[Adjenda]
Software Requirements Specification
Version <1.0>

Revision History

Date	Version	Description	Author
<10/18/2021>	<1.0>	SRS 1.0	Group-2
<date here>	<version here>		
<date here>	<version here>		
<date here>	<version here>		

Software Requirements Specification

1. Introduction

The production and revision of this SRS document serves to translate the business requirements of our classroom collaboration software, “Adjenda” into a set of features and functionalities that provide technical definition to said requirements. This document can help us organize our ideas in a way that will make implementation feasible while also targeting the needs of the product owner and retaining the original requirements from the problem statement in the system description.

1.1 Purpose

The purpose of this SRS document is to define specific criteria for the production of “Adjenda”. Decisions regarding the technology suite for the production of our software solution can be assessed and refined within this document to better ensure compatibility of all the features which we intend to deliver in the final product. This document can contain all of our design concepts and allow us to easily weigh the pros and cons of implementation on a per feature basis. Organizing our technical concepts can assist in ensuring our software is compliant with the user’s needs as well as being straightforward for the ease of user interaction. Also, we can further meet the potential needs and use cases of the users of this system by keeping the user experience in mind as we refine technical requirements.

This SRS should effectively provide definition and documentation for users, clients, product owners and all parties associated with the development of “Adjenda”.

1.2 Scope

The scope of this project through the current semester will include all phases of the software development life cycle up to, and including some of, implementation. The website’s design and user interface will both be completed, with the student and instructor views displaying all of their planned pages and navigational functions. Much, if not all, of the database itself will also be designed and written out, although it will not be entirely interfaced with the website’s front end before the end of this semester.

Fully functioning course displays, class roster views, classroom attendance logging, collaborative student groups, calendar events, and instructor created polls will not be completed before the end of this semester, and thus are out of the current scope. However, these features will still be visible to a certain extent within the website’s layout.

1.3 Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification

SQL	Structured Query Language
HTML	HyperText Markup Language

2. Overall Description

As more and more institutions are seeking to optimize their online operations. Having a reliable and efficient student portal is becoming an imperative must. With the current events going on as institutions are beginning to regain their on-campus operations, it has become clear that there needs to be a reliable system that can help streamline classroom activities and functionalities. As well as aiding students with their current methodologies of garnering educational success with tools that provide opportunities of a better collaborative experience amongst other students and their instructors. The COVID-19 pandemic has been an educational experience in itself for students and institutions. And as classes begin to transition from a virtual online environment back to traditional in-person classrooms. It is favorable to retain some virtual tools and improvisations made during the lockdown era of education and implement them in current in-person classes in an unprecedented level of hybridization furthering the symbiotic relationship between student and classroom technology. This is where the proposed system-to-be comes into play.

The proposed software application, Adjenda, is a student portal that institutions and individual instructors can provide to their students to better organize and conduct classes and assignments. However, Agenda is capable of being more than just the typical student portal. Adjenda will be loaded with an abundance of quality of life features such as taking attendance, creating polls, and an intuitive calendar system that can help students and instructors conduct and organize their classes and schedules with an unprecedented level of detail just to name a few. Adjenda makes it an emphasis to provide the best user collaboration experience in that students will be able to create groups to collaborate, share documents, share calendar schedules, and communicate with each other. With all of its features Adjenda takes the best of both worlds of online classroom collaboration and an in person educational experience.

3. Specific Requirements

The specific requirements are –

3.1 Functionality

This subsection contains the requirements for the software-to-be. These requirements are organized by the features discussed in the Proposal Document.

3.1.1 ***Account Logins.***

3.1.1.1 The system shall allow users to login to an account.

*3.1.2 **Create Accounts.***

3.1.2.1 The system shall allow users to register a new account.

3.1.2.2 The system shall assign users to a student or teacher account based upon their provided email address.

*3.1.3 **Provide access to class rosters.***

3.1.3.1 The system shall allow all users to view their courses' class rosters.

3.1.3.2 The system shall allow instructors to create class rosters.

3.1.3.3 The system shall allow instructors to edit their class rosters.

*3.1.4 **Log Attendance.***

3.1.4.1 The system shall allow enrolled students who are in class to sign-in for the day.

3.1.4.2 The system shall allow instructors to generate room codes for student attendance.

3.1.4.3 The system shall allow students to enter a code to log their attendance.

3.1.4.4 The system shall allow instructors to view the attendance roster.

3.1.4.5 The system shall allow instructors to download the attendance roster.

*3.1.5 **Create Collaboration Groups.***

3.1.5.1 The system shall allow students to produce groups in a given class.

3.1.5.2 The system shall allow students to send invitations to join a group.

3.1.5.3 The system shall allow students to request access to join a group.

3.1.5.4 The system shall allow students to accept or deny requests to join a group.

*3.1.6 **Access to a Calendar.***

3.1.6.1 The system shall allow instructors and students to view courses on a calendar.

3.1.6.2 The system shall allow instructors and students to create and edit events on a calendar.

3.1.6.3 The system shall allow instructors and students to view their activities.

3.1.7 ***Provide poll functionality.***

3.1.7.1 The system shall allow instructors to create polls for their students within a given course.

3.1.7.2 The system shall allow students to view created polls for their courses.

3.1.7.3 The system shall allow students to submit their response to polls.

3.1.7.4 The system shall allow instructors to view the results of their polls.

3.1.7.5 The system shall allow instructors to download the results of their polls.

3.1.8 ***Navigation Bar***

3.1.8.1 The system shall allow users to navigate to different pages via the navigation bar.

3.1.8.2 The system shall allow users to select the dashboard page.

3.1.8.3 The system shall allow users to view their enrolled classes on the dashboard.

3.1.8.4 The system shall allow users to select the calendar page.

3.1.8.5 The system shall allow users to select the settings page.

3.1.8.6 The system shall allow users to select the option to log out of the system.

3.1.9 ***Class Creation***

3.1.9.1 The system shall allow instructors to create courses

3.1.9.2 The system shall allow instructors to determine a schedule for the given course

3.2 Usability

3.2.1 ***Graphical User Interface***

The user interface shall provide a navigation system on every page for convenient access to all functions.

The user interface shall provide a consistent look and feel between all pages of the site.

The user interface shall provide a comprehensive and consistent calendar view.

3.2.2 Accessibility

The system shall allow users to access the settings page via the navigation system.

The system shall allow users to configure the settings of their account.

The system shall allow users to configure minor details of the software.

3.3 Reliability & Availability

The system shall be hosted from a persistent server allowing for non-local access to its various functionalities.

The system shall persist all of the user accounts on the database.

The system shall allow registered users to login at any given time.

3.4 Performance

The system shall be web based and lightweight enough to run on a wide array of devices.

3.5 Security

3.5.1 Account authentication

The system shall verify only accounts which register using a montclair.edu email.

The system shall automatically assign student and professor accounts based on the email provided to prevent students from changing class rosters, grades, etc…

The system shall allow students to only view group collaboration pages if they have accepted an invitation to said group and are considered a member.

The system shall automatically encrypt user credentials within the database of the system.

3.5.2 Sign in verification

The system shall verify valid student attendance recording via a randomly generated sign in code, this code is given to the instructor who then gives it to the students.

3.5.3 Password Modification

The system shall allow users to change the password associated with their individual account.

3.6 Design Constraints

3.6.1 Development Tools

The system will be built using standard web page and database development languages such as HTML, Javascript, Typescript and SQL.

3.6.2 System Type

The system will begin as a web-based application with potential plans to expand to a mobile application.