Software Requirements Specification

for

Bearly Free Scheduler

**Version 1.0 approved**

**Prepared by Nick Chambers, Henry Gustafson, Matt Roselli**

**Thundercat Software Dev Team**

**November 2, 2016**

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Product Scope

1.5 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3. External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4. System Features**

4.1 System Feature 1

4.2 System Feature 2 (and so on)

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

5.5 Business Rules

**6. Other Requirements**

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of The Bearly Free Scheduler is to provide an easy to use environment for students and staff of the writing help room to schedule appointments and replace the current pen-and-paper system.

## Document Conventions

No Special conventions used.

## Intended Audience and Reading Suggestions

This SRS is made to be read by the investors who decide our fate, Professor Schilling, developers of the project, writing room staff if interested, and documentation writers. This SRS will be a rundown of the project, how it will work, who will be using it, who will be implementing it, how it will be implemented, and things we hope to accomplish in the given time frame. I suggest reading the overview and high-level descriptions of the project first, then browsing the UML diagrams before turning to the low-level descriptions. This will provide the deepest and broadest understanding of the project in the least amount of reading.

## Product Scope

The scheduler will be replacing the current system of scheduling appointments that the writing help room uses. Currently, they are using pen and paper first come first serve appointments to aid students in paper revisions. They also require the paper to be physically in the room by way of digital copy or printed copy. This can be done by the student bringing it with them or emailing it directly to the aide helping them if they know who they will be helped by. We wish to improve this by requiring the file to be submitted when the appointment is made so that the aide has time in advance to proofread. Not only will this simplify and streamline the process of appointment making and getting help, but speed up the help itself by having aides prepared in advance for appointments. The extra preparation will allow for better revisions as well; thus allowing both parties to function on a higher level.

## References

Nothing to see here. No references to outside sources were used.

# Overall Description

## Product Perspective

Our scheduler will be a replacement of the current help room scheduler. Using the UC database of login information, we plan to require students to log in to make appointments, then have a whitelist to determine if that student is of the help room staff or another student. This will decide if they see the staff or student interface. We would like to make the scheduler a part of Canvas if possible, or the UC site itself if it is not.

## Product Functions

Students may create appointments and submit drafts of their papers with them. The drafts will be mandatory to create an appointment. Students may change certain aspects of the appointments after submitting them such as their preferred fellow, the time of their appointment, and the length of their appointment. All of these must be approved by the staff member before being finalized. Staff members may edit currently scheduled appointments, and must approve or accept requests for appointments made by students. The staff will then determine the length of time needed to meet with the student once they have an understanding of the student’s paper. If possible, we would like email notifications to alert students when their pending appointment requests have been updated.

## User Classes and Characteristics

Users need only to be able to read and work a computer to use the program. Assuming that they are able to navigate to Canvas and sign in, they should be able to figure out how the program functions. Staff may need a short introduction but need only to know how to operate a computer as well, the help button will explain anything not immediately made clear.

## Operating Environment

We would like to make the program a part of Canvas to allow for cross platform use. It will work on any device able to use canvas, including phones, computers, and any device with a browser.

## Design and Implementation Constraints (Optional)

There are a few things we would like to implement into our design that we may not have time for. Time will be our biggest constraint. Researching how to get our program on Canvas will be time consuming and possibly not viable. Email notifications are also something we would like to implement but may not have time for.

## User Documentation

We will be implementing a small help button in the GUI of the program that will come up with a short user’s guide on how to perform the functions of the program.

## Assumptions and Dependencies

Our assumed factors include the ability to use the login database of UC students and to be able to get our project onto Canvas for all UC students to use. We also assume we will finish our project in the given time frame.

# External Interface Requirements

## User Interfaces

There will be two major interfaces for the software. Staff will see one interface with options to approve, deny, edit or cancel appointment requests submitted by students, students will see which writing fellows are available and when. They will then be able to submit a request for an appointment to be approved or denied by the staff members. All appointments will include a typed draft or pictures of the paper being proofread. The design will follow the colors of Ursinus and Canvas, using similar design styles to match the platform. We hope to add it seamlessly so that it will look familiar without having ever used it.

Administrator teachers will have two extra options that staff do not, to change the whitelist and edit the open appointments list. They can change open hours of the room to match student’s schedules and add or remove staff from the writing room whitelist. Other than that, they will have the same abilities of other staff.

## Hardware Interfaces (Optional)

There will be no need for external hardware other than the computer that the program is running on. No other physical hardware will be supported.

## Software Interfaces

Using Canvas, we hope to use all operating systems that support Canvas. The databases we will use would be the login database for UC students.

Input to the system will be given by students and give feedback in the form of removing an open time slot from a list. This will change the list for all students who see it, and staff will see that the slot has been filled. We hope to provide email feedback to confirm appointments have been made or cancelled.

## Communications Interfaces

Email notifications would be ideal for our project. The web browser running Canvas is essential, and would require network connection to update the page for all users.

# System Features

## System Feature 1

Approve Appointment (REQ-1)

4.1.1 Description and Priority

This feature allows the writing fellow to accept the appointment request.

Importance: 10

4.1.2 Stimulus/Response Sequences

In order to arrive at this feature a fellow will receive an appointment request and have the ability to accept or decline it, send a notification back to student

4.1.3 Functional Requirements

The system shall allow a writing staff member to approve appointments that meet certain criteria and follow the guidelines set by the staff members.

Decline Appointment (REQ-2)

4.2.1 Description and Priority

This feature allows the writing fellow to decline the appointment request.

Importance: 10

4.2.2 Stimulus/Response Sequences

In order to arrive at this feature a fellow will receive an appointment request and have the ability to accept or decline it, send a notification back to student

4.2.3 Functional Requirements

The system shall allow staff to decline appointments that do not meet their criteria or are not able to be put on the schedule.

Show Available Times of Fellow (REQ-3)

4.3.1 Description and Priority

This feature allows the student to view available times for different fellows.

Importance: 9

4.3.2 Stimulus/Response Sequences

In order to arrive at this feature student will follow option to make an appointment and this will be one of the next options to view/select.

4.3.3 Functional Requirements

The system shall display times that the staff member is free to meet. This will vary as lengths of approved appointments vary.

Ability to choose a specific fellow (REQ-4)

4.4.1 Description and Priority

This feature allows the student to view available times for different fellows.

Importance: 7

Advantage: Students may prefer to choose a specific fellow

Risk: Limits the open time for a student to meet with a fellow by only having access to availability of one fellow

4.4.2 Stimulus/Response Sequences

While going through the specifics of making an appointment the student will be given an option to choose a specific fellow

4.4.3 Functional Requirements

The system shall have an option to meet with a specific staff member of the writing room. This will show the times that said staff member is available and remove other staff member’s schedules from view of the user.

Ability to add draft (REQ-5)

4.5.1 Description and Priority

This feature allows the student to add a draft to send to a requested fellow to review before meeting as to save time and have a paper to go over upon face to face fellow session

Importance: 8

Advantage: Saves both fellow and student time because there is no wasted time during session if there is already something to go over

Risk: Students draft may not be long enough for review

4.5.2 Stimulus/Response Sequences

While going through the specifics of making an appointment the student be asked by default to add a draft via attachment.

4.5.3 Functional Requirements

The system shall force a user to submit a draft of their paper with the scheduling of an appointment. This ensures the staff member helping them will be prepared beforehand to speed up and strengthen the editing process.

Ability to change chosen fellow (REQ-6)

4.6.1 Description and Priority

This feature allows the student to view available times for different fellows.

Importance: 6

Advantage: Students may prefer to change fellows at some time

Risk: Other chosen fellow may have time slots filled, may cause time slot mix-up between fellows

4.6.2 Stimulus/Response Sequences

After already sending an appoint request, a student can return to an appointment status page and follow a link to change assigned fellow

Student will have to go through most of process again such as attaching draft, choosing specific time, etc.

4.6.3 Functional Requirements

The system shall allow students to submit a request to change to their appointment by asking to meet with a different fellow. This will need to be approved by staff before becoming final.

Ability to change time (REQ-7)

4.7.1 Description and Priority

This feature allows the fellow to change meeting time

Importance: 8

Advantage: Schedule of student may change and could be very necessary

Risk: May cause time slot mix-up and confusion amount fellow/student

4.7.2 Stimulus/Response Sequences

After already requesting an appointment on the appointment status page, student may follow a link to change time, student will then be able to choose from a list of different available time slots

4.7.3 Functional Requirements

The system shall allow students to submit a request to change their appointment time. This will need to be approved by a staff member before becoming final.

Ability to change time duration (REQ-8)

4.8.1 Description and Priority

This feature allows the fellow to change the default time duration of the appointment

Importance: 6

Advantage: Students may prefer to choose a specific length, saves time for both student and fellow

Risk: Changes default time slot length, may cause mixup, student may need more time than this slot and cause overtime

4.8.2 Stimulus/Response Sequences

After already requesting an appointment on appointment status page student can follow a link to change duration of the appointment.

4.8.3 Functional Requirements

The system shall allow students to request a change in their length of appointment. If students cannot be in the room for the allotted time, they may request to shorten or lengthen it. This must be approved by a staff member before becoming final.

# Other Nonfunctional Requirements

## Performance Requirements

The lists shall update every 5 seconds to reflect appointments being taken or cancelled. Students need to know rapidly when a spot has been taken or freed up so that they may request the most convenient appointment for themselves.

## Safety Requirements

The screen shall not flash rapidly because of the implementation of the program. Seizures are forbidden. The program shall not store personal information indefinitely, to protect individual privacy.

## Security Requirements

Security will follow protocols enacted by Canvas.

## Software Quality Attributes

The program shall be reusable with slight changes to code and skin. The database used and the destination of appointment requests should be changed to change the implementation. The GUI would need to be reskinned to not be Ursinus colors.

Simplicity is key for our system. We do not want a complex, hard to use scheduler that nobody can figure out. A few buttons and well-labeled options will be necessary to provide ease of use.

The system should be reliable, but will depend on staff of the writing room to check the pending appointments.

## Business Rules

Writing staff will be required to check pending appointments when the room opens for the day. This will begin the scheduling process to get appointments approved for the rest of the day and coming days. Appointments need to be approved a good amount of time in advance, so staff must stay on top of the list.

# Other Requirements

The program shall require the Canvas database for Ursinus Student’s login info.

We will need permission to put our project on Canvas, and to implement it. The backup plan is to try to put the scheduler on the Ursinus website as a tab on the homepage after logging in.

**Appendix A: Glossary**

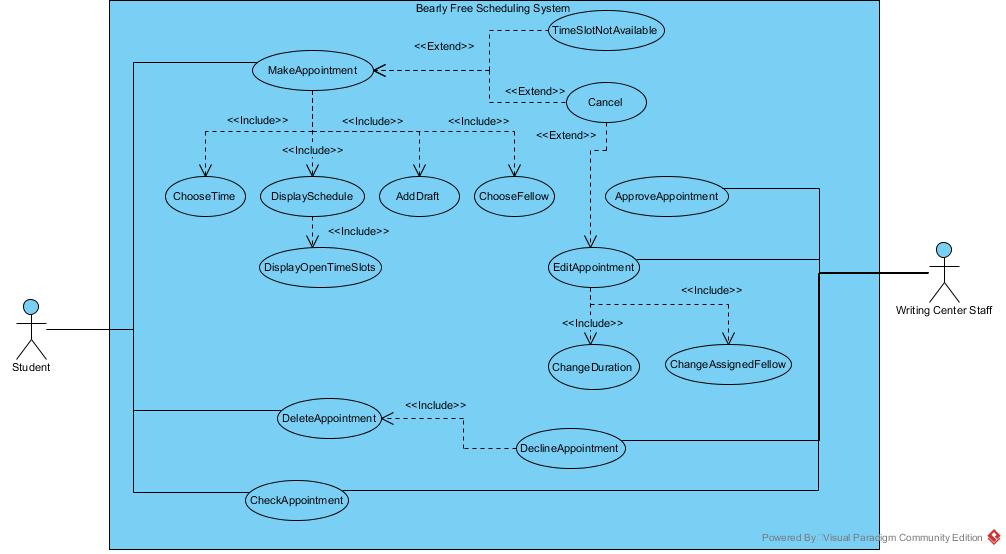
White List: a list of specific users who have access to different functions.

Reskinning: Changing the look of a GUI without changing the functionality. Ex: changing the colors of a button to better suit your business.

GUI: Graphic User Interface.

Fellow/ Writing fellow: a term used for the staff of the UC writing room.

**Appendix B: Analysis Models**



This is our current use case diagram. It may change over time with new challenges or time constraints.

**Textual Descriptions:**

[**Textual Descriptions.docx**](Textual%20Descriptions.docx)

The descriptions themselves would lengthen the SRS too much and make it unbearable. Pun intended. The file is available here to read at your leisure.

**Appendix C: To Be Determined List**

1. Ability to use UC student login database.
2. Ability to put scheduler on Canvas.
3. Ability to send email notifications to students.