Software Requirements Specification

for

<School Registrar>

Version 1.0 approved

Prepared by <Everett Hinckley, Frank Torre, Erik Martinez, Rory Japal>

<3/3/24>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to explain the functionality and use cases of the School Registrar Application. This document explains to users how to use the application. It also explains in detail the specifications of the software.

1.2 Document Conventions

The Document was created with the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document shareholders with an interest in the School Registrar Product.
Users, who fall into three categories, Students, Professors, and Administrators.
Programmers who will work on developing the School Registrar Software Application.

1.4 Product Scope

The School Registrar is a system which helps manage a University. It allows students to sign up for their desired courses and create a schedule online. Students can also view their grades for their classes. The School Registrar also helps the University Staff in tracking student registration in courses and helps Professors track their Student's grades.

1.5 References

School Registrar GitHub Page: https://github.com/franktorre99/School-Registrar/tree/main

Figma Low Fidelity Designs:
https://www.figma.com/file/bN4rDvIR5uTicx8Uw8TTdC/Project-Proposal-Mockup?type=design&node-id=0-1&mode=design&t=lX9WGVzANrgaYcy7-0

2. Overall Description

2.1 Product Perspective

The School Registrar is an application that will allow students of a University to register for classes and stay up to date with their grades and academic standing. Students will be able to design their schedules so that they can choose the best times, Professors, and classes available to them. The Students will also be able to easily track their grades in the classes they are enrolled in.

2.2 Product Functions

Type of user followed by the functions they can use.

Login:

All Users:

- Login Request - User sends a request to login at login screen
- Authenticate User - The credentials the user inputs are authenticated.
- Change Password - Function that allows the user to change their password.
- Verify Identity - Verifies the type of user they are, Student, Professor, Admin.

Student Registration:

Student:

- Add Class - The Student adds a course to their schedule.
- Remove Class - The Student removes a course from their schedule.
- Check Requirements - Checks that the student has all the necessary prerequisites.
- Check Availability - Checks that the class is not filled and the student can sign up for that class.
- Check Schedule Conflict - Checks if the student already has a class in that time slot.
- Check Class Exists - Verifies that the class the student inputs exists.
- Search For Classes: Allows students to search for classes by time, professor, and department.

Grade Assignment:

Professor:

- Add Grade - Allows the Professor to store a student’s grade on the system.

- Edit Grade - Allows the Professor to edit a student's grade for an assignment.

- Announcement - Allows the Professor to send messages to all their students at the same time.

Student:
- View Grade - Returns the student's grade. Verifies that the student is the correct user first.

- Contact Professor - Allows students to send a message to their professor.

Adding Classes:

Administrator:
- Insert Class - The Admin creates the classes the University is offering and inputs attributes, for example time, capacity, Professor name

- Edit Class Attributes - The Admin Changes the class attributes

- Delete Class - The Admin removes a class the University is no longer offering.

- Notify Students - Function called at the end of Edit Class Attributes or Delete Class. It informs the students signed up for those classes that a Class Attribute is changed or that the University is no longer offering the course.

Adding Users:

Administrator:
- Create User - The Administrator creates a username and password for students when they first sign up for the University.

2.3 User Classes and Characteristics

Students: Users that want to register for classes, view their grades, and create a class schedule for a semester. Students can't add or edit grades, and also can't create classes.
Professors: Users that can add grades for students of the classes they teach.
Administrators: Users that add classes that can be signed up for by students. Administrators can't edit the grades of students.

2.4 Operating Environment

Windows 7
Windows 8
Windows 10
Mac OS X
Linux

2.5 Design and Implementation Constraints

We will be using Java and JavaFx for the main application functionality and Google Firebase for storage. There will likely be a maximum storage capacity in Google Firebase and a maximum amount of simultaneous requests that can be sent to the School Registrar application.

2.6 User Documentation

We will develop a guide for all the different user types on how to use the system when it is further developed.

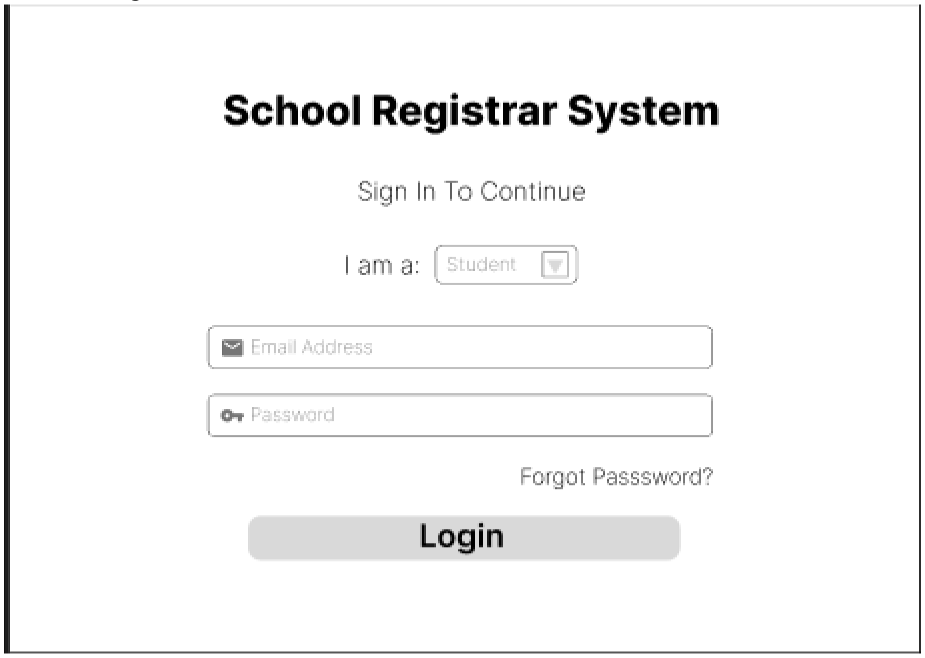
2.7 Assumptions and Dependencies

The School Registrar runs on Java and JavaFx and assumes that all users of the system has both programs installed.

3. External Interface Requirements

3.1 User Interfaces

1. Login Screen:



School Registrar System

Sign In To Continue

I am a: Student

Email Address

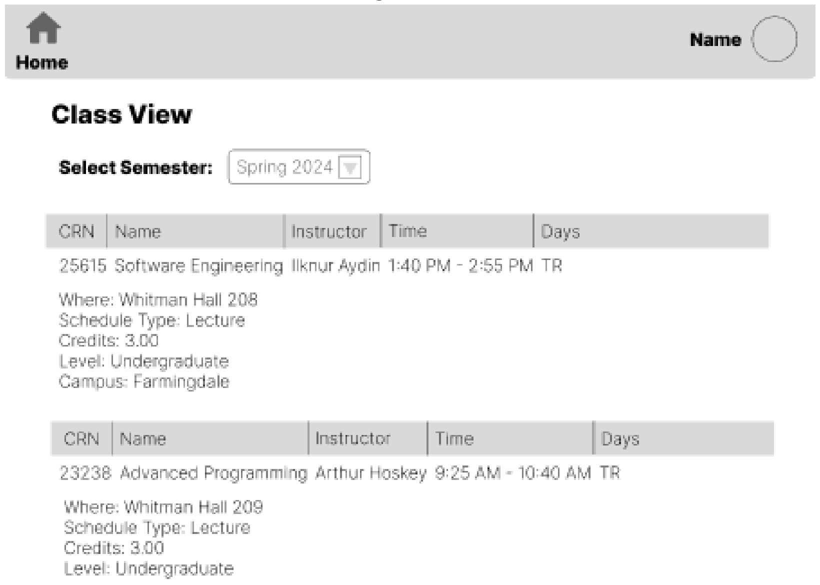
Password

Forgot Passsword?

Login

2. Student Home Page: (insert picture later in semester)

3. Student Schedule Page:



Home

Name

Class View

Select Semester: Spring 2024

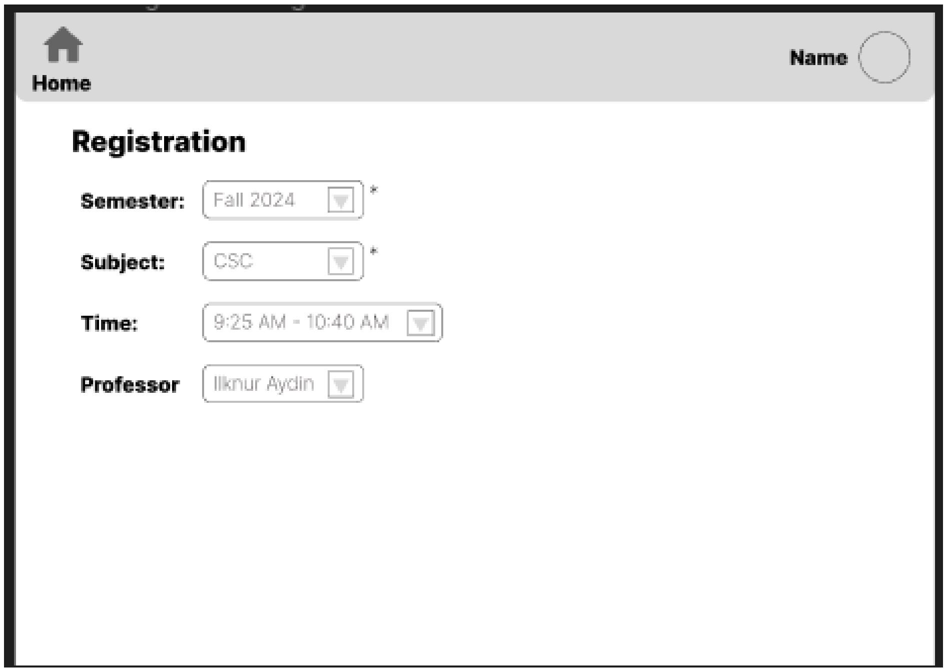
CRN	Name	Instructor	Time	Days
25615	Software Engineering	Ilknur Aydin	1:40 PM - 2:55 PM	TR

Where: Whitman Hall 208
Schedule Type: Lecture
Credits: 3.00
Level: Undergraduate
Campus: Farmingdale

CRN	Name	Instructor	Time	Days
23238	Advanced Programming	Arthur Hoskey	9:25 AM - 10:40 AM	TR

Where: Whitman Hall 209
Schedule Type: Lecture
Credits: 3.00
Level: Undergraduate

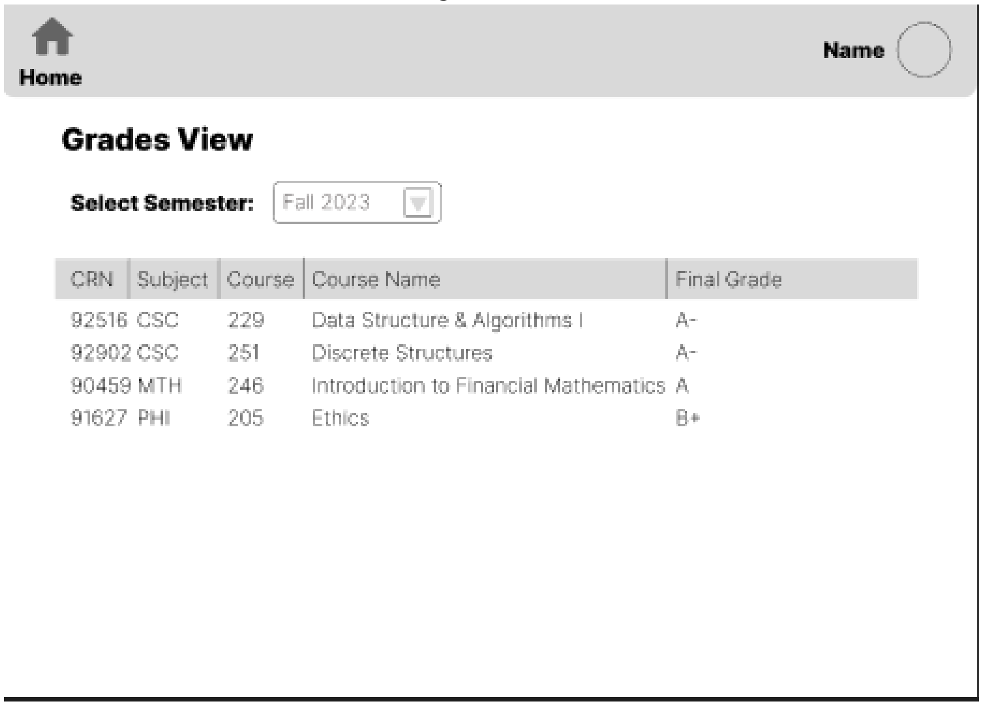
4. Student Registration Page:



Registration

Semester:	Fall 2024 *
Subject:	CSC *
Time:	9:25 AM - 10:40 AM
Professor	Ilknur Aydin

5. Past Semester Grades Page:

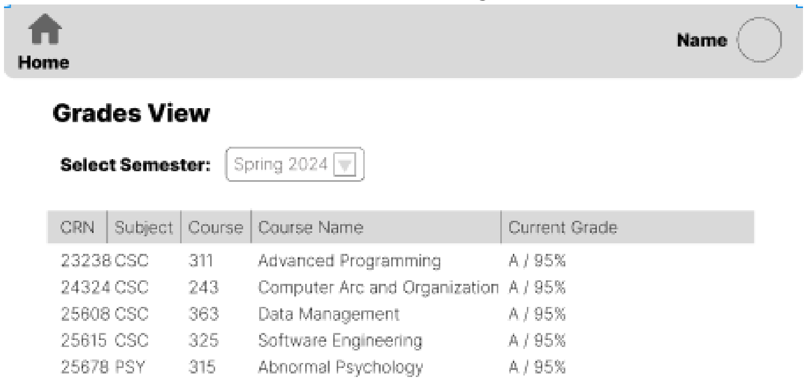


Grades View

Select Semester: Fall 2023

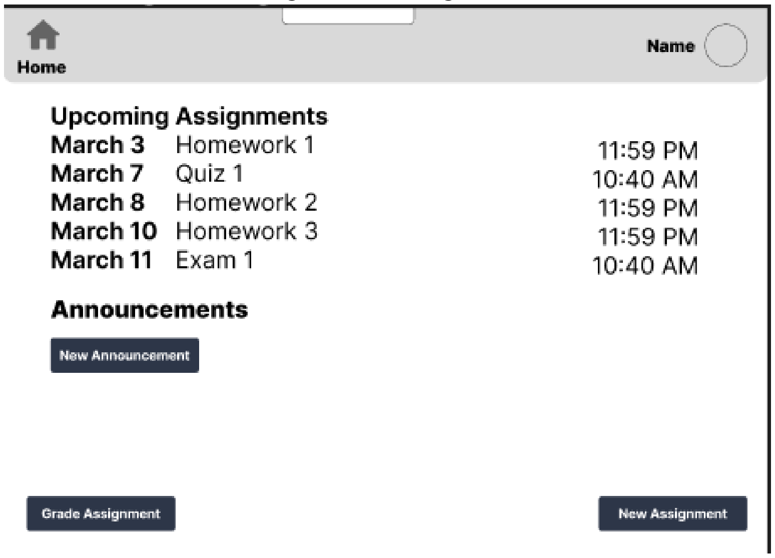
CRN	Subject	Course	Course Name	Final Grade
92516	CSC	229	Data Structure & Algorithms I	A-
92902	CSC	251	Discrete Structures	A-
90459	MTH	246	Introduction to Financial Mathematics	A
91627	PHI	205	Ethics	B+

6. *Current Semester Grades Page:*



CRN	Subject	Course	Course Name	Current Grade
23238	CSC	311	Advanced Programming	A / 95%
24324	CSC	243	Computer Arc and Organization	A / 95%
25608	CSC	363	Data Management	A / 95%
25615	CSC	325	Software Engineering	A / 95%
25678	PSY	315	Abnormal Psychology	A / 95%

7. *Professor Assignments Page:*



Upcoming Assignments

March 3	Homework 1	11:59 PM
March 7	Quiz 1	10:40 AM
March 8	Homework 2	11:59 PM
March 10	Homework 3	11:59 PM
March 11	Exam 1	10:40 AM

8. *Administrators Add Classes Page: (insert picture later in semester) (Add more features later)*

3.2 Hardware Interfaces

The School Registrar system is not very complex and does not use up a large amount of memory, so any modern(Windows 7 or higher, MAC OS, or Linux) CPU would be adequate for the use of this system.

3.3 Software Interfaces

The School Registrar system requires Java 17(for use of the JavaFx SDK) or higher to be installed on the system for its use. More information can be found in section 2.7.

The School Registrar system can be connected with Google Firestore to hold information on all classes.

3.4 Communications Interfaces

The School Registrar system uses e-mail to verify the login information of all users. The password should be encrypted so as to not allow for information to be easily stolen.

The School Registrar system requires an internet connection to be able to connect to the Google Firestore database to synchronize class information.

4. System Features

This section shows the School Registrar system's most important features and explains how they are used and what the user sees while using them

4.1 Student Registration Page

4.1.1 Description and Priority

The user(a student) will be able to register for a class by selecting the semester, subject(both of which are required), time and professor of the class.

High priority feature

4.1.2 Stimulus/Response Sequences

If the class the user selected is valid and is not already full and already has the prerequisites for it, a message will show saying that they have been successfully registered: (insert picture later in semester)

If the class entered is invalid, a message will appear saying invalid class entered: (insert picture later in semester)

If the class is valid but they do not have the prerequisites for it, a message should appear that says they cannot yet take this class: (insert picture later in semester)

If the class is valid but the class is full, a message saying the class is full: (insert picture later in semester)

If the class the user is trying to register for is at the same time as one they already have, a message will appear saying they already have a class scheduled for that time:(insert picture later in semester)

4.1.3 Functional Requirements

413-1-1: The user must be able to select the class they want with its current semester, course, time, and professor.

413-2-2: The system must be able to tell if a class is valid or not, if the class is currently full, if the user has the prerequisites for the course, or if the user is already taking a course at the selected time

4.2 Professor Assignments Page

4.2.1 Description and Priority

The user(a professor) will be able to see their current assignments, add new assignments, grade existing assignments, and create announcements for the class.

High priority feature

4.2.2 Stimulus/Response Sequences

If the user clicks the New Announcement button, a screen is brought up that allows the user to type in the announcement and post it for the class: *(insert picture later in semester)*

If the user clicks the Grade Assignment button, a screen is brought up that allows the user to pick which assignment they want to be graded and allows the user to give a grade to each student in the class: *(insert picture later in semester)*

If the user clicks the New Assignment button, a screen is brought up allowing the user to enter the name of the assignment and when the assignment is due(If the date and time is set to a date and time before the current date, an error message should be shown*(insert picture later in semester)*): *(insert picture later in semester)*

4.2.3 Functional Requirements

423-1-1: The user must be able to add a new announcement to the page

423-2-1: The user must be able to grade assignments of all the students

423-3-1: The user must be able to add new assignments

423-4-2: The system must be able to tell if the date and time the new assignment is set to be due is later than the current date and time or not

4.3 Administrators Add Classes Page

4.3.1 Description and Priority

The user(an administrator) can add a new class to the system by selecting the semester, course, time , and professor.

Medium priority feature

4.3.2 Stimulus/Response Sequences

If the class entered is invalid, a message will appear saying invalid class entered: (insert picture later in semester)

If the class that is attempted to be added already exists, a message will appear saying class already exists:*(insert picture later in semester)*

If the class is valid and does not already exist, the class will be added to the database and a message will show stating the class has been added:(insert picture later in semester)

4.3.3 Functional Requirements

433-1-1: The user must be able to add a new class to the system by entering its semester, course name, time, and professor

433-2-2: The system must be able to tell a class that was entered by the user is valid or not and if it already exists in the system

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The School Registrar system is not very complex and does not use up a large amount of memory, so any modern(Windows 7 or higher, MAC OS, or Linux) CPU would be adequate for the use of this system.

5.2 Safety Requirements

The user's data(email and password) could be stolen if someone were to hack into the system, so all the user data must be encrypted so as to not allow other users to affect the system.

The user data could not be fully entered into the system if the system were to crash in the middle of the data entry, so this should be noted when it happens so it could be fixed later on.

5.3 Security Requirements

The use of the School Registrar system requires the use of a school created email address, meaning only those who are affiliated with the college can use the system.

5.4 Software Quality Attributes

The School Registrar system is very basic and intuitive, so no user should have trouble using it. It being basic is not only for ease of use, but it means that the system is likely to have zero to few bugs that would cause problems for the user.

5.5 Business Rules

Only students can look at their grades from previous semesters and register for classes.

Only professors can grade assignments, add new assignments, and make announcements.

Only administrators can create new classes.

6. Other Requirements

Google Firebase is required for the storing of the class data, so it must be connected with the project to be able to integrate it within the project.

Appendix A: Glossary

All terms used right now seem to be pretty self-explanatory

Appendix B: Analysis Models

(Add 2 UML diagrams later in semester)

Appendix C: To Be Determined List

Add more features to 3.1
Add pictures to section 4
Add more Functional requirements to 4.3
Add terms to Glossary
Add UML Diagrams to Appendix B