System Requirements Specification

for

Majorizer

Version 1.0

Andrew Caruso, Cole Caruso, Ceilidh Kiegle, Ethan Matzek, Payton Shafer

Software Surge

March 21st, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Kiegle	4/6/2023	Completed remaining sections of document	1.0

1. Introduction

1.1 Purpose

The process of determining a student's path is one that requires a student to work with the assistance of the advisor of their major, along with advisors for a second major and/or additional minors. Ensuring that a student is on track to graduate can be a daunting process, and even with assistance human error can lead to mistakes in a student's educational plan that can leave them confused, or worse, unable to graduate on time.

The goal of the Majorizer application is to allow students and advisors to create educational plans for a student's majors and minors with ease, and allow for those plans to be shared with their respective students and advisors for quick approval. With the ability to quickly create multiple plans for different combinations of majors and minors and a forecasting algorithm that takes the student's previous courses into account, Majorizer's goal is to help remove some of the confusion from a student's schedule so they can feel confident in their decisions for their academic roadmap at Clarkson.

1.2 Intended Audience and Reading Suggestions

The system requirements specification document describes what the system will do and how it will perform each function. The intended audiences for this document are system developers, customers, and as a reference for marketing, testing, and documentation. Shown below is the suggested reading sequence for viewing this document based on the intended audience.

System Developer: A system developer should use the system requirements specification as a master document to outline the scope and functionality while designing the system. A system developer should view the Project Description first if they are new to the system followed by the External Interface Requirements, System Features, and Other Nonfunctional Requirements.

Customer: The customer should use this document to understand the functionality of the system and how each segment of the system could be integrated into other systems. It is suggested that a customer view the Overall Description and optionally view the System Features for more detail on Functionality.

Marketing: A marketing representative should use the system requirements specification as a reference to understand the intended functionality and uses of the system. It is suggested that a marketing representative use the Overall Description and System Features.

1.3 Product Scope

The Majorizer functions as a website with a connection to a backend that contains the forecasting code. The user interface will differ depending on whether the user is a student, advisor, or admin, and will provide different functionalities for each. The backend will be integrated with the frontend so that the schedule creation request from a user on the frontend will operate like a black box in the backend and return the completed schedule. This algorithm will be tested by the developers to ensure the forecasted schedules created are accurate, and both the frontend and backend will be tested to ensure they meet performance requirements.

1.4 References

Please refer to our team's website for more information on testing plans, requirements interview notes, and weekly team reports:
https://webspace.clarkson.edu/classes/softwaresurge/public_html/

2. Overall Description

2.1 Product Perspective

Majorizer is a website that is designed to be used by Clarkson students, advisors, and administrators. This website will allow users to build forecasted schedules for different combinations of majors and minors that also takes into account the courses the student has already taken. These schedules can be saved by the students and viewed by their advisors. Functionalities to allow for students to send advisors course requests and for advisors to approve those requests, along with admin control over advisor-student connections, are also included.

2.2 Product Functions

The main functionality of the Majorizer is in its ability to forecast schedules for students. This algorithm takes into account the student's previous courses, along with a minimum of 1 major and a maximum of 2 majors and 2 minors. Once created these schedules are saved onto the student's view schedule page, where they can be accessed at a later time by the student. The schedules built by the students can also be viewed by their advisor. Advisors also have the ability to build schedules on behalf of their students. Administrators will have the ability to connect advisors and students.

Our stretch functionality goals are the ability for the students and advisors to download a schedule they've created as a PDF, and for the admin to be able to add courses and majors to the system.

2.3 User Classes and Characteristics

The goal of developing this website is to provide an easy system that Clarkson students and advisors can use to build schedules. Considering the number of students that advisors have to advise, it's easy for confusion to cause problems when attempting to build a schedule for a student by hand. By letting the Majorizer system take away that complexity and build schedules on their behalf, we hope to take some of the guesswork out of student course advising at Clarkson.

2.4 Operating Environment

Majorizer is designed to operate as a website within the Clarkson intranet, where users can log in using their Clarkson username and password. The website should be accessible to anyone with a Clarkson account and access to the internet.

2.5 Design and Implementation Constraints

Time: Since the entirety of Majorizer has to be built within this semester, we will have a very limited amount of time to build the entire system; with this in mind the requirements will likely have to be revisited further in the semester to reevaluate what we will be able to complete before the end of the semester, and which requirements will have to be tabled.

Team Background: Our team members all have different backgrounds in terms of Computer Science experience, and therefore the assignments within the project will have to take experience into account, along with requirements for the project as a whole if there are any areas where no one on the team has experience related to one of our requirements.

2.6 User Documentation

- Requirements Presentation
- Mid-Semester Presentation
- Design Presentation

2.7 Assumptions and Dependencies

The platform the website is running on is not down.
The platform the backend is running on is not down.

3. External Interface Requirements

3.1 User Interfaces

The main user interface is the Majorizer website. The frontend has designed the website so that the user's interactions depend on whether they are a student, advisor, or admin. All connections to the website's functionalities that require the backend will be delivered to the frontend and user interface using APIs.

3.2 Hardware Interfaces

The only hardware interface this system requires is a computer that has access to the internet.

3.3 Software Interfaces

The code for our system is housed on GitHub. This system has been built using Javascript, CSS, and React for the frontend. For integration we used REST API. Our backend was built using Python, Django, JWT, PostgreSQL, and AWS.

3.4 Communications Interfaces

All communication between the backend and frontend of the system will be handled using APIs. Communication between users, such as sending messages to an advisor, will also be handled using APIs.

4. System Features

4.1 Schedule Forecasting

4.1.1 Schedule Forecasting Capability

The system shall be able to forecast student schedules given a set of major(s) and minor(s), along with the student's previously taken courses. The forecasted schedule will provide all courses that the student would need to take in their 8 semesters at Clarkson in the order they should be taken. The forecast will also determine if the student will be able to graduate within 8 semesters, and if not, will inform the student that they will not be able to graduate on time with the given academic plan the student requested. Schedules created shall be saved and able to be viewed at a later time by Students and Advisors through the View Schedule Page.

4.1.2 Two Minors Two Majors Input

The system shall be able to forecast a schedule with a minimum of one major given to a maximum of two majors and two minors given. For this version of the Majorizer, the majors available for selection are Computer Science and Psychology, and the minors available for selection are Math and Literature and the Arts.

4.2 Student Functionalities

4.2.1 Student can send Course Requests to their Advisor

The system shall allow students to send requests to their advisors to approve schedules or add or drop courses through the Send Request Page.

4.2.2 Visual Calendar/Schedule

The system shall show a visual calendar on the home page that displays the student's most recent schedule. The calendar will update when new schedules are created.

4.3 Advisor Functionalities

4.3.1 Advisor can Approve Course Requests

The system shall allow Advisors to view the requests sent by the students they are advising, and shall allow the Advisors to either approve or reject those requests on the View Request Page.

4.3.2 Advisor can Build Schedules for Students they are Advising

The system shall allow Advisors to build schedules on the Build Schedule page on behalf of a student that they are advising. The Forecasting process shall work the same for the advisor, just with the addition of a select student feature.

4.4 Admin Functionalities

4.4.1 Admin Controls Advisor-Student Connections

The system shall allow Admins to control the creation and removal of Advisor-Student connections on the Majorizer system on the Connection Page.

4.5 Additional Functionalities

4.5.1 1 Admin, 4 Advisors, and 4 Students are able to use the Website

The system shall have 1 admin, 4 advisors, and 4 student accounts successfully implemented and working in the Majorizer system.

4.6 Stretch Functionality Goals

4.6.1 Admin can add Majors and Courses

If there is enough time, the system shall allow for Admins to control the creation of new majors and courses within the Majorizer system.

4.6.2 Download Schedule as a PDF

If there is enough time, the system shall allow for Students and Advisors to download a schedule they have created as a PDF. This functionality would be available on the View Schedule Page.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response Time: The system should be able to build a schedule upon request in under 30 seconds.

5.2 Ease of Use Requirements

The system shall have a help page accessible at all times for all users that will describe how to use all of the functions of the Majorizer system.

5.3 Software Quality Attributes

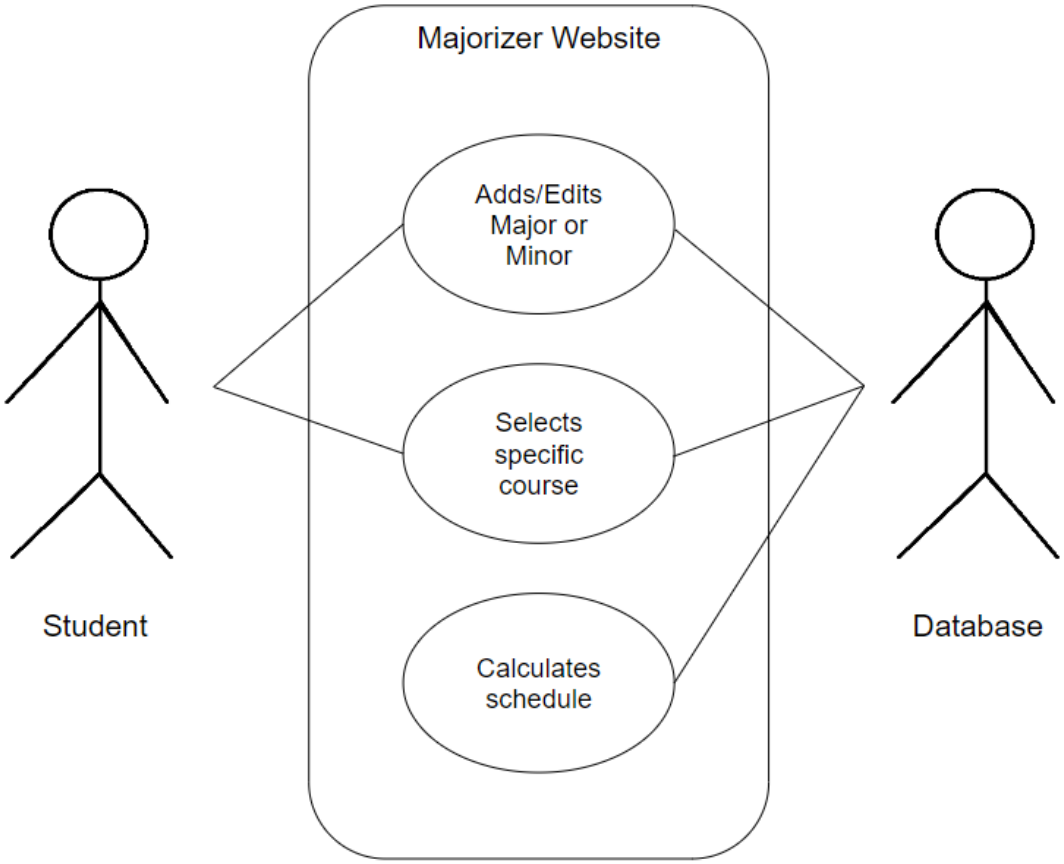
Testability: This system will be tested thoroughly to meet both Functional and Nonfunctional Requirements.

Reliability: The forecasting process will always create accurately forecasted schedules.

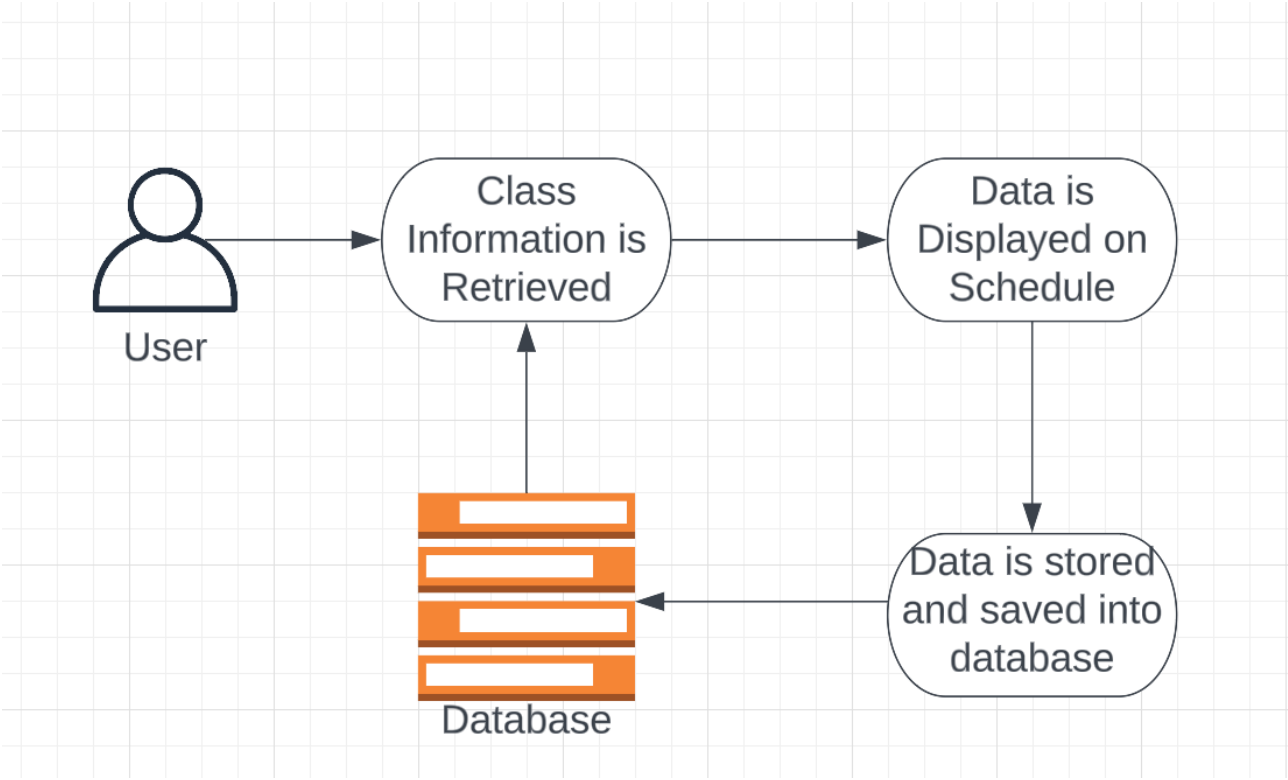
6. Appendix

6.1 Use Case Diagrams

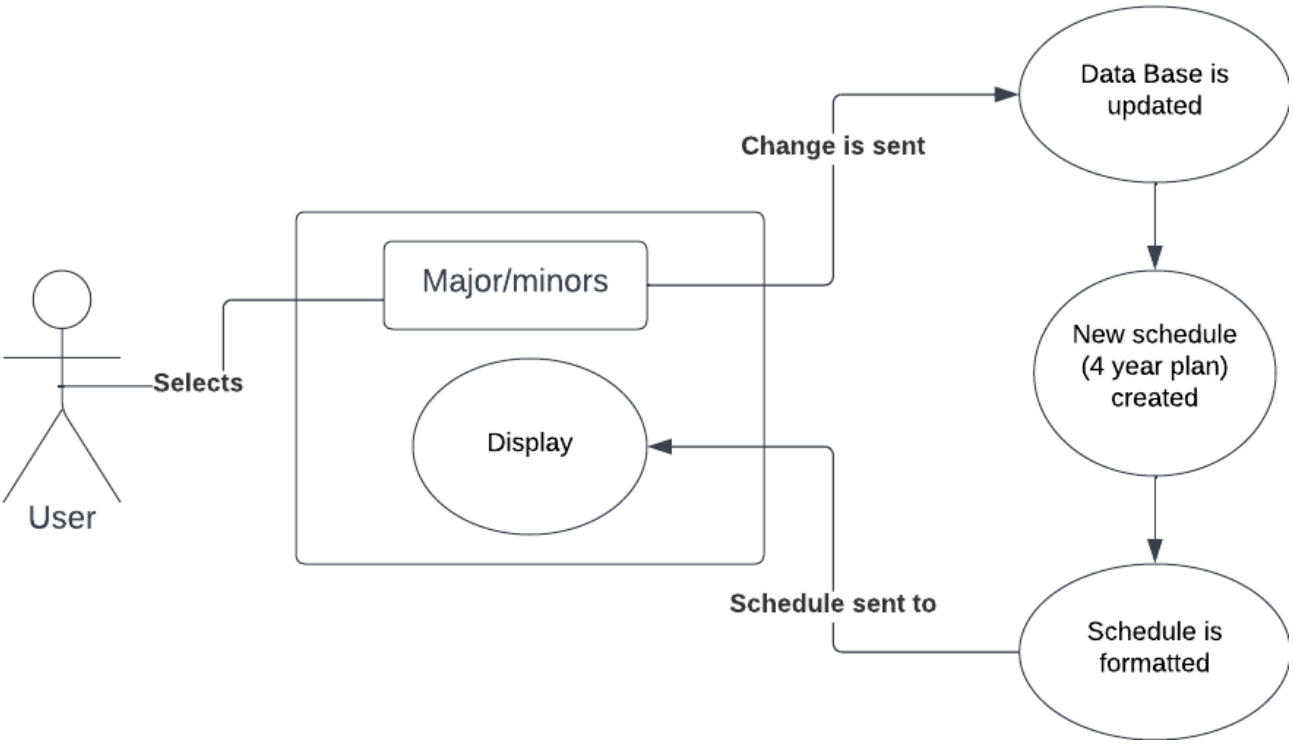
6.1.1 Use Case 1 - Forecasting Courses:



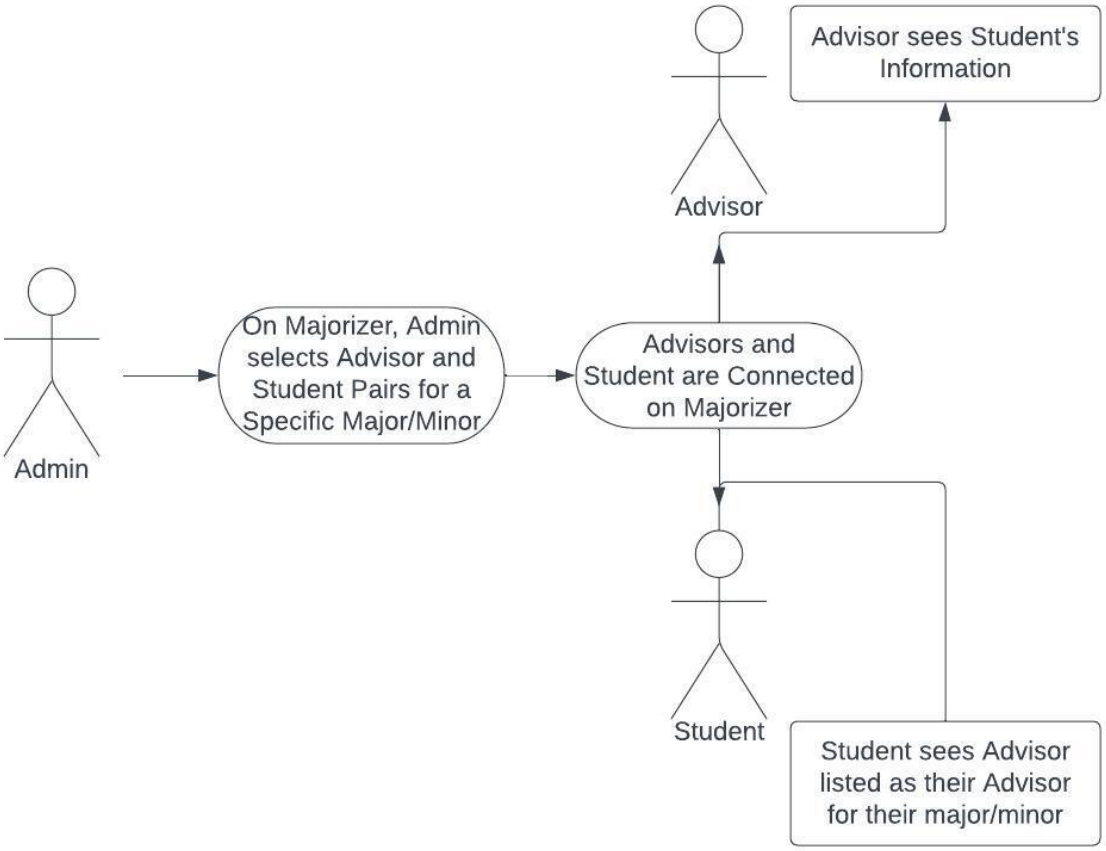
6.1.2 Use Case 2 - Visual Calendar:



6.1.3 Use Case 3 - Major/Minor Input:



6.1.4 Use Case 4 - Admin Control:


Advisor sees Student's Information
Advisor
Admin
On Majorizer, Admin selects Advisor and Student Pairs for a Specific Major/Minor
Advisors and Student are Connected on Majorizer
Student
Student sees Advisor listed as their Advisor for their major/minor