Software Requirements Specification

for

LA Tech CS Advising Web Application

Version 1.1 Approved

Prepared by: Group 1, Group 2, Group 3, Group 4, Group 5
Department of Computer Science, Louisiana Tech
3
February 19, 2015

Client Approval:

Table of Contents

1 Introduction

1.1 Purpose

This software is a Louisiana Tech Advising Web application used on Personal Computers and Mobile Smartphones. It will be primarily used by the Computer Science department at first, but can be adapted to feature many different departments at the college if it catches on. The scope of the product is for students not currently enrolled in the Computer Science curriculum, students in the Computer Science curriculum, Advisors of students in the Computer Science curriculum, and the administrator or administration team. This SRS describes the whole system, and provides instructions to all teams involved in every aspect of the software creation.

1.2 Document Conventions

<N/A>

1.3 Intended Audience and Reading Suggestions

This SRS is a reference document intended for everyone involved in the project. Each of the following sections deals with the descriptions of the product, the different interfaces, and the functional and non-functional requirements. Every part of the document is important to all of the teams, but the development team should pay careful attention to the functional and non-functional requirements. The design team should focus on the interfaces, and the testing team should focus on the quality of each use case, and the testing of each use case as well.

1.4 Product Scope

This is going to be used to provide an easier, more accessible way to access student and advisor records, and also to register for advising, planning out your schedule for the rest of your time at the University, and to access a calendar of events held by different organizations in the Computer Science department. This will be beneficial to many people, such as students confused about advising, advisors, and it make the process much more streamlined and simple. The mobile functionalyity will be limited from the full experience on the desktop website.

1.5 References

<N/A>

2 Overall Description

2.1 Product Perspective

This app is intended to act as a companion for the advising process. To that end, it will piggyback onto the existing advising system and utilize databases already kept by the University. Self-contained databases should be created and populated with existing data, including program information (curricula), student information (transcripts and program enrollment), and the current course offerings.

2.2 Product Functions

The key functions of this product are to allow advisors to set up advising appointments for students to then sign up for. Additionally, another primary function of the product is for advisors to create a checklist for a particular student where the student's transcript is auto populated into the checklist so the advisor can see what classes are missing. The last main function of the software product is for students (or advisors) to interactively create their own course schedule to

outline what classes they will take each quarter for their college career. All remaining functionality to to provide administrative assistant and display useful information to aid the primary functions. This includes an activity log to keep digital record of all advising appointments and other activity on the application.

2.3 User Classes and Characteristics

Administrator:

Other users, following registration, can be given higher privileges by the administrator of the system.

Program Chair

The program chair can manipulate curricula for his associated program and push to the app's database.

The program chair can also submit future course offerings, which affects suggested course offerings for students.

Advisor

An advising schedule can be created by the adviser, specifying appointment blocks that can be chosen by students.

The adviser can view a check sheet, created by the app from the advisee's curriculum and transcript, which shows his current progress through the program.

Advisee

Appointments for advising can be selected from a schedule created by the adviser.

Suggested term schedules can be generated from prerequisite chains, current and future course offerings, and the student's program progress.

These term schedules aid the student in choosing courses for the next term.

Advising forms can be generated, automatically or manually, which are forwarded to the adviser upon completion.

Guest

Guests are any user that has not authenticated into the system yet. Guests have access to view any publicly available information on the application

**Note that administrators, program chair, and advisors all have same basic access control on the application.

2.4 Operating Environment

The operating environment will be a web application hosted on some sort of web server with web or mobile based clients connecting to it as needed. An example of one that might be used is Zam. While aspects of the application are designed with the intent of being mobile-friendly, other parts of the application will be intended for desktop use only (i.e. advisor setting up advising schedule)

2.5 Design and Implementation Constraints

This app is intended for use with existing data that is maintained by Louisiana Tech University. As a result, if the University's databases are restructured, or data is added or removed, the app's developers must account for the change, in case the restructure might affect the process, as the app is intended for full integration into the advising system. Security is also a concern, as FERPA protects the privacy of some information involved. It should also be noted that the ADA requires that disabilities are accommodated in the usage of this app. Also, a bulk update feature will also be available to keep the app up to date with the BOSS system. This will occur once a quarter.

2.6 User Documentation

On-line documentation will be provided, as full integration into the advising system implies that all users must be able to take advantage of the software.

2.7 Assumptions and Dependencies

It is assumed in the creation of this document that all processes described are both relevant and comprehensive. That is, the advising process can benefit from everything the app does and should not require any more input from it in order to function properly. The app is not meant to replace the advising process, but to enhance it.

This app does depend on existing data to pre-populate its own databases. This data is retrieved at regular intervals from the BOSS mainframe and is preloaded into the database. It also relies on regular updates to said databases to function properly. One previously developed portion of software regarding preference-based schedule prediction will also contribute to the app.

3. External Interface Requirements

3.1 User Interfaces

The default screen for the user is the “guest” home screen, which provides the user all options that do not require the user to be logged in. It also provides an option to log in the web app’s header.

Once logged in as any user, the homescreen is populated with the appropriate options for that user and the header option to “login” is replaced by options to “logout” and “change password”.

Options are generally selected in a point-and-click fashion. When using certain tools, such as the interactive course progression tool, the user will be able to interact with the system using double-click, drag, drop, and right-click events, and will interact with common internet controls such as checkboxes and text fields.

One critical success factor for the website will be to minimize the amount of clicking that needs to be done to navigate the website.

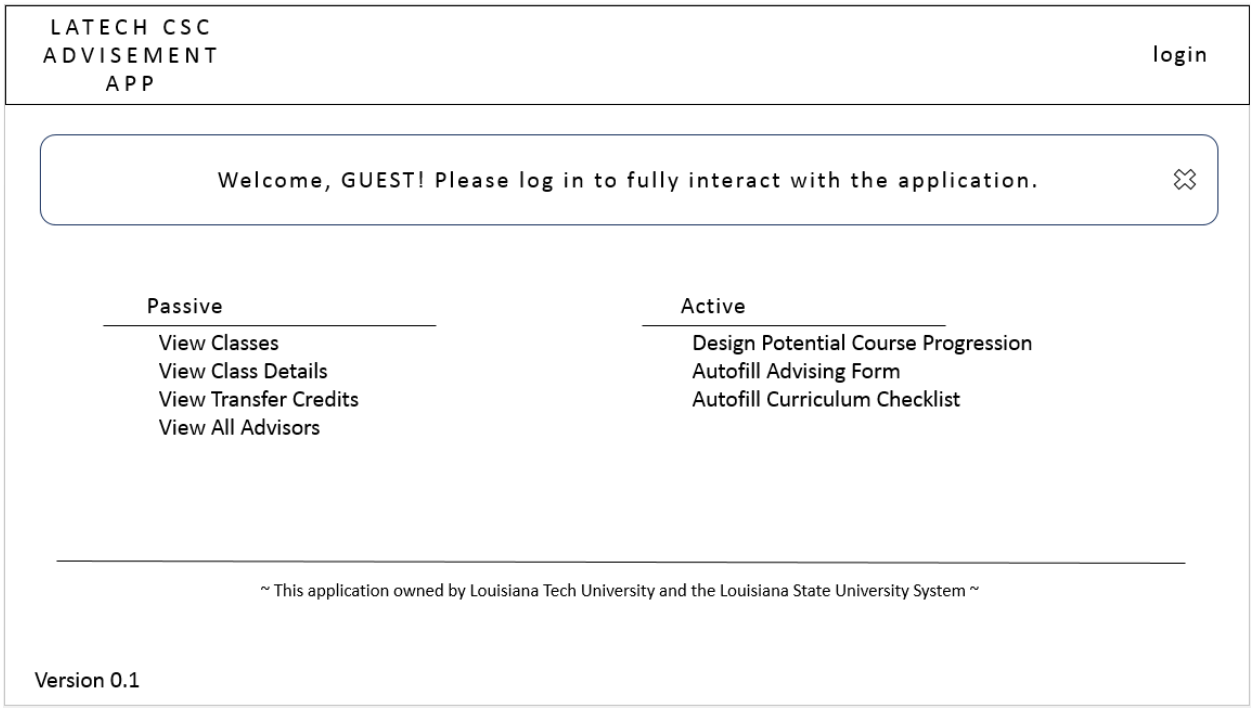


Figure 3.1 The Guest (logged-out user) home screen

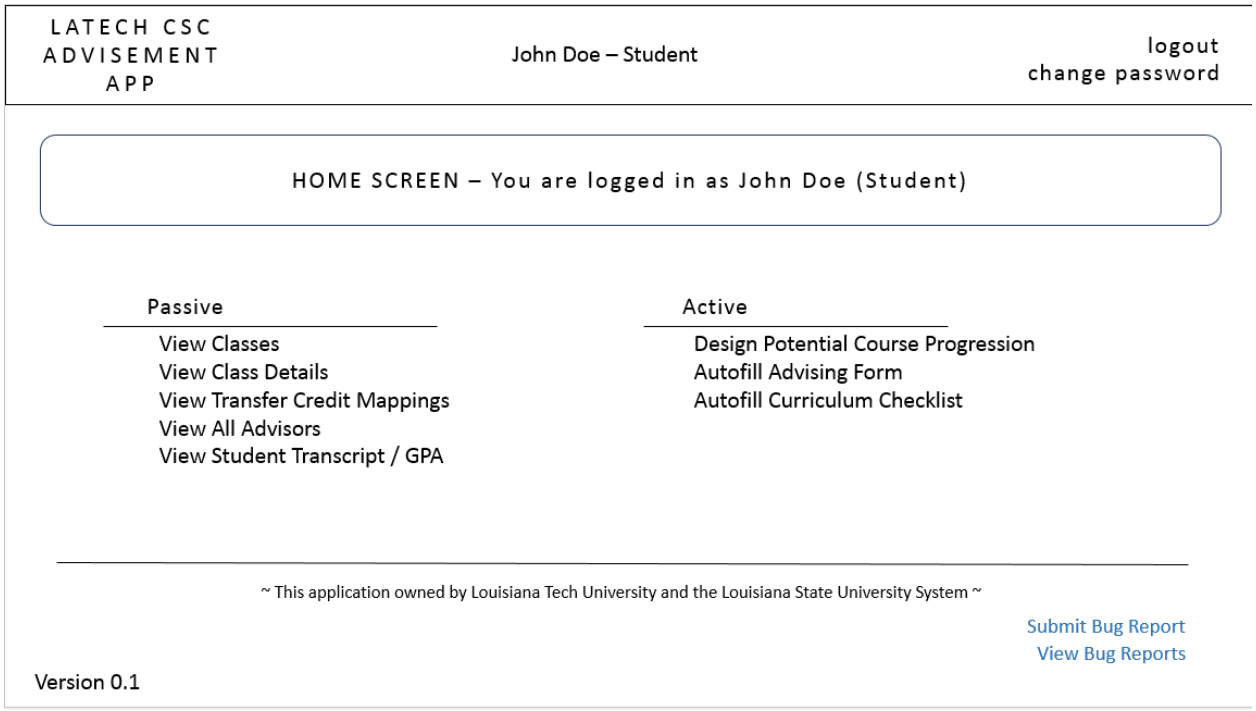


Figure 3.2 The student user home screen & its associated options

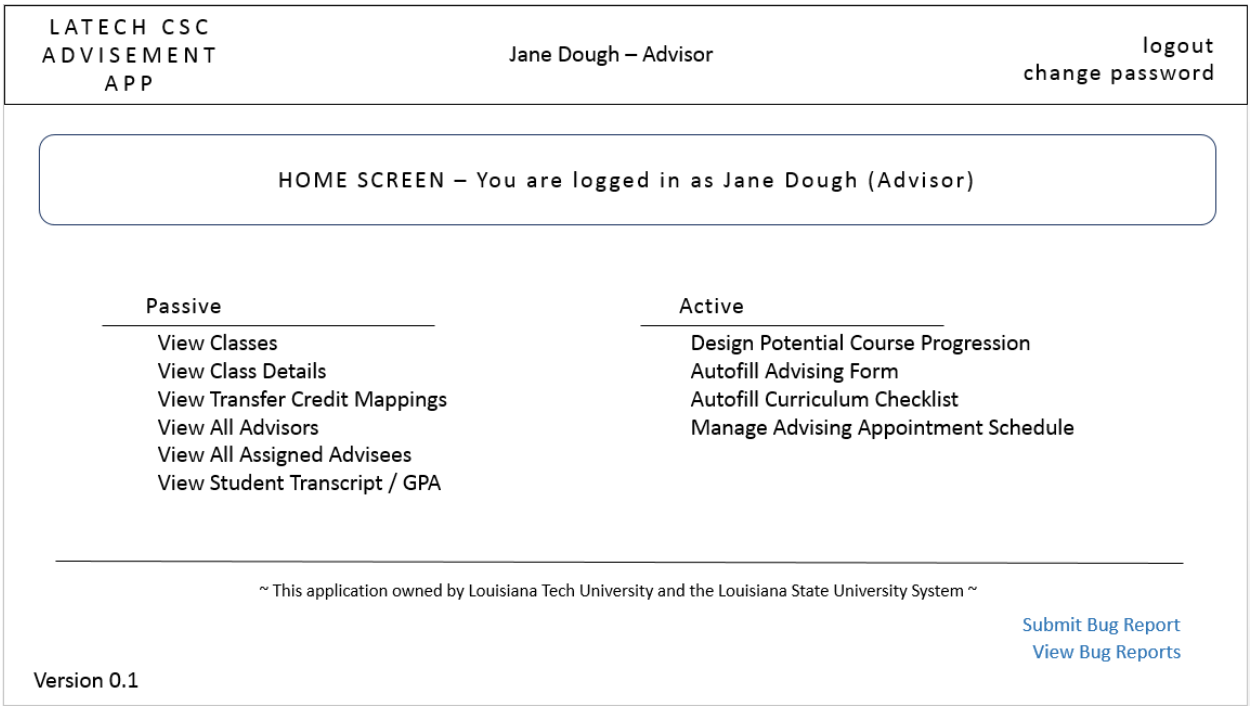


Figure 3.3 The advisor user home screen & its associated options

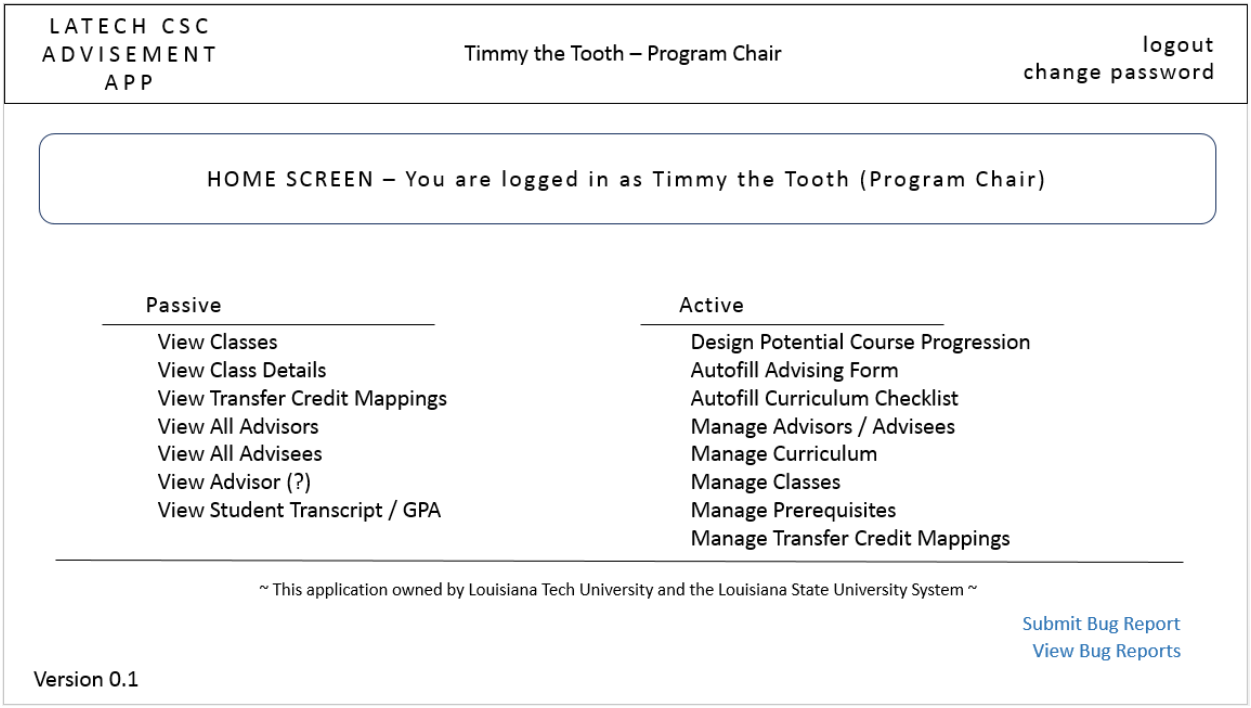


Figure 3.4 The program chair user home screen & its associated options

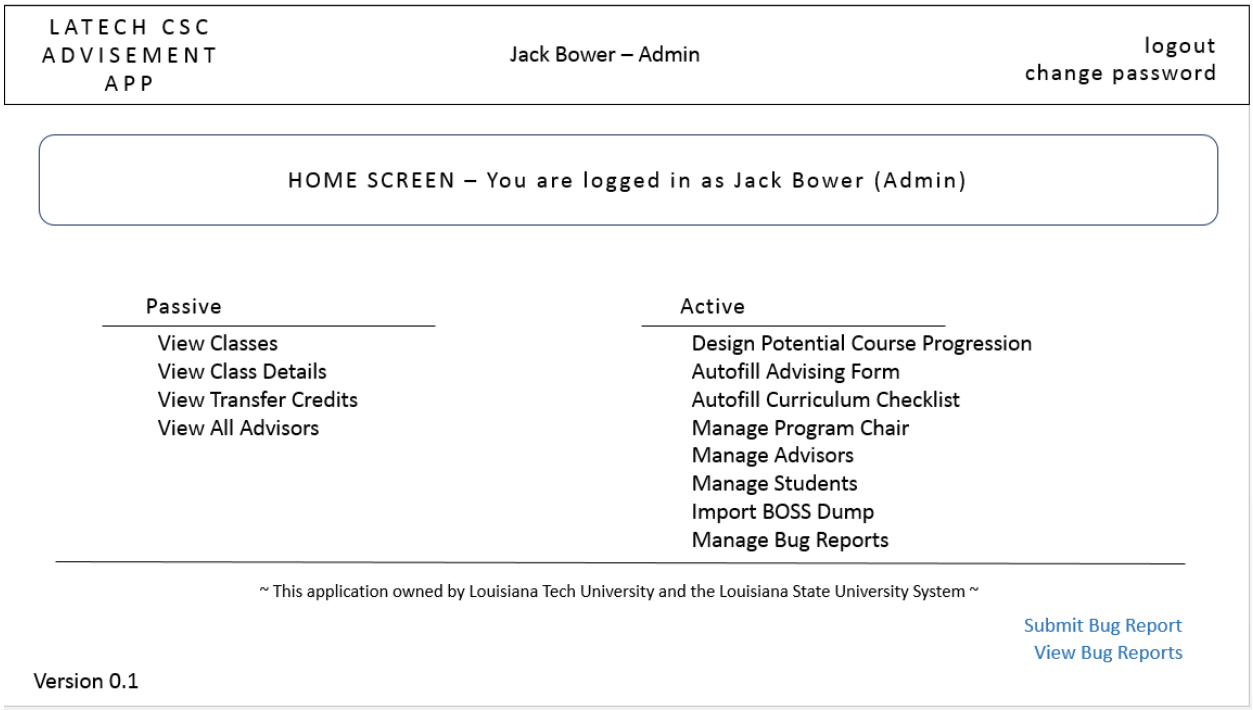


Figure 3.5 The admin user home screen & its associated options

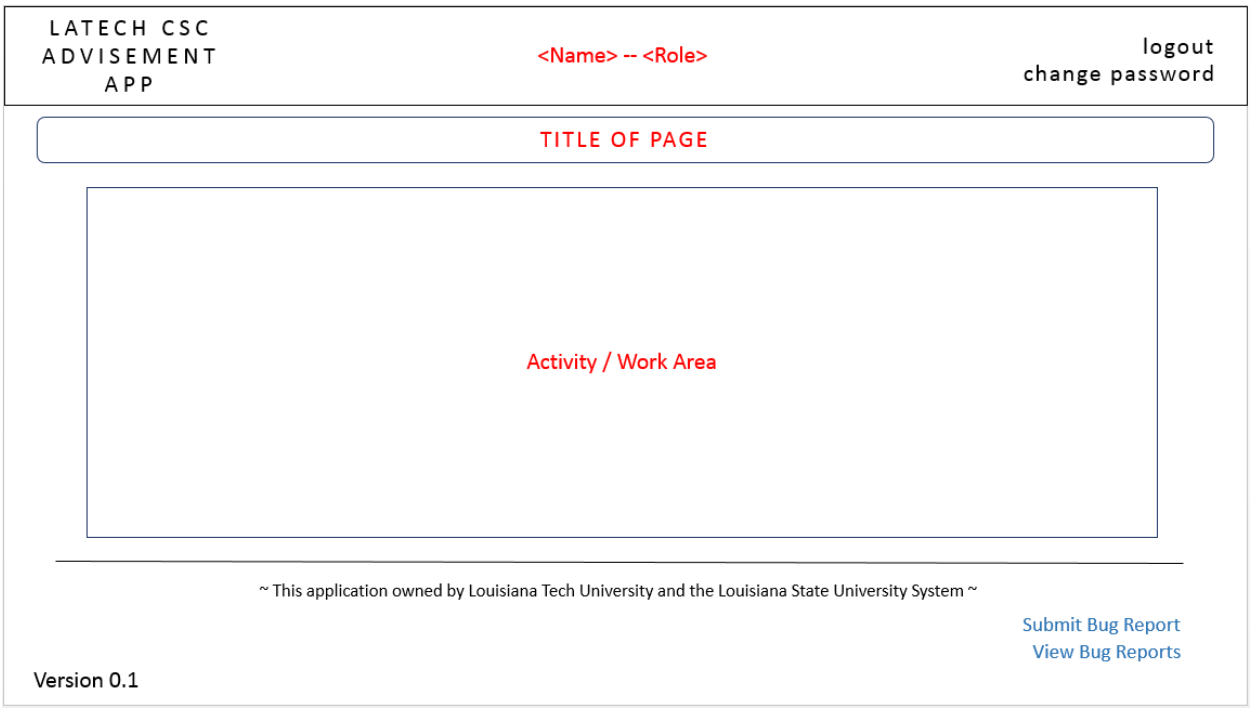


Figure 3.6 The generalized use case screen, where data & tools appear in the activity / work area

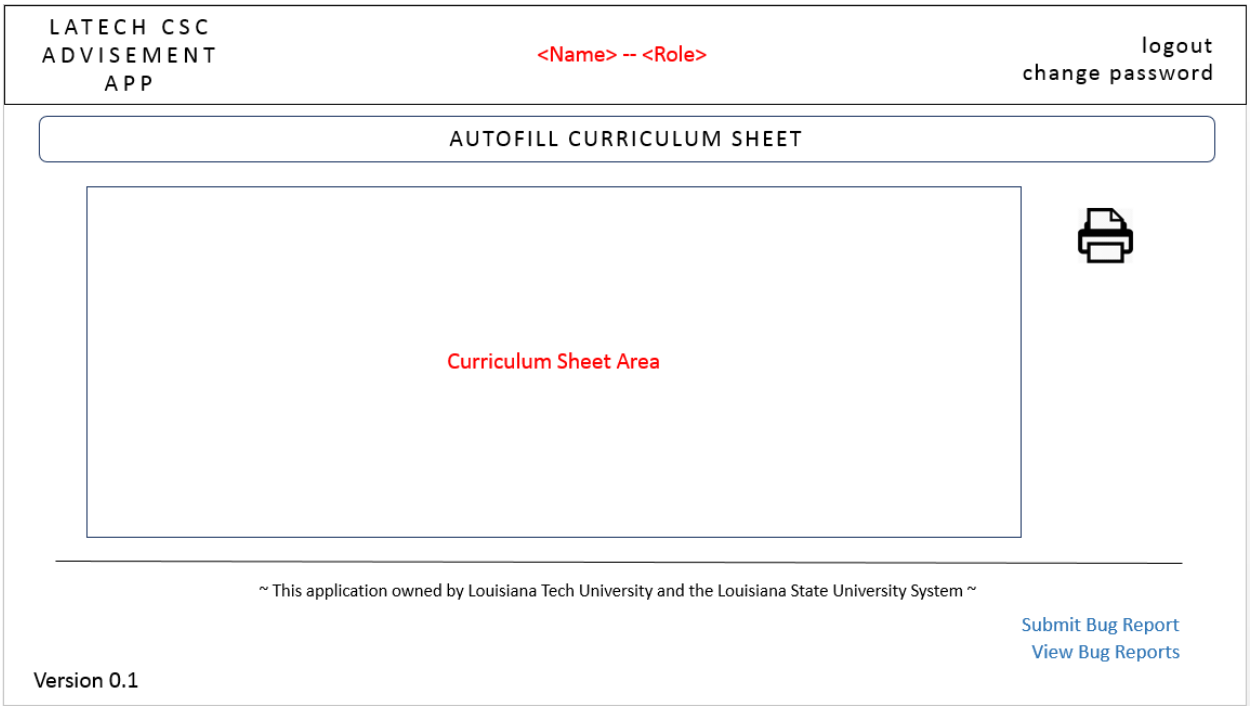


Figure 3.7 A specific use case screen ("Autofill curriculum sheet") as an example

3.2 Hardware Interfaces

The application will operate on current hardware with the potential to be hosted on a web server and operated through a web browser. The hardware includes but is not limited to:

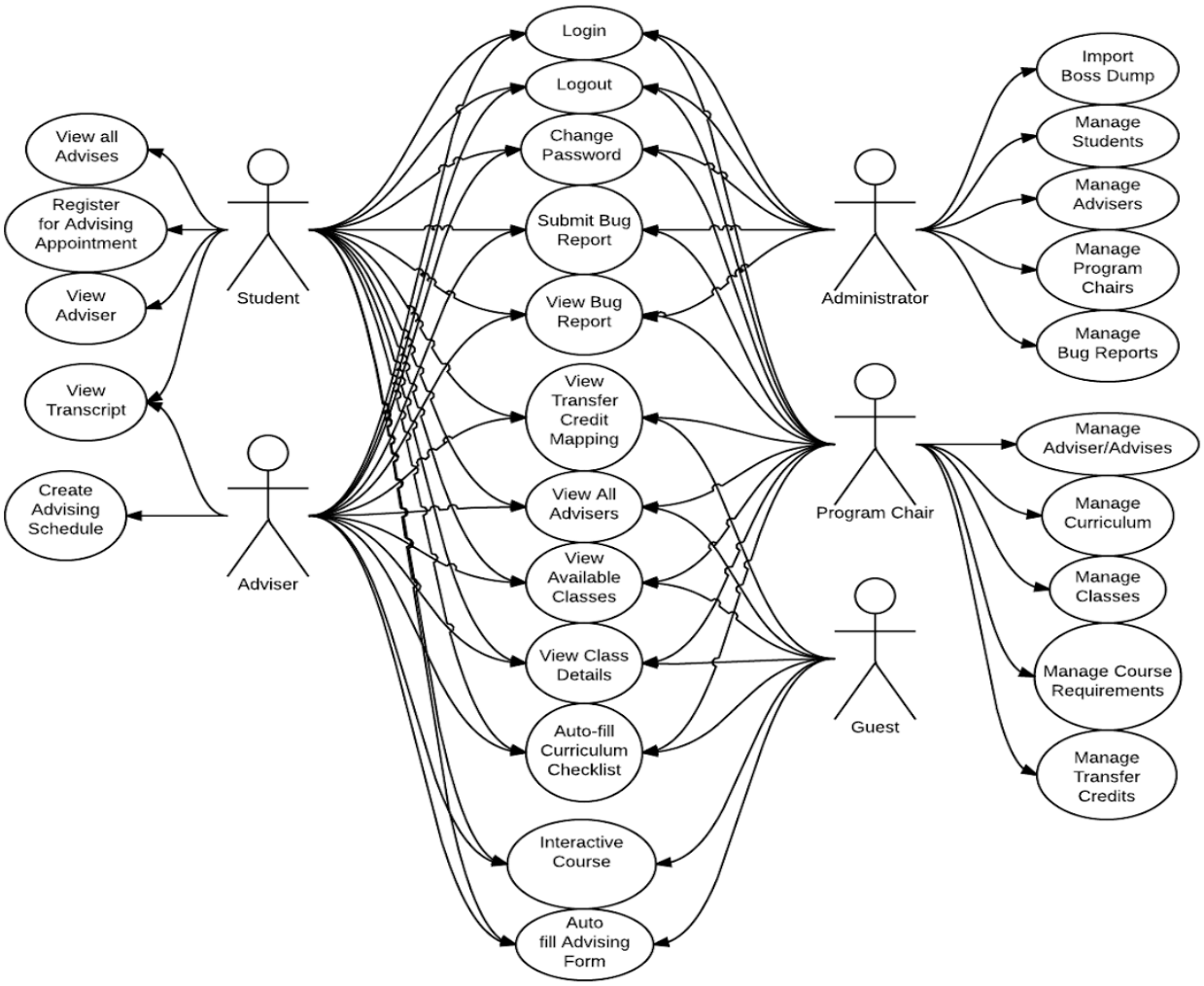
- Monitor
- Keyboard
- Mouse
- CPU w/ ability to access existing system
- Smart Phone w/ Internet Access

3.3 Software Interfaces

The application can be deployed either as a web application with a mobile friendly browser interface or as native mobile application unto mobile devices.

3.4 Communications Interfaces

The database should communicate with MySQL. The front end for the user to interact with should be HTTPS, being encrypted whenever the user is logged in as their information will be highly sensitive and subject to script security requirements from FERPA.



Correction: View All Advisees should be an Adviser usecase, not a Student usecase as reflected in the chart.

5. Use-Cases

5.1 Login

5.1.1 Actors: Student, Administrator, Adviser, Program Chair

5.1.2 Goal: Authenticates the user.

5.1.3 Input: User inputs username and password.

5.1.4 Output: System authenticates users and displays main menu.

5.1.5 Main Scenario: Username and password are authenticated, and user is logged in.

5.1.6 Pre-condition: User is not already logged in.

5.1.7 Steps:

5.1.7.1- User enters username and password, and clicks Login.

5.1.7.2- If authenticated, takes user to main menu.

5.1.8 Post-condition: Based on your role (Student, Advisor, etc), the specialized menu is displayed for the user.

5.1.9 Exceptional Scenarios:

5.1.9.1- User is not registered with the system.

5.1.9.2- Username or password aren't correct.

5.2 Logout

5.2.1 Actors: Student, Administrator, Adviser, Program Chair

5.2.2 Goal: revoke access.

5.2.3 Input: User clicks the "Logout" .

5.2.4 Output: Display the login screen.

5.2.5 Main Scenario: The user access is revoked and returned to the login screen until the login again.

5.2.6 Pre-condition: User is logged in.

5.2.7 Steps:

5.2.7.1- User enters clicks Logout.

5.2.7.2- User is sent to the login screen.

5.2.8 Post-condition: Login screen is displayed for the user.

5.2.9 Exceptional Scenarios:

5.2.9.1- Cannot connect to server. Return to login screen and do not allow access unless the user logs in again.

5.3 Interactive Course Schedule

5.3.1 Actor: Guest, Student, Adviser

5.3.2 Goal: The goal is for a student to generate a potential schedule of classes for their next academic quarter.

5.3.3 Input: The student (user) is expected to provide what classes they are interested in taking (and any backup classes in case of conflicts or full classes). Student transcript is fetched from the database to check for prerequisites.

5.3.4 Output: The output is a complete prospective schedule for next quarter and any alerts / warnings regarding course conflicts or prerequisite failures. This use case also has some part in 5.9 and 5.10.

5.3.5 Main Scenario: The user selects courses they wish to take from a list of overall available classes. Once all desired lasses and potential backup classes have been selected, the user hits ‘create schedule’ and the scheduled is generated based off of the user input. The user can then look over the schedule and save it or go back and adjust their input to change the resulting schedule. The user can also add in preferences to the schedule, such as no morning classes or a professor they do not enjoy.

5.3.6 Pre-condition: The user must be logged in and their transcript data must be available for prerequisite checking. Additionally, the courses for next quarter must be available.

5.3.7 Steps:

5.3.7.1 User selects ‘Build Schedule’ from main screen.

5.3.7.2 User selects desired classes from overall list of courses

5.3.7.3 User clicks ‘Create Schedule’ to finalize list and generate a prospective schedule.

5.3.7.4 User reviews generated schedule and saves it or goes back to course list to create a new schedule.

5.3.8 Post-condition User saves a schedule for review later. Schedule is forwarded to Advisor.

5.3.9 Exceptional Scenario 1 Prerequisite conditions are not met.

If prerequisite conditions are not met, course is not added to schedule and user is alerted.

Students selects courses with conflicting times.

The first course selected from the list is added to the schedule and the second course is not. The user is alerted about the time conflict.

If the student is currently in a course, and that course a prerequisite for a course that they want to take.

5.4 Register for Advising Appointment

5.4.1 Actor: Student

5.4.2 Goal: The goal is for a student to register for an advising appointment at a designated time slot provided by the advisor.

5.4.3 Input: The student (user) chooses an available appointment time slot. Available time slots are fetched from the database.

5.4.4 Output: Whether or not the student has been successfully registered for the desired time slot.

5.4.5 Main Scenario: The user is presented with a schedule showing all available advising appointment time slots. The user then clicks on the desired time slot to register for that time. If successful, an email confirming the appointment is sent to the student and the advisor. Advisees are also able to cancel appointments as well, sending an e-mail to the advisor when it is done.

5.4.6 Pre-condition: The user must be logged in and the student's advisor must have posted their available advising appointment time slots.

5.4.7 Steps:

5.4.7.1 User selects 'Advising Appointment' from main screen.

5.4.7.2 User selects desired advising appointment time slot from the set of available time slots.

5.4.7.3 Confirmation email sent to student and advisor

5.4.8 Post-condition: Confirmation email is sent to advisor and student. The selected time slot is marked as not available (filled).

5.4.9 Exceptional Scenario 1

Advisor has not published available time slots

If there are not time slots published, the user is told no time slots available and the advisor's email address is presented to the user.

No remaining available time slots

If all available time slots have been filled, the user is provided with the advisor's email address for contact

Student already has scheduled time slot

If student already secured a time slot, they may select a different time slot from the set of remaining available time slots. Upon selecting a new time slot, the student's old time slot is marked available again and the new time slot is filled. Student and advisor are sent an email detailing the time slot change.

5.5 Manage Curriculum

5.5.1 Actor: Program Chair

5.5.2 Goal: The goal is for a program chair to craft or update a curriculum associated with his program.

5.5.3 Input: The program chair (user) inputs classes (codes, names, credits) that are in the curriculum, as well as their prerequisites, if applicable. Slots for electives and minor/concentration credits are also allocated based on program chair discretion.

5.5.4 Output: The result is a curriculum that displays all classes required for graduation in the program, with their prerequisite courses, empty slots with descriptors for possible electives, and total number of credit hours in the program.

5.5.5 Main Scenario:The user walks through an interface for easy course addition. The department code, course number, course name, prerequisite department code and course number, and credits are supplied, and an entry is made. An option, aside from course addition, adds a blank entry under an elective or minor/concentration category. Once all entries are made, the user can review the completed curriculum, delete entries if desired, and push the curriculum to the associated program.

5.5.6 Pre-condition:

The user must be logged in and verified as the program chair. He can only push to his associated program.

5.5.7 Steps:

5.5.7.1 User selects ‘Create Curriculum’ from main screen

5.5.7.2 If desired, an existing curriculum can be imported.

5.5.7.3 Entries are added with associated wizards. Selected entries can be deleted or edited.

5.5.7.4 User reviews curriculum and either discards changes or pushes the created/modified curriculum as the official program curriculum

5.5.8 Post-condition: User and advisors in the program are emailed a copy of the curriculum for review.

5.5.9 Exceptional Scenario 1

Total hours in curriculum lower than the University's requirements.
The curriculum cannot be pushed, but a copy can be sent by email to the user for offline revision.
Class to be added does not exist.
A placeholder class with arbitrary information set by the user can be added to the curriculum, but the entry appears in red, and the curriculum cannot be pushed. Again, a copy can be sent to the user for revision.

5.6 Create Advising Schedule

5.6.1 Actor: Advisor

5.6.2 Goal: The goal is for an advisor to be able to select advising appointment schedule time blocks.

5.6.3 Input: The advisor must select times out of the week that he/she is available for advising. Time blocks are provided in 15 minutes blocks (or a different block size specified by the advisor) throughout the week and the advisor picks the blocks that they are available for. This input can be provided from a google calendar import.

5.6.4 Output: The output of this is that the advisor's available advising schedule is published and available for all students under that advisor.

5.6.5 Main Scenario: The advisor logs into the system and select "Choose Advising Schedule". Once there, the advisor is presented with a set of 15 minute (default) time blocks throughout the week of advising. Those time blocks can be changed, at the advisor's discretion. The advisor selects the available time blocks (ensuring the number of time blocks equals or exceeds the number of advising students).

5.6.6 Pre-condition: The user must be logged in and verified as an advisor.

5.6.7 Steps:

5.6.7.1 User selects ‘Choose Advising Schedule’ from options.

5.6.7.2 All available 15 minute time slots are presented to the advisor

5.6.7.3 Advisor selects all time slots available for their schedule.

5.6.7.4 Advisor selects “Finish” and the available time slots are published and made available to all students under that advisor

5.6.8 Post-condition: Advisor’s advising schedule is published for students to view and select advising appointments from.

5.6.9 Exceptional Scenario 1:

Advisor selects less time slots less than the number of students under that advisor.

The advisor is given a warning about the number of time slots and the schedule can’t be published.

5.7 Change Password

5.7.1 Actor: Administrator, Program Chair, Advisor

5.7.2 Goal: The user, once authenticated, will be able to change the user's password credential for login. User passwords can be changed regularly to ensure security of user.

5.7.3 Input: The new password for the user account.

5.7.4 Output: Whether or not the new password was successfully changed.

5.7.5 Main Scenario: The user, already authenticated, can select the option to manage the account. In the account management state the user can select the "change password" option. The user will be prompted for re-authentication, and,

once verified, the system will request that the new password be input twice. If the two new password entries match, the new password will be set as the user's new password, and the user will be logged out.

5.7.6 Pre-condition: The user must be logged.

5.7.7 Steps:

5.7.7.1 The user selects "change password" from home screen.

5.7.7.2 The system will prompt the user to re-authenticate with their "old" credentials

5.7.7.3 The user will be prompted to enter the new password.

5.7.7.4 The user account password will be changed to the new password.

5.7.8 Post-condition: The user account password will be changed to the new password.

5.7.9 Exceptional Scenario 1

If the user, logged in as an user, chooses the "change password" option and fails to re-authenticate, the user will be logged out

If the user is re-authenticated, and both of the requested new password entries do not match, the "password change" will fail and the user will be returned to the "manage account" screen

5.8 View Transcript

5.8.1 Actors: Student, Adviser

5.8.2 Goal: The goal is for a student or advisor to view all classes the student has available on their transcript.

5.8.3 Input: The student or advisor (user) selects the view transcript option and (in the case of advisors) selects which student transcript to view

5.8.4 Output: The output is a complete transcript of the student.

5.8.5 Main Scenario: The advisor selects which student transcript to view or the student clicks view transcript and the transcript is presented to the user for viewing.

5.8.6 Pre-condition: The user must be logged in and their transcript data must be available.

5.8.7 Steps:

5.8.7.1 User selects 'View Transcript' from main screen

5.8.7.2 User inputs student ID if user is advisor

5.8.7.3 Transcript is shown.

5.8.8 Post-condition: The student's transcript is shown.

5.8.9 Exceptional Scenario 1: If student transcript data is not available, no transcript is shown.

5.9 Auto-Fill Curriculum Check Sheet

5.9.1 Actors: Guest, Program Chair, Advisor, Student

5.9.2 Goal: The goal is for a user to view the CS curriculum and if the student transcript is available, auto-populate the check list

5.9.3 Input: The user provides no information. If the user is logged in or a student id is selected, the curriculum pulls the student's transcript for auto populating the fields.

5.9.4 Output: The output is a complete curriculum sheet with the classes auto-populated if the information is available.

5.9.5 Main Scenario: The user selects curriculum check sheet and if logged in the classes are pulled from the database to be filled into the curriculum

5.9.6 Pre-condition: The user must be logged in to have the classes auto-populated

5.9.7 Steps:

5.9.7.1 The user selects "Curriculum Check Sheet".

5.9.8 Post-condition: User can save the curriculum as a PDF for printing, or a XLS spreadsheet to keep the format with the Registrar.

5.9.9 Exceptional Scenario 1: N/A

5.10 Autofill Advising Form

5.10.1 Actor: Guest, Student, Advisor

5.10.2 Goal: The goal is for a student to generate an advising form with the recommended classes for the student to take next quarter

5.10.3 Input: The student (user) has their transcript information pulled from the database along with the courses available next quarter.

5.10.4 Output: The output is an advising form filled with the recommended classes for the student.

5.10.5 Main Scenario: The user selects the automatic advisor tool and if the classes for next quarter are available, a screen showing an auto-generated advising form with recommended classes filled in is displayed.

5.10.6 Pre-condition: The user must be logged in and their transcript data must be available for prerequisite checking. Additionally, the courses for next quarter must be available.

5.10.7 Steps:

5.10.7.1 User selects ‘Automatic’ Advisor’ from main screen.

5.10.7.2 User receives the displayed auto-generated advising form

5.10.7.3 User can print or save the form if needed

5.10.8 Post-condition: User saves the form for later use if needed.

5.10.9 Exceptional Scenario 1

If no course information is found for next quarter, the form cannot be generated and the user is sent back to the main screen.

5.11 View Advisees

5.11.1 Actor: Advisor

5.11.2 Goal: Show list of advisees

5.11.3 Input: Click the “View Advisee” button

5.11.4 Output: Display a clickable list of the advisor’s advisees with an edit button.

5.11.5 Main Scenario: User clicks “View Advisee” an is shown a list of their advisees

5.11.6 Pre-Condition: User is logged in as an advisor

5.11.7 Steps:

5.11.7.1 Step 1: Click the “View Advisee” button

5.11.7.2 Step 2: The list of advisees are retrieved from the server

5.11.7.3 Step 3: Advisees, if any, are displayed

5.11.8 Post-Condition: Advisor is shown a list of their advisees

5.11.9 Exceptional Scenario:

5.11.9.1 The advisor has no advisees. Display “You have no advisees”

5.11.9.2 The advisee list cannot be found/no connection to database

Display “Advisee list unavailable”

5.12 View All Advisors

5.12.1 Actor: Program Chair

5.12.2 Goal: for a user to view all advisors.

5.12.3 Input: Whether a user wants to view all advisors’ information.

5.12.4 Output: all advisors contact info.

5.12.5 Main Scenario:The user selectsto view all advisors and the required contact information is then presented.

5.12.6 Pre-condition: N/A

5.12.7 Steps:

5.12.7.1 User selects ‘View All Advisors’ from main screen.

5.12.7.2 User selects whether or not to view their advisor

5.12.8 Post-condition: The appropriate advisor contact information is displayed

5.12.9 Exceptional Scenario 1: N/A

5.13 View Transfer Credit Mapping

5.13.1 Actor: Program Chair, Advisor

5.13.2 Goal: for a user to view Transfer Credit Mapping.

5.13.3 Input: Whether a user wants to view Transfer Credit Mapping.

5.13.4 Output: Transfer Credit Mappings.

5.13.5 Main Scenario: The user selects to view Transfer Credit Mappings

5.13.6 Pre-condition: N/A

5.13.7 Steps:

5.13.7.1 User selects ‘Transfer Credit Mappings’ from main screen.

5.13.7.2 User selects whether or not to view Transfer Credit Mappings

5.13.8 Post-condition: Transfer Credit Mappings are displayed

5.13.9 Exceptional Scenario 1: N/A

5.14 View Class Details

5.14.1 Actors: Student, Advisor, Program Chair, Guest

5.14.2 Goal: Display a description and list of the prerequisites and corequisites required for a particular course

5.14.3 Input: Selection of a course on the “View Available Courses” page

5.14.4 Output: A description and list of prerequisites and corequisites required for the selected course

5.14.5 Main Scenario: A pop-up bubble displays next to the selected course, showing the course requirements

5.14.6 Pre-condition: A user is on the “View Available Classes” page

5.14.7 Steps:

5.14.7.1 Step1: A user navigates to the “View Available Classes” page

5.14.7.2 Step 2: The user selects a course from a list of courses

5.14.7.3 Step 3: The selected course’s description and requirements will be displayed in a pop-up bubble next to the course.

5.14.8 Post-condition: NA

5.14.9 Exceptional Scenario: The course’s requirements are missing: Display an error message and recommend that the user contact an administrator.

5.15 Import Boss Dump

5.15.1 Actors: Administrator

5.15.2 Goal: Incorporate the latest Boss Dump into the database

5.15.3 Input: User clicks “import boss dump”

5.15.4 Output: System imports Boss dump and displays that the import is completed

5.15.5 Main Scenario: User imports the boss dump and the latest boss dump is incorporated into the database

5.15.6 Pre-condition: User is logged in.

5.15.7 Steps:

5.15.7.1- User clicks “Import Boss Dump”.

5.15.7.2- When import complete, display “Import completed”.

5.15.8 Post-condition: Latest Boss Dump is incorporated into the database and “Import completed” is displayed.

5.15.9 Exceptional Scenarios 1:

Unable to connect to Boss. Display “unable to connect”

5.16 Manage Transfer Credit Mapping

5.16.1 Actor: Program Chair

5.16.2 Goal: The goal is for the program chair to apply transfer credit mappings.

5.16.3 Input: The student (user) is expected to provide what transfer credit mappings they want to add. These will be applied to the new curriculum.

5.16.4 Output: The output is a transfer credit mapping

5.16.5 Main Scenario: The user adds New transfer credit mappings

5.16.6 Pre-condition: The user must be logged in as program chair.

5.16.7 Steps:

5.16.7.1 User selects ‘Manage Transfer Credit Mappings’ from main screen.

5.16.7.2 User modifies transfer credit mappings

5.16.7.3 User clicks ‘Save’ to finalize changes and create new mapping

5.16.8 Post-condition: A new mapping is added and available to be published.

5.16.9 Exceptional Scenario 1: N/A

5.17 Manage Program Chair

5.17.1 Actor: Administrator

5.17.2 Goal: The goal is for the Administrator to be able to assign the role of program chair.

5.17.3 Input: The user adds/removes program chairs from the database

5.17.4 Output: The output is the present program chair

5.17.5 Main Scenario: The user adds/removes program chair from database

5.17.6 Pre-condition: The user must be logged in as Administrator.

5.17.7 Steps:

5.17.7.1 User selects ‘Manage Program Chair’ from main screen.

5.17.7.2 User modifies the Program Chair

5.17.7.3 User clicks ‘Save’ to finalize changes

5.17.8 Post-condition: A new program chair is made available

5.17.9 Exceptional Scenario 1: N/A

6 Other Non-functional Requirements

6.1 Performance Requirements

The application will be required to handle an increasing number of CS and CYEN students and faculty. At this time. An appropriate number of around 500 total users will be a good benchmark for the performance of this system.

The application will use a shadow database in order to increase its availability beyond the availability of BOSS, where the shadow database's data originates from.

6.2 Safety Requirements

The system should create account backups on a per account basis so that unfinished work is preserved and critical information is not lost.

6.3 Security Requirements

System should log failed login attempts for observation and protection against attacks. Additionally, the application will maintain up to date web server security to prevent student information from leaking to non-authorized users. FERPA and ADA will also be involved in the security aspects of the application. All authentication procedures should follow industry standard security protocols, such as hashing with salts and HTTPS.

6.4 Software Quality Attributes

All operations must be implemented as specified in this document. Any bugs and problems should be eliminated, with bugs that interfere with the functionality of the program fixed first. Upon completion, a packing and installation system will be created and documented so the server and client applications can be easily and readily distributed onto any compatible platform.