Software Requirements Specification

Mentorship Network Management System

Jaisal Friedman
Daniel Watson
Gabriel Garcia

Table of Contents

1. Introduction

This section provides the purpose, scope, and overview of the SRS document.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Manage.Mentorship software system. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the Mentorship Network Executive Team for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

The Mentorship Network Management System is a web-application which helps the leaders of the Mentorship Network at NYU Abu Dhabi better perform their weekly, monthly, and yearly tasks to upkeep the program. The application will be accessible on the web and login-protected. The application will deliver 10 intertwined key features. The goal of this software system is to allow the mentorship network to support more pairs of mentees and mentors with the same amount of executive team members. The goal of the system is also to ensure that data is protected and preserved for each year in the program.

1.3 Overview

The SRS contains 3 sections, the 1st section is the introduction, it provides the purpose, scope, and overview of the document itself.
The 2nd section provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product. Customers should refer to this 2nd for better understanding of the overall system.
The 3rd section provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify

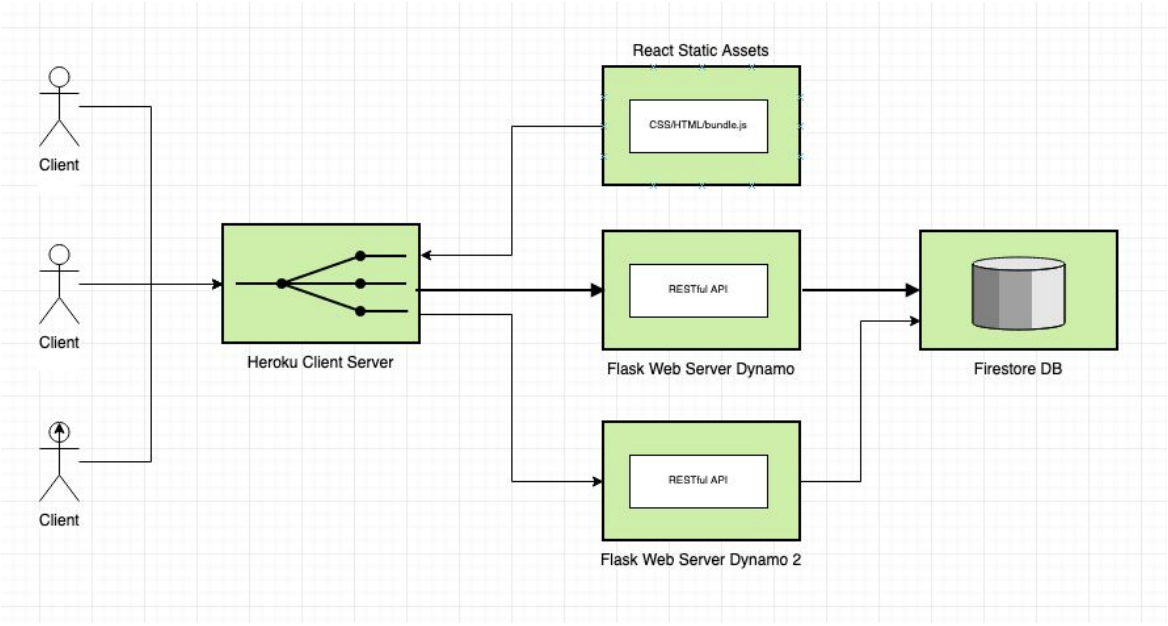
the requirements more precisely for different audiences. This will be used for developers and designers to translate the requirements to a working system. Section 3 is a more technical and in-depth expansion of section 2. Developers and designers should refer to Section 3.

2. Product Overview

This section will give an overview of the entire system, introduce its functionalities, and show how it interacts with other systems. It will also describe what type of stakeholders that will use the system and what functionality is available for each stakeholder. Finally, the constraints and assumptions for the system will be provided.

2.1 Product Perspective

The product will be akin to other mentorship management platforms. However, the product will focus on mentorship management and not an interface for mentors and mentees. This is an important distinction. The Mentorship Network has expressed its desire to remain a face-to-face program, with little technological reliance with mentors/mentees. This software is meant to aid the executive team's processes and thus is unique.



The diagram shown above displays the major components of the system.

2.2 Product Functions

The product functions can be summarized to the following 10 features: more detailed descriptions for each can be found in section 3.

1. Registration (Login/Logout) of System
2. Upload Data
3. Edit Users
4. Display Database
5. Check Database
6. Download Database
7. Resolve Match Requests
8. Send Emails
9. Generate Statistics
10. Submit Match Request

2.3 User Characteristics

There are 3 users which interact with the system

1. Admins
2. Board Members
3. Students

Admins and Board Members are often described as users together. Admins are a subset of Board Members which privileged access. Board members are members of the leadership board of the Mentorship Network. Admin includes the director of the career development center and the leadership team's executives of the mentorship network. Students are participants in the Mentorship Network. They do not have password protected access and can only use the feature to request with a mentor.

2.4 Constraints

- Board members will change each semester.
- No members of the team will be assumed to be technically inclined and able to debug any application errors
- No members of the team will be assumed to be able to use the command line

2.5 Assumptions

- The application will not need to scale beyond the use of 50 users.
- The application will not need to handle more than 10 processes running at the same time.
- The database will not need to hold more than 500 entries combined with: board members, mentors, mentees, and match requests.
- There will always be admins among the board members: CDC representative or executive's in the leadership team

- The board members will not abuse the system and perform deletions from the database with malintent

3. Specific Requirements

3.1 Functional Requirements

Registration (Login/Logout) of System

The users (board members and admins) should be able to login and log out of the system. Furthermore, they should be allowed to select if he or she wants the application to use a cookie to remember them. This will allow them to bypass the login screen on the same browser next time they reach the website. In the event that an incorrect or null username or password is passed, the system should tell the user that the login failed.

Update Database

The application should support admin actions including upload new data to the dataset. When new data is uploaded which contains existing items, the admin should be allowed to select whether he/she wants to wipe the dataset or merge the database. Merging the database will update the database with supplied data on existing keys.

Edit users

The application should allow the admin to add and remove users. Upon the addition of a user the type of user should be specified: board member or admin. This will be done in the admin dashboard.

Display Database

The application should display a dashboard of the database to logged in users. This database will include the mentees in the left column and mentors on the right column. There will be a search bar for both mentors and mentees. Users will be able to search through all keys and on certain keys for entries in the database. The users will also be able to select all or select specific mentors or mentees and then download or delete entries from the database. This dashboard will be located on the main landing page of the application once the user is logged in.

Check Database

The application should allow the user to search, filter, and delete from the database through the database dashboard. Please check the display database section above for how check database features will be accessible through the display database.

Download Database

The application should allow the user to delete selected mentors or mentees from the database. The delete feature will be accessible through the display dashboard. Please check the display database section above for how delete database features will be accessible through the display database.

Resolve Match Requests

The application will display match requests and allow users to view and delete them individually. The display will be at the top of the logged in dashboard. It will be located above the database display. Once clicked on, a dropdown will show all match requests. Users will then be able to see the match requests and delete them individually once processed.

Send Email

The application should support mail merge email functionality. This will send an email through the user's gmail account. The interface for the email functionality will be integrated through the database dashboard display. Once mentors or mentees are selected, a send email button will pop up. The user will be able to start sending an email to the selected mentors once that button is clicked. The email functionality will allow users to add a subject line, a body, and attach attachments.

Generate Statistics

The application will generate a program statistics dashboard with the data in the database. The front-end should display program statistics based on the uploaded data. These include but are not limited to:

- Number of different nationalities in the program
- Male to Female ratio in the program
- Industry Breakdown
- Major Breakdown

The display should be interactive allowing users to query based on mentors, mentees and specific subsets of mentors and mentees.

Submit Match Request

The application should include a dashboard for students to request to match with a mentor. This will be located on the home page before the login screen. This will only be accessible for non-logged in students. These students will be able to view all mentor information without their email. Students can then select a mentor. Once they select a mentor, they will be asked to add

their name, net id, and reason for wanting to connect with said mentor. They will then submit the request. This will be processed in the backend and stored in the database.

3.2 External Interface Requirements

Google Mail

Google mail will need to be integrated in the application. This will so that users can send emails from the application. The standard for google mail logins is to use google oauth. This will require a special verification by users to allow the application to send emails on their behalf.

CSV Downloads/Uploads of Data

The mentors and mentees database tables should be both uploadable and downloadable in csv format. The csv will contain a row of headers as the first row.

3.4 Non-Functional Requirements

Testing

Testing will be an important non-functional requirements of the application and system. The system should support testing in isolation. This means that a new database will be spawned for testing and a new environment will be used for the codebase during testing. This will allow testing to be separated from production code.

Running in Production

The application must be able to run in production as a web-accessible application. This means that the front and back end will be hosted. The server will also be able to support up to 10 simultaneous users. The database will be able to hold up to 500 entries of mentors, mentees, and matches combined. The application will be run in Heroku for scalable production. Heroku is a PaaS, platform as a service.

Security - Encryption, Https

The application should be a secure, encrypted web application. This means that an SSL certificate, secure encryption, and https will need to be employed for the system.

Logging

The system should log requests on the backend to keep track of application usage and track down hard to find bugs.

3.5 Changes to the Requirements

The requirements have morphed and changed as the project has developed through each agile sprint cycle. For this reason, the system request has different requirements then the requirements listed here. The requirements listed here are an updated version of the project requirements compared to the system request.