Software Requirements Specification Document

Boardman Computer Science Lab Web Portal

Client: Mr. Christopher Dufour

Development Company: In-House Operations

Development Team:
Klei Bendo
Jack Brisson
Alex Landry
Samuel Morse
Aaron Schanck
Forrest Swift

Date: 10/18/2021

Boardman Computer Science Lab Web Portal
System Requirements Specification

Table of Contents

1. Introduction

The Boardman Computer Science Lab Web Portal is an all encompassing tool for computer science students at the University of Maine. It is designed to ensure better help for students seeking aid in both specific inquiries, and broad subject areas at the Boardman Computer Science Lab through use of an interactive calendar, individual and group meeting scheduling, forum posting, and news updates. The Web Portal will make the Boardman Computer Science Lab more accessible and easy to use for University of Maine Computer Science Students.

1.1 Purpose of This Document

The purpose of this document is to define and describe the purposes, processes, and goals encapsulated within the development of the Boardman Computer Science Lab Web Portal project. It is intended for viewership by the client for confirmation of procedure for the development team, and other interested parties for official documentation of the application. This document includes background information on the project's development, requirements of the application both functional and non-functional, necessary future documentation and development, and signed agreements of both the development team and client for adequate directive of project scope and features.

1.2. References

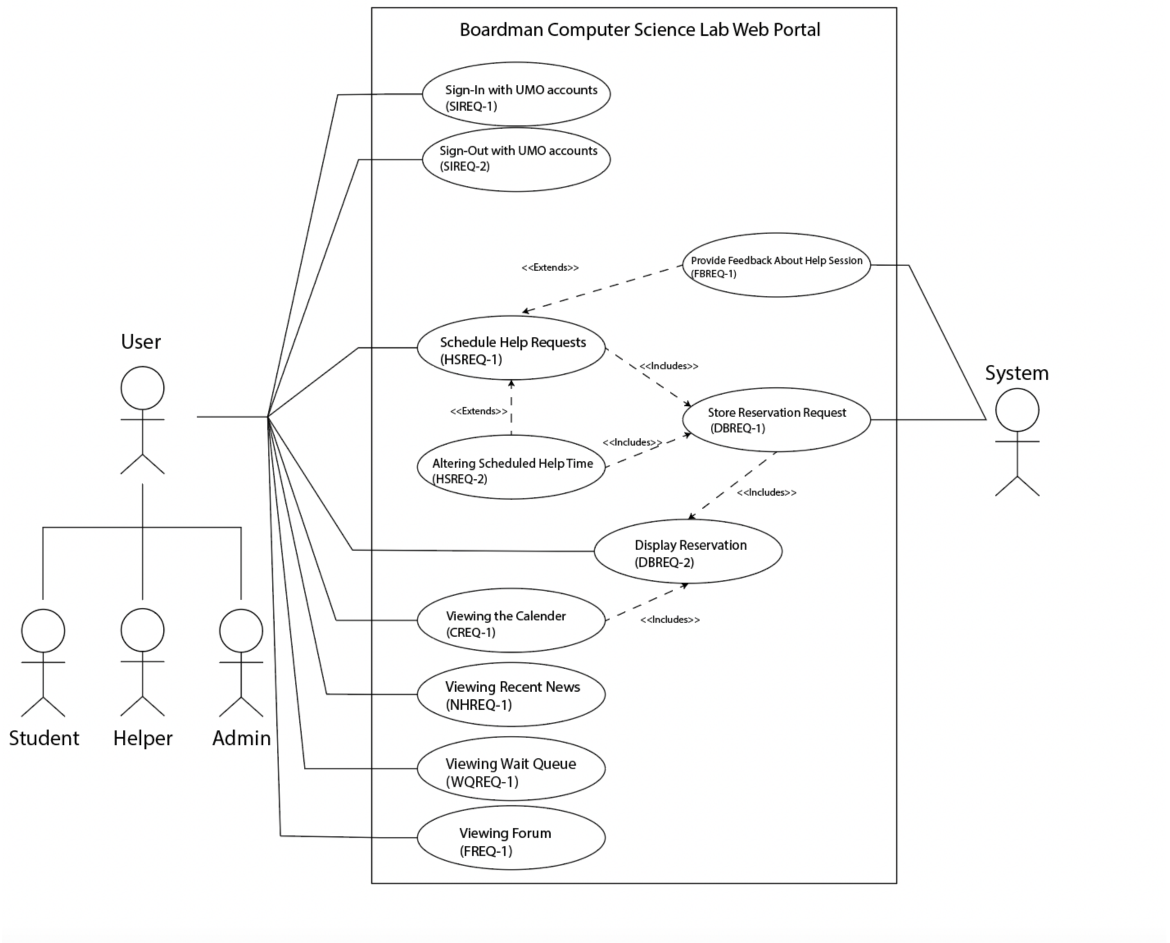
TBD - note: references will be added as they are created

1.3. Purpose of the Product

The Boardman Computer Science Lab is a long-standing resource for UMaine Computer Science students to receive help with computer science and other related issues, and as a meeting place for student groups. One common complaint facing the lab is a lack of clarity in regards to when lab helpers are available, and the areas in which these helpers are able to lend assistance. The Boardman Computer Science Lab Web Portal is a program aimed to both alleviate this complaint in a variety of ways, and to also track user data related to wait times and traffic within the labs to better address student needs moving forward.

1.4. Product Scope

This section identifies the boundary between the system under development and the outside world. To demonstrate this, a use case diagram is provided to display scope of functionality at the topmost level. This diagram displays the major features of the application and their interactions with the users and the system.


Boardman Computer Science Lab Web Portal
Sign-In with UMO accounts (SIREQ-1)
Sign-Out with UMO accounts (SIREQ-2)
<<Extends>>
Provide Feedback About Help Session (FBREQ-1)
User
Schedule Help Requests (HSREQ-1)
<<Includes>>
System
<<Extends>>
Store Reservation Request (DBREQ-1)
<<Includes>>
Altering Scheduled Help Time (HSREQ-2)
<<Includes>>
Display Reservation (DBREQ-2)
Viewing the Calender (CREQ-1)
<<Includes>>
Viewing Recent News (NHREQ-1)
Student
Helper
Admin
Viewing Wait Queue (WQREQ-1)
Viewing Forum (FREQ-1)

2. Functional Requirements

Included in this section are the functional requirements intended for the Boardman Computer Science Lab Web Portal. Each section is denoted by a major feature of the application. It contains a description and priority (1 = lowest, 5 = highest), stimulus/response sequences of the major requirements, the remainder of the associated functional requirements, and a use-case diagram of the feature. See the top-level use case diagram referred to in Section 1.4. for more context of the scope of this application.

2.1 Sign in / Sign out

2.1.1 Description and Priority

This feature will allow users to use their UMaine login credentials for the Boardman Lab website. This will allow the system to maintain security as well as store information about scheduled meeting times, requests for help and other information. This functionality is priority 5.

2.1.2 Stimulus/Response Sequence

Sign-In

Number	SIREQ-1
Name	Sign-In with UMO accounts
Summary	The system shall allow the user to enter the site with UMaine Login credentials
Priority	High
Preconditions	User has a umaine account.
Postconditions	User is signed into their umaine account and has access to the site
Primary Actors	Users
Secondary Actors	Google OAUTH, systems database
Triggers	Visiting the site
Main Scenario	1. User goes to the Boardman Lab website and selects “login” 2. User enters their UMaine credentials in the login screen 3. Using the OAUTH api, the system verifies the UMAINE user account and gives them access to the site.
Extensions	2.1 The user enters invalid credentials 2.2 The user is not allowed entry into the site
Open Issues	None

Sign-Out

Number	SIREQ-2
Name	Sign-Out of UMO account
Summary	The system shall allow the user to log out of the users account
Priority	High
Preconditions	Users are logged into their UMaine account on the boardman lab site.
Postconditions	User is signed out of their UMaine account and no longer have access to the site.

Primary Actors	Users
Secondary Actors	Google OAUTH, systems database
Triggers	Selecting “logout” button
Main Scenario	1. An onclick event calls signout function in OAuth API 2. System updates user authorization status 3. Site redirects user to the sign in page.
Extensions	1.1 User is signed in but inactive for 20 minutes 1.2 System calls signout function 2. -
Open Issues	None

2.1.3 Functional Requirements

SIREQ-3: The system shall utilize university of maine sign-in accounts for login.

SIREQ-4: The system shall utilize OAuth to manage logins.

SIREQ-5: The system shall allow the user to logout of their account.

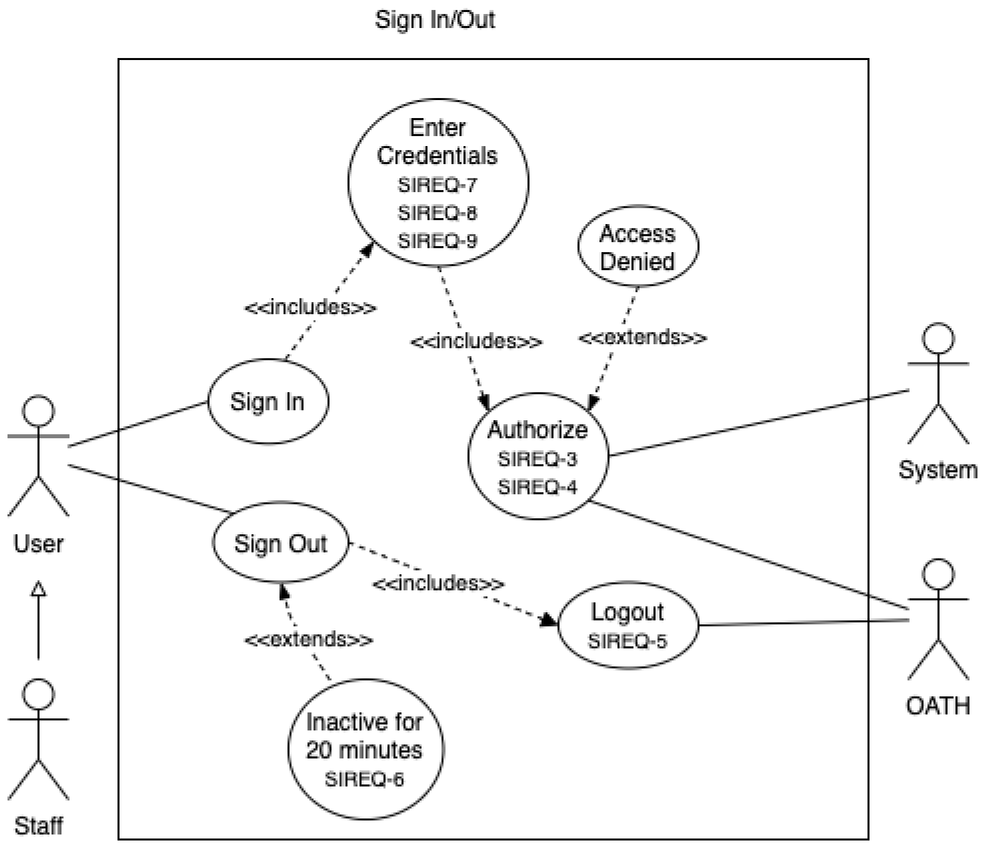
SIREQ-6: The system shall lock access to an account when not in use for a set amount of time.

SIREQ-7: The system shall have a user type for student.

SIREQ-8: The system shall have a user type for helper.

SIREQ-9: The system shall have a user type for administrator.

2.1.4 Use case diagram



2.2 Help Scheduling

2.2.1 Description and Priority

This feature will allow users to select time slots that are available on the calendar and create an instance of a help session. The user will be able to specify which class or topic they would like help with and be able to check availability of helpers. This functionality is priority 3.

2.2.2 Stimulus/Response Sequence

Schedule Help Request

Number	HSREQ-1
Name	The system shall allow users to Schedule help Requests
Summary	User selects time they would like to help and the system allows them to schedule that time with a helper either online or in person
Priority	Medium

Preconditions	User is signed in
Postconditions	System has created new item on the schedule for that student
Primary Actors	Users
Secondary Actors	System
Triggers	Selecting the “schedule help” button on the UI
Main Scenario	1. The user selects a date and time that they would like to get help by selecting an available time from the UI prompt. 2. The system checks to see if there is space available in the designated time slot wait queue.
Extensions	1.1 The user enters the class that they need help with in the prompt within the UI 1.2 The user selects the helper that they would like in the prompt within the UI 1.3 The user selects the mode of help: online or in person and one-on-one or group help. 2.1 If the selected time slot has too many students in the wait queue, the system displays a message saying that the selected time is busy. 2.2 If the selected time slot has space available in the wait queue, the system will add the student to the wait queue for that time slot.
Open Issues	None

Alter a Scheduled Help Request

Number	HSREQ-2
Name	The system shall allow the user to Alter a scheduled help time
Summary	User decided to cancel or move their scheduled time for a help session
Priority	High
Preconditions	User has a scheduled help session
Postconditions	User alters the scheduled help session
Primary Actors	Users
Secondary Actors	System

Triggers	User selects “Edit Scheduled Help Session”
Main Scenario	1. System displays the edit help session page
Extensions	1.1.1 User selects cancel help session 1.1.2 System cancels the help session and removes the user from the wait queue 1.2.1 User changes the scheduled help time 1.2.2 System cancels help session and removes user from the wait queue 1.2.3 System creates new scheduled help session with the new time.
Open Issues	None

2.2.3 Functional Requirements

HSREQ-3: The system shall allow requests for help at a specific date and time.

HSREQ-4: The system shall allow requests to schedule recurring meetings.

HSREQ-5: The system shall allow requests to search for available times matching a criteria.

HSREQ-6: The system shall allow requests to meet with a specific helper.

HSREQ-7: The system shall allow requests for help with a specific topic or course.

HSREQ-8: The system shall allow help reservations to have various modifications.

HSREQ-9: The system shall allow for remote or in person reservations.

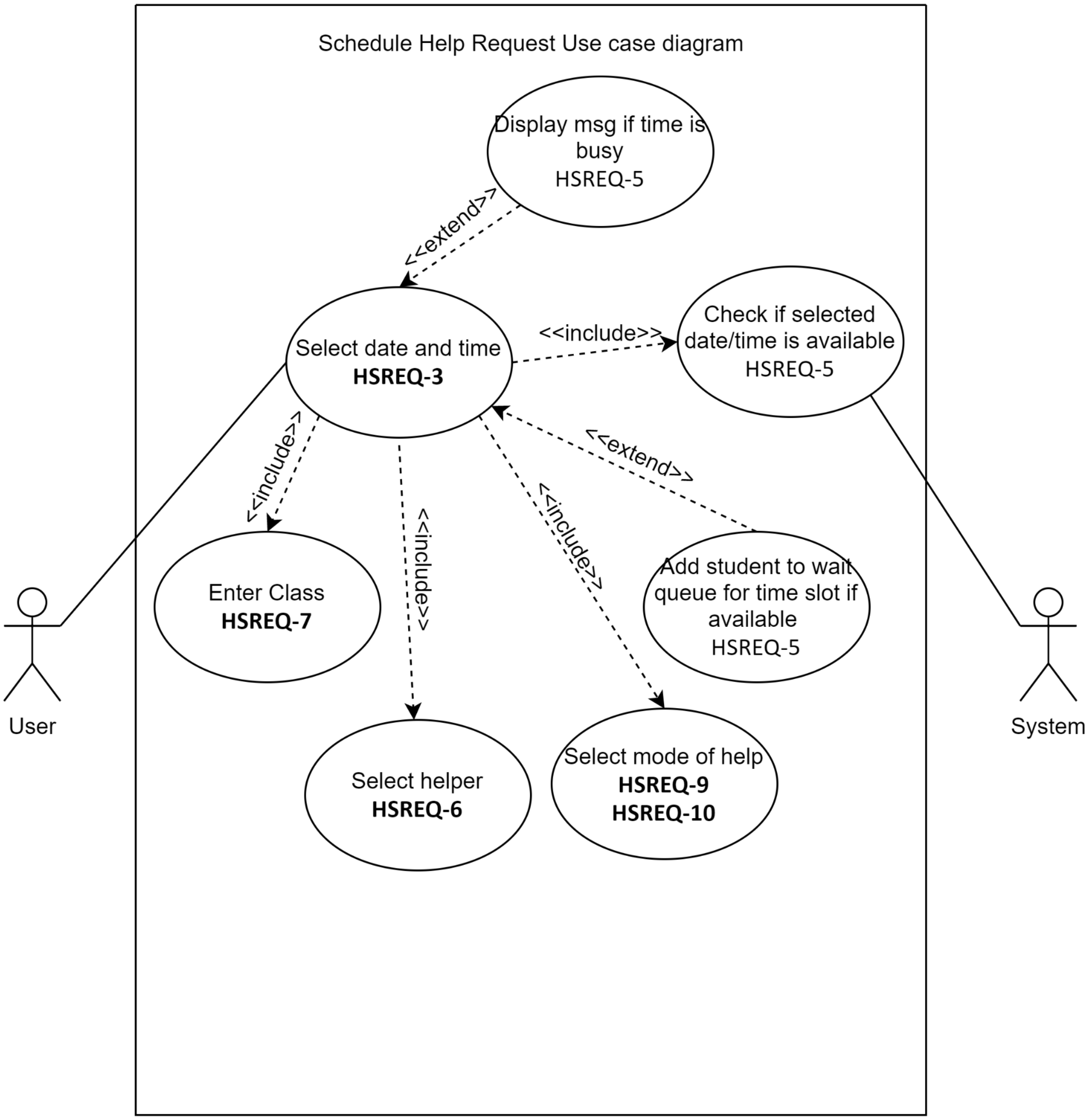
HSREQ-10: The system shall allow for solo or group help.

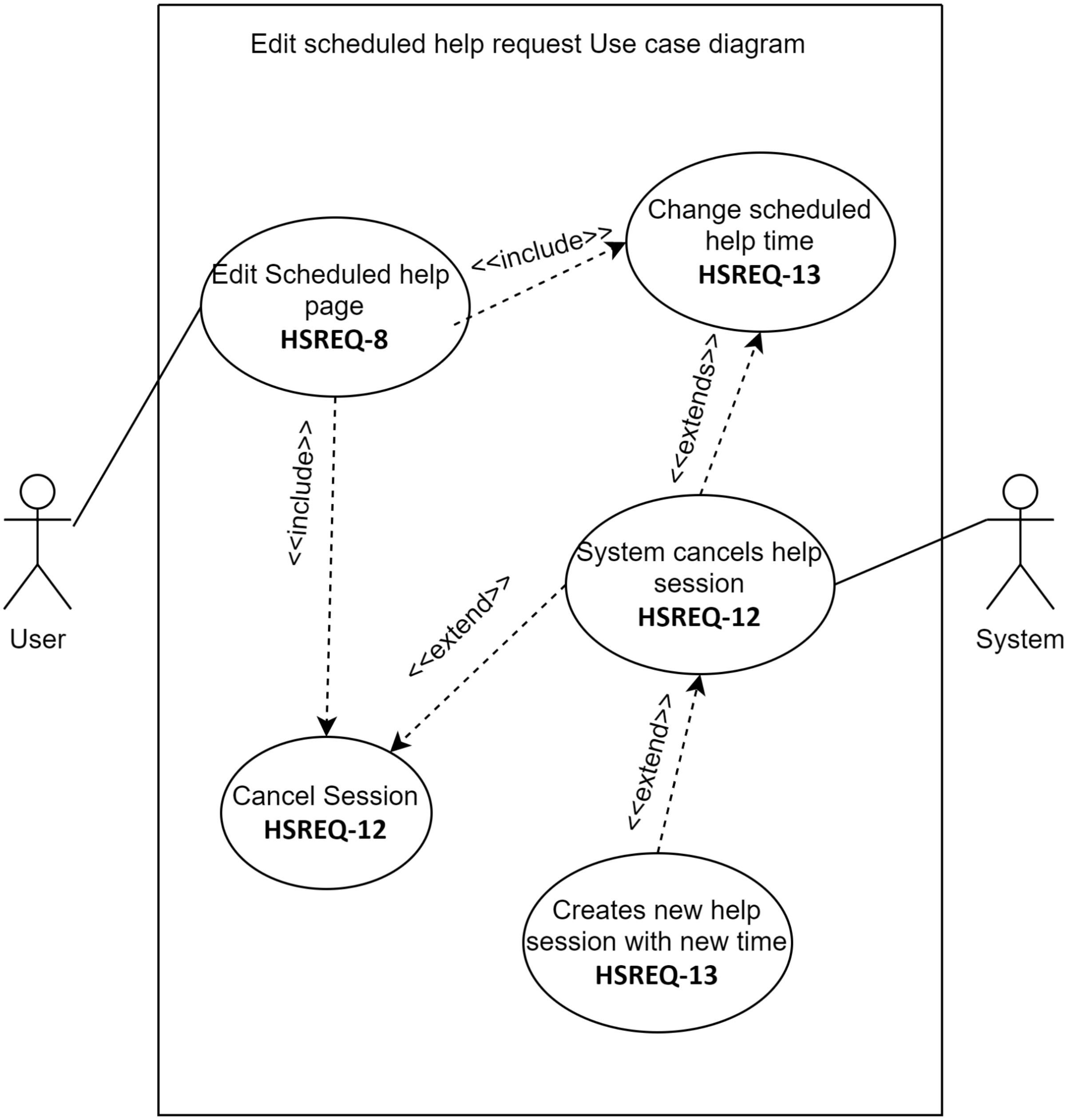
HSREQ-11: The system shall allow a limited number of students to join in a study group meeting

HSREQ-12: The system shall allow users to cancel a reservation.

HSREQ-13: The system shall allow users to move a reservation to a new valid time.

2.2.4 Use Case Diagrams


Schedule Help Request Use case diagram
Display msg if time is busy
HSREQ-5
<<extend>>
Select date and time
HSREQ-3
<<include>>
Check if selected date/time is available
HSREQ-5
<<extend>>
<<include>>
<<include>>
<<include>>
Enter Class
HSREQ-7
Add student to wait queue for time slot if available
HSREQ-5
User
System
Select helper
HSREQ-6
Select mode of help
HSREQ-9
HSREQ-10


Edit scheduled help request Use case diagram
Edit Scheduled help page HSREQ-8
<<include>>
Change scheduled help time HSREQ-13
<<extends>>
<<include>>
System cancels help session HSREQ-12
<<extend>>
<<extend>>
Cancel Session HSREQ-12
Creates new help session with new time HSREQ-13
User
System

2.3 Feedback

2.3.1 Description and Priority

The application will have opportunities for students to give feedback about their help sessions by accessing a history help session they have attended. These reviews will be accessible to the professors and TAs of any given class. This functionality is priority 4.

2.3.2 Stimulus/Response Sequence

Feedback for Help Session

Number	FBREQ-1
Name	Provide Feedback about Help Session
Summary	The system shall allow Students to give feedback on the helper who led a help meeting.
Priority	High
Preconditions	Student has attended a help session.
Postconditions	Feedback is saved and ready for review.
Primary Actors	Users
Secondary Actors	System
Triggers	Student selects 'Provide Feedback' for a help session in their history.
Main Scenario	1. System displays an embedded form within the website for the student to fill. 2. Student presses Submit. 3. System saves the response.
Extensions	1.1: Student enters name of help provider. 1.2: Student enters a short paragraph of what was done during the session. 1.3: Student enters the amount of time in minutes for how long the interaction lasted 1.4: Student enters an integer from 0-10 describing their satisfaction.
Open Issues	None

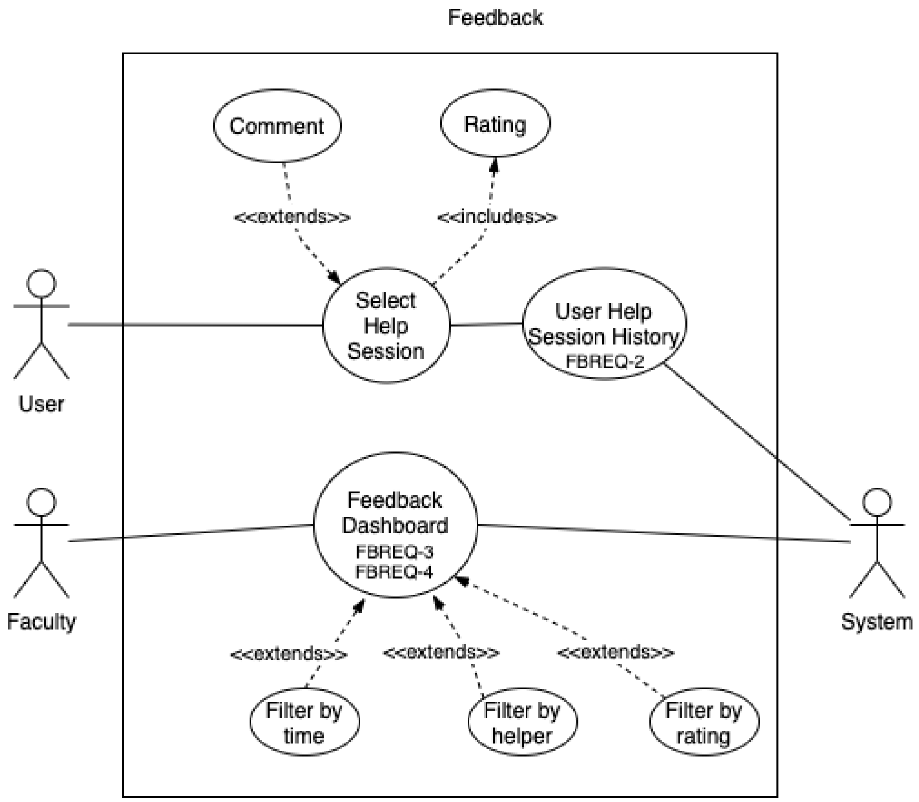
2.3.3 Functional Requirements

FBREQ-2: The system will display history of help sessions attended by the student.

FBREQ-3: The system shall allow for anonymous reviews of help sessions.

FBREQ-4: The system shall allow for anonymous reviews of helpers.

2.3.4 Use Case Diagram



2.4 Calendar

2.4.1 Description and Priority

The calendar view is a quick glance at who is in the lab at any given time and what they are most likely able to help with. This gives students and faculty vision on the help availability for any given day. The calendar is meant to be an interactive element with links to more information about a staff member. This functionality is priority 4.

2.4.2 Stimulus/Response Sequence

View Calendar

Number	CREQ-1
Name	Viewing the Calendar
Summary	The system shall allow a user to view a calendar of schedules and events
Priority	High
Preconditions	User is logged in
Postconditions	User has desired information about the calendar
Primary Actors	Users

Secondary Actors	System
Triggers	User clicks on the “Calendar” button
Main Scenario	1. The system displays a view of the calendar to the user 2. The user selects a time on the calendar 3. The user selects the week view 4. The user selects the month view
Extensions	2.1 The system displays the staff that are attending to the lab at that given time, their specialization, class they teach, and position 3.1 The system displays a summary for help availability for any given week 4.1 The system displays a summary for the help availability for any given month
Open Issues	None

2.4.3 Functional Requirements

CREQ-2: The system shall display a schedule of helpers and events.

CREQ-3: The system shall display the names of lab helpers that are available

CREQ-4: The system shall display the times that lab helpers are available.

CREQ-5: The system shall display the classes that the lab helper can assist with.

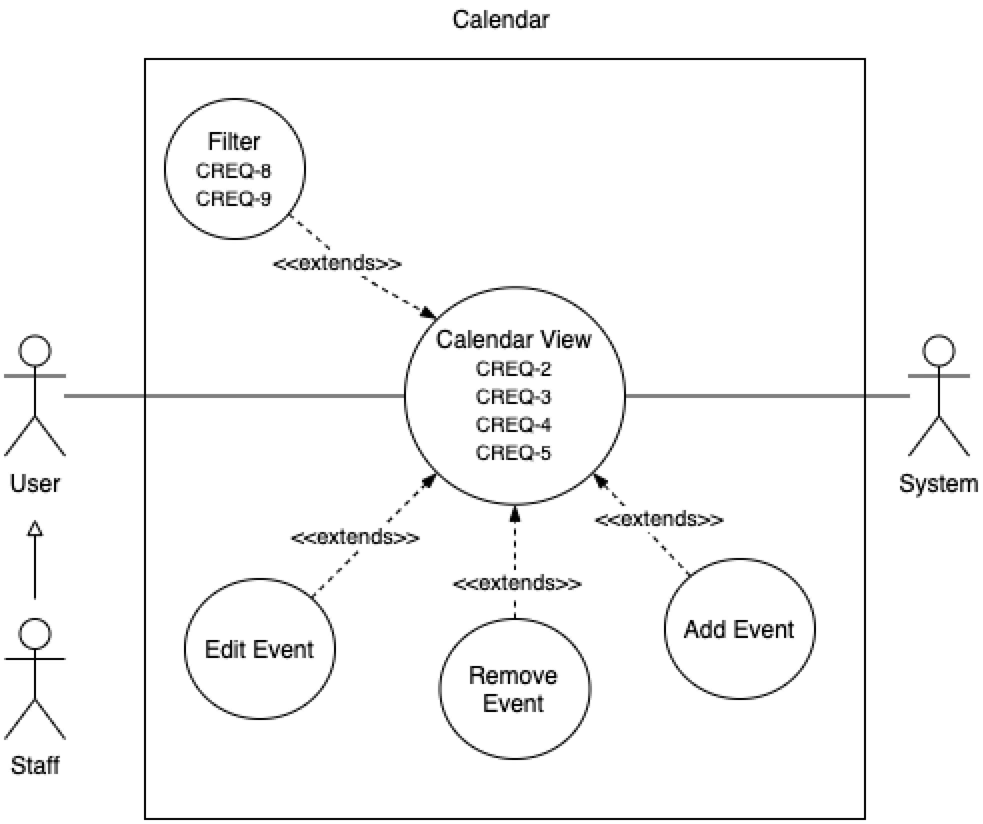
CREQ-6: The system shall display any scheduled group help sessions.

CREQ-7: The system shall display any scheduled refresher lectures.

CREQ-8: The system shall provide an option to view the calendar by week.

CREQ-9: The system shall provide an option to view the calendar by month.

2.4.4 Use Case Diagram



2.5 Database

2.5.1 Description and Priority

The database stores all pertinent information related to the calendar, scheduled help sessions, and user information. The database is also able to generate reports based on specified criteria for analytical purposes. This functionality is priority 5.

2.5.2 Stimulus/Response Sequence

Store Reservation Request

Number	DBREQ-1, 2, 3, 4
Name	Store Reservation Request
Summary	The system shall store user reservation requests.
Priority	High
Preconditions	New reservation has been created

Postconditions	Reservation is stored in a help time instance within the calendar
Primary Actors	User
Secondary Actors	System
Triggers	User submits reservation via the Schedule Help module
Main Scenario	1. The system adds an instance of a reservation to the desired help time object within the database 2. The system updates the calendar 3. The system stores the instance of the reservation to the user's profile. 4. The system displays the new number of users attending the designated help session
Extensions	1.1 The reservation is invalid (overlap). 1.2 The reservation is canceled and an error displayed.
Open Issues	

Edit Reservation

Number	DBREQ-5
Name	Edit Reservation
Summary	The system shall allow Administrators, Helpers, or Students to edit a previously established reservation.
Priority	High
Preconditions	An established reservation.
Postconditions	The reservation is changed according to user specifications.
Primary Actors	Administrator, Student
Secondary Actors	System
Triggers	User selects edit reservation option.
Main Scenario	1. A student selects the edit reservation option. 2. A list of all reservations made by the student is displayed. 3. The student selects a reservation.

	4. The student makes modifications to the reservation. 5. The reservation is replaced.
Extensions	1.1 An administrator selects the edit reservation option. 1.2 A list of all reservations is displayed.
Open Issues	

Store Help Session Data

Number	DBREQ-6, 7, 8, 9
Name	Store Help Session Data
Summary	The system shall store: name of student, the course that the student was helped with, name of staff member that helped the student, the topic and the duration of the session.
Priority	High
Preconditions	Information on help session has been submitted.
Postconditions	A new entry has been added to a table within the database.
Primary Actors	Administrator, Helper
Secondary Actors	System
Triggers	Data on a help session is submitted to the database.
Main Scenario	1. Information on a help session is entered by an administrator, helper, or recorded by the system. 2. A database entry is created recording the help session for future reports.
Extensions	
Open Issues	None

Display Reservation

Number	DBREQ-10, 11, 12, 13
Name	Display Reservation
Summary	The system shall send notifications and viewing options associated with help reservations to applicable

	users.
Priority	High
Preconditions	A valid help reservation has been submitted and processed
Postconditions	The reservation information is displayed to all applicable parties
Primary Actors	Users
Secondary Actors	System
Triggers	User submits reservation via the Schedule Help module
Main Scenario	1. The system adds an instance of a reservation 2. The system updates the calendar 3. The system stores the instance of the reservation to the user's profile. 4. The system displays the new number of users attending the designated help session
Extensions	1.1 The user enters an invalid meeting request 1.2 The system displays a warning to the user, asking them to change their entry.
Open Issues	None

Search Reservations

Number	DBREQ-14, 15, 16, 17, 18
Name	Search Reservation
Summary	The system shall display reservation instances based on search criteria such as: course, topic, time window, helper
Priority	High
Preconditions	Reservations table must be populated inside the Database.
Postconditions	A list of reservations is returned.
Primary Actors	Administrator
Secondary Actors	System
Triggers	One or multiple criteria have been selected, and the

	Search button has been pressed
Main Scenario	1. Administrator selects one or multiple search criterias. 2. Administrator enters text for each search criteria selected. 3. System retrieves all reservations that match the text from each of the selected criteria. 4. System displays the retrieved reservations.
Extensions	
Open Issues	None

Generate Report

Number	DBREQ-19, 20, 21, 22, 23
Name	Generate Report
Summary	The system shall allow the user to generate a variety of reports to analyze the data gathered.
Priority	High
Preconditions	A populated database
Postconditions	A report has been generated based on specifications provided.
Primary Actors	Administrator
Secondary Actors	
Triggers	The generate report option is selected.
Main Scenario	1. Administrator selects generate report 2. Administrator selects report mode (by course, by topic, by window of time, or by helper). 3. System retrieves all data that matches selected criteria. 4. System generates a formatted report of information for analytical use by administrators.
Extensions	
Open Issues	

2.5.3 Functional Requirements

Store Reservation Request

DBREQ-1: The system shall store user reservation requests.
DBREQ-2: The system shall manage reservation requests
DBREQ-3: The system shall check reservation requests for conflict.
DBREQ-4: The system shall store valid reservations.

Edit Reservation

DBREQ-5: The system shall allow for edits to reservations already in the database.
Store Help Session Data
DBREQ-6: The system shall store walk-ins data entered by administrators.
DBREQ-7: The system shall store the course the student was helped with.
DBREQ-8: The system shall store the topic the student was helped with.
DBREQ-9: The system shall store the duration of each help session.

Display Reservations

DBREQ-10: The system shall send notifications and viewing options associated with help reservations to applicable users.
DBREQ-11: The system shall display reservations**.**
DBREQ-12: The system shall display all reservations on a schedule to administrators.
DBREQ-13: The system shall display a student's own reservations to them.

Search Reservations

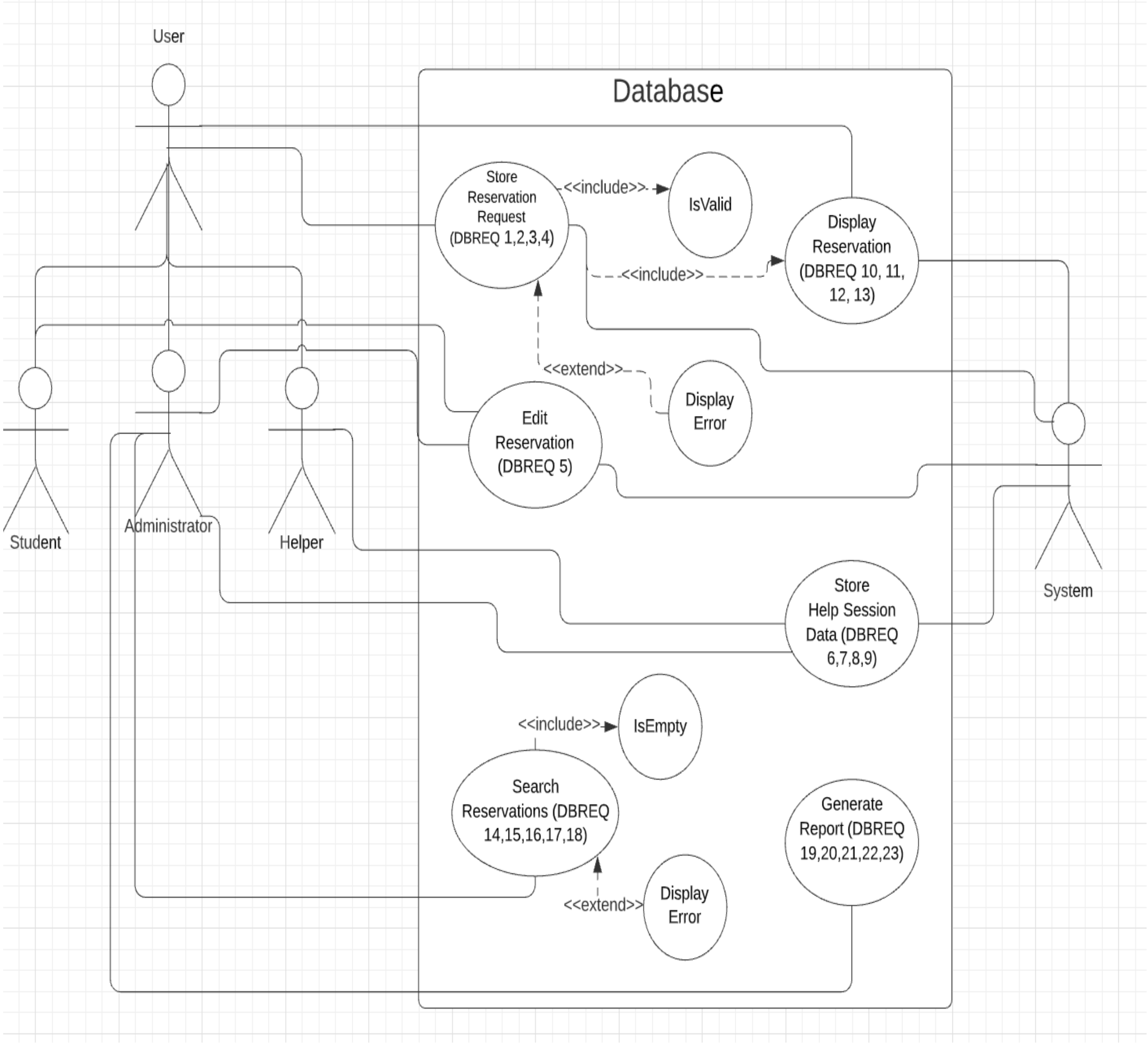
DBREQ-14: The system shall allow administrators to search reservations with criteria.
DBREQ-15: The system shall allow for searches by course.
DBREQ-16: The system shall allow for searches by topic.
DBREQ-17: The system shall allow for searches by window of time.
DBREQ-18: The system shall allow for searches by helper.

Generate Report

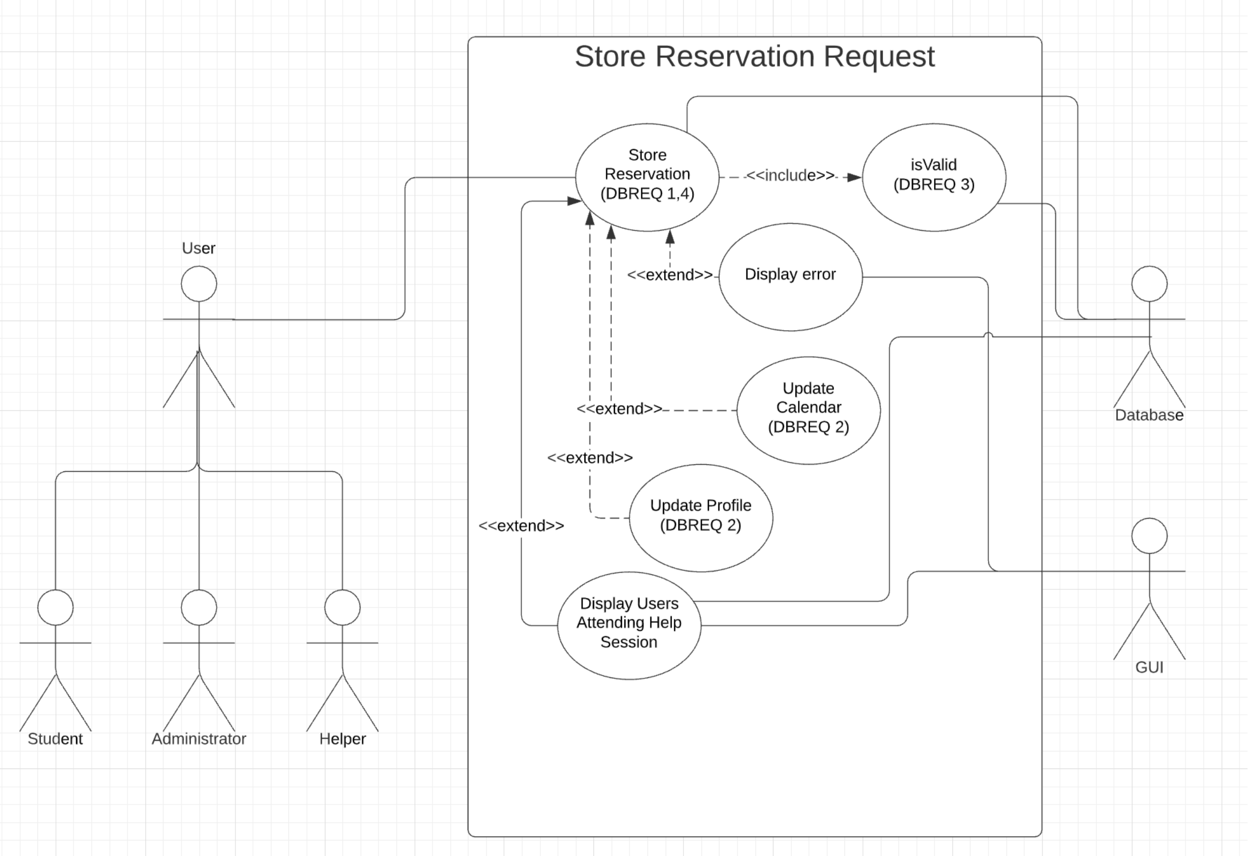
DBREQ-19: The system shall generate reports based on data from a specified window.
DBREQ-20: The system shall generate frequency reports based on course.
DBREQ-21: The system shall generate frequency reports based on topic.
DBREQ-22: The system shall generate frequency reports based on windows of time.
DBREQ-23: The system shall generate frequency reports based on helper.

2.5.4 Use Case Diagram

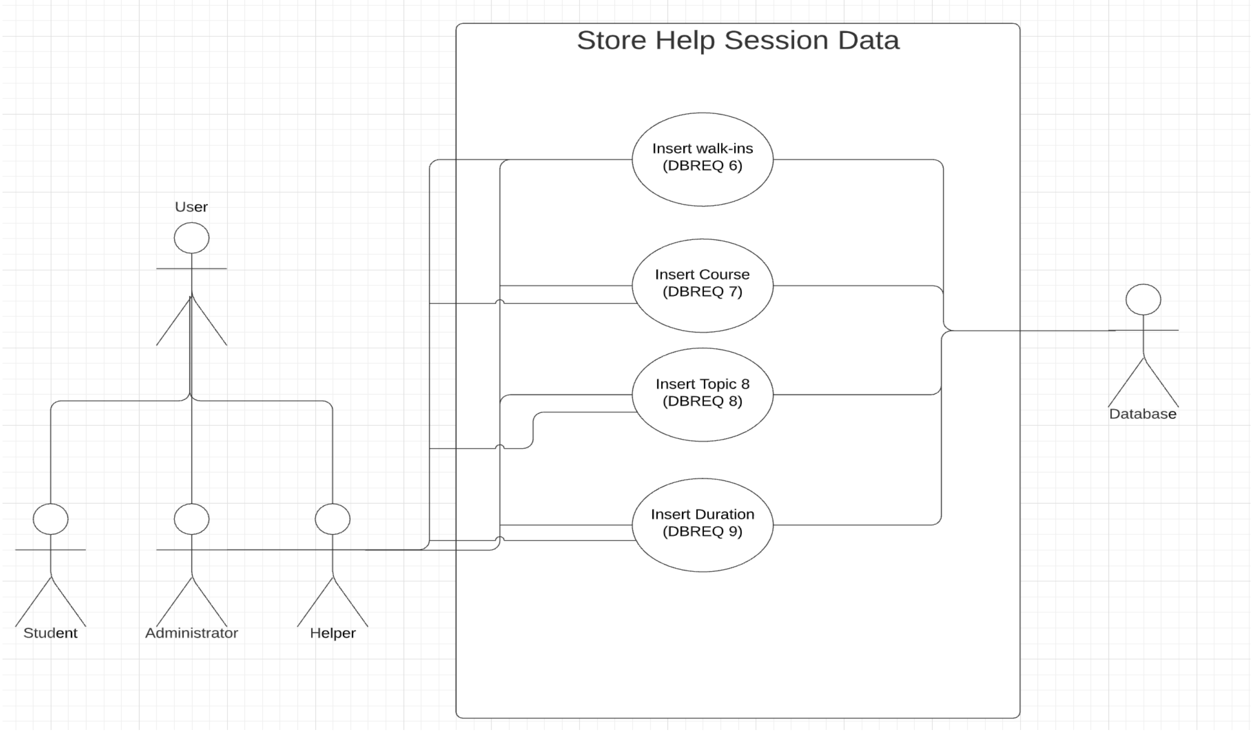
2.5.4.1 Top Level View


User
Database
Store Reservation Request (DBREQ 1,2,3,4)
<<include>>
IsValid
<<include>>
Display Reservation (DBREQ 10, 11, 12, 13)
<<extend>>
Display Error
Edit Reservation (DBREQ 5)
Student
Administrator
Helper
System
Store Help Session Data (DBREQ 6,7,8,9)
<<include>>
IsEmpty
Search Reservations (DBREQ 14,15,16,17,18)
Generate Report (DBREQ 19,20,21,22,23)
<<extend>>
Display Error

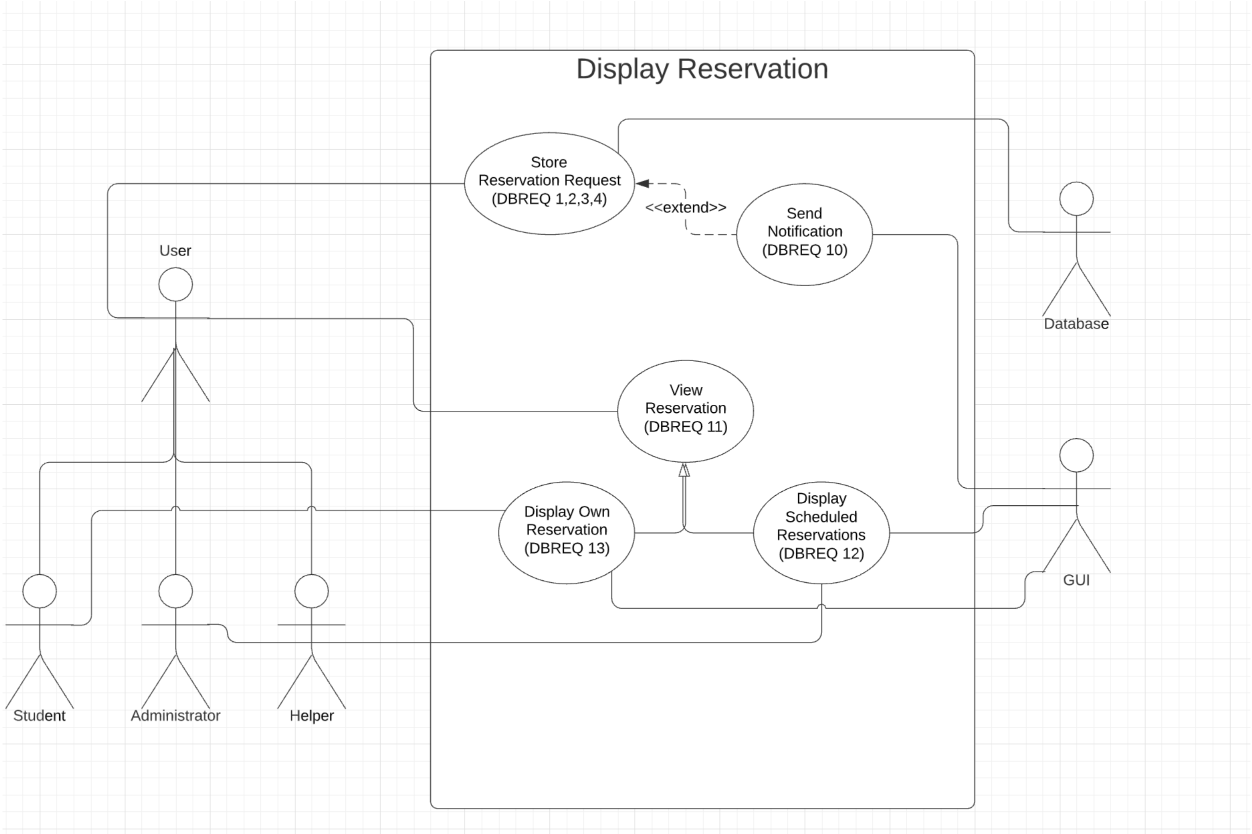
2.5.4.2 Store Reservation Request



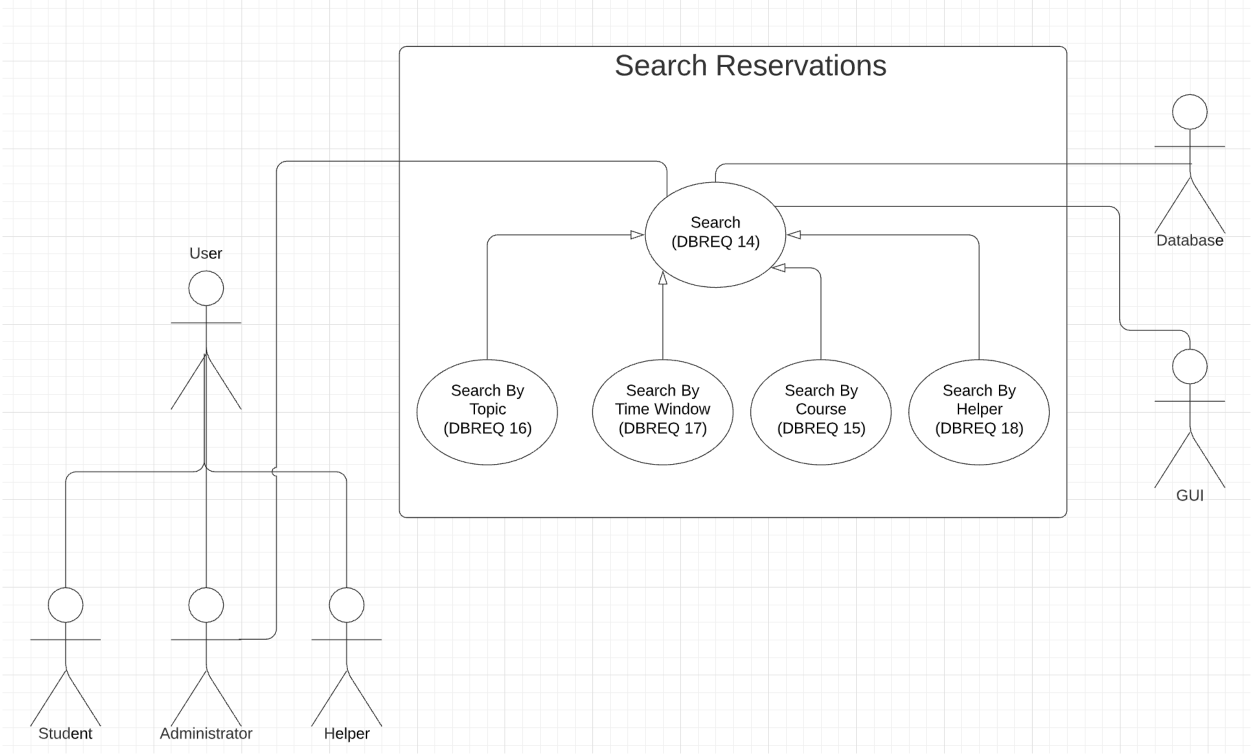
2.5.4.3 Store Help Session Data



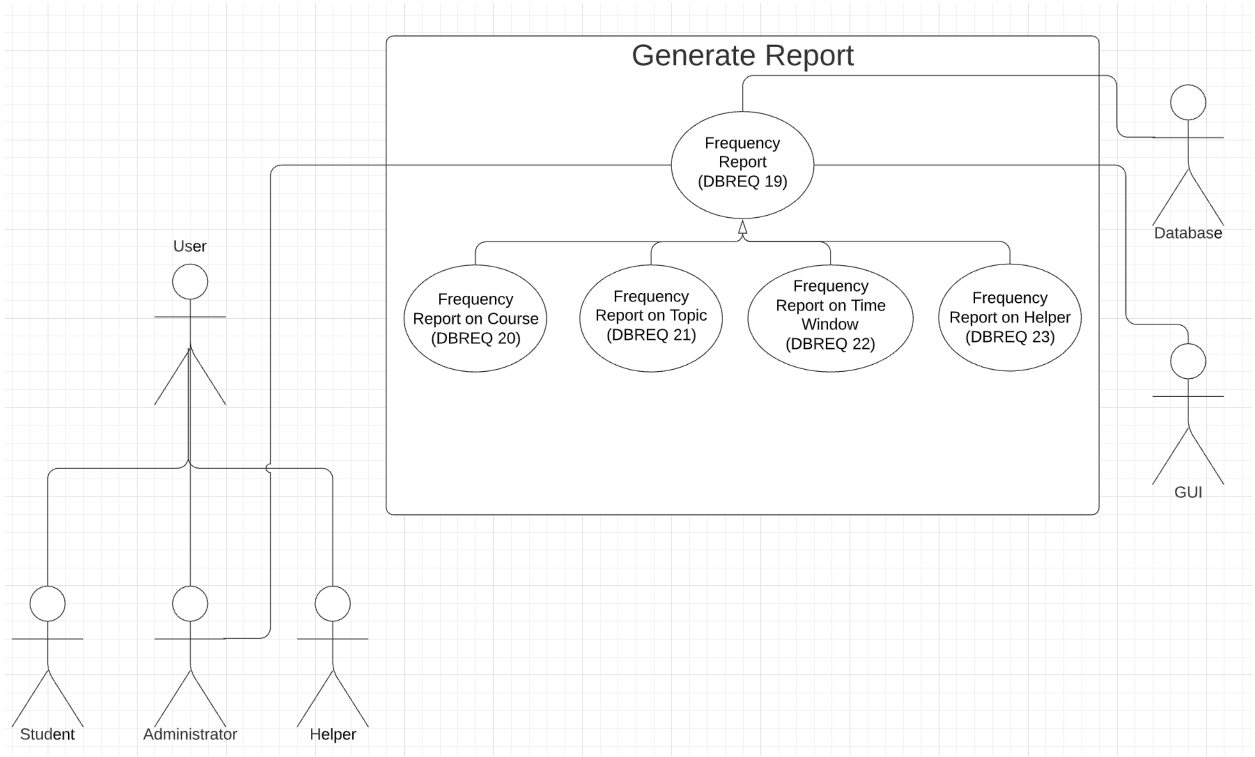
2.5.4.4 Display Reservation


Display Reservation
Store Reservation Request (DBREQ 1,2,3,4)
<<extend>>
Send Notification (DBREQ 10)
View Reservation (DBREQ 11)
Display Own Reservation (DBREQ 13)
Display Scheduled Reservations (DBREQ 12)
User
Student
Administrator
Helper
Database
GUI

2.5.4.5 Search Reservations


Search Reservations
Search (DBREQ 14)
Search By Topic (DBREQ 16)
Search By Time Window (DBREQ 17)
Search By Course (DBREQ 15)
Search By Helper (DBREQ 18)
User
Student
Administrator
Helper
Database
GUI

2.5.4.6 Generate Report


Generate Report
Frequency Report (DBREQ 19)
Frequency Report on Course (DBREQ 20)
Frequency Report on Topic (DBREQ 21)
Frequency Report on Time Window (DBREQ 22)
Frequency Report on Helper (DBREQ 23)
User
Database
GUI
Student
Administrator
Helper

2.6 Stretch Goals

The functional requirements listed in this section are identified as ideas for potential development if time permits, but are not necessarily required for the client's specifications. If the previously listed requirements are met before the final deadline, this section will be filled out and development will begin on these requirements. These functionalities are priority 1.

2.6.1 News / Homepage

2.6.1.1 Description and Priority

TBD

2.6.1.2 Stimulus/Response Sequence

USE CASE

TBD

2.6.1.3 Functional Requirements

NHREQ-1: The system shall allow the user to view recent news.
NHREQ-2: The system shall display a list of recent computer science updates.
NHREQ-3: The system shall allow helpers to make new posts to the news and updates page.

NHREQ-4: The system shall filter recent news and updates by class.

2.6.2 Wait Queue

2.6.2.1 Description and Priority

TBD

2.6.2.2 Stimulus/Response Sequence

USE CASE

TBD

2.6.2.3 Functional Requirements

WQREQ-1: The system shall allow the user to view a wait queue of students waiting to receive help.
WQREQ-2: The system shall display a number of students currently receiving help.
WQREQ-3: The system shall display the current average wait time of being helped.

2.6.3 Forum

2.6.3.1 Description and Priority

TBD

2.6.3.2 Stimulus/Response Sequence

TBD

2.6.3.3 Functional Requirements

FREQ-1: The system shall display a forum of student asked questions.
FREQ-2: The system shall allow students to post questions to the forum page.
FREQ-3: The system shall allow students to post replies to forum questions
FREQ-4: The system shall allow Helpers to post replies to forum questions
FREQ-5: The system shall allow students to filter the forum by class tag

3. Non-Functional Requirements

Included in this section are the non-functional requirements intended for the Boardman Computer Science Lab Web Portal including NFRs specific to product requirements, organizational requirements, and external requirements. Each NFR is organized by use of a unique identifier, priority scale (1 = lowest, 5 = highest), a brief description of the NFR, and the thest(s) associated with the NFR.

3.1. Non-Functional Requirements Expanded

NFR1: The system shall not be down for more than 5 mins at a time between the hours of 6am and 10pm EST 96% of the time.
Priority: 4
Description: Major application maintenance is only expected to occur between the hours of 10pm and 6am EST, so server downtime is required to be less than 5 minutes outside this window.
Test(s): Test every 5 minutes for server availability for 24 hours, if more than one consecutive ping fails, test fails.

NFR2: The system shall be able to handle at least 500 requests per second 95% of the time.
Priority: 5
Description: At any given point in time, the server will not experience overload for up to 500 requests in a second.
Test(s): Make 500 requests in a second, if the system is able to handle these requests without crashing, test success.

NFR3: The system shall be able to process a request within 10 seconds 99% of the time.
Priority: 4
Description: At any given point in time, the system will receive and execute user requests within 10 seconds.
Test(s): Time the process time in milliseconds, a value over 10000 is failure.

NFR4: The system shall be able to load pages within 2 seconds when ping < 1000ms 96% of the time.
Priority: 3
Description: At any given point in time, new pages will load within 2 seconds upon user request if their ping is less than 1000ms.
Test(s): Test on a connection with a ping of less than 1000ms and time the page load in milliseconds. A value of over 2000 fails the test.

NFR5: The system shall allow no more than three errors per second 99% of the time.
Priority: 3
Description: At any given point in time, the system will experience a maximum of 3 errors in a second.
Test(s): Test all functions repeatedly for an hour. If the number of errors in the hour is greater than 10,000, test fails.

NFR6: The system shall not share account information with other users.
Priority: 5
Description: Login information and user data will not be shared with users other than the associated user, likewise, user data will not be used for any other purpose than internal analysis.
Test(s): Attempt to access another user’s account without the correct password, attempt to access another user’s reservations without the correct password.

NFR7: The system shall account for a list of popular resolutions (1920x1080, 1366x768, 1280x720, 360x640, 414x896, 1536x864, 375x667).
Priority: 2
Description: The website will fit accurately within most common screen size resolutions.
Test(s): Attempt to display the website on each of the listed resolutions. If there are graphical distortions or errors, the test fails.

NFR8: The system shall use 16 px font size for the body text.
Priority: 1
Description: All body text in the website will use size 16 font for purposes of uniformity.
Test(s): Search for font size declarations in document, each body text declaration should point to 16pt font.

NFR9: The system shall be able to store reservations for any point from 1970-2099.
Priority: 3
Description: When a student or helper reserves a meeting time, dates and time will be available for selection from years 1970 - 2099 so that updating will likely be unnecessary in perpetuity.
Test(s): Make a reservation for each year from 1970-2099. If any years are missing from the reservation table, the test fails.

NFR10: The system shall be compatible with mobile and web based devices.
Priority: 2
Description: The website will be usable for both desktop (latest ubuntu, Windows 10, and MACOS) and mobile devices (latest android, latest IOS).
Test(s): Test the system on each of the listed devices. If there are any graphical distortions or errors, the test fails.

NFR11: The system shall be able to run on Google Chrome, Firefox, Microsoft Edge, and safari.
Priority: 2
Description: The website will be usable for all common web browsers.
Test(s): Test the system on latest Chrome, Firefox, Edge, and Safari browsers. If there are any graphical distortions or errors, the test fails.

NFR12: The system shall be scalable, to accommodate for growth in the number of users (up to 1 million users).
Priority: 5
Description: The website will be able to accommodate any number of new users.
Test(s): Attempt to add 1 million users to the database system. If there are 1 million users in the database, the test succeeds.

NFR13: The system shall be able to file a new meeting time in less than 3 minutes 98% of the time.
Priority: 4
Description: When a request for a new meeting is received, the request will be processed within 3 minutes.

Test(s): Attempt to file a new meeting, time the response. If the response occurs in less than 3 minutes, the test succeeds.

NFR14: The system shall be able to file a change in a meeting in less than 3 minutes 98% of the time.
Priority: 4
Description: When a request for a meeting change is received, the request will be processed within 3 minutes.
Test(s): Attempt to file a change to a meeting, time the response. If the response occurs in less than 3 minutes, the test succeeds.

NFR15: The system shall be able to update the calendar in less than 3 minutes 98% of the time.
Priority: 4
Description: When a request to update the calendar is received, the request will be processed within 3 minutes.
Test(s): Attempt to update the calendar, time the response. If the response occurs in less than 3 minutes, the test succeeds.

NFR16: The system shall be able to update a forum request in less than 3 minutes 98% of the time.
Priority: 4
Description: When a request to post to the forum is received, the request will be processed within 3 minutes.
Test(s): Attempt to update a forum request, time the response. If the response occurs in less than 3 minutes, the test succeeds.

NFR17: The system shall be able to upload a feedback request in less than 3 minutes 98% of the time.
Priority: 4
Description: When a request to upload a feedback form is received, the request will be processed within 3 minutes.
Test(s): Attempt to upload a feedback form, time the response. If the response occurs in less than 3 minutes, the test succeeds.

NFR18: The system shall be able to update the walkin-wait times in less than 3 minutes 98% of the time.
Priority: 4
Description: When a request to update the wait times for help is received, the request will be processed within 3 minutes.
Test(s): Attempt to update walkin wait times, time the response. If the response occurs in less than 3 minutes, the test succeeds.

NFR19: The system shall be able to report wait time data within 3 minutes of a new entry 98% of the time.
Priority: 4
Description: When there is a change in wait time, that data will be saved for analysis within 3 minutes
Test(s): Attempt to update walkin wait times, time the response for the walkin wait update. If the response occurs in less than 3 minutes, the test succeeds.

NFR20: The system shall be able to report help request data within 3 minutes of a new entry 98% of the time.
Priority: 4
Description: When there is a request for help, the subject matter of request and time of request will be saved for analysis within 3 minutes
Test(s):
Test the time required that it takes for the request to be saved after it has been submitted. If less than 3 minutes, then the test succeeded.

4. User Interface

This section describes and explains the user interface intended for this project, or a reference to exterior documentation that better describes the user interface.

4.1. User Interface Expanded

See "User Interface Design Document" for the Boardman Computer Science Lab Web Portal.

5. Deliverables

This section includes a list of all deliverable items for this project. Additionally, it includes the tentative date of completion and the format that it will be submitted in.

5.1. Deliverables Expanded

Hard copies of each of the following:

- Systems Requirement Specification
  Date Expected: 10/25
- System Design Document
  Date Expected: 11/10
- User Interface Design Document
  Date Expected: 11/29
- User Manual
  Date Expected: Second Semester
- Administrator Manual
  Date Expected: Second Semester
- Copies of all Biweekly Status Reports
  Date Expected: Second Semester

An electronic file containing the following:

- Systems Requirement Specification
  Date Expected: 10/25
- System Design Document
  Date Expected: 11/10
- User Interface Design Document

Date Expected: 11/29

- User Manual

  Date Expected: Second Semester
- Administrator Manual

  Date Expected: Second Semester
- All source code

  Date Expected: Second Semester
- The executable program

  Date Expected: Second Semester
- Any other software required for installation and execution of the delivered program

  Date Expected: Second Semester

6. Open Issues

This section displays Issues that have been raised and do not yet have a conclusion. These issues will be addressed later in the development process.

6.1. Open Issues Expanded

TBD - note: Open Issues will be added as they are created

Appendix A – Agreement Between Customer and Contractor

This section denotes that both the client and the development team have agreed upon the information contained within this document. It will be used as both a guideline and as an end goal in terms of the requirements needed for the application to function to the clients vision.

In the case that an addition or edit be needed after the completion and signing of this document, the change or addition must be agreed upon by both client and development team and included in **Appendix D - Document Additions** with the title of the addition, date, brief description, and signature from both parties.

-Client-

Name: Mr. Christopher Dufour

Date: October 26, 2021

Signature: Christopher G Dufour

Christopher G Dufour (Oct 26, 2021 15:07 EDT)

-Development Team-

Name: Klei Bendo

Date: October 26, 2021

Signature:

Kleition Bendo (Oct 26, 2021 16:14 EDT)

Name: Jack Brisson

Date: October 26, 2021

Signature: John H Brisson

John H Brisson (Oct 26, 2021 21:01 EDT)

Name: Alex Landry

Date: October 26, 2021

Signature: Alex Landry

Alex Landry (Oct 26, 2021 20:36 EDT)

Name: Samuel Morse

Date: October 26, 2021

Signature:

Samuel Morse (Oct 26, 2021 20:53 EDT)

Name: Aaron Schanck

Date: October 26, 2021

Signature: Schanck

aaron schanck (Oct 26, 2021 21:06 EDT)

Name: Forrest Swift

Date: October 26, 2021

Signature: Forrest Swift

Forrest Swift (Oct 26, 2021 20:58 EDT)

Client Comments (Continues on next page if needed):

Client Comments Cont.

Appendix B – Team Review Sign-off

This section denotes that all members of the In-House Operations development team have reviewed this document and agree on its content and format. If any minor disagreements in content and format are present, they are listed below the development team signatures.

Name: Klei Bendo
Date: October 26, 2021
Signature:
Kleition Bendo (Oct 26, 2021 16:14 EDT)

Name: Jack Brisson
Date: October 26, 2021
Signature: John H Brisson
John H Brisson (Oct 26, 2021 21:01 EDT)

Name: Alex Landry
Date: October 26, 2021
Signature: Alex Landry
Alex Landry (Oct 26, 2021 20:36 EDT)

Name: Samuel Morse
Date: October 26, 2021
Signature:
Samuel Morse (Oct 26, 2021 20:53 EDT)

Name: Aaron Schanck
Date: October 26, 2021
Signature: Schanck
aaron schanck (Oct 26, 2021 21:06 EDT)

Name: Forrest Swift
Date: October 26, 2021
Signature: Forrest Swift
Forrest Swift (Oct 26, 2021 20:58 EDT)

Minor Disagreements in Content and Format (if any):

Appendix C – Document Contributions

This section denotes the contributions of each team member to this document. It includes the sections each member worked on and their percentage contributed in parentheses.

Name: Klei Bendo
Sections worked on (percentage contributed):
Section 2 - Functional Requirements (15%) (Database section, use-cases and diagram)
Section 3 - Non-Functional Requirements (20%) (requirement and test writing)

Name: Jack Brisson
Sections worked on (percentage contributed):
Section 2 - Functional Requirements (15%) (Sign-In/Sign-Out section, use-cases and diagram)
Section 3 - Non-Functional Requirements (20%) (requirement and test writing)

Name: Alex Landry
Sections worked on (percentage contributed):
Section 2 - Functional Requirements (15%) (Help Scheduling section, use-cases and diagram)
Section 3 - Non-Functional Requirements (20%) (requirement and test writing)

Name: Samuel Morse
Sections worked on (percentage contributed):
Section 2 - Functional Requirements (35%) (calendar and feedback sections, use-cases and diagram, and most of the Stimulus/Response diagrams)
Section 3 - Non-Functional Requirements (5%) (requirement and test writing)

Name: Aaron Schanck
Sections worked on (percentage contributed):
Section 1 - Introduction (100%)
Section 2 - Functional Requirements (5%) (Introduction and layout)
Section 3 - Non-Functional Requirements (5%) (Introduction and layout)
Section 4 - User Interface (100%)
Section 5 - Deliverables (100%)
Section 6 - Open Issues (100%)
Appendices (100%)

Name: Forrest Swift
Sections worked on (percentage contributed):
Section 2 - Functional Requirements (15%) (List of Functional Requirements, Database section use-cases)
Section 3 - Non-Functional Requirements (20%) (requirement and test writing)

Appendix D – Document Additions

No Document additions to date.