archiving past communication and progress records.

FUNC\_REQ-6: The system shall allow users to complete a career interest questionnaire covering preferred industries, roles, technologies, and target companies.

FUNC\_REQ-7: The system shall recommend users with mentors based on questionnaire responses, skills, and career goals.

FUNC\_REQ-8: The system shall display mentor profiles including bio, experience, areas of expertise, and availability.

## Progress Tracking Management

4.2.3 Description and Priority

\* Medium Priority

This section describes functionality related to progress tracking for mentors and users. It allows users to create and manage career goals with defined milestones to measure achievement over time. Mentors can view progress updates and provide guidance to help users stay on track.

4.2.3 Stimulus/Response Sequences

User creates a new career goal and defines milestones -> System stores goal and milestone and displays goal and progress to user.

User updates progress -> System stores progress and updates progress on dashboard.

4.2.4Functional Requirements

-FUNC\_REQ-9: The system shall allow users to update their progress on a dashboard.

-FUNC\_REQ-10: The system shall allow users to create career goals and define milestones to track their progress toward completion.

## Scheduling Management

4.2.4 Description and Priority

This section describes all requirements related to scheduling and managing meetings between mentors and mentees. Users will be able to view their mentor’s availability, request meeting times, and receive confirmations or updates through the system. This functionality ensures efficient communication and coordination, helping both parties maintain consistent mentorship interactions.

4.2.4 Stimulus/Response Sequences

Users views mentors schedule -> System displays mentors meeting slots and availability.

User requests a meeting during a mentors open slot -> System sends email to mentor and mentee to confirm meeting.

Users scheduled meeting -> System sends reminder email to both mentor and mentee day of about the meeting.

*4.2.4* Functional Requirements

FUNC\_REQ-11: The system shall allow users to view their assigned mentor’s meeting availability schedule.

FUNC\_REQ-12: The system shall allow users to request meetings on CareerWise that will be scheduled on Zoom.

FUNC\_REQ-13: The system shall send notice and reminder emails regarding meeting times.

## Rating System

4.2.4 Description and Priority

\*Low priority

This section includes all requirements regarding the rating system of mentors. This system allows users to rate mentors after meeting to help careerwise gauge mentor quality and will allow admins to remove mentors if needed.

4.2.4 Stimulus/Response Sequences

User rates mentor after meeting -> system stores rating and updates mentors overall rating.

Admin gets notification that a mentor's profile has dropped below a 3.0 -> System prompts admin with option to remove an account.

Admin removes mentor profile -> System removes mentor profile from profile pool and removes them from profiles that have already matched/meet with them.

*4.2.4* Functional Requirements

FUNC\_REQ-14: The system shall allow users to rate mentor profiles on a 1–5 star scale and optionally provide written feedback.

FUNC\_REQ-15: The system shall allow administrators to remove or flag mentor profiles that have an average rating below 3.0.

## Discussion Forum

4.2.4 Description and Priority

\*Low priority

This section encapsulates all requirements relating to the discussion board. This will allow users to communicate on a public forum with others and share useful information to those in similar positions.

4.2.4 Stimulus/Response Sequences

User post on forum-> System displays post for all on the same forum.

User replies to another persons post -> System displays reply for all on the same forum

*4.2.4* Functional Requirements

FUNC\_REQ-16: The system shall allow users to make posts and replies on a public discussion forum.

# **Other Nonfunctional Requirements**

NONFUNC\_REQ1: The system shall be available 24/7 with a minimum of 99% uptime

NONFUNC\_REQ2: The system shall not share user data with third parties.

NONFUNC\_REQ3: The system shall employ manual moderation tools to allow admins to remove posts if they present abuse as defined in the Trust & Safety Professional Association’s curriculum.

<https://www.tspa.org/curriculum/ts-fundamentals/policy/abuse-types/>

NONFUNC\_REQ4: The system shall employ automated moderation tools to not allow users to post to the discussion forum if the post contains abuse as defined in the Trust & Safety Professional Association’s curriculum.

<https://www.tspa.org/curriculum/ts-fundamentals/policy/abuse-types/>

NONFUNC\_REQ5: The system shall use PayPal’s secure payment to allow users to pay for mentorship services using PayPal or linked payment methods.

NONFUNC\_REQ6: Page load times shall not exceed 2 seconds 95% of the time.

NONFUNC\_REQ7: The system shall support up to 10,000 concurrent users while maintaining response times under 2 seconds for 95% of requests.

NONFUNC\_REQ8: The system shall provide a user interface that allows at least 90% of users to learn within 10 minutes how to attain a mentor.

NONFUNC\_REQ9: The system shall prevent users from being billed unwillingly by allowing them to archive a mentorship in a maximum of three clicks.

NONFUNC\_REQ10: The system shall employ a verification process to ensure that all mentors available to users are legitimate.

## **Performance Requirements**

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## **Safety Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## **Security Requirements**

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## **Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*