Software Requirements Specification

for

University Advising Tool

Version 1.0 approved

Prepared by Hannah Heflin

Brandon Johnson

Tia Malley

Makayla McClain

Gavin Sharp

Brandon Temple

Mitchell Toth

Created March 22, 2023

Table of Contents

Table of Contents ii
Revision History ii
1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 2
1.5 References 2
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 3
2.4 Operating Environment 3
2.5 Design and Implementation Constraints 4
2.6 User Documentation 4
2.7 Assumptions and Dependencies 4
3. External Interface Requirements 4
3.1 User Interfaces 4
3.2 Hardware Interfaces 7
3.3 Software Interfaces 7
3.4 Communications Interfaces 7
4. System Features 7
4.1 Account Creation, Login, and Logout 7
4.2 Create/Delete Potential Schedule 10
4.3 Request Course Substitution or Waiver 13
4.4 Schedule Meeting with Advisor 15
4.5 Add/Change Minor or Concentration 17
4.6 Perform What-If Reports 19
4.7 View Course History and Grades 22
4.8 Request Transcript 24
4.9 Calculate Estimated GPA 26
4.10 View Major/Minor Requirements 29
4.11 Messaging System 31
4.12 Edit Dashboard 33
5. Other Nonfunctional Requirements 36
5.1 Performance Requirements 36
5.2 Safety Requirements 37
5.3 Security Requirements 37
5.4 Software Quality Attributes 37
Appendix A: Glossary 38
Appendix B: To Be Determined List 38

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document presents a detailed overview of the University Advising Tool (UAT) Version 1.0. It will describe the functions of the subsystem, the environment in which the software will operate, constraints that will limit the product's development, the features that will be included in the user interface, and the attributes and requirements that will be incorporated into the product.

1.2 Document Conventions

When reading this document, certain words will be in different typefaces and font sizes. For readability, titles and subtitles are bolded and have larger font sizes. Any italicized words or abbreviations will be defined in Appendix A: Glossary. The project's name, University Advising Tool (UAT), is tentative and will be reconsidered in the future.

Every requirement for the system has its own priority.

These priorities are defined as:
- High: It is a critical requirement that the product must perform.
- Medium: It is a core functionality that the product should perform.
- Low: It is a functional enhancement not explicitly stated that the project should have.

1.3 Intended Audience and Reading Suggestions

This document is intended for stakeholders to understand the project's functionality and scope and for developers to aid in its development. For the stakeholders, the order for reading this document should be Part 1 – Introduction, Part 4 – System Features, Part 5 – Other Nonfunctional Requirements, Part 3 – External Interface Requirements, and Part 2 – Overall Description, while skipping to Appendix A: Glossary for the definition of italicized words and abbreviations and Appendix B: Analysis Models to view any models mentioned in this document. The development team should read over the entire document but jump to any section relevant to the task they are accomplishing. Below is a summary of each part of the document:

- Part 1 – Introduction
  - This section includes a summary of the benefits, objectives, and goals for Version 1.0 of the University Advising Tool.
- Part 2 – Overall Description
  - This section describes how the system will be developed. It includes information on the functions, user classes, operating environments, and constraints.
- Part 3 – External Interface Requirements
  - This section provides information on user, hardware, software, and communications interfaces.
- Part 4 – System Features
  - This section includes the core features that will be implemented on the UAT website, as well as step-by-step instructions on how these features will work.
- Part 5 – Other Nonfunctional Requirements

    - This section discusses the performance, safety, and security requirements. It also notes the quality attributes and business rules that will be included in the software.
- Appendix A: Glossary
    - This section includes additional information for italicized words and abbreviations that may be helpful to readers.
- Appendix B: To Be Determined List
    - This section includes a list of the to be determined references for the project.

1.4 Product Scope

The software project specified in this document is a tool meant to aid college students with their schedules and graduation requirements throughout their time in school. It will not only be of use to students but also to advisors. One of the goals in mind when designing this software is for it to be able to save advisors a significant amount of time when it comes to scheduling meetings and analyzing each student's progress.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

This project idea originated in a software engineering senior design class at Mississippi State University. It is meant to be a better, more efficient replacement for existing advisement tools that universities may be using. If a university is not using any advising assistance tool, then this will serve as a replacement for the time that advisors spend on advising every semester.

Assuming the fact that this software will be used by a university, it will need to be integrated into the universities system. This way the tool will have access to the students' records and course information and be able to work more efficiently in accordance with a specific college.

The language of choice is still being determined. However, the website will be made in either Python with a Django framework, or JavaScript with a JavaScript framework like React or Node.js. The front-end will mostly be coded in HTML and CSS, and this project will be using a MySQL database regardless of the framework used.

2.2 Product Functions

The functionalities listed below are of high priority and do not encompass all the features that will be in UAT. For a detailed description of all features please refer to Section 4, "System Features."

- The user must be able to create an account and log in to said account.

- The user must be able to see their degree progress in the sense of what class they have taken and are currently taking.
    - A student user will only see their own degree progress.
    - An advisor user will be able to see all their advisee's degree progress.
    - An administrator will be all to see every student's degree progress.
- The user must be able to see the required classes for a specific degree program.

2.3 User Classes and Characteristics

Student: Students are the primary user of this tool, as they will use it to plan for upcoming semesters, schedule appointments, and receive guidance from their advisor. They can access information about registration deadlines, major/minor requirements, and their course history. Students are also able to request transcripts and course substitutions/waivers, compare academic programs, and add/change their minor or concentration.

- Students will have a lower privilege level compared to advisors and administrators. They will only be able to access certain functions and their own data.
- No expertise will be required for this user class, as the tool will provide a user-friendly interface and intuitive navigation that will enable students to easily access and utilize its features.

Advisor: Advisors are the secondary user of this tool and will assist students in scheduling their upcoming semester and keep them on track to graduate on time. They can access all the information about their advisees with a few exceptions. Advisors will receive notifications of scheduled appointments and form submissions via the message systems, as well as email. They will also be able to accept or decline forms that require their approval.

- Advisors will have a medium privilege level-higher than students, but lower than admins. They will only be able to access information about students that they are assigned to advise, or within their department.
- No expertise will be required for this user class, aside from their knowledge as an advisor, as the tool will provide a user-friendly interface and intuitive navigation that will enable advisors to easily access and utilize its features.

Administrator: Administrators are another secondary user of this tool and will assist in overseeing the advising process of the university or their department. Assuming access to the universities database is not granted, the administrators will need to manually input the student's information and course progression into the system. They will be able to see the information on every student in their college and be able to approve or deny certain forms.

- Admins will have the highest privilege compared to students and advisors. They will be able to access the information of every student in their college.
- Some knowledge of how to add data to databases will be needed if the admins are to manually input student's information in case of a lack of access to the universities database.

2.4 Operating Environment

UAT will operate in a web-based environment and will be accessible to users through any standard web browser, such as Google Chrome, Safari, or Edge. The hardware platform requirements for accessing the website will be minimal and will include a computer, tablet, or smartphone with an internet connection. The website will be compatible with all major operating systems, including

Windows, macOS, and Linux. No additional software will need to be installed to access the website, and any libraries or frameworks that are used will be included in the website's codebase.

2.5 Design and Implementation Constraints

As of right now, we have not secured access to a universities database. Therefore, any student data will need to be manually entered into our MySQL database by a developer or administrator. If access is acquired, data will be retrieved directly from the university's database, and this will no longer be a constraint. In addition, because our application will be utilizing student's private information, security is our upmost priority, and nothing should be done to compromise it. In addition, we will be working within the limits of our chosen languages and frameworks, which are still to be determined. Because MySQL database is being used, we will also be working within its limitations, as well as the limitations of Mississippi State's server for development and testing.

2.6 User Documentation

A thorough FAQ section will be available on the UAT website. Each user class will be provided with quick and easy access to answers to common questions regarding the features provided to them. The website will also include tutorials for administrators on how to use the various features of the tool. These tutorials will be easy to follow and will include step-by-step instructions and screenshots to guide administrators on how to use the website effectively. The FAQ section and tutorials will be regularly updated to reflect any updates to the website.

2.7 Assumptions and Dependencies

- The user has their own device to access the application.
- The user is in a supported browser and has access to the internet.
- The user is associated (student/advisor/admin) with a partnered university.
- MySQL database is running as expected.
- Relevant university server (Mississippi State University for development) is running as expected.

3. External Interface Requirements

3.1 User Interfaces

Login Page:

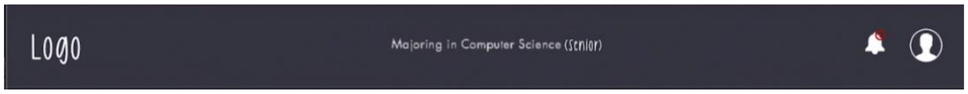
- The login page will contain two fields for the user to enter their netID and password.
  - An error message will be displayed if either the netID or password is invalid.
  - The user will be redirected to their dashboard upon successful login.
- A "Forgot Password" button will also be included that will redirect the user to a page to reset their password.
- No prototype has been made yet.

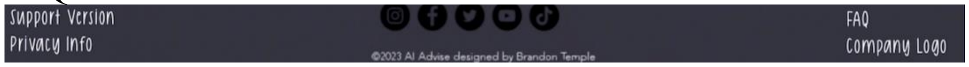
Information Provided on Every Screen:

- A header which includes:
  - University's Logo
  - User's Category

- Student – Majoring in Computer Science
- Advisor – Computer Science Advisor
- Administrator – Their position (i.e. department head)

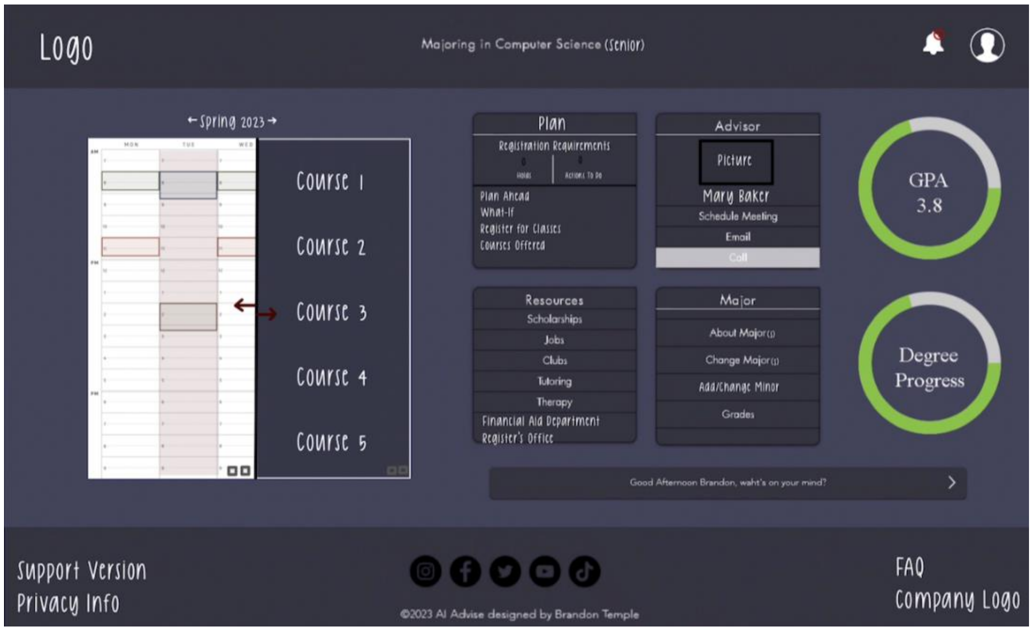
- Notification Button
- Account Setting Button

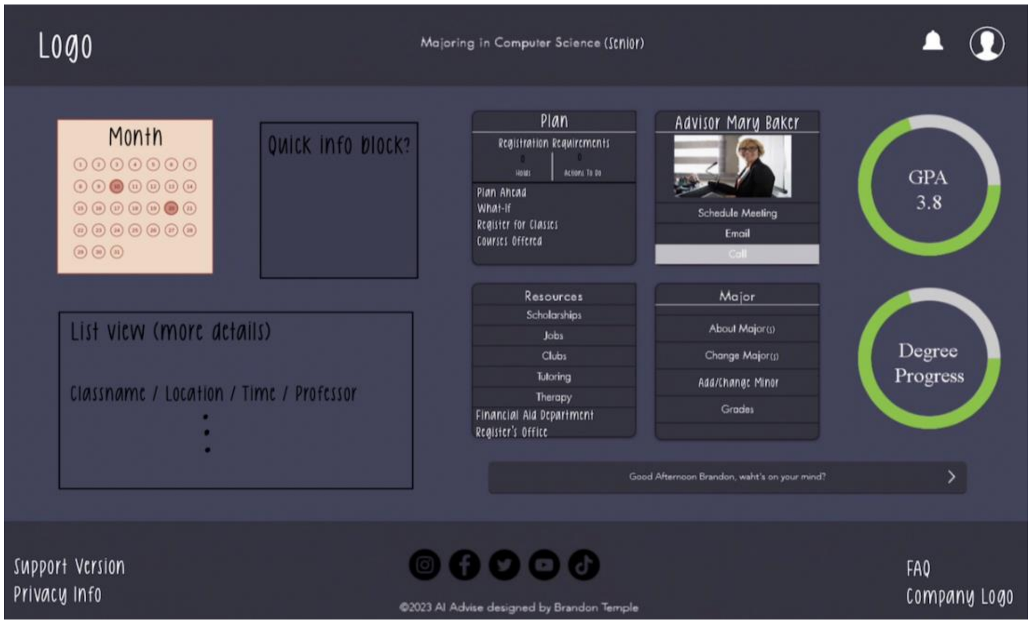


- A foot which includes:
  - Support Version
  - Privacy Notice
  - FAQ Section
  - 

Student Dashboard:

- Will provide containers for:
  - These containers will be customizable, allowing the student to choose which containers will be displayed on their dashboard and the container's location.
    - Month/Week View
    - Quick Information
    - Detailed Schedule
    - Planning
    - Resources
    - Advisor
    - Major
    - GPA
    - Degree Progress
    - Etc.





Advisor Dashboard:

- Will be formatted the same as the student's dashboard, but containers will be relevant to advisor features.
- No prototype has been made yet.

Administrator Dashboard:

- Will be formatted the same as the student's dashboard, but containers will be relevant to administrator features.
- No prototype has been made yet.

Other pages: TBD

3.2 Hardware Interfaces

Laptop/Desktop:
The user may access the software through a web browser on a desktop or laptop. This is assuming they are in a traditional environment, are using a supported browser, and are connected to the internet. They will be able to interact with a keyboard and mouse/touchpad.

Mobile Device:
The user may access the software through an app on a supported mobile device, such as a smartphone or tablet. The user must have these devices connected to Wi-fi or mobile data for some functionalities of the app to work.

3.3 Software Interfaces

MySQL Database interface:
The system will interact with the database internally. There will also be an admin login for maintenance and testing.

Windows, MacOS, and Linux:
All will interact with the web interface through a web browser to display the data to the user.

IOS and Android:
Both will display a mobile interface to the user and interact using a touch screen to display data to the user.

3.4 Communications Interfaces

- **Email Communication:** The website will need to send emails to users to when appointments and form submissions have been made. An email must also be sent to a user to change their password.
- **Electronic Forms:** The website will require users to fill out electronic forms to request a course substitution/waiver or transcript. These forms will need to be formatted in a way that is compatible with the website's backend systems.
- **Web Browser:** The website will communicate with the web server using HTTP to display the website.

4. System Features

Student Features

4.1 Account Creation, Login, and Logout

4.1.1 Description and Priority

High Priority:

- A student must be able to:

- Create a new account.
- Login and logout of their account.

4.1.2 Stimulus/Response Sequences

Account Creation:

1. Student clicks the "Create Account" button.
2. The system presents the student with the account creation screen.
3. Student enters required information into the provided fields and clicks the "Submit" button.
4. System checks to make sure the username does not already exist, and the password meets the requirements.
    - If the input is valid, the system stores the student data in the database and automatically logs in the student.
    - If the input is invalid, an error message will appear stating the problem, and the student will be prompted to enter the information again.

*TBD: Students might be manually inputted into system

Login:

1. Student clicks the "Login" button.
2. Systems presents the student with the login screen.
3. Student enters required information into the provided fields and clicks the "Submit" button.
4. System attempts to match the student-entered data to an account in the database.
    - If the input is valid, the system will redirect the student to their dashboard.
    - If the input is invalid, the student will be prompted to enter the information again.
        - The student has limited attempts after their first failed attempt to enter their correct information. If this limit is met, their account will be locked for a currently undetermined time.

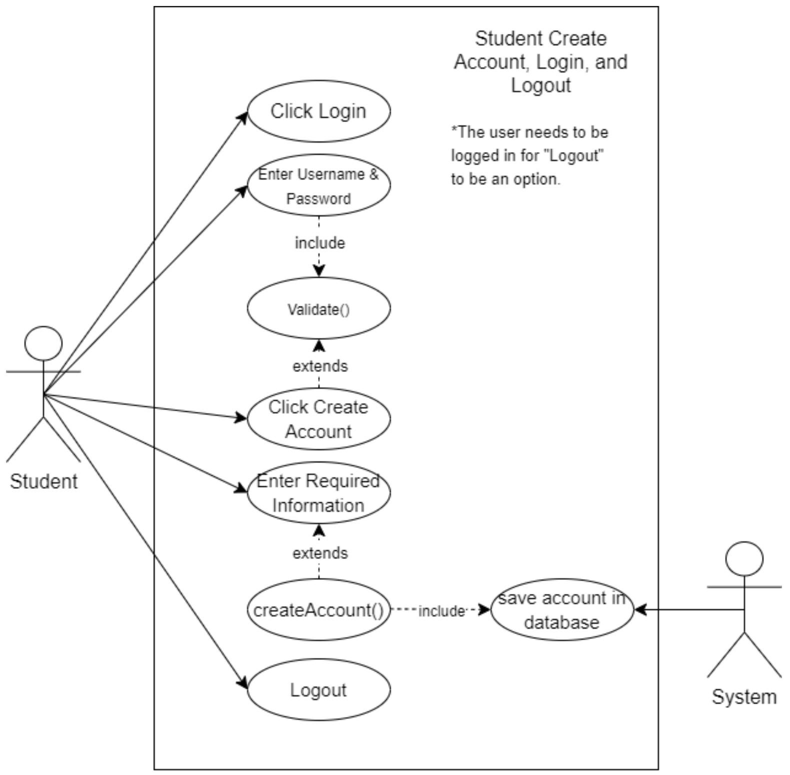
Logout:

1. Student clicks the "Logout" button.
2. System changes user status to "logged out" and the user is redirected to the login screen.

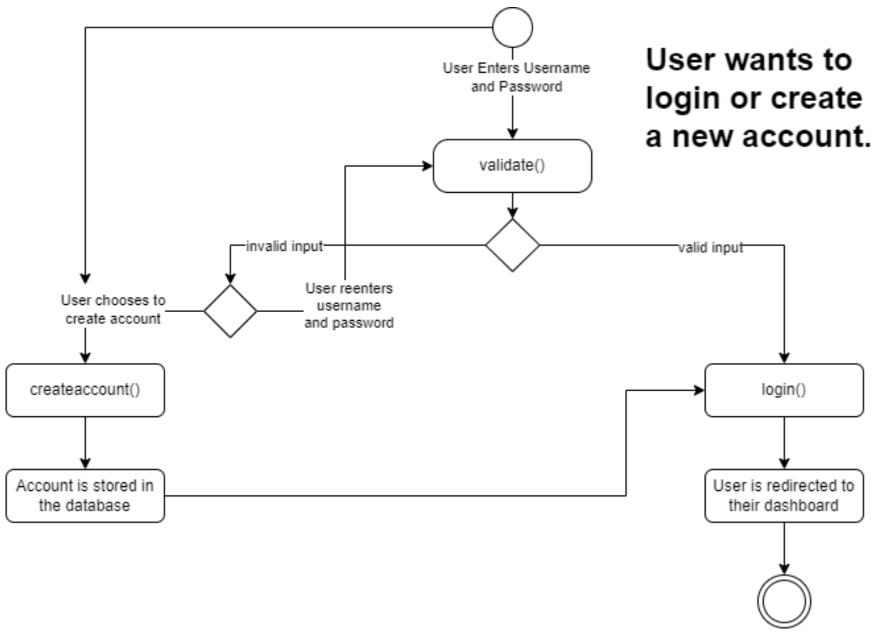
4.1.3 Functional Requirements

REQ-1: System has two "states" a student can be in: logged in and logged out.
REQ-2: System has functional database for storing student information.

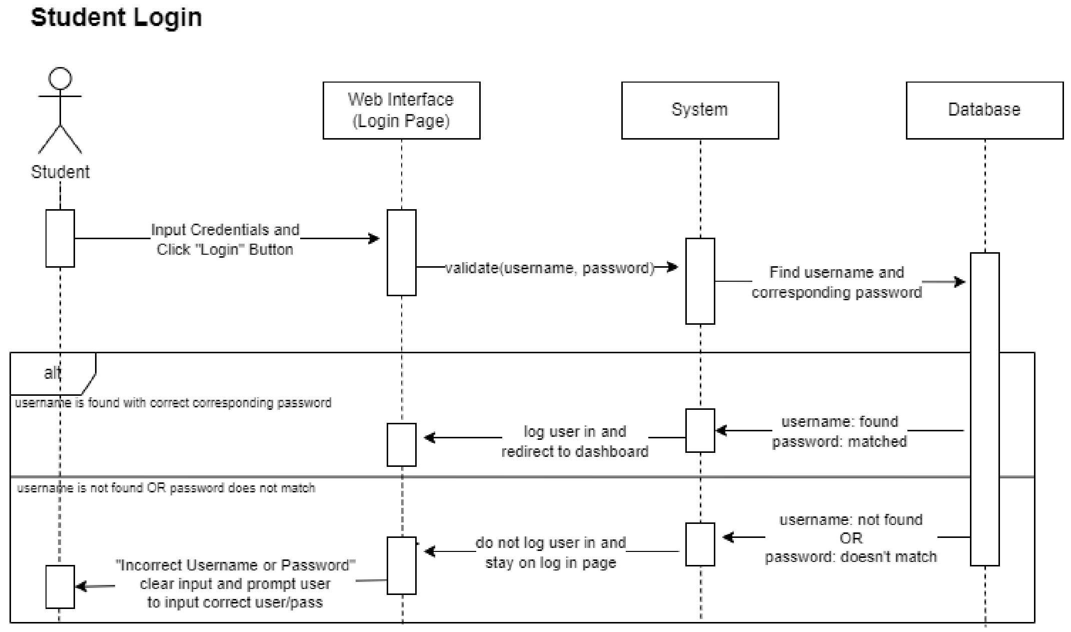
4.1.4 Use Case Diagram


Student Create Account, Login, and Logout
*The user needs to be logged in for "Logout" to be an option.
Click Login
Enter Username & Password
include
Validate()
extends
Click Create Account
Enter Required Information
extends
createAccount()
include
save account in database
Logout
Student
System

4.1.5 Activity Diagram


User wants to login or create a new account.
User Enters Username and Password
validate()
invalid input
valid input
User chooses to create account
User reenters username and password
createaccount()
Account is stored in the database
login()
User is redirected to their dashboard

4.1.6 Sequence Diagram


Student Login
Student
Web Interface (Login Page)
System
Database
Input Credentials and Click "Login" Button
validate(username, password)
Find username and corresponding password
alt
username is found with correct corresponding password
log user in and redirect to dashboard
username: found password: matched
username is not found OR password does not match
do not log user in and stay on log in page
username: not found OR password: doesn't match
"Incorrect Username or Password" clear input and prompt user to input correct user/pass

4.2 Create/Delete Potential Schedule

4.2.1 Description and Priority

<u>High Priority</u>:

- A student must be able to:
  - Create multiple potential schedules for the upcoming semester.
  - Add courses to the potential schedule.
  - Remove courses from the potential schedule.
  - Delete a potential schedule for the upcoming semester.

4.2.2 Stimulus/Response Sequences

<u>Create Potential Schedule:</u>

1. Student clicks the “Plan Ahead” button on their dashboard.
2. System presents the “Plan Ahead” page and lists any existing plans already made by the student.
3. Student clicks the “Create New Plan” button.
4. System presents a blank schedule and filter options to the student.
   - Student can filter available courses for the upcoming semester by course number, keyword, department, availability, time, etc.
5. Student selects the course they would like to add to their schedule and clicks the “Add Course” button.

- The selected course is added to the visual schedule.

6. Student can remove the course from their potential schedule by clicking the "Remove Course" button.
    - The selected course is removed from the visual schedule.
7. Once finished, the student clicks the "Save" button and enters a title for the plan. The system stores the potential schedule data in the database.

<u>Edit Potential Schedule:</u>

1. Student clicks the "Plan Ahead" button on their dashboard.
2. System presents the "Plan Ahead" page and lists any existing plans already made by the student.
3. Student selects an existing plan.
4. System presents the selected plan and filter options to the student.
    - Student can filter available courses for the upcoming semester by course number, keyword, department, availability, time, etc.
5. Student selects the course they would like to add to their schedule and clicks the "Add Course" button.
    - The selected course is added to the visual schedule.
6. Student can remove the course from their potential schedule by clicking the "Remove Course" button.
    - The selected course is removed from the visual schedule.
7. Once finished, the student clicks the "Save" button and the system updates the potential schedule data in the database.

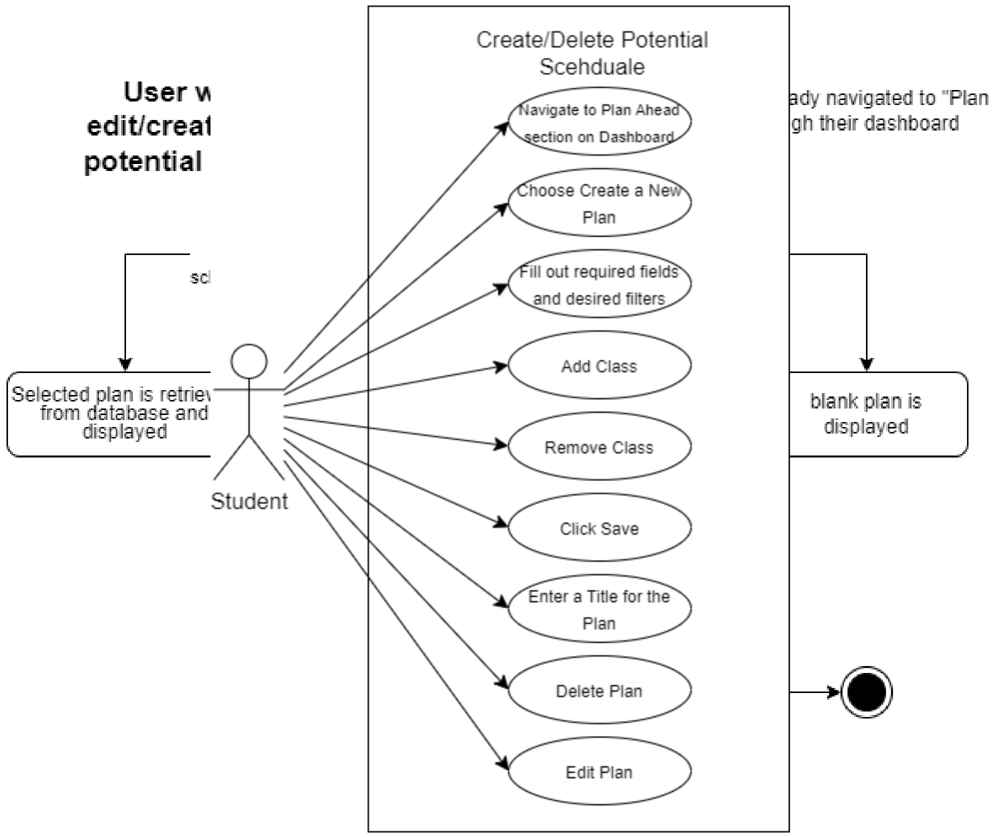
<u>Delete Potential Schedule:</u>

1. Student clicks the "Plan Ahead" button on their dashboard.
2. System presents the "Plan Ahead" page and lists any existing plans already made by the student.
3. Student clicks the selects an existing plan and clicks the "Remove Plan" button.
4. System deletes all data associated with the plan selected from the database.

4.2.3 Functional Requirements

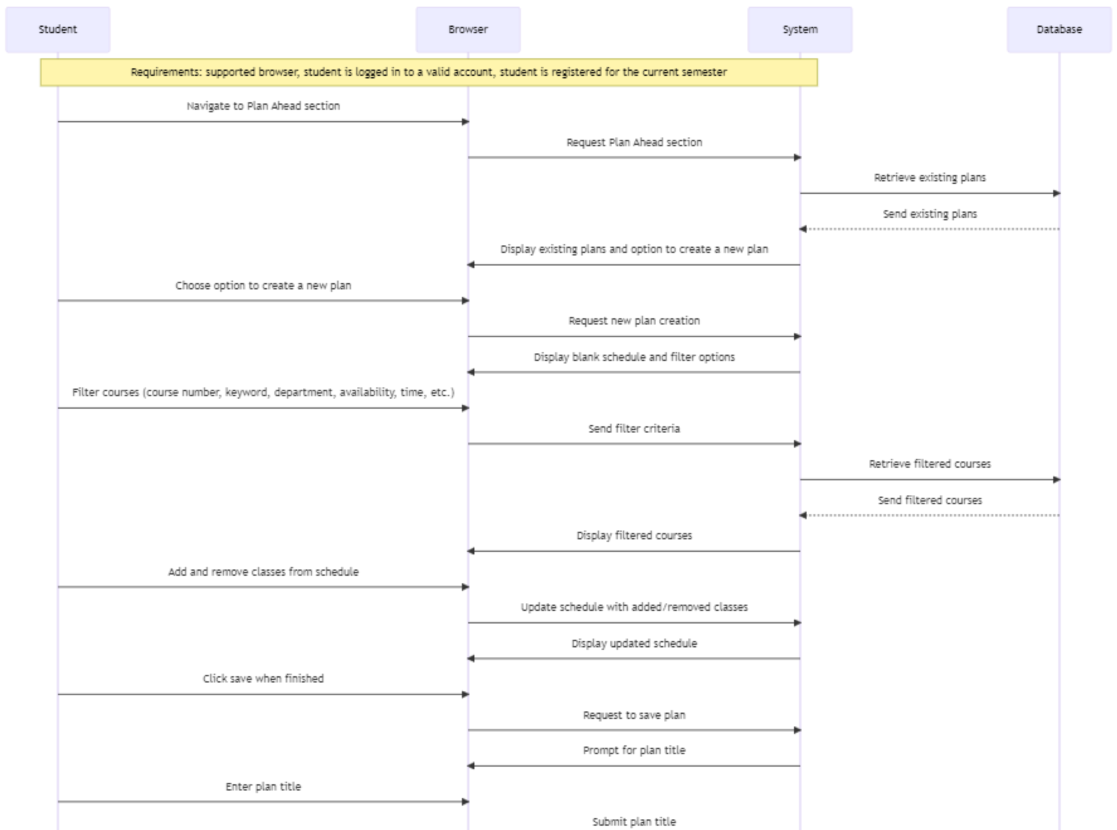
REQ-3: System must be able to display all potential schedules associated with the student.
REQ-4: System must be able to save all potential schedules data in the database.
REQ-5: System must be able to edit any existing potential schedules data in the database.
REQ-6: System must be able to delete any existing potential schedule from the database.
REQ-7: System must be able to filter class by course number, keyword, department, availability, time, etc.

4.2.4 Use Case Diagram


Create/Delete Potential Scehduale
Navigate to Plan Ahead section on Dashboard
Choose Create a New Plan
Fill out required fields and desired filters
Add Class
Remove Class
Click Save
Enter a Title for the Plan
Delete Plan
Edit Plan
Student
User w
edit/creat
potential
sc
Selected plan is retriev from database and displayed
ady navigated to "Plan
gh their dashboard
blank plan is displayed

4.2.5 Activity Diagram

4.2.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account, student is registered for the current semester
Navigate to Plan Ahead section
Request Plan Ahead section
Retrieve existing plans
Send existing plans
Display existing plans and option to create a new plan
Choose option to create a new plan
Request new plan creation
Display blank schedule and filter options
Filter courses (course number, keyword, department, availability, time, etc.)
Send filter criteria
Retrieve filtered courses
Send filtered courses
Display filtered courses
Add and remove classes from schedule
Update schedule with added/removed classes
Display updated schedule
Click save when finished
Request to save plan
Prompt for plan title
Enter plan title
Submit plan title

4.3 Request Course Substitution or Waiver

4.3.1 Description and Priority

Medium Priority:

- A student must be able to:
  - Request for a course substitution or waiver to their advisor for consideration.

4.3.2 Stimulus/Response Sequences

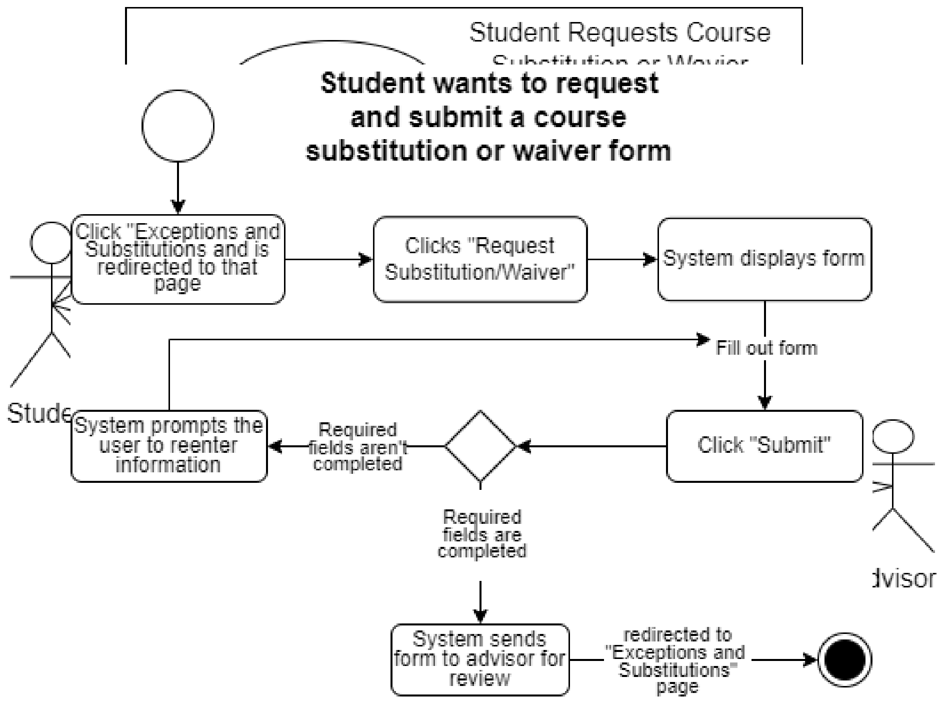
Request Substitution/Waiver:

1. Student navigates to the "Exceptions and Substitutions" section on their dashboard and clicks the "Request Substitution/Waiver" button.
2. The system presents the student with the request substitution/waiver form.
3. Student enters required information into textboxes and clicks the "Submit" button.
   a. If all required textboxes are answered, the system sends an email and notification to the advisor for review.
   b. If any required textboxes are left empty, the student will be prompted to enter the information again.

4.3.3 Functional Requirements

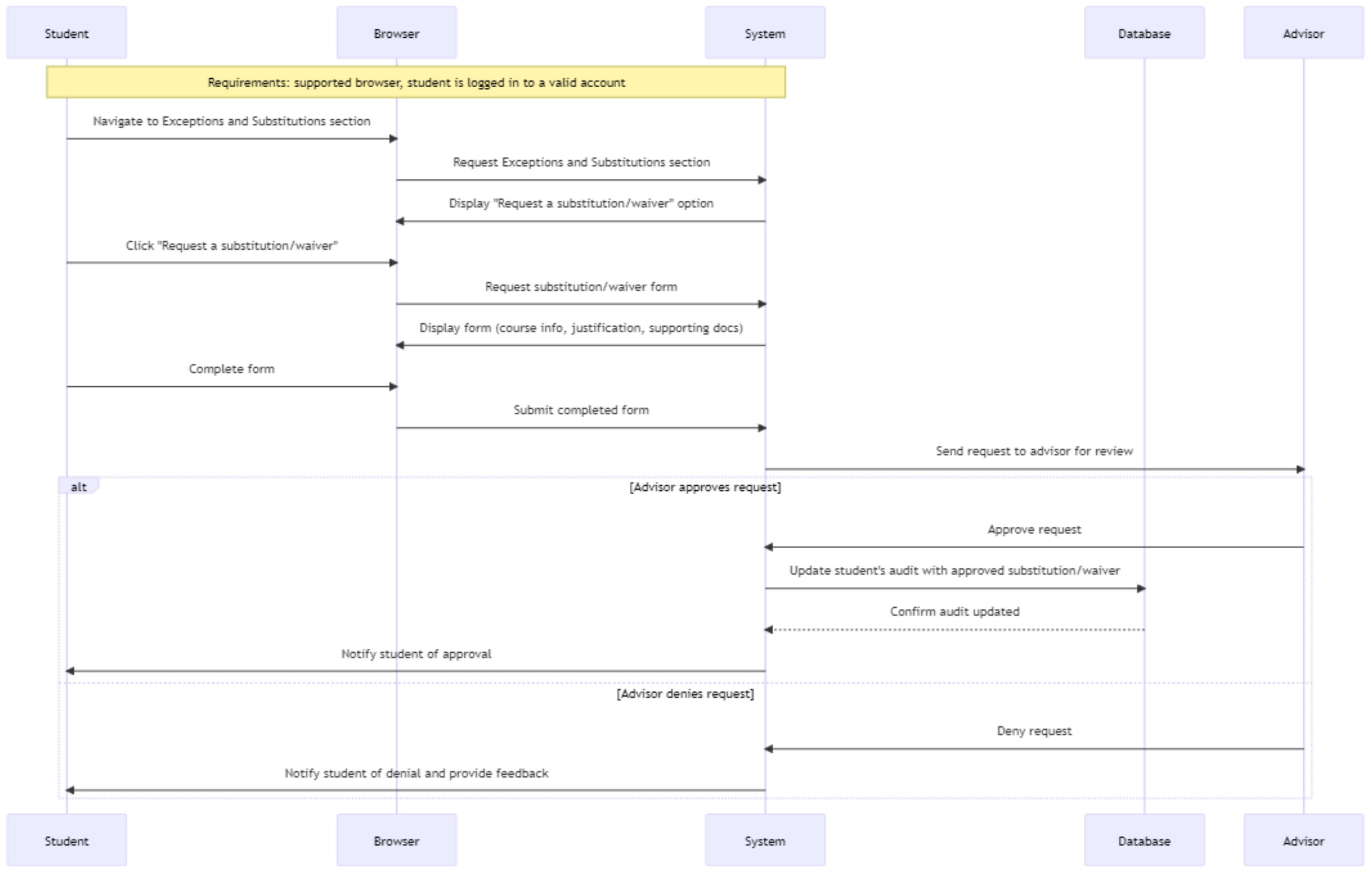
REQ-8: System must be able to keep a record of all forms submitted by the student.
REQ-9: System must keep track of the student's advisor.
REQ-10: System must be capable of sending an email and notification to the student's advisor.

4.3.4 Use Case Diagram


Student Requests Course
Substitution or Wavier
Student wants to request and submit a course substitution or waiver form
Click "Exceptions and Substitutions and is redirected to that page
Clicks "Request Substitution/Waiver"
System displays form
Fill out form
Click "Submit"
Stude
System prompts the user to reenter information
Required fields aren't completed
Required fields are completed
dvisor
System sends form to advisor for review
redirected to "Exceptions and Substitutions" page

4.3.5 Activity Diagram

4.3.6 Sequence Diagram


Student
Browser
System
Database
Advisor
Requirements: supported browser, student is logged in to a valid account
Navigate to Exceptions and Substitutions section
Request Exceptions and Substitutions section
Display "Request a substitution/waiver" option
Click "Request a substitution/waiver"
Request substitution/waiver form
Display form (course info, justification, supporting docs)
Complete form
Submit completed form
Send request to advisor for review
alt
[Advisor approves request]
Approve request
Update student's audit with approved substitution/waiver
Confirm audit updated
Notify student of approval
[Advisor denies request]
Deny request
Notify student of denial and provide feedback
Student
Browser
System
Database
Advisor

4.4 Schedule Meeting with Advisor

4.4.1 Description and Priority

Medium Priority:

- A student must be able to:
  - View their advisor's available meeting times.
  - Schedule a virtual or in-person meeting with their advisor.

4.4.2 Stimulus/Response Sequences

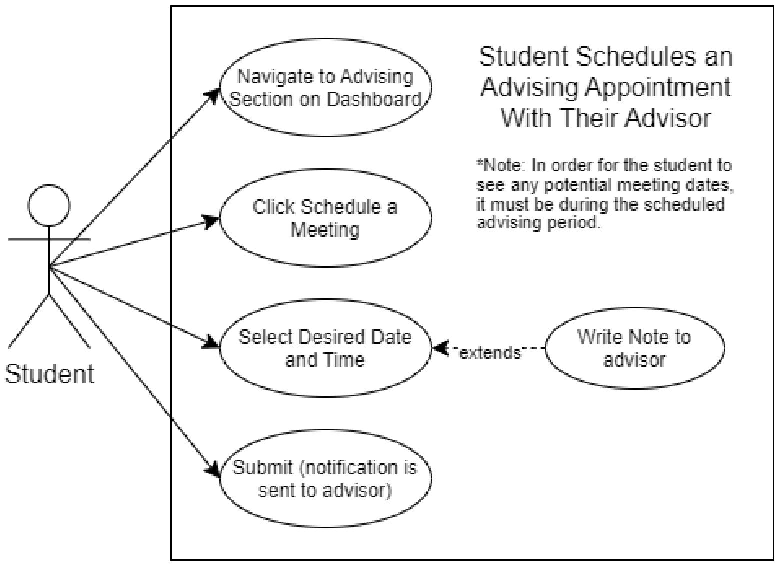
Schedule Meeting:

1. Student navigates to the "Advising" section on their dashboard and clicks the "Schedule a Meeting" button.
2. The system presents a calendar block that shows their advisor's available dates and times.
3. Student selects a date and time.
4. The system presents a confirmation screen stating the date and time selected and provides a textbox for the student to include a note.
5. Student clicks the "Submit" button.
6. System will process this submission, remove the date and time from the advisor's available meeting times, and send a notification and email to the advisor.

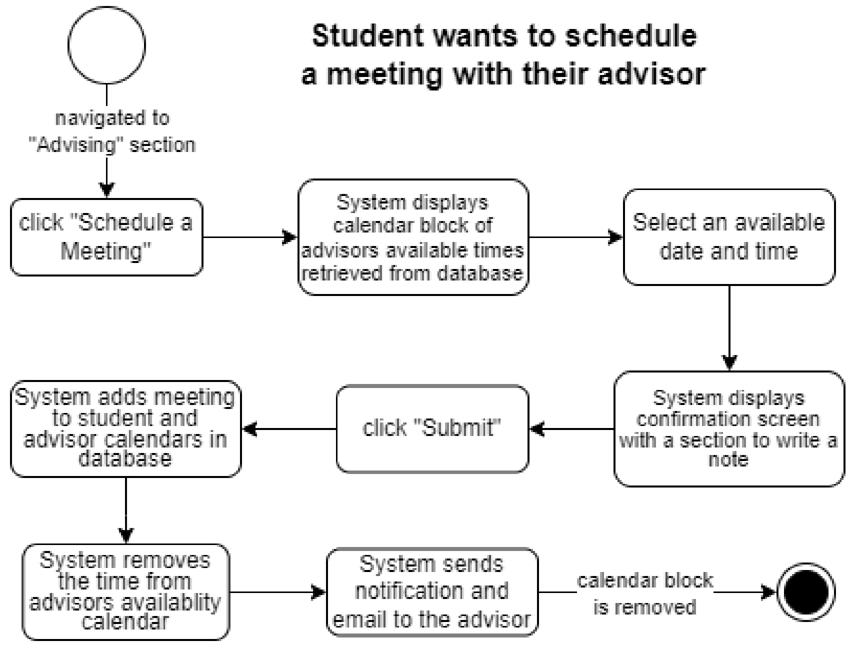
4.4.3 Functional Requirements

REQ-11: System must be able to keep an updated list of all available times for the calendar block.
REQ-12: System must be capable of handling scheduling requests from multiple users at a time.

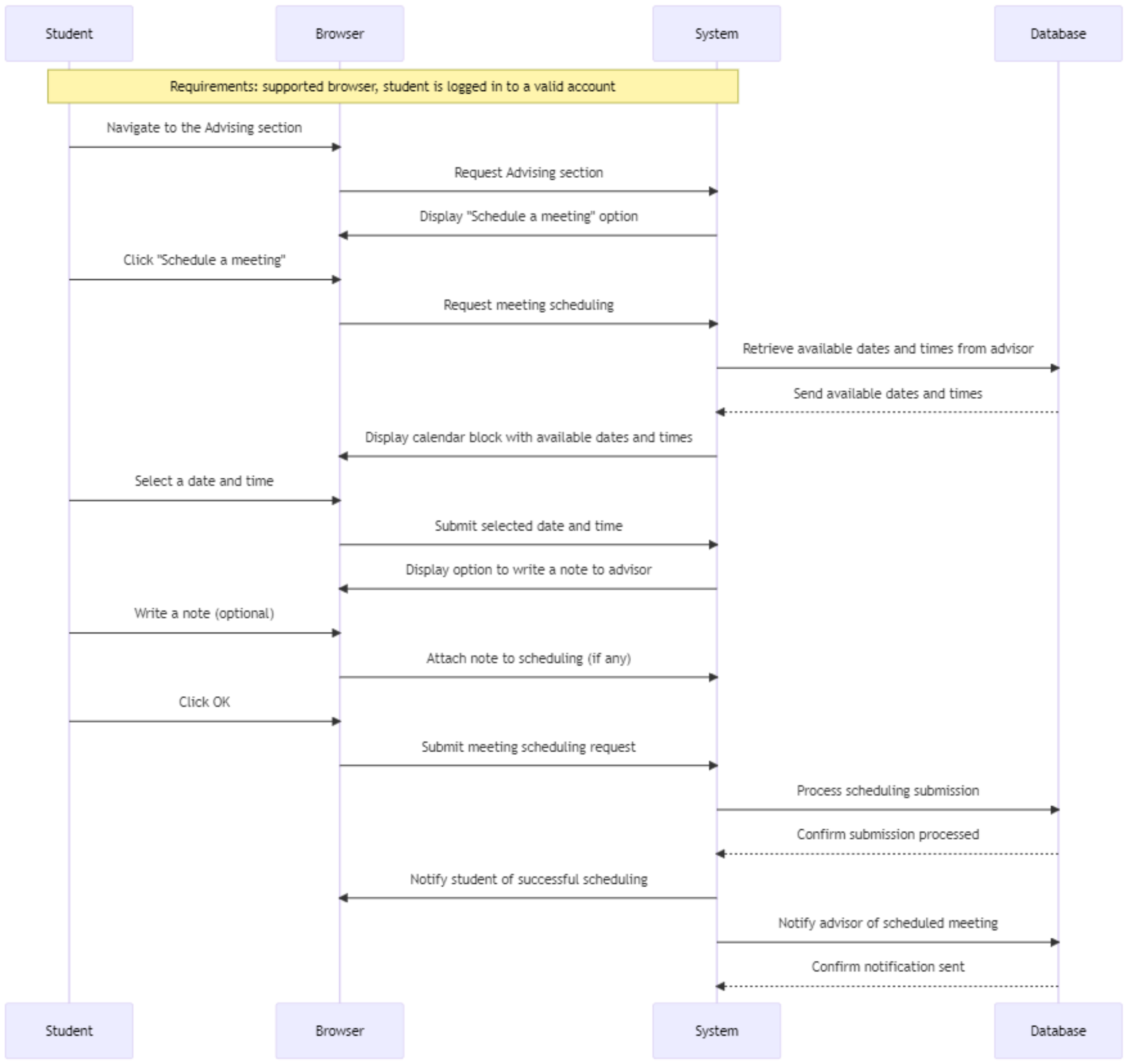
4.4.4 Use Case Diagram


Student Schedules an Advising Appointment With Their Advisor
*Note: In order for the student to see any potential meeting dates, it must be during the scheduled advising period.
Navigate to Advising Section on Dashboard
Click Schedule a Meeting
Select Desired Date and Time
extends
Write Note to advisor
Submit (notification is sent to advisor)
Student

4.4.5 Activity Diagram


Student wants to schedule a meeting with their advisor
navigated to "Advising" section
click "Schedule a Meeting"
System displays calendar block of advisors available times retrieved from database
Select an available date and time
System displays confirmation screen with a section to write a note
click "Submit"
System adds meeting to student and advisor calendars in database
System removes the time from advisors availablity calendar
System sends notification and email to the advisor
calendar block is removed

4.4.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to the Advising section
Request Advising section
Display "Schedule a meeting" option
Click "Schedule a meeting"
Request meeting scheduling
Retrieve available dates and times from advisor
Send available dates and times
Display calendar block with available dates and times
Select a date and time
Submit selected date and time
Display option to write a note to advisor
Write a note (optional)
Attach note to scheduling (if any)
Click OK
Submit meeting scheduling request
Process scheduling submission
Confirm submission processed
Notify student of successful scheduling
Notify advisor of scheduled meeting
Confirm notification sent
Student
Browser
System
Database

4.5 Add/Change Minor or Concentration

4.5.1 Description and Priority

Medium Priority:

- A student must be able to:
  - Add or modify a minor or concentration within their degree program.

4.5.2 Stimulus/Response Sequences

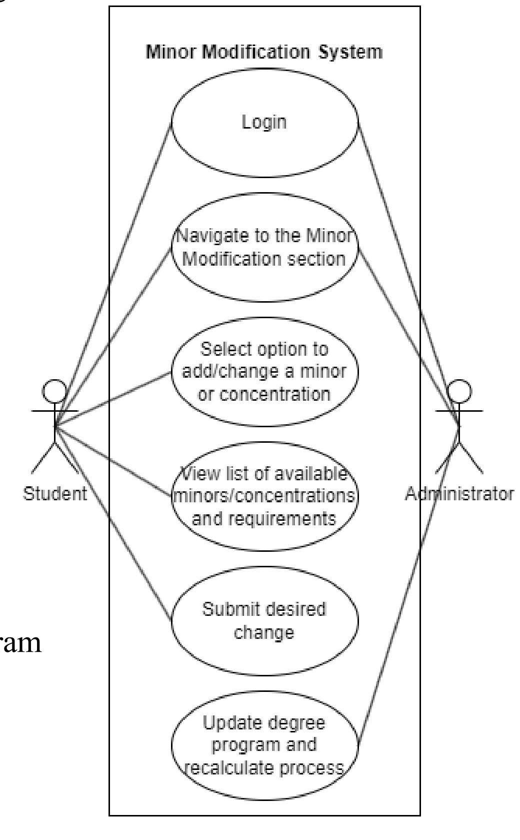
Add/Change Minor or Concentration:

1. Student navigates to the “Minor Modification” section on their dashboard and selects the “Add or Change a Minor or Concentration” option.
2. System presents a list of available minors or concentrations, along with any applicable requirements.
3. Student selects the desired minor or concentration and clicks the “Submit” button.
4. System will process this request and send a notification and email to the administrator.

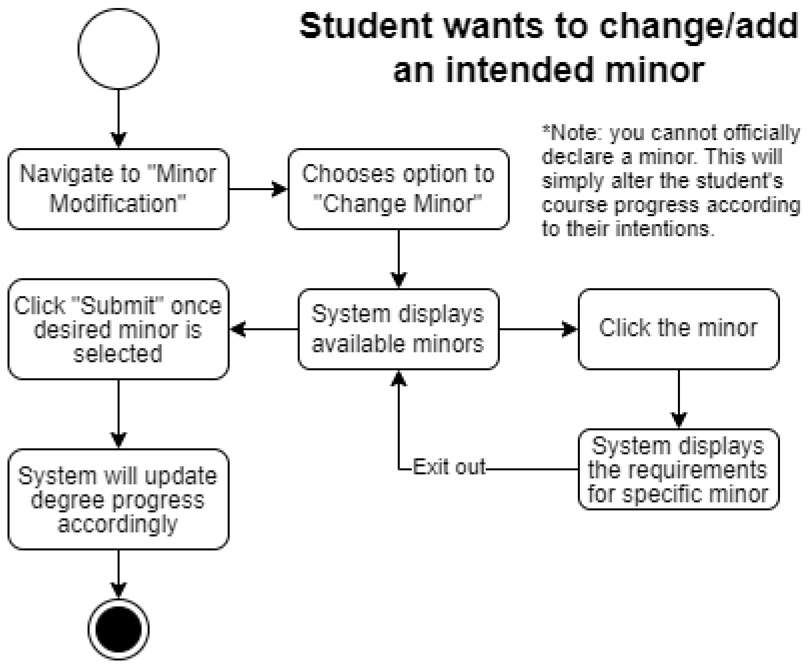
4.5.3 Functional Requirements

REQ-13: System must be able to display all available minors/concentration at the university.
REQ-14: System must be capable of sending an email and notification to the student’s administrator.
REQ-15: System must be able to update a student’s minor/concentration.

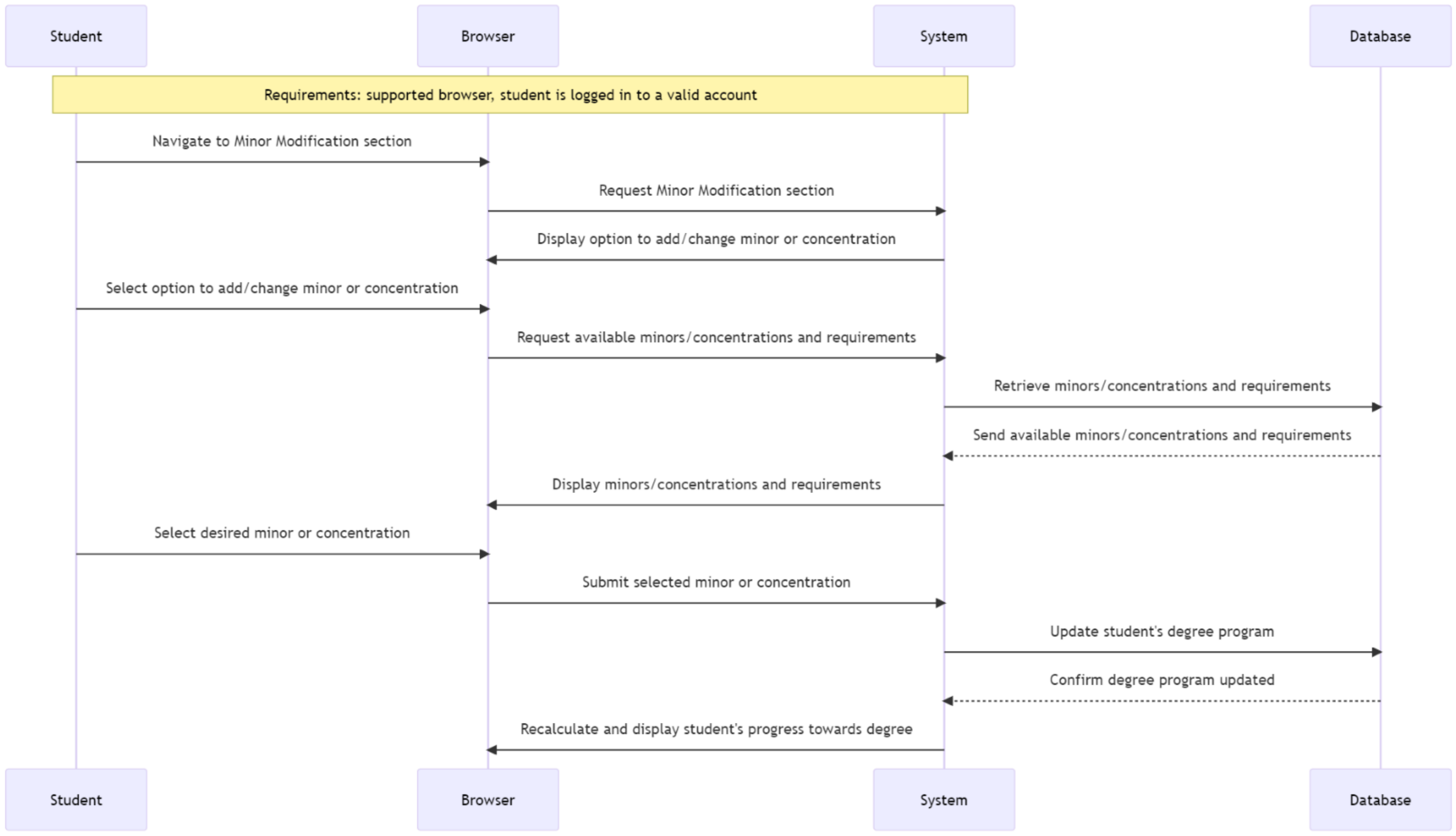
4.5.4 Use Case Diagram


Minor Modification System
Login
Navigate to the Minor Modification section
Select option to add/change a minor or concentration
View list of available minors/concentrations and requirements
Submit desired change
Update degree program and recalculate process
Student
Administrator

4.5.5 Activity Diagram


Student wants to change/add an intended minor
Navigate to "Minor Modification"
Chooses option to "Change Minor"
*Note: you cannot officially declare a minor. This will simply alter the student's course progress according to their intentions.
Click "Submit" once desired minor is selected
System displays available minors
Click the minor
System will update degree progress accordingly
Exit out
System displays the requirements for specific minor

4.5.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to Minor Modification section
Request Minor Modification section
Display option to add/change minor or concentration
Select option to add/change minor or concentration
Request available minors/concentrations and requirements
Retrieve minors/concentrations and requirements
Send available minors/concentrations and requirements
Display minors/concentrations and requirements
Select desired minor or concentration
Submit selected minor or concentration
Update student's degree program
Confirm degree program updated
Recalculate and display student's progress towards degree
Student
Browser
System
Database

4.6 Perform What-If Reports

4.6.1 Description and Priority

Medium Priority:

- A student must be able to:
  - See how changing their major would affect their degree progress.
  - Compare multiple academic programs.

4.6.2 Stimulus/Response Sequences

Effect of Changing Major:

1. Student navigates to the "What If" section on their dashboard.
2. System presents the "What If" page and lists any existing What If reports already made.
3. Student clicks the "Generate New What If" button.
4. System presents a block and filter options to the student.
5. Student selects their desired answers for the appropriate filters and click the "Submit" button.
6. System generates and presents an analysis of the student's expected graduation date and degree progress based on the student's current progress and their input.
7. System stores the generation data (date of generation, who generated it, physical report) in the database.

Compare Academic Programs:

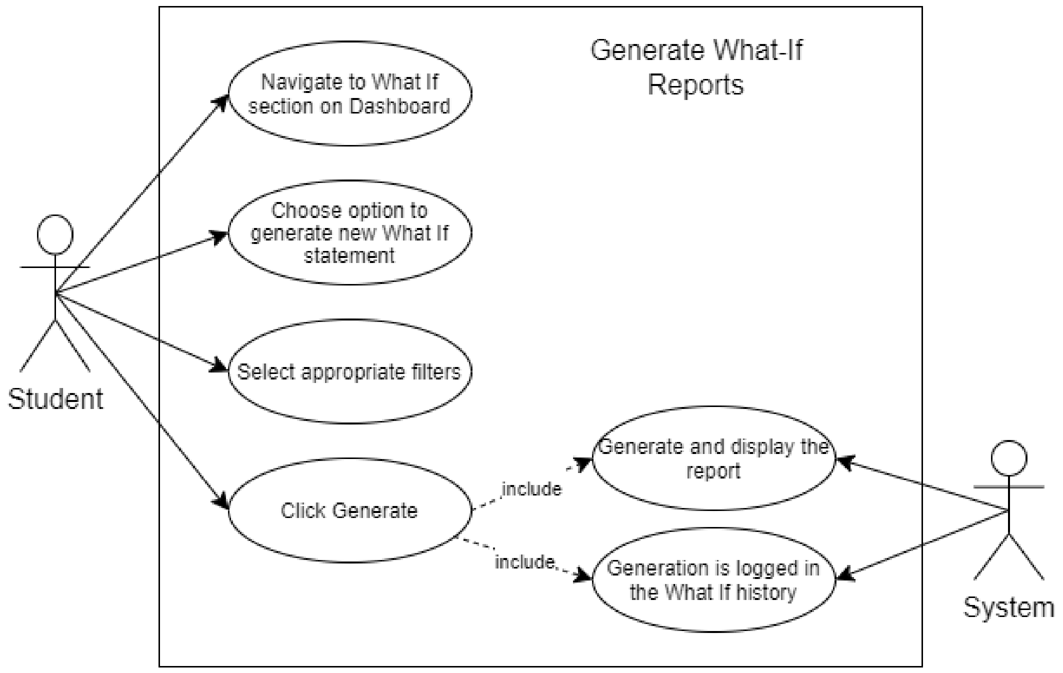
1. Student navigates to the "What If" section on their dashboard.
2. System presents the "What If" page.
3. Student clicks the "Program Comparison" button.
4. System presents a list of available programs to compare.
5. Student selects the programs they want to compare.
6. System generates and presents a side-by-side comparison of the selected programs, including degree requirements, course offerings, and potential career paths.
7. Student can:
   - Click the "Save" button to download a PDF file.
     - System will generate and present a PDF file of the comparison report.
   - Click the "Share" button to share with their advisor.
     - System will present a message box with a textbox for the student to include a note.
     - Student clicks the "Submit" button.

- System will process this submission and send a message and notification to the advisor.

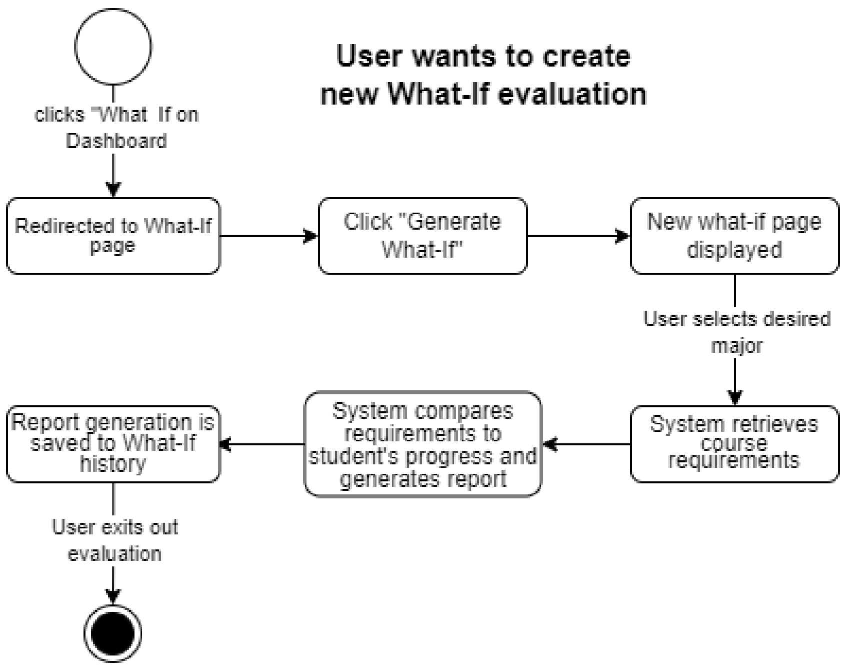
4.6.3 Functional Requirements

REQ-16: System must be able to display all available programs at the university.
REQ-17: System must be able to compare selected programs.
REQ-18: System must be able to generate a PDF report.

4.6.4 Use Case Diagram

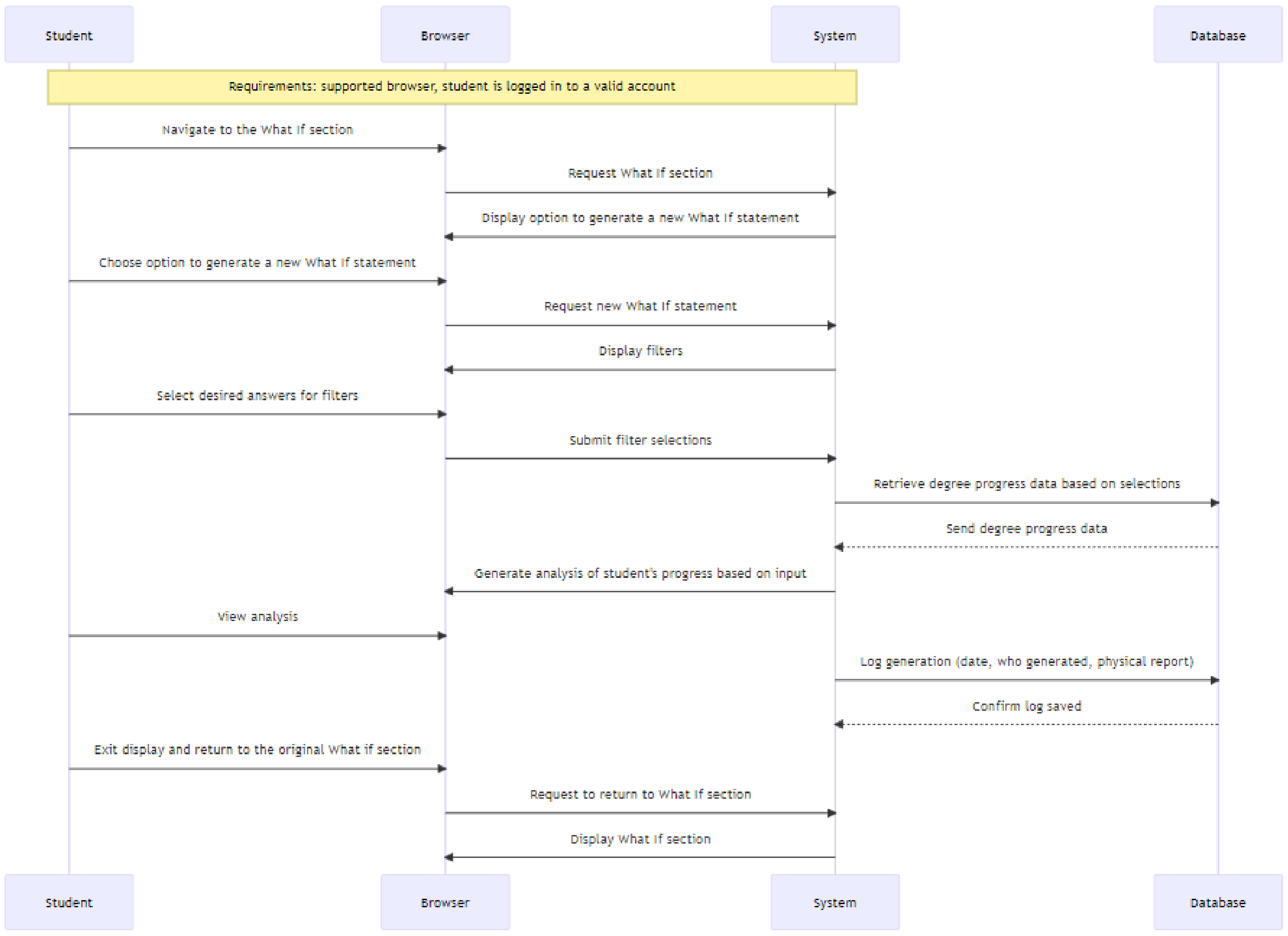

Generate What-If Reports
Navigate to What If section on Dashboard
Choose option to generate new What If statement
Select appropriate filters
Click Generate
include
include
Generate and display the report
Generation is logged in the What If history
Student
System

4.6.5 Activity Diagram

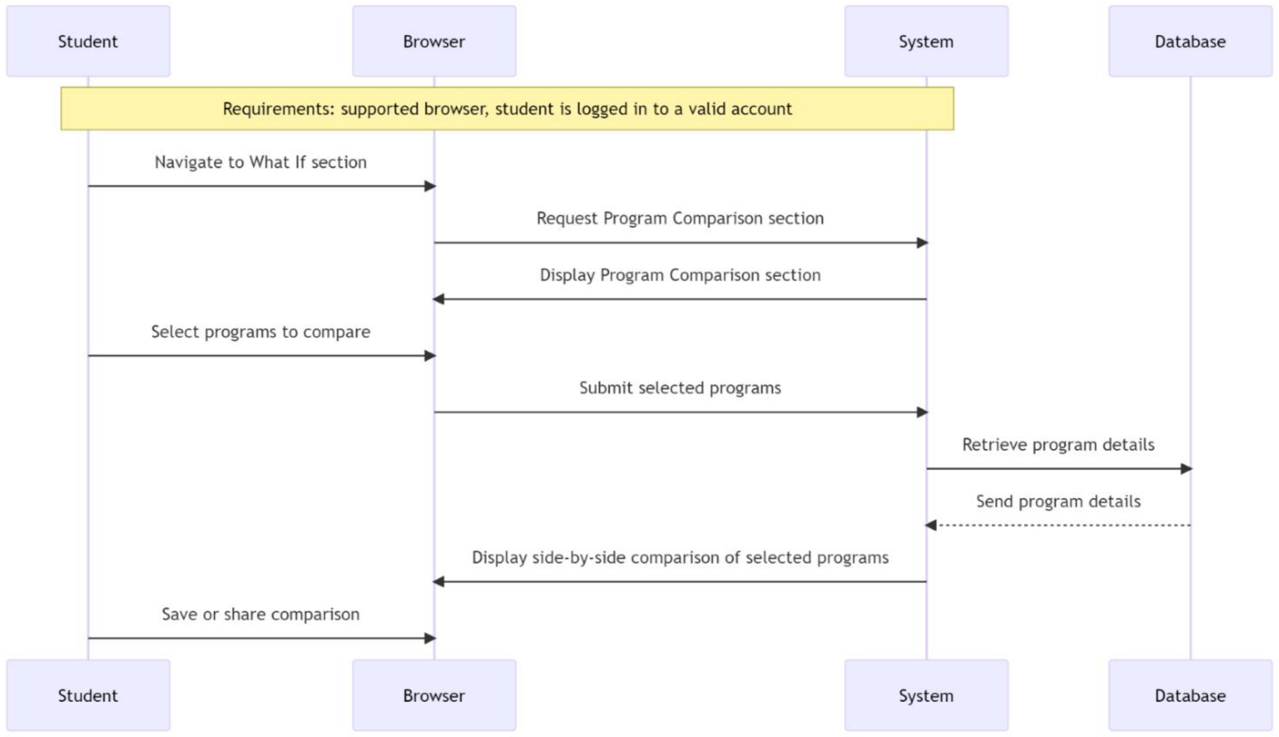

User wants to create new What-If evaluation
clicks "What If on Dashboard
Redirected to What-If page
Click "Generate What-If"
New what-if page displayed
User selects desired major
System retrieves course requirements
System compares requirements to student's progress and generates report
Report generation is saved to What-If history
User exits out evaluation

4.6.6 Sequence Diagram

<u>Effect of Changing Major:</u>


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to the What If section
Request What If section
Display option to generate a new What If statement
Choose option to generate a new What If statement
Request new What If statement
Display filters
Select desired answers for filters
Submit filter selections
Retrieve degree progress data based on selections
Send degree progress data
Generate analysis of student's progress based on input
View analysis
Log generation (date, who generated, physical report)
Confirm log saved
Exit display and return to the original What if section
Request to return to What If section
Display What If section
Student
Browser
System
Database

Compare Academic Programs:


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to What If section
Request Program Comparison section
Display Program Comparison section
Select programs to compare
Submit selected programs
Retrieve program details
Send program details
Display side-by-side comparison of selected programs
Save or share comparison
Student
Browser
System
Database

4.7 View Course History and Grades

4.7.1 Description and Priority

Medium Priority:

- A student must be able to:
    - Access and review previous courses.

4.7.2 Stimulus/Response Sequences

View Course History:

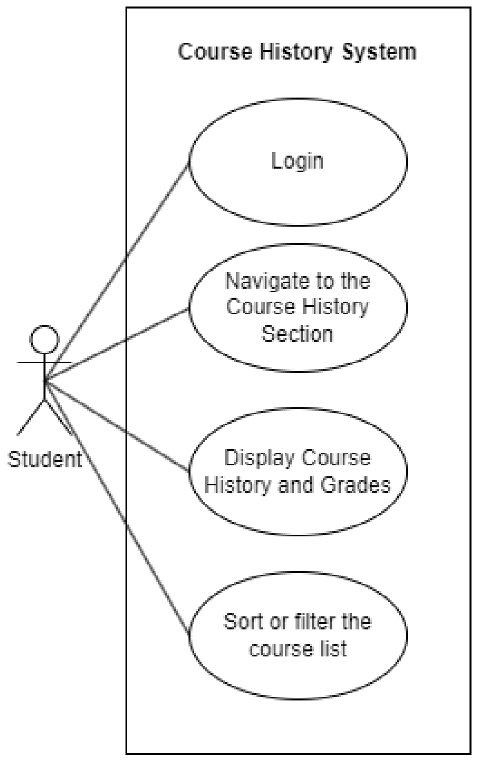
1. Student navigates to the "Course History" section on their dashboard.
2. System presents a list of all previously taken courses, including course names, course codes, grades, and semester taken.
3. Student can sort or filter the list by semester, department, grade, etc.

4.7.3 Functional Requirements

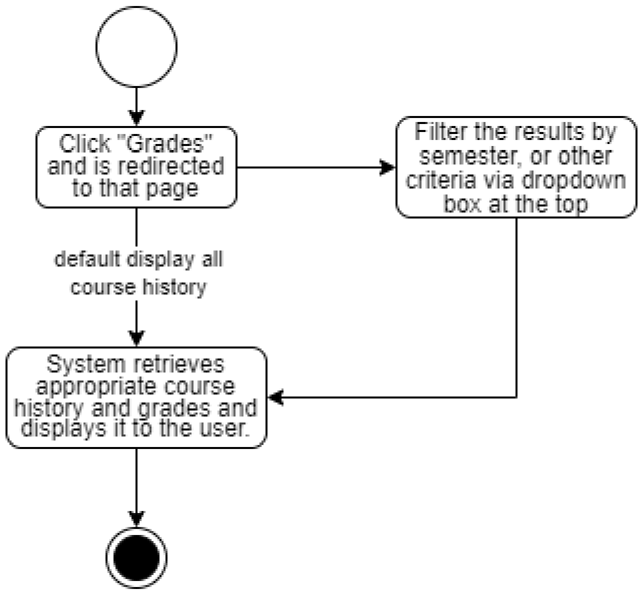
REQ-19: System must be able to keep a record of the students previously taken courses.

REQ-20: System must be able to sort or filter courses by semester, department, grade, etc.

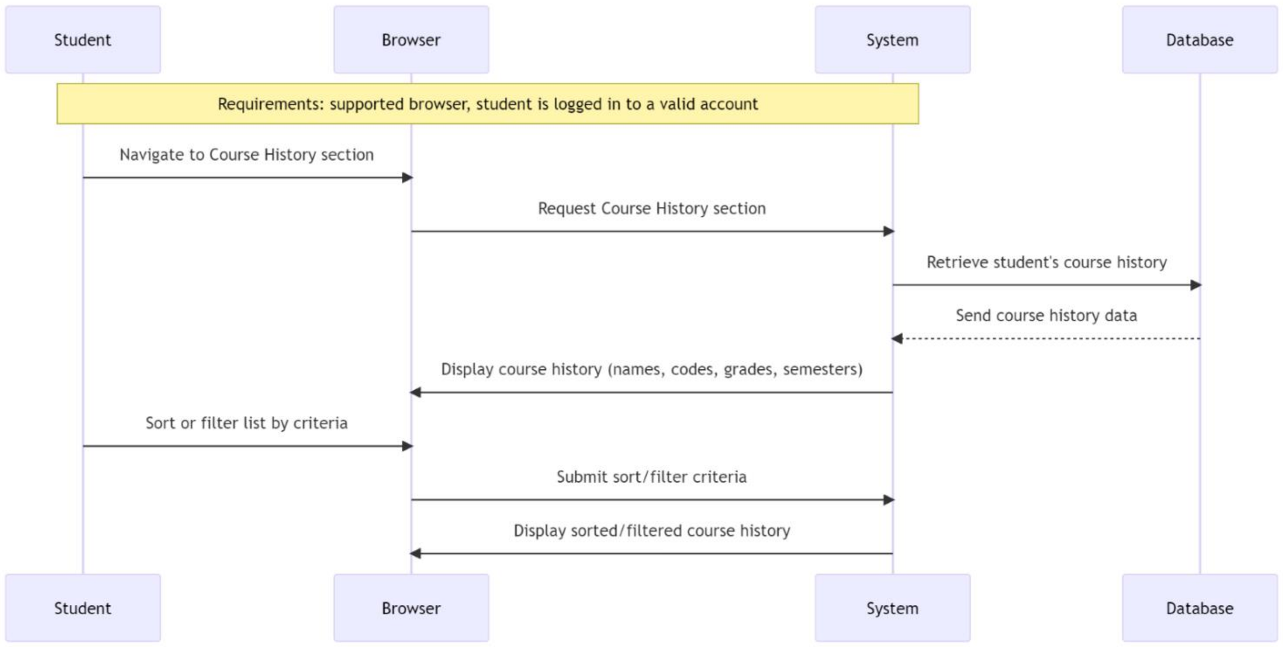
4.7.4 Use Case Diagram


Course History System
Login
Navigate to the Course History Section
Display Course History and Grades
Sort or filter the course list
Student

4.7.5 Activity Diagram


Click "Grades" and is redirected to that page
Filter the results by semester, or other criteria via dropdown box at the top
default display all course history
System retrieves appropriate course history and grades and displays it to the user.

4.7.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to Course History section
Request Course History section
Retrieve student's course history
Send course history data
Display course history (names, codes, grades, semesters)
Sort or filter list by criteria
Submit sort/filter criteria
Display sorted/filtered course history
Student
Browser
System
Database

4.8 Request Transcript

4.8.1 Description and Priority

<u>Medium Priority</u>:

- A student must be able to:
  - Request an official transcript to be sent to their desired recipient.
  - View an unofficial transcript.

4.8.2 Stimulus/Response Sequences

<u>Request Official Transcript</u>:

1. Student navigates to the "Transcript Request" section on their dashboard.
2. System presents the "Transcript Request" page.
3. Student selects "Request an Official Transcript."
4. System presents the student with the request official transcript form.
5. Student enters all required information into the provided fields and clicks the "Submit" button.
   - If all required information is entered and valid, the system will process this submission and send a notification to the administrator.
   - If all required information is not entered or is invalid, an error message will appear stating the problem, and the student will be prompted to enter the information again.

Request Unofficial Transcript:

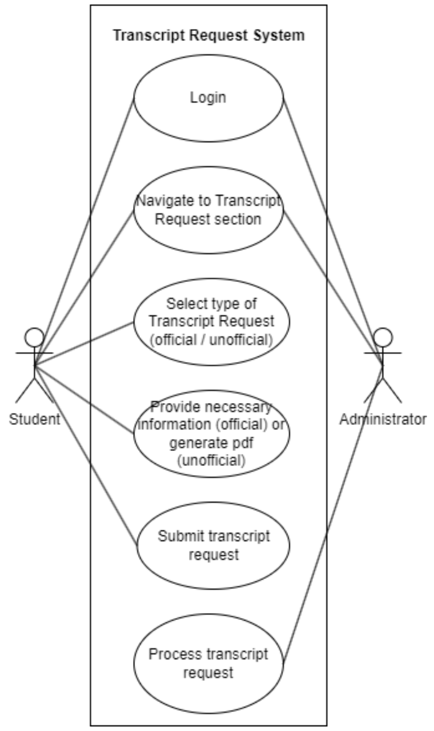
1. Student navigates to the "Transcript Request" section on their dashboard.
2. System presents the "Transcript Request" page.
3. Student selects "Request an Unofficial Transcript."
4. System generates and presents a PDF file that the student can download.

4.8.3 Functional Requirements

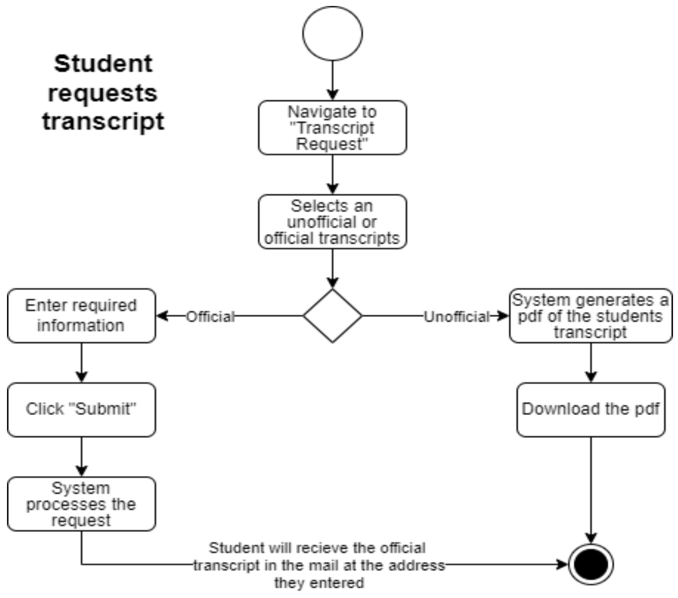
REQ-21: System must be able to keep a record of all requests submitted by the student.

REQ-22: System must be able to display the student's unofficial transcript.

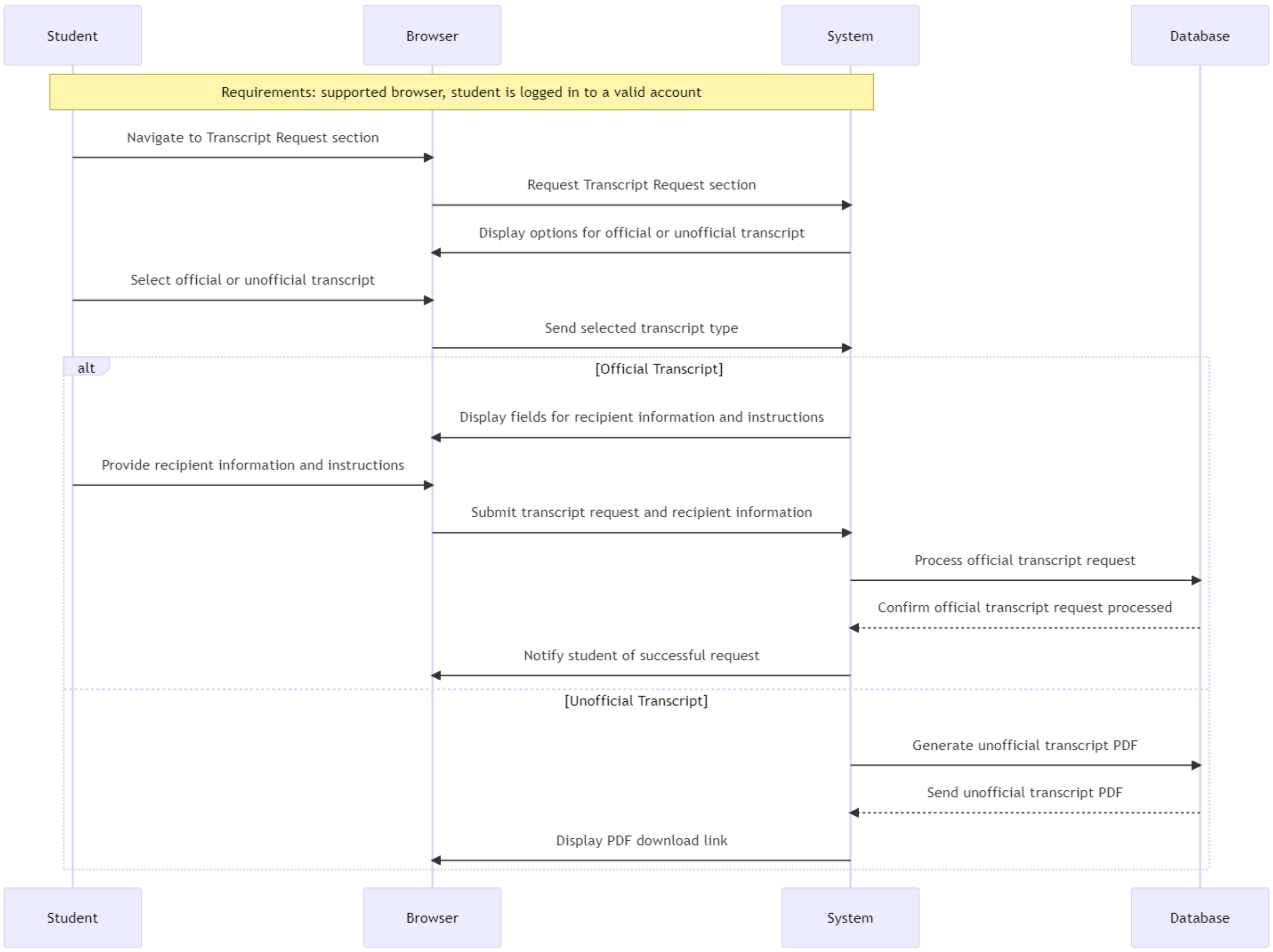
4.8.4 Use Case Diagram


Transcript Request System
Login
Navigate to Transcript Request section
Select type of Transcript Request (official / unofficial)
Provide necessary information (official) or generate pdf (unofficial)
Submit transcript request
Process transcript request
Student
Administrator

4.8.5 Activity Diagram


Student requests transcript
Navigate to "Transcript Request"
Selects an unofficial or official transcripts
Official
Unofficial
Enter required information
Click "Submit"
System processes the request
System generates a pdf of the students transcript
Download the pdf
Student will recieve the official transcript in the mail at the address they entered

4.8.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to Transcript Request section
Request Transcript Request section
Display options for official or unofficial transcript
Select official or unofficial transcript
Send selected transcript type
alt
[Official Transcript]
Display fields for recipient information and instructions
Provide recipient information and instructions
Submit transcript request and recipient information
Process official transcript request
Confirm official transcript request processed
Notify student of successful request
[Unofficial Transcript]
Generate unofficial transcript PDF
Send unofficial transcript PDF
Display PDF download link

4.9 Calculate Estimated GPA

4.9.1 Description and Priority

Medium Priority:

- A student must be able to:
  - Input their current course and expected grades to estimate their GPA for the current semester.

4.9.2 Stimulus/Response Sequences

Calculate Estimated GPA:

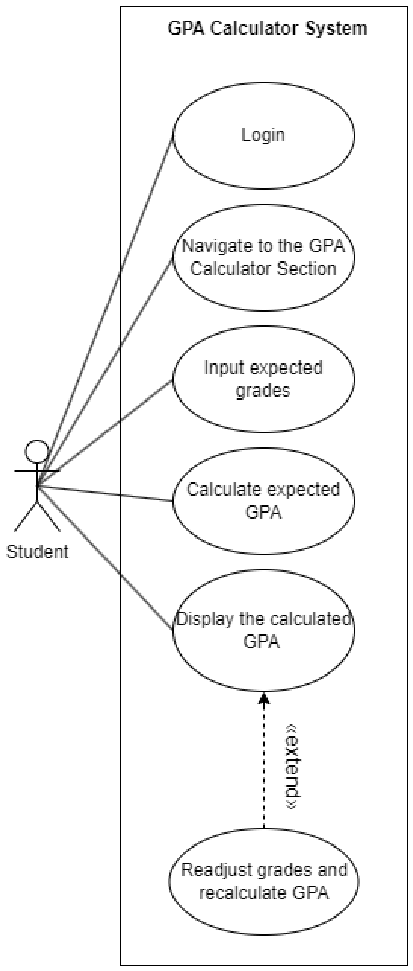
1. Student navigate to the “GPA Calculator” section on their dashboard.
2. System presents the “GPA Calculator” page.

3. Student enters their current course and expected grades into the provided fields.
4. System calculates the estimated GPA and presents the results to the student.
5. Student views the results and can adjust the expected grades to see how different outcomes may affect their GPA.

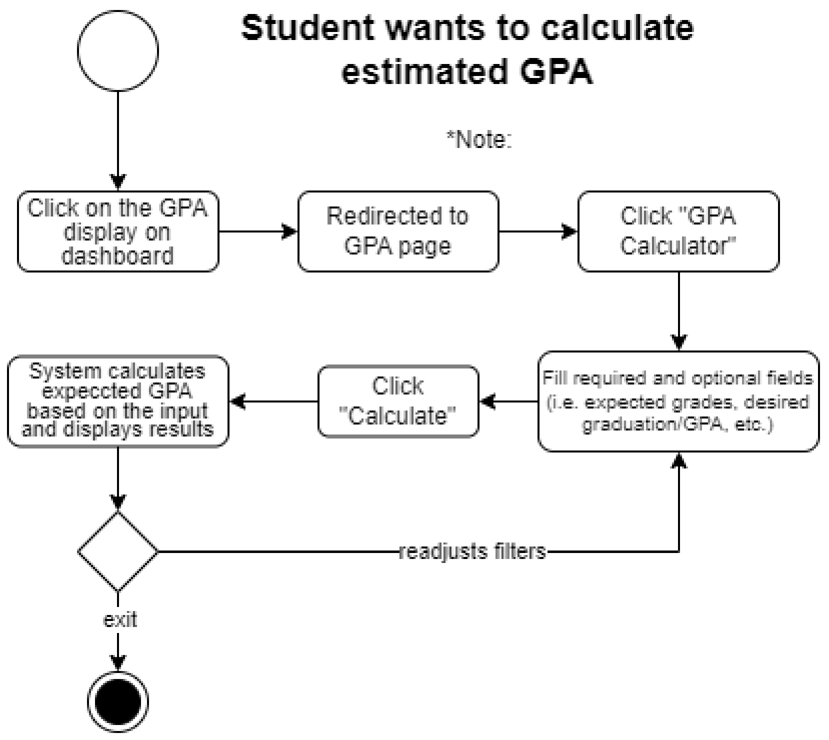
4.9.3 Functional Requirements

REQ-23: System must be able to calculate the estimated GPA.

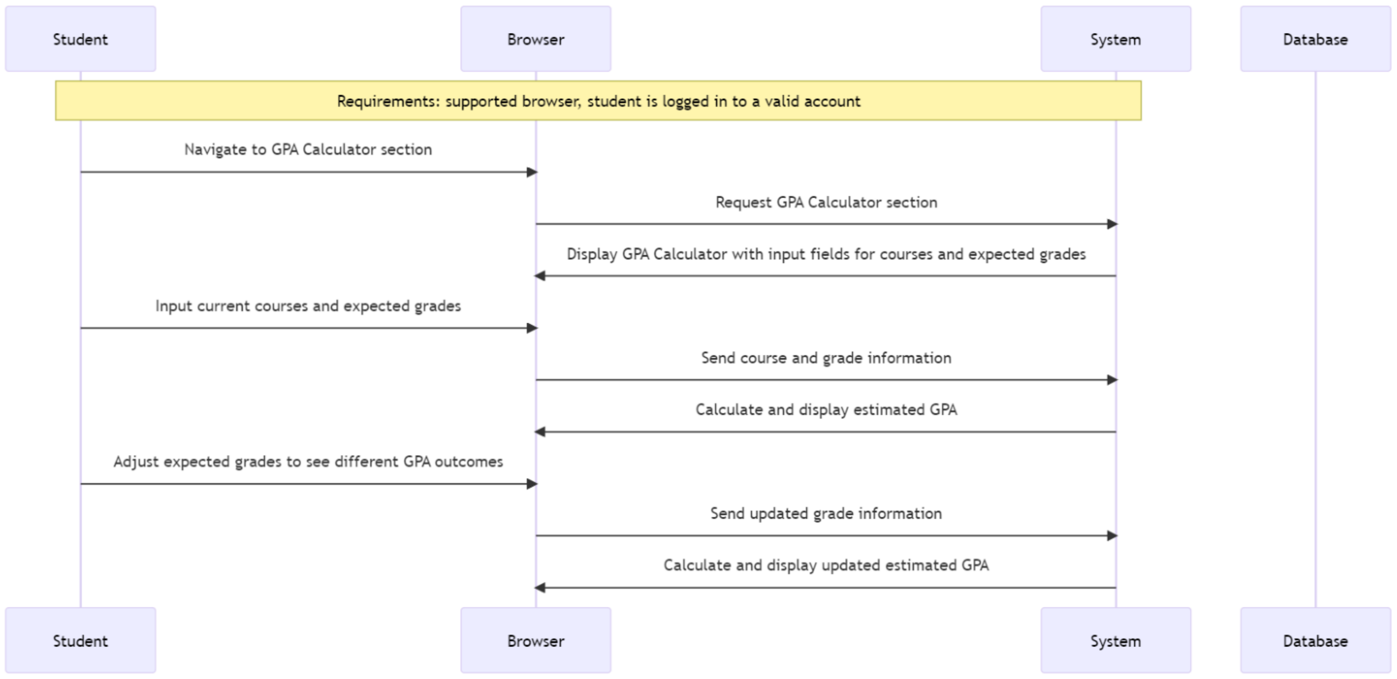
4.9.4 Use Case Diagram


GPA Calculator System
Login
Navigate to the GPA Calculator Section
Input expected grades
Calculate expected GPA
Display the calculated GPA
«extend»
Readjust grades and recalculate GPA
Student

4.9.5 Activity Diagram


Student wants to calculate estimated GPA
*Note:
Click on the GPA display on dashboard
Redirected to GPA page
Click "GPA Calculator"
Fill required and optional fields (i.e. expected grades, desired graduation/GPA, etc.)
Click "Calculate"
System calculates expeccted GPA based on the input and displays results
readjusts filters
exit

4.9.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to GPA Calculator section
Request GPA Calculator section
Display GPA Calculator with input fields for courses and expected grades
Input current courses and expected grades
Send course and grade information
Calculate and display estimated GPA
Adjust expected grades to see different GPA outcomes
Send updated grade information
Calculate and display updated estimated GPA
Student
Browser
System
Database

4.10 View Major/Minor Requirements

4.10.1 Description and Priority

Medium Priority:

- A student must be able to:
  - View the requirements of any major/minor offered by their university.

4.10.2 Stimulus/Response Sequences

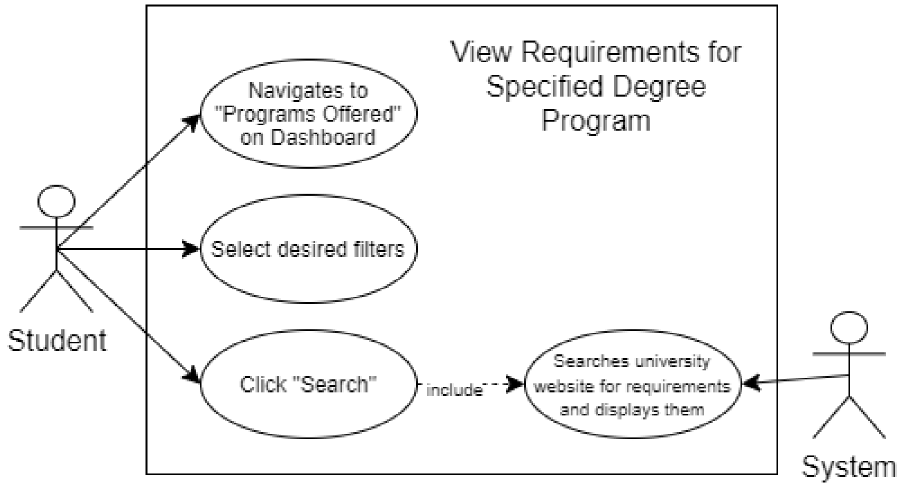
View Requirements of Specified Program:

1. Student Navigates to the “Programs Offered” section on their Dashboard.
2. Student selects certain filters, such as type of program (i.e., Bachelor’s), department, and whether they are looking for major, minor, etc.
3. The student clicks “Search.”
4. The system gets the degree program requirements from the university’s website.
5. The system displays the requirements to the user.

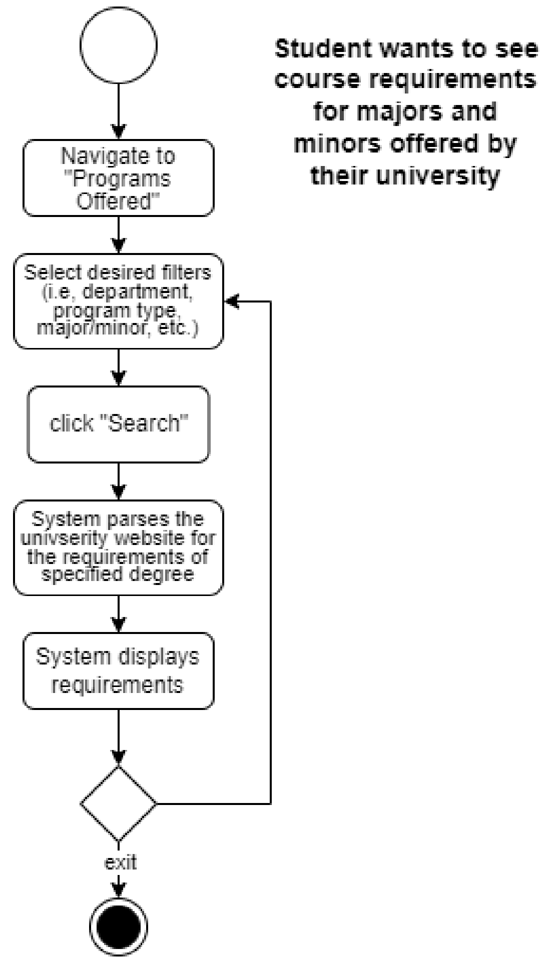
4.10.3 Functional Requirements

REQ-24: System must be able to filter programs offered at the university by type of program, department, etc.
REQ-25: System must be able to retrieve and display the degree program’s requirements.

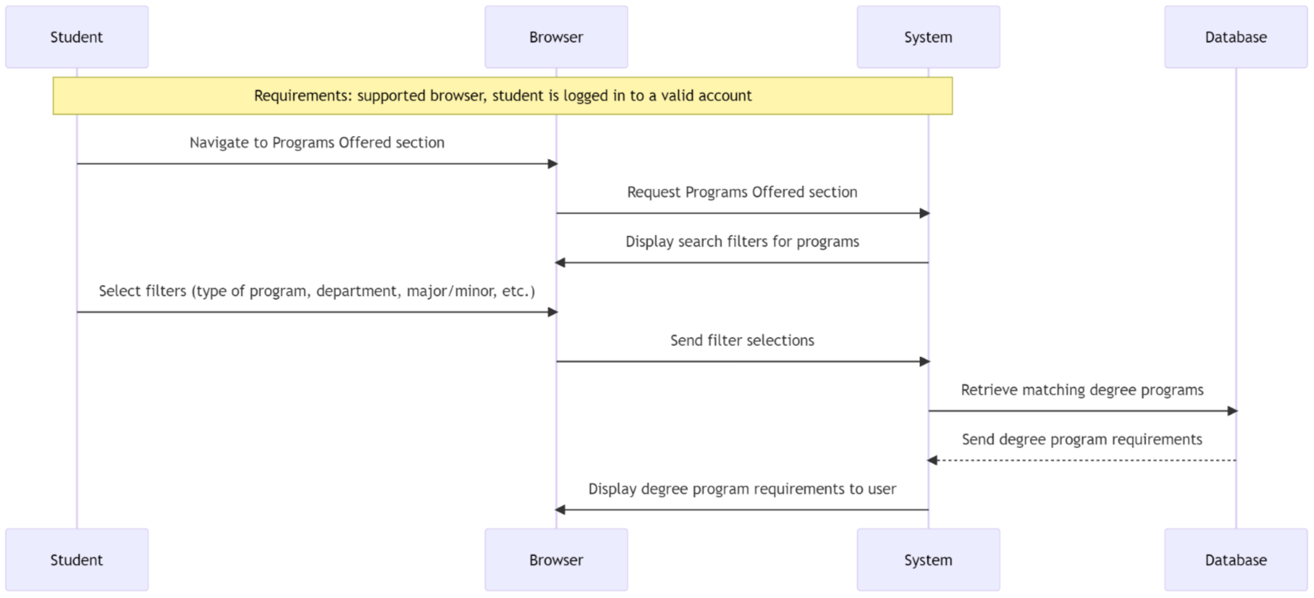
4.10.4 Use Case Diagram


View Requirements for Specified Degree Program
Navigates to "Programs Offered" on Dashboard
Select desired filters
Click "Search"
include
Searches university website for requirements and displays them
Student
System

4.10.5 Activity Diagram


Student wants to see course requirements for majors and minors offered by their university
Navigate to "Programs Offered"
Select desired filters (i.e, department, program type, major/minor, etc.)
click "Search"
System parses the univserity website for the requirements of specified degree
System displays requirements
exit

4.10.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Navigate to Programs Offered section
Request Programs Offered section
Display search filters for programs
Select filters (type of program, department, major/minor, etc.)
Send filter selections
Retrieve matching degree programs
Send degree program requirements
Display degree program requirements to user
Student
Browser
System
Database

4.11 Messaging System

4.11.1 Description and Priority

Medium Priority:

- A student must be able to:
  - Send a message with a subject and a body to their advisor and their advisor should receive the message and a notification.

4.11.2 Stimulus/Response Sequences

Send Message:

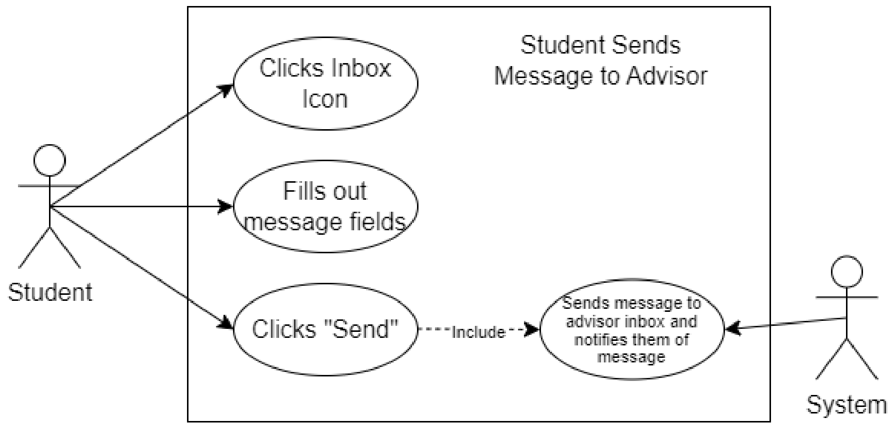
1. Student clicks the inbox icon on their Dashboard.
2. Student clicks the plus button.
3. System displays an empty message box.
4. Student fills out the subject and body of the message.
5. Student clicks "Send."
   a. If either the subject or body is empty the system will ask if the student is sure they want to send the message.
6. System sends the message to the student's advisor's inbox.
7. System will also send an email notification to the advisor via email.

View Messages:

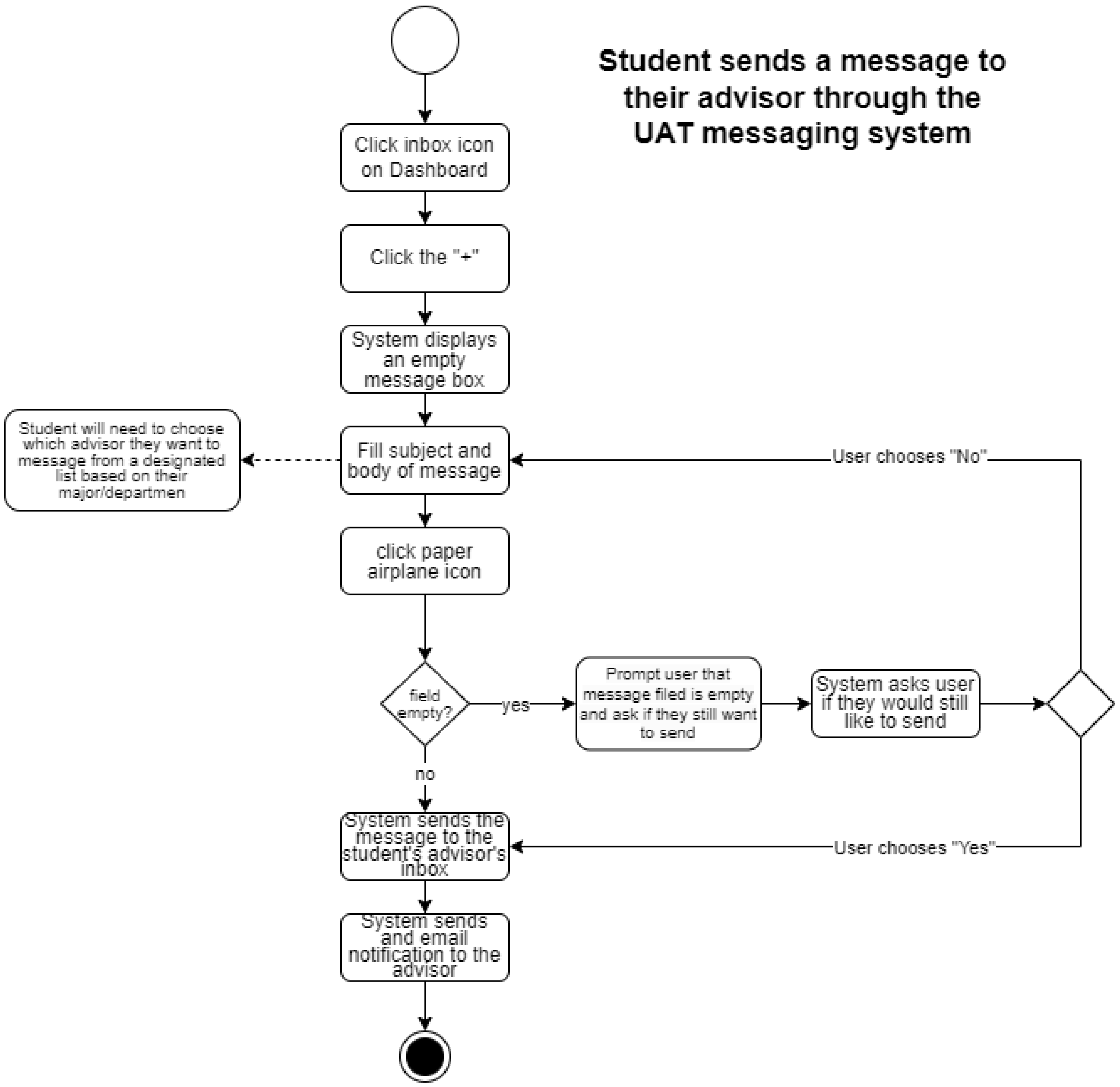
4.11.3 Functional Requirements

REQ-26: System must be able to retrieve and display the user's message data.
REQ-27: System must be able to send a message from one user to another through the internal inbox message system.

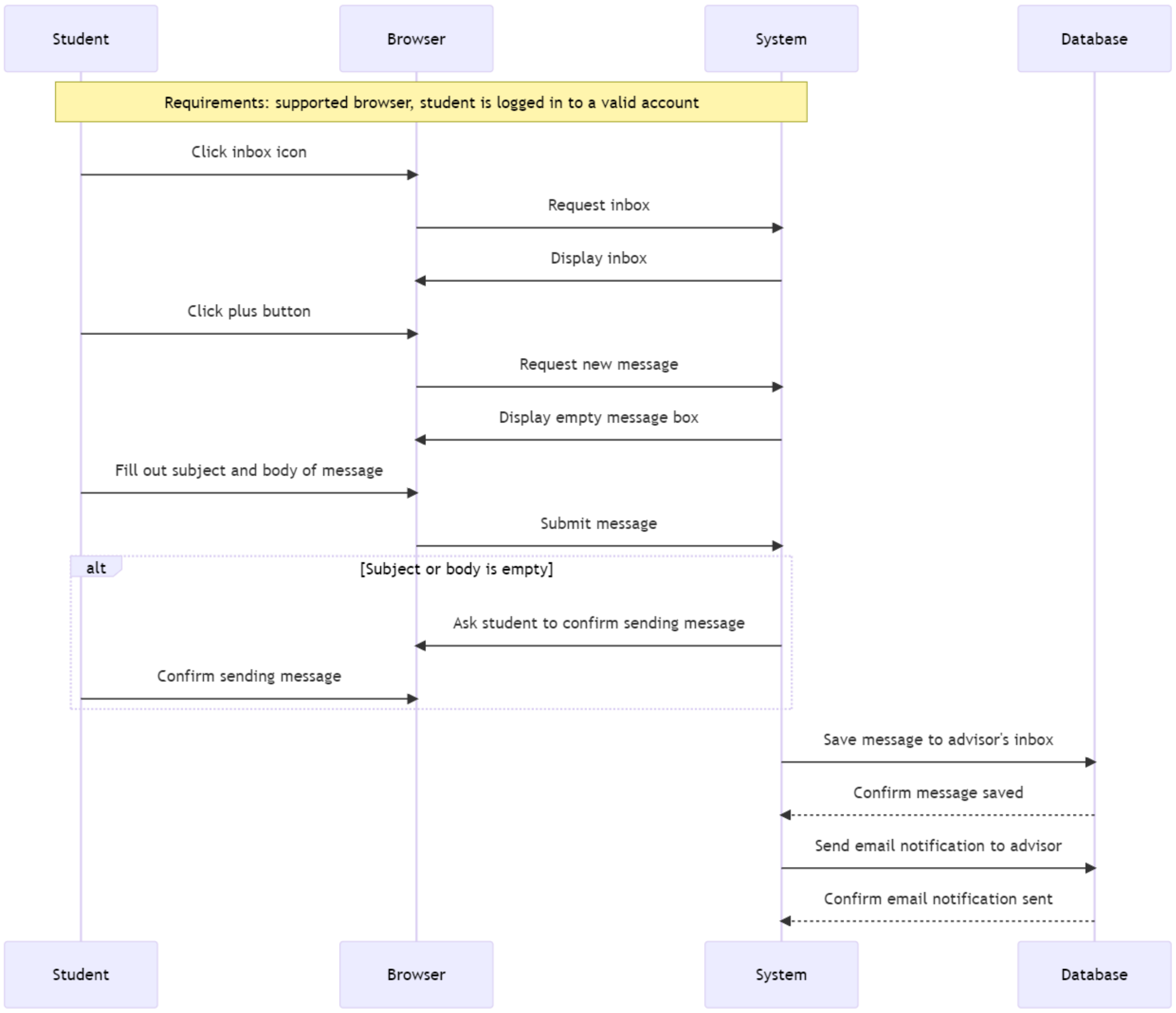
4.11.4 Use Case Diagram


Student Sends Message to Advisor
Clicks Inbox Icon
Fills out message fields
Clicks "Send"
Include
Sends message to advisor inbox and notifies them of message
Student
System

4.11.5 Activity Diagram


Student sends a message to their advisor through the UAT messaging system
Click inbox icon on Dashboard
Click the "+"
System displays an empty message box
Fill subject and body of message
Student will need to choose which advisor they want to message from a designated list based on their major/departmen
click paper airplane icon
field empty?
yes
Prompt user that message filed is empty and ask if they still want to send
System asks user if they would still like to send
User chooses "No"
User chooses "Yes"
no
System sends the message to the student's advisor's inbox
System sends and email notification to the advisor

4.11.6 Sequence Diagram


Student
Browser
System
Database
Requirements: supported browser, student is logged in to a valid account
Click inbox icon
Request inbox
Display inbox
Click plus button
Request new message
Display empty message box
Fill out subject and body of message
Submit message
alt
[Subject or body is empty]
Ask student to confirm sending message
Confirm sending message
Save message to advisor's inbox
Confirm message saved
Send email notification to advisor
Confirm email notification sent
Student
Browser
System
Database

4.12 Edit Dashboard

4.12.1 Description and Priority

Medium Priority:

- A student must be able to:
  - Edit the widgets on their dashboard to their preference.

4.12.2 Stimulus/Response Sequences

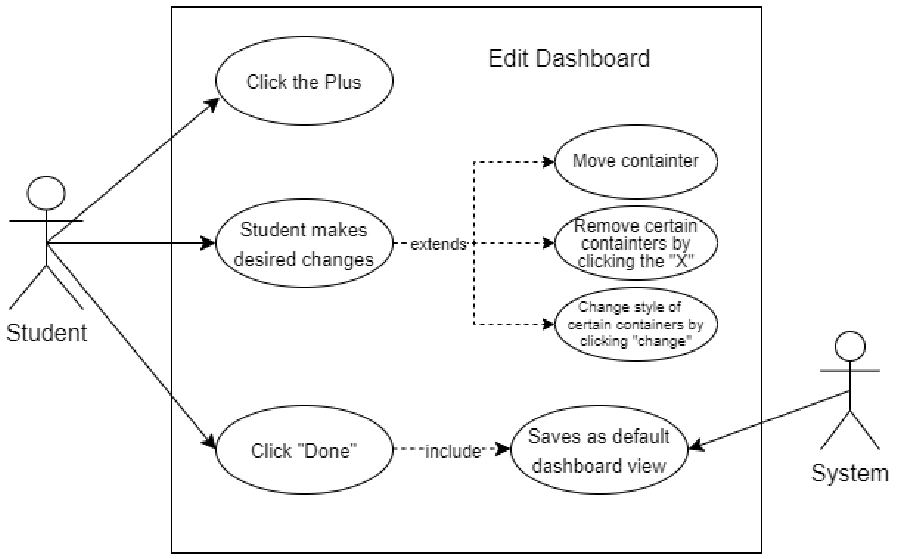
Edit Dashboard:

1. Student clicks the edit button on their Dashboard.
2. The system puts the student's dashboard in edit mode.
3. The student can drag and change the size of widgets, as well as delete certain widgets until they are satisfied with the layout.
4. Student clicks "Done."
5. System saves the dashboard layout as the student's default view.

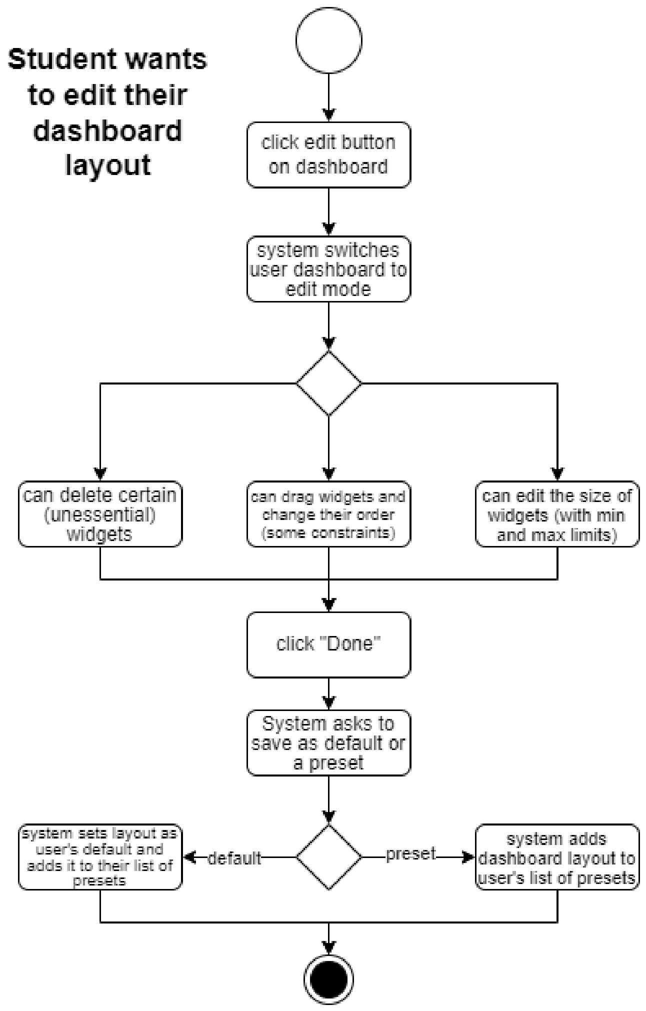
4.12.3 Functional Requirements

REQ-28: System must be able to override and save a default dashboard of the student's choosing.
REQ-29: System must allow for the user to edit widgets location, size, and viewability on their dashboard.

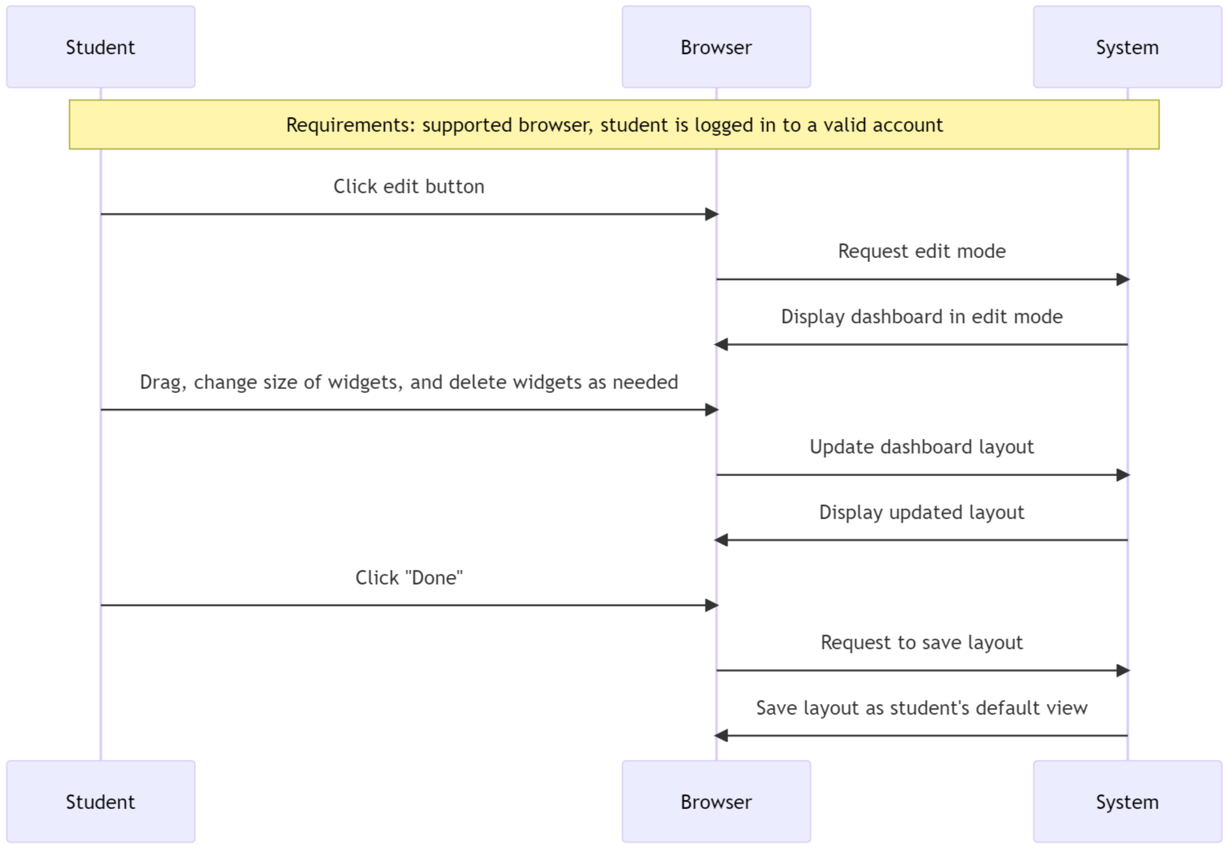
4.12.4 Use Case Diagram


Edit Dashboard
Click the Plus
Student makes desired changes
extends
Move containter
Remove certain containters by clicking the "X"
Change style of certain containers by clicking "change"
Click "Done"
include
Saves as default dashboard view
Student
System

4.12.5 Activity Diagram


Student wants to edit their dashboard layout
click edit button on dashboard
system switches user dashboard to edit mode
can delete certain (unessential) widgets
can drag widgets and change their order (some constraints)
can edit the size of widgets (with min and max limits)
click "Done"
System asks to save as default or a preset
system sets layout as user's default and adds it to their list of presets
default
preset
system adds dashboard layout to user's list of presets

4.12.6 Sequence Diagram


Student
Browser
System
Requirements: supported browser, student is logged in to a valid account
Click edit button
Request edit mode
Display dashboard in edit mode
Drag, change size of widgets, and delete widgets as needed
Update dashboard layout
Display updated layout
Click "Done"
Request to save layout
Save layout as student's default view
Student
Browser
System

Advisor Features

TBD.

Administrator Features

TBD.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The UAT website will be designed and implemented to handle a high volume of users, simultaneous requests, and complex queries, without compromising its responsiveness, speed, or availability.

- The website should have a minimum uptime of 99% to ensure that it is always available to users.
- The website should have a fast-loading time, with each page loading within a maximum of three seconds, to reduce user frustration.
- The website should be able handle simultaneous requests, such as appointments scheduling, and provide confirmation to users within a maximum of three seconds.
- The website should be able to handle peak traffic loads without a significant decrease in performance.

5.2 Safety Requirements

The UAT website will be designed in such a way that it is free of any malware, viruses, and other harmful software that could damage the user's device. Appropriate security measures will be in place to prevent unauthorized access to users' information. Lastly, consistent maintenance and updates will be performed to ensure the website's safety and security.

5.3 Security Requirements

The UAT website will adhere to the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CPPA) to guarantee that user data is handled in a secure and confidential manner. Security safeguards will be put in place to prevent data breaches or any other security threat. This will include encryption of sensitive data, regular software updates, and the use of strong passwords. To ensure the authenticity of users, two-factor authentication will be required to login to the website.

5.4 Software Quality Attributes

- **Adaptability:** The website should be able to adapt to changing customer needs and preferences. One measure of adaptability is the number of new/updated features per quarter, which should be at least 3.
- **Maintainability:** The website should be easy to maintain and update. One measure of maintainability is the time required to implement an update, which should be no more than 2 hours.
- **Portability:** The website should be able to run on a variety of platforms and browsers. One measure of portability is the number of supported browsers and platforms, which should be at least 3.
- **Reliability:** The website should be available and responsive at all times. One measure of reliability is the uptime percentage, which should be at least 99%.
- **Robustness:** The website should be able to handle unexpected inputs and errors without crashing or producing incorrect results. One measure of robustness is the number of critical bugs found per quarter, which should be no more than 10.
- **Testability:** The website should be easy to test and validate. One measure of testability is the percentage of code covered by automated tests, which should be at least 80%.
- **Usability:** The website should be easy to use and navigate, with clear and intuitive instructions. One measure of usability is the success rate of users completing tasks without errors, which should be at least 90%.

The most important attributes for users will be usability and reliability as they directly impact the user experience. For developers, maintainability and testability will be the most important for ensuring the website can be updated and validated efficiently.

Appendix A: Glossary

- UAT – University Advising Tool
- TBD – To be determined
- Use Case Diagram – graphical description of a user's interaction with the system
- Activity Diagram – graphical description of the workflow of a functionality
- Sequence Diagram – graphical description of the interaction between processes in regards to a specific functionality

Appendix B: To Be Determined List

1. The name of the system (currently UAT)
2. Advisor features
3. Administrator features
4. Interface designs (excluding dashboard)
5. Language/Framework