Software Requirements Specification

for

<University Faculty Course Assignment System>

Term Project - Group 06

Prepared by <Group Members >

Name: Rotan hawlader pranto	**ID:2013193642**
Name: Sharjina Jahan	**ID: 2013399642**
Name: Md. Mohibul Islam	**ID: 2021077642**
Name: Iftekher Mahbub Rafi	**ID: 2021463642**

Section: CSE 327.3

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to specify the requirements for the development of a web-based application that will be used for the allocation of course sections to faculty members based on certain business rules. This document defines the scope of the project and outlines the functional and non-functional requirements that need to be met in order to satisfy the project's objectives. The SRS covers the entire application and its various subsystems. It is intended to be used as a reference by the development team and stakeholders to ensure that the final product meets the stated requirements.

1.2 Document Conventions

The IEEE SRS standard is followed.

1.3 Intended Audience and Reading Suggestions

Intended Audience:

This document's primary intended audience are the software developers. The different users (Teachers and students) can read the introduction and system features for better understanding.

Reading Suggestions:

***Developers**: The developers should start by reading the overview sections, including the purpose, product scope, and product perspective. Then, they should focus on the functional requirements and non-functional requirements sections to understand the technical details of the system.*

***Project Managers:** The project managers should begin by reading the purpose, product scope, and business rules sections to understand the objectives and constraints of the project. Then, they should focus on the functional and non-functional requirements to ensure that the project meets the specified requirements.*

***Testers**: The testers should start by reading the functional and non-functional requirements sections to understand what features and performance criteria need to be tested. They should also review the business rules section to ensure that all the rules are enforced properly.*

***Documentation Writers:** The documentation writers should begin by reading the purpose and product scope sections to understand the objectives of the project. Then, they should focus on the functional and non-functional requirements to understand the features and performance criteria of the system.*

1.4 Product Scope

The software is intended to improve the efficiency of the allocation process while ensuring that it adheres to certain business rules, such as the maximum number of credits that can be assigned to a faculty member, one faculty can not be assigned multiple class at the same time, one faculty cannot view other faculty's schedule and the avoidance of scheduling conflicts, faculty can request to change section if any inconvenient situation occurs. The benefits of this software include reducing the workload of the allocation staff, ensuring a fair distribution of course sections among faculty members, and optimizing the use of resources.

1.5 References

Software requirement specification (SRS) format.

https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-

document

IEEE srs format

https://goo.gl/nsUFwy/.

2. Overall Description

2.1 Product Perspective

The Course Section Allocation System is a new software application that helps educational institutions assign courses to faculty members. The system replaces manual processes and interfaces with existing student information systems and user authentication systems. It generates *schedules based on business rules and generates reports and class routines based on faculty and course schedules. The system is a standalone application that exports data to other systems as needed.*

2.2 Product Functions

The Course Section Allocation System must be able to perform the following major functions:

- *Allow administrators to input and manage course information, including course name, room number. number of credits, section, day and times.*
- *Allow administrators to input and manage faculty information, including faculty name, designation, email,EXT, room number, department, mobile number and maximum number of credits that can be assigned.*
- *Generate schedules based on business rules, such as not assigning more than 11 credits to a faculty member and not assigning multiple classes in the same slot for a particular faculty member .*
- *Generate reports based on faculty and course schedules, including faculty-wise, course-wise, parallel course-wise, date-wise and room-wise schedules.*
- *Interface with existing student information systems to ensure that schedules are accurate and up-to-date.*
- *Interface with user authentication and authorization systems to ensure secure access to the application.*
- *Export data to other systems as needed.*

2.3 User Classes and Characteristics

The Course Section Allocation System is mainly designed for administrators in educational institutions who will use the system to assign course sections to faculty members. These administrators may have varying levels of computer expertise but should be comfortable using a web-based interface. Other users, such as faculty members and department heads and students may also interact with the system in a more limited way. The administrators are the most important user class, as they will rely on the system to accurately and efficiently allocate course sections.

2.4 Operating Environment

The Course Section Allocation System is a web-based application that can be accessed through common web browsers on any desktop or laptop computer. It requires a web server and a database server to run, and users will need an internet connection to access it. The system should work alongside other software used by the educational institution, such as student information systems and user authentication systems, without any issues.

2.5 Design and Implementation Constraints

● The web application must be accessible through the latest internet browsers.
● The website frontend will be built using react-js. The backend will be built with Node-js for databases, We will use MongoDB. The backend and the frontend will communicate via api calls.
● Functions such as course assigning , updating course , generating course everything will be properly arranged.

2.6 User Documentation

The user manual will be provided to the teacher and student on how to use the software.

2.7 Assumptions and Dependencies

Assumptions:

- *We assume that the data we have is correct and complete.*
- *We assume that we have access to the necessary tools to build the system.*
- *We assume that other tools we need from third-party companies will work with our system.*
- *We assume that our team has the skills they need to build the system.*

***Dependencies**:*

- *We may need to use tools from other companies to do certain things, like check that users are who they say they are.*

- *Our system might need to work with other systems that already exist, like student information systems. This might need more work to make it happen.*
- *We might depend on things outside our control, like making sure all our tools work together well.*

3. External Interface Requirements

3.1 User Interfaces

There will be separate login options for the administrators, teachers and students. Each of the options will contain different different functions such as

- *admin: All the information of the teacher and courses will be provided. and a comment box will be provided where he can see the teacher's request.*
- *teacher. A teacher can only see his information and assigned sections . if there is any inconvenient situation there will be an option to change the section.*
- *students: There will be some options to generate courses.*

3.2 Hardware Interfaces

A laptop or desktop that can support latest web browsers(chrome, firefox)

3.3 Software Interfaces

The web-based application will be built using the MERN stack, which includes the following software components:

- *MongoDB (version 4.4 or later) as the database management system*
- *Express.js (version 4.x or later) as the web application framework*
- *React.js (version 17.x or later) as the front-end JavaScript library*
- *Node.js (version 14.x or later) as the server-side JavaScript runtime environment*

The application will communicate with the MongoDB database using the MongoDB Node.js driver (version 4.x or later). The Express.js application will expose RESTful APIs that the React.js front-end will consume using the Axios HTTP client (version 0.21.x or later).

The application will run on a web server that supports Node.js applications, such as Apache or Nginx, and will require access to a MongoDB database server. The application will be developed using Visual Studio Code (version 1.56 or later) as the primary integrated development environment (IDE) and will be version-controlled using Git.

3.4 Communications Interfaces

For this project, the web-based application will require internet connectivity to communicate with the server to retrieve and store data. The application will use HTTP protocol to communicate with the server, and data transfer will be secured using encryption mechanisms such as SSL or TLS. The application will also require a compatible web browser to access and use the application.

4. System Features

4.1 Course Section Allocation

4.1.1 Description and Priority

The Course Section Allocation feature will allow the system to allocate sections of different courses to faculty members based on various business rules. This feature is of High priority as it is critical to the core functionality of the system.

4.1.2 Stimulus/Response Sequences

The following is the sequence of actions and responses required to stimulate the behavior of the Course Section Allocation feature:

- *User selects a course and clicks on the "Assign Faculty" button*
- *System displays a list of available faculty members based on their credits and schedule*
- *User selects a faculty member and clicks on "Assign" button*
- *System assigns the selected faculty member to the selected course section*

4.1.3 Functional Requirements

1. User Management: The application should allow administrators to create and manage user accounts for faculty members and administrators.

2. Course Management: The application should allow administrators to create and manage courses, sections, and prerequisites.

3. Faculty Management: The application should allow administrators to manage faculty member profiles, including personal and professional information.

4. Allocation Management: The application should allow administrators to allocate sections to faculty members based on business rules.

5. Reporting: The application should generate reports based on the following criteria:

- *Faculty-wise routine*
- *Course-wise routine*
- *Room-wise routine*
- *Parallel course-wise routine*
- *Date & time-wise routine*

6. Schedule Management: The application should allow administrators to create and manage schedules for each course section, including start and end times, days of the week.

7. Conflict Management: The application should be able to detect and resolve scheduling conflicts between faculty members, courses, and rooms.

8. Course Section Availability: The application should allow administrators to specify the availability of each course section for allocation, based on factors such as enrollment limits, room capacity, and faculty workload.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software can be run on any desktop or laptop with at least 1.3 gigahertz CPU and 2 gigabytes of RAM.

5.2 Safety Requirements

Hopefully no error will occur while using the software.

5.3 Security Requirements

Teachers can't view other teachers information and students can't view teachers personal information and must not able to do any task except generating courses

5.4 Software Quality Attributes

software have to be convenient for everyone and follow every functional and nonfunctional requirements

5.5 Business Rules

The following business rules should be considered for the web/mobile application:

- *A faculty member cannot be assigned more than 11 credits.*
- *A faculty member cannot be assigned multiple classes in the same time slot.*
- *Only authorized personnel can access and modify the course and faculty data.*
- *Courses can only be assigned to faculty members who are qualified to teach them.*
- *Course allocations should be fair and unbiased.*
- *The application should validate the input data to ensure accuracy and consistency.*
- *The application should maintain a log of all user actions for audit purposes.*