Software Requirements Specification

for

Advisely

Version 1.1 approved

Prepared by Sam Elliott, Jens Hansen, Connor Langlois, Eli Legere, Eric Schessler, and Spencer Ward

Big Muscle Boys

Sunday, February 24, 2019

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements of the entirety of Advisely, an education support application. Advisely plans to guide student's futures towards academic success by providing a streamlined and extensive way to plan course schedules.

1.2 Document Conventions

Large font size and boldface text represent areas of importance or significance. Requirements are assumed to have their own priority unless where otherwise noted.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is Professor Ghanavati and the members of Big Muscle Boys, including but not limited to project managers, developers, testers, documentation writers, and users.

1.4 Product Scope

Advisely is meant for all students at the University of Maine. Advisely seeks to assist university students in planning. Students will benefit by having access to all potential classes on a single platform and all degree plans on a single platform. They will be provided with tools to plan their schedules based on major, time availability, and personal preference. Advisely will inform them of any schedule conflicts and suggest new classes or degree plans. Currently, there are no platforms like this for University of Maine students.

1.5 References

Material UI: https://material.io/design/introduction/

2. Overall Description

2.1 Product Perspective

Advisely is a new, self-contained product that interfaces with the University of Maine's degree path and course databases. It uses the information it gathers from these sources to provide the basis of its functionality, mainly degree path and course planning for students.

2.2 Product Functions

- The system shall suggest schedule options for the student.
- The system shall build a student's schedule.
- The system shall allow the user to specify course constraints.

2.3 User Classes and Characteristics

The only class of users is University of Maine students. In terms of those students, we've identified three subclasses and their respective characteristics. There are new students, who have no previous schedules and thus will need more guidance then returning students. The returning students will have experience setting up of semester's schedule and know their degree plan. For that reason returning students will use less of the functions that Advisely provides. There are also transfer students. These are the students new to the university but have college credits. They will then have to learn about the University's requirements.

2.4 Operating Environment

The software will operate as a web application, allowing its use in any modern browser on any modern operating system. Specifically, the software will target Google Chrome 72, Mozilla Firefox 65, Microsoft Edge 18, Safari 12, Opera 58, and Microsoft Internet Explorer 11 on operating systems Windows, Mac OS, and Linux where each respective browser is available.

2.5 Design and Implementation Constraints

There are currently no design or implementation constraints.

2.6 User Documentation

There will be a help section that provides instructions on how the user can utilize all the functions of Advisely. Each function will have a section that provides a clear and concise tutorial on its use.

2.7 Assumptions and Dependencies

The software assumes and depends on the existence and reliability of the University of Maine's degree path and course databases. In addition, the software depends on React as the framework for the software's code and the continuance of browser support.

3. External Interface Requirements

3.1 User Interfaces

The software will utilize the React UI framework Material-UI, which follows Material UI guidelines, to create a seamless experience across browsers, both desktop and mobile. See “Material UI” in “References” for details.

3.1.1 User Interfaces Requirements

Requirement 1: The system shall be navigable by the user within 1 hour of usage 95% of the time.

Requirement 2: The system shall respond to user input within 1 second 95% of the time.

3.2 Hardware Interfaces

Advisely is a web-based application and thus will run on any device so long as that device has a supported browser (see section 2.4). The database of classes and degree plans will be stored on a server. That will be accessed by the users when requested.

3.3 Software Interfaces

Client-side environment:

The client-side will be written with the React 16.8.3 JavaScript library. React is a framework for building user interfaces and single-page applications. All communication with the server will be done with the JSON data format.

Server-side environment:

The server-side will be written in JavaScript with Node.js 11.10.0 through Firebase Cloud Functions. Node.js is a web server written in JavaScript. Firebase is a BaaS (Backend-as-a-Service) that allows easy integration with single-page applications. All communication with the client will be done with the JSON data format.

3.4 Communications Interfaces

Advisely requires communication through the web browser, as it is designed as a web application. The system communicates through local client JavaScript to the server, hosted on

Firebase. Communication is done through the JSON data format to both ends across HTTP and HTTPS as needed for security.

4. System Features

4.1 Degree Planning

4.1.1 Description and Priority

Degree Planning is the backbone of Advisely. This feature consists of the macro and micro scales of the application. At the macroscale, Advisely will take the student's major into consideration and find the required courses they must take to complete that degree. Furthermore, it will recommend a schedule for the degree where it makes sense to take each course.

At the microscale, Advisely allows the user to search for courses they would like to take that fit their schedule. Additionally, it will recommend additional courses they may like to take.

Priority: High

4.1.2 Stimulus/Response Sequences

To take advantage of degree planning, users must be logged into the system and currently located on the homepage of Advisely. Users will then navigate to the "Degree Planning" page to search for other courses and receive recommendations from Advisely.

4.1.3 Functional Requirements

Requirement 1: The system shall provide users with a suggested full-degree schedule.

Requirement 2: The system shall allow the user to build new schedules for each semester.

Requirement 3: The system shall suggest elective courses that are required for degree completion when available.

4.2 Degree Progress

4.2.1 Description and Priority

Degree Progress tracking is one of the many ways that Advisely will help alleviate the stress of degree planning. It will take into account the courses that users have taken, and make sure that the users are on track to graduating on time. Thus this feature will consist of tracking courses users have entered into their Advisely schedule, the courses they have entered, and the degree paths they have selected to ensure optimal completion. the priority of this feature is very high, as this is the backbone of Advisely.

Similarly, it will detect when users have nearly completed any minors or majors outside of their main degree. If so, it will recommend courses to students so that they

might complete extra degrees. This feature is of medium priority as it is an important but secondary feature of Advisely.

4.2.2 Stimulus/Response Sequences

To track their degree progress, users need only log into their account and navigate the website's homepage in order to see their selected degrees progress. And to receive degree recommendations, users need only enter courses that they've taken into the website. When a degree is detected which is feasible for user completion, a popup will appear wherever the user is on the website.

4.2.3 Functional Requirements

Requirement 1: The system shall suggest secondary majors to users when users have completed at least half of the courses for the suggest major.

Requirement 2: The system shall calculate the number of credits remaining until degree completion.

Requirement 3: The system shall allow the user to enter all the classes they have already completed.

Requirement 4: The system shall allow users to identify the majors and minors they intend to complete.

Requirement 5: The system shall suggest secondary minors once users have completed at least half of the required courses.

Requirement 6: The system shall display the number of credits remaining until degree completion.

4.3 Data Persistence

4.3.1 Description and Priority

Users will be able to create Advisely accounts using OAuth using their university email accounts in order to access their schedules on demand. Thus the schedules that they build will be available throughout the entirety of their degree and beyond. As well as this, the system will support importing and exporting of schedules into various formats. These features are tantamount to the application, as they allow for degree planning in general. Thus, this is a high priority feature.

4.3.2 Stimulus/Response Sequences

Whenever a user logs into their Advisely account and alters any of the information on their account, the system shall store that information within the database.

4.3.3 Functional Requirements

Requirement 1: The system shall allow users to create an account using their university email address.

Requirement 2: The system shall store all changes made by the user.

Requirement 3: The system shall allow the user to import schedules that others have made.

Requirement 4: The system shall allow the user to export their custom schedules for sharing.

5. Other Nonfunctional Requirements

Requirement 1: The system shall only collect course completion data.

Requirement 2: The system shall have an uptime 99%.

Requirement 3: The system shall have a response time of fewer than 4 seconds 96% of the time.

Requirement 4: The system shall be available on all modern web browsers.

Requirement 5: The system shall log users out after 15 minutes of inactivity.

Requirement 6: The system shall be accessible such that after 45 minutes of use, users will make fewer than 1 mistake per hour.

Requirement 7: The system shall be compliant with the California Consumer Privacy Act of 2018.

Requirement 8: The system shall store only encrypted passwords of the user.