Software Requirements Specification

for

Computer Science Schedule Generating System

Version 1.0

Prepared by Jiaqi Yang, Runzhi Zhou, Zijun Liu, Tao Huang, Jieyu Ren

CWRU EECS393

September 23, 2019

Table of Contents

1. Introduction

1.1 Purpose: This SRS describes software functional and nonfunctional requirements for the upcoming release of the Computer Science Schedule Generating System. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0, which is the first demo on October 24.

1.2 Vision: For students major in Computer Science at Case Western Reserve University who are tired of double-checking the major bulletin and confused by the complicated course requirements, we introduce a web application that collects their information and generates their next-semester schedules in a semi-automatic manner.

1.3 References

Re-1: "2019-20 General Bulletin," Case Western Reserve University. [Online]. Available: http://bulletin.case.edu/collegeofartsandsciences/computerscience.

Re-2: "SAGES Requirements," Case Western Reserve University. [Online]. Available: https://case.edu/ugstudies/programs-and-requirements/sages-requirements.

Re-3: "The Identity Standard," Okta. [Online]. Available: https://www.okta.com/.

2. Overall Description

2.1 Product Perspective: The purpose of this web application is to enable Computer Science Major students in Case Western Reserve University to quickly manage and plan their course schedule. The application will allow users to register and input their information, which consists of their tracks and courses they have taken. After storing their information, they could automatically generate their partial schedules of high-priority courses, which include courses in General Requirements, Core Requirements, Depth Requirements, Breadth Requirements, and pick the most suitable one from all generated partial schedules. From there, they could manually select from a list of low-priority courses, which includes courses in SAGES Requirements and Technical Requirements; the list will dynamically filter the low-priority courses conflicting with selected courses in the schedule. Then the system will generate a schedule based on the selection during the semi-automatic scheduling process.

2.2 User Classes and Characteristics

Student - A student, specifically majoring in Computer Science major at Case Western Reserve University, who wishes to manage and plan their course schedules for next semester in a more intuitive manner. Students are expected to use our application at least once a semester in order to generate their next-semester course schedules.

Administrator - An administrator will be an officer from the Electrical Engineering and Computer Science Department at Case Western Reserve University or a student volunteer. The administrator will acquire an admin account and update the course information, including time slot, availability, prerequisite, and so on, in the admin interface every semester. The administrator will also receive feedback and suggestions from the users like reports of incorrect course information.

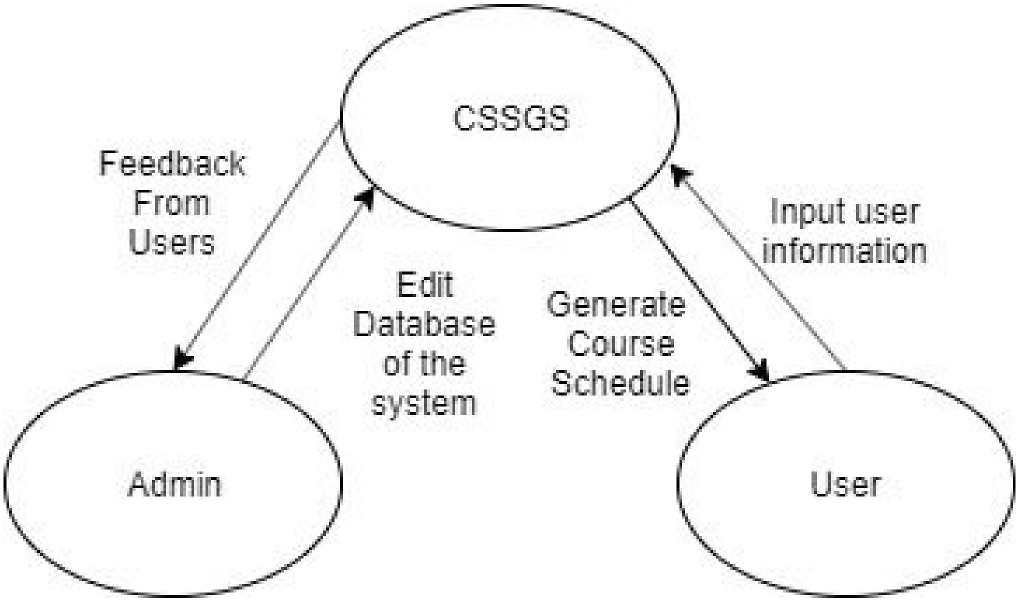


Figure 1
Context diagram for the first demo of the Computer Science Schedule Generating System

2.3 Operating Environment: We will focus on most of the major browsers and operating systems.

OE-1: The Computer Science Schedule Generating System shall operate with the following Web Browsers: Microsoft Edge version 44, Firefox version 69, and Google Chrome version 77.

OE-2: The Computer Science Schedule Generating System shall operate with the following operating system: Windows OS, Mac OS, and Linux Ubuntu OS.

2.4 Design and Implementation Constraints

CO-1: The Computer Science Schedule Generating System shall use the open-source PostgreSQL server, version 12.

CO-2: All HTML code shall conform to HTML 5 standard.

CO-3: All CSS code shall conform to CSS 3 standard.

CO-4: All application functionality and classes shall be written in Java and JavaScript.

2.5 User Documentation

UD-1: The Computer Science Schedule Generating System will have a help subsite of the basic user manual that will provide documentation and frequently asked questions to assist users.

2.6 Assumptions and Dependencies

AS-1: The developers assume that users will double-check the course availability and graduation status with their advisors and within the official SIS system.

AS-2: Students should send course permission requests by themselves whenever a course from the generated schedule is full.

AS-3: The administrators update the course information in a timely manner.

DE-1: The users have access to a stable Internet connection in order to store their input information into the database.

3. System Features

3.1 Registration: The system shall let a computer science major student at Case Western Reserve University to create a new account.

3.1.1 The user shall be able to set a unique ID and a password.

3.1.2 The user shall be able to log into the system with the account created.

3.2 Information Collection: The system shall collect the necessary information from users and save it in the database.

3.2.1 The user shall be able to input their tracks and courses taken.

3.2.2 The user shall be able to add, delete and modify the information entered.

3.2.3 The users shall not have to enter information every time they log in.

3.3 Administration: The system shall provide an admin account and an independent interface. The administrator shall be able to manage the system.

3.3.1 The system shall provide an independent and manageable interface for the administrator.

3.3.2 The administrator shall be able to add, delete and modify course information in the database through the interface.

3.4 Semi-automatic Course Scheduling: The system shall schedule high-priority courses automatically and the user shall be able to select low-priority courses manually.

3.4.1 The system shall consider the high-priority courses based on priority (General Requirements > Core Requirements > Depth Requirements > Breadth Requirements) and generate multiple plans of recommended combinations of High Priority Courses according to users' information.

3.4.2 The user shall be able to choose one of the plans for the high-priority courses generated by the system.

3.4.3 The system shall provide all the available low-priority courses that have not been taken by the user and are not conflict with the existing schedule and present them in a list view.

3.4.4 The user shall be able to select low-priority courses provided by the system to be added to the schedule.

3.4.5 The system shall dynamically remove courses with time conflict and update list the courses that are still available to be added to the schedule.

3.4.6 The user shall be able to add courses available from the list and delete courses from the schedule.

3.4.7 The system shall generate a table of the final schedule of the courses.

4. External Interface Requirements

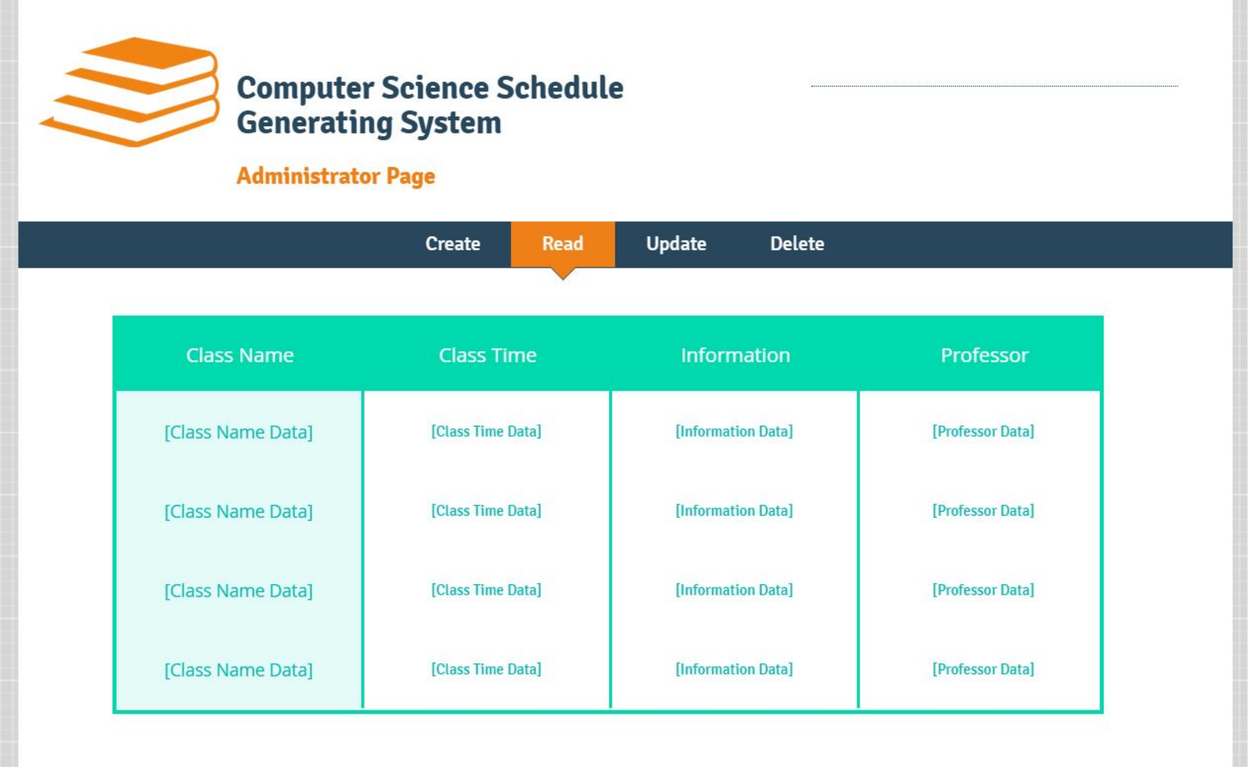
4.1 User Interfaces

UI-1:   The Computer Science Schedule Generating System screen displays shall show users several recommended schedule.

UI-2: The system shall offer users with a help link for each HTML page to explain how to use that page.

UI-3: The Web pages shall allow users to fill in personal information and to select classes using keyboard alone, in addition to using both mouse and keyboard.

UI-4: The admin page shall allow the administrator to add, delete and modify the database of the system.



Class Name	Class Time	Information	Professor
[Class Name Data]	[Class Time Data]	[Information Data]	[Professor Data]
[Class Name Data]	[Class Time Data]	[Information Data]	[Professor Data]
[Class Name Data]	[Class Time Data]	[Information Data]	[Professor Data]
[Class Name Data]	[Class Time Data]	[Information Data]	[Professor Data]

Figure 2
Concept graph of the administrator page for the first demo of the Computer Science Schedule Generating System

4.2 Hardware Interfaces

No hardware interfaces have been identified.

4.3 Software Interfaces

No software interfaces have been identified.

4.4 Communication interfaces

CI-1: The web pages will provide users with an email address to contact the administrator.

5. Other Non-functional Requirements

5.1 Usability Requirements

US-1: The system should be efficient to use. In particular, 80% of the registration phase should take no longer than 1 minute; the whole process, including the user checking the courses already been taken and the preferred courses, should take no longer than 10 minutes.

US-2: The system should give effective guidance so that the whole process is easy to follow and understood by 90% of the users.

US-3: 80% of the users shall be satisfied with the final schedule generated. In particular, the courses selected should be in most priority or preferred by the users.

5.2 Reliability Requirements

RE-1: 99% of the generated schedules should provide rational results. In other words, it should only select courses available during the upcoming semester. If the user does not meet the prerequisites of certain course, it should not appear on the generated schedule.

RE-2: 99% of the course information (i.e., course prerequisites, regular class period, class capacity) stored in the system's database should be correct and easy to maintain or change.

5.3 Serviceability Requirements

SV-1: The system should provide access for users to give feedback and suggestions.

SV-2: The technical support staff should be noticed whenever feedback is given.

5.4 Security Requirements

SC-1: The system shall be able to differentiate student user accounts and administrator accounts. Only administrators are able to change the course information.

SC-2: Users shall be required to log in to the Computer Science Schedule Generating System for all operations.

5.5 Availability Requirements

No availability requirements have been identified.

6. Scope and Limitations

6.1 Scope of Initial Release: The scope of the first release, which will be presented in the first demo on October 24, focuses on putting a product out for our own use and probably our friends' use for testing and feedback. In later releases, the web application would be scaled up to the entire campus.

System Features	Scope of Initial Release
3.1	We will be stored in our local database instead of using the third party login service such as Okta [3].
3.2	We will not update the user information based on the generated schedule; instead, the users need to update their information like courses they have taken each semester.
3.3	We will only provide a CRUD (Create, Read, Update, and Delete) admin panel to update course information, but not provide any means to reset users' passwords.
3.4	We will take course prerequisites, time conflict, and course priority into consideration when generating course schedules, but not include user preference, course rate, and course difficulty.

6.2 Limitations

LI-1: The Computer Science Course Generating System does not provide password reset function, therefore if the user forgets their username or password, they will not be able to retrieve the account.

LI-2: The Computer Science Course Generating System does not have any software interfaces with any official software provided by Case Western Reserve

University, therefore the course information might not be updated in a timely manner, and the future update of major bulletin might not be reflected in a timely manner.

7. Inspection Report

Name	Inspection Findings	Resolution
Jiaqi Yang	● Lackness of graphs makes it unclear how our system operates.	● Adding Application Flowchart and
Zijun Liu	● The nonfunctional requirements (section 5.1 and 5.2) are not very realistic. (i.e., It's almost impossible to make every user satisfied with our system.)	● The requirements should be quantitative.
Tao Huang	● There will be no software interfaces in the system. ● Missing information on communication interfaces.	● Deleted section 4.2 and added information about communication interfaces.
Jieyu Ren	● Section 2.1 contains redundant information.	● Went through the section and deleted redundant information.
Runzhi Zhou	● The functional requirements stated in section 3.4 are not clear and specific.	● Went through the section and updated the requirements with specific details.

8. Revision History

Document Authors	Date	Reason For Changes	Version
Jiaqi Yang, Zijun Liu, Jieyu Ren, Tao Huang, Runzhi Zhou	09/18/2019	Initial draft	1.0 draft
Jiaqi Yang, Zijun Liu, Jieyu Ren, Tao Huang, Runzhi Zhou	09/23/2019	baseline following changes after inspection	1.0

Appendix A: Acronym and Terminology Dictionary

- **Track** - Track entered by a user includes 6 tracks listed as Areas of Computer Science Depth Requirement in Case Western Reserve University official bulletin [1].
- **Course Priority** - Course priority, defined by Computer Science Course Generating System, ranks the different requirements in the decreasing order of General Requirements, Core Requirements, Depth Requirements, Breadth Requirements, SAGES Requirements, Technical Requirements.
- **High priority courses** - High priority courses include courses in General Requirements, Core Requirements, Depth Requirements, and Breadth Requirements.
- **Low priority courses** - Low priority courses include courses in SAGES Requirements and Technical Requirements.
- **General Requirements** - General Requirements include 10 courses listed as Major Requirements and 1 course listed as Statistics Requirement in Case Western Reserve University official bulletin [1].
- **Core Requirements** - Core Requirements include 6 courses listed as Computer Science Core Requirements in Case Western Reserve University official bulletin [1].
- **Depth Requirements** - Depth Requirements include 4 courses of the user-input track listed as Computer Science Depth Requirements in Case Western Reserve University official bulletin [1]. If a course can be used to meet both Depth

Requirements and Breadth Requirements, it will be preferentially counted toward Depth Requirements if the user’s Depth Requirements have not been satisfied.

- **Breadth Requirements** - Breadth Requirements include 5 out of 7 courses listed as Computer Science Depth Requirements in Case Western Reserve University official bulletin [1].
- **SAGES Requirements** - SAGES Requirements include 2 out of 3 SAGES courses from USNA, USSO, USSY listed as University Seminar Requirements in Case Western Reserve University SAGES Requirements bulletin [2].
- **Technical Requirements** - Technical Requirements include at least 3 courses from Group 1 and at most 2 courses from Group 2 listed as List of Approved Technical Electives in Case Western Reserve University official bulletin [1].

Appendix B: Application Flowchart

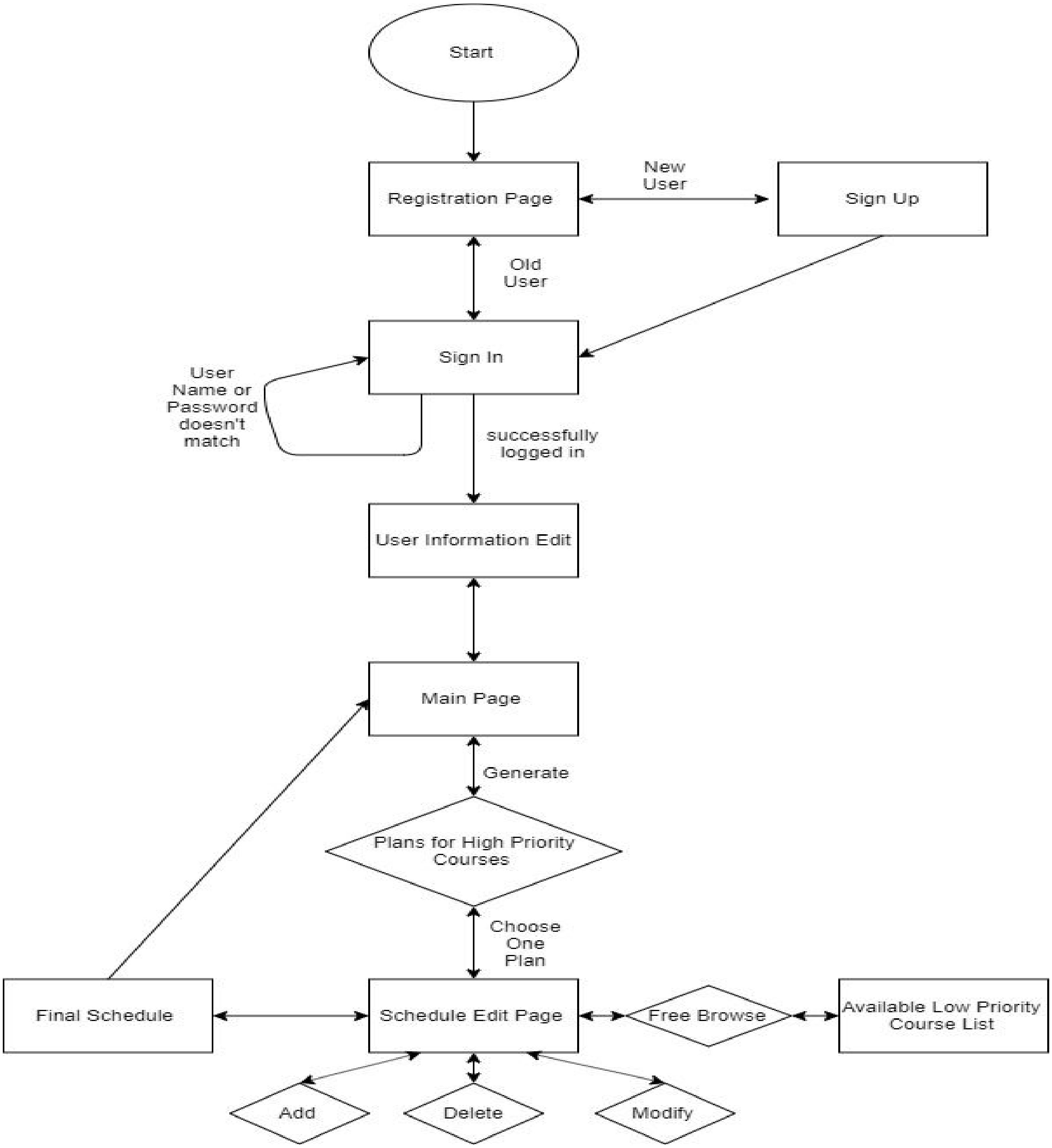


Figure 3

Application Flowchart for the first demo of the Computer Science Schedule Generating System