

1. General overview of the application, what it should do, and how it should perform

This application will provide Computer Science students at Columbus State University the ability to create a profile which contains degree information. They can choose which degree and calendar year they are registered for and the application will pull those requirements for the curriculum. The desktop interface will provide the ability to upload a .csv file for classes they have already taken to bulk update their profile of existing classes, or manually add them one by one. The system will show what classes are being offered on schedule and what classes are needed by the student, for them to choose classes they would like to take. Once the student has decided on the classes they must submit them to their advisor for approval or adjustments. Advisors have the ability of seeing all students that have created a profile and reported them as their advisor. The student now has a digital record of what was suggested for them to take along with what classes they still need.

2. Mobile Implementations

a) Mobile Implementation

- i. The mobile implementation for Cougar Advisor begins as typical mobile applications begin, with a splash screen. This splash screen displays the Columbus State University logo and the title of the application.(see screen shot below)



- ii. Following the splash screen is a welcome screen that uses the splash screen as a background and prompts the user to create a profile for the application.



- iii. In the profile creation screen the user is provided a form that consists of two textboxes and one radio group and five dropdown menus. These selections are used to determine what information needs to be displayed to the user throughout the application.

Two side-by-side screenshots of the 'Create Profile' screen. The left screenshot shows the form with fields for First Name, Last Name, Level (radio buttons for Undergraduate and Graduate), Classification (dropdown for Freshman), Track (dropdown for Applied), Major (dropdown for Computer Science), Starting Term (dropdown for Fall), and Starting Year (dropdown for 2014). The right screenshot shows the same form with the values filled in: First Name: Ortiz, Last Name: Green, Level: Graduate, Classification: Freshman, Track: Software Development, Major: Computer Science, Starting Term: Fall, and Starting Year: 2014. Both screens have a 'Create Profile' button at the bottom.

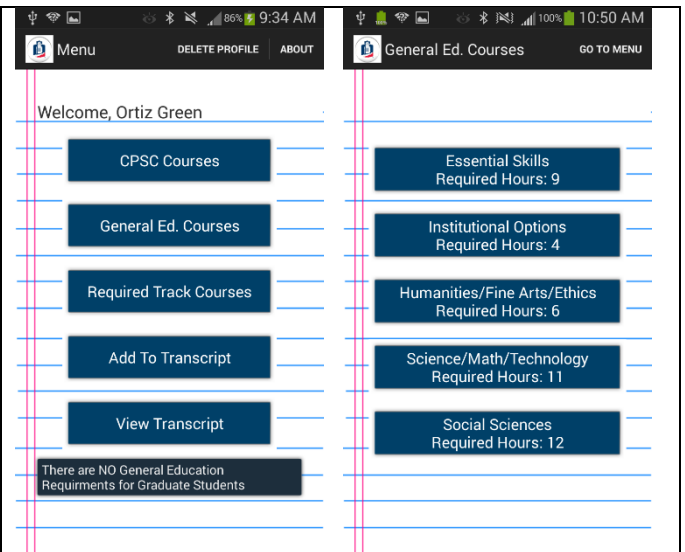
- iv. After the users clicks the Create Profile button they will be prompted to confirm their entries and then they are presented with the main menu of the application from this menu they are able to choose what they would like to view. This Main menu consists of five main buttons (CPSC Courses, General Ed. Course, Required Track Courses, Add to Transcript, View Transcript) as well as two menu buttons (Delete Profile, About)

Two side-by-side screenshots. The left screenshot shows a 'CONFIRM' dialog box with the following text: 'Name: Ortiz Green', 'Major: Computer Science', 'Level: Graduate', 'Starting Term: Fall 2014', 'Concentration: Software Development'. It has 'REDO' and 'SAVE' buttons. The right screenshot shows the 'Menu' screen with a 'WELCOME, Ortiz Green' message and five main buttons: 'CPSC Courses', 'General Ed. Courses', 'Required Track Courses', 'Add To Transcript', and 'View Transcript'. There are also 'DELETE PROFILE' and 'ABOUT' buttons at the top right.

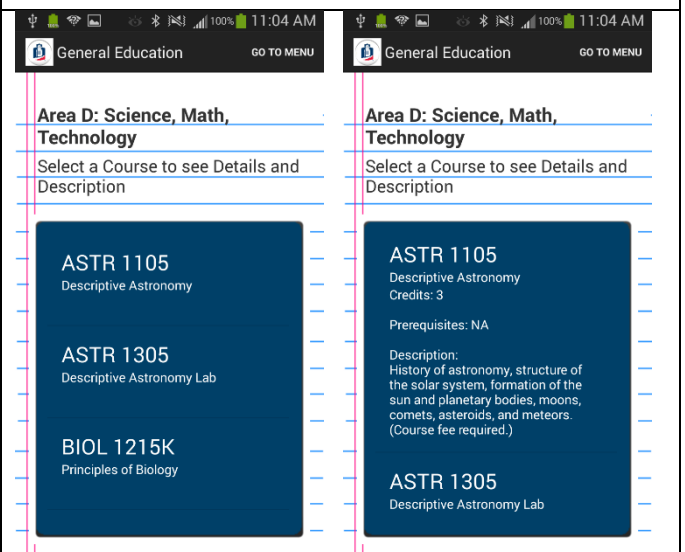
- v. Starting with the first option of the main menu CPSC Courses this section of the application will list all of the available CPSC courses for the user depending on the level they selected (Undergraduate or Graduate) in this list of classes the user can select a class and the list item will expand providing the user the vital information for that class (Course Number, Name, Credit Hours, Prerequisites, and description) to collapse this information the user just simply selects the item again.

Two side-by-side screenshots of the 'Computer Science Classes' screen. The left screenshot shows a list of 'Graduate CPSC Courses' with four items: 'CPSC 5115 Algorithm Analysis and Design', 'CPSC 5125 Computer Graphics', 'CPSC 5135 Programming Languages', and 'CPSC 5138'. The right screenshot shows the expanded view for 'CPSC 5115', displaying its description, prerequisites (CPSC 2108, MATH 5125), and a detailed description of the course content.

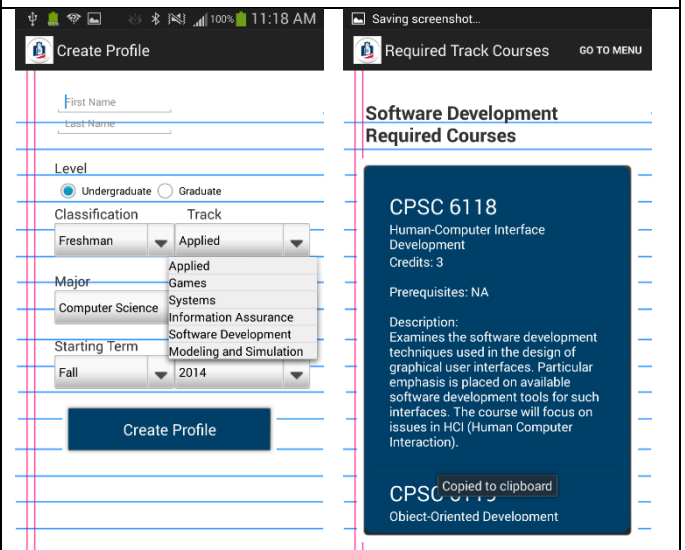
vi. In the second section of the application (General Ed. Courses) the user will receive one of two possible results based on the level that they chose when creating their profile. If the user is a graduate student then they will receive a message alerting them that there are no General Education requirements for graduate students. The other functionality that only undergraduate students will see is a secondary menu that breaks the general education requirements into their dedicated fields.



vii. The General Ed. Course Sub Menu as stated above breaks down the general ed courses into their respective fields. Once a field selection is made the application follows the same functionality as the CPSC Course list above and lists the classes for that section and allows the user to expand and collapse to show and hide the class's vital information. For convenience purposes each application window that is launched from the main menu there is an application menu button "Go To Menu" that will take the user back to the main Menu



viii. The next section of the application "Required Track Courses" functions similarly to the CPSC Courses section with the exception that the course list will consist of the required courses for the Track selected when the user created their profile. Available Concentrations are "Applied", "Games", "Systems" (Undergraduate) and "Information Assurance", "Software Development", "Modeling and Simulation" (Graduate)

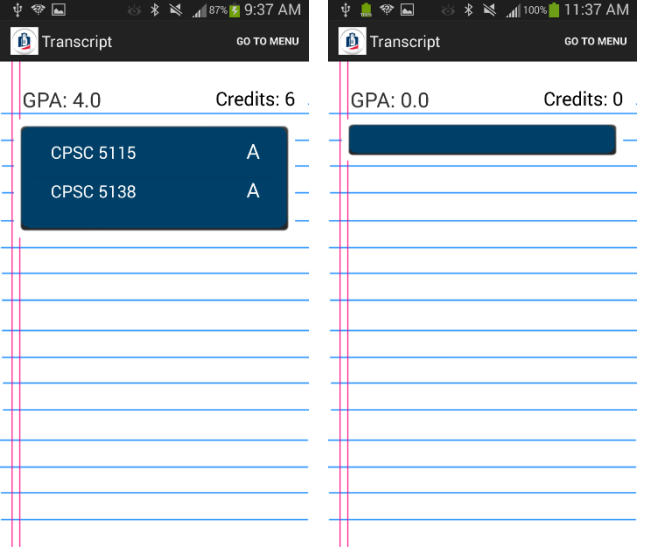
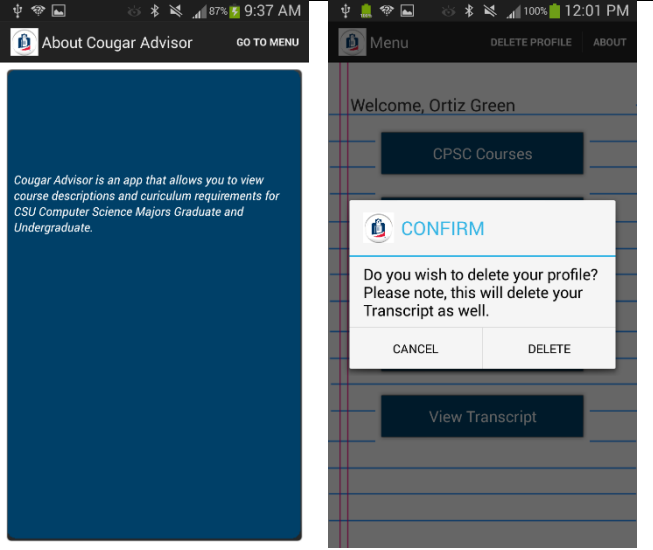


- ix. To allow the user to maintain a record of the classes they have taken the next section “Add to Transcript” allows the user to select courses from a dropdown menu and provide the grade the received in that course. From this screen a user can add a new course “Add to Transcript” or update a course they have retaken “Update Transcript”.

The left screenshot shows the 'Add to Transcript' app interface. At the top, there's a header with 'Add to Transcript' and a 'GO TO MENU' button. Below the header, the user's name 'Ortiz Green' is displayed. A 'Select a Class' dropdown menu is open, showing a list of CPSC courses. The right screenshot shows the same interface, but the 'Grade' field is now populated with the letter 'A'. Below the 'Grade' field, there are two buttons: 'Add To Transcript' and 'Update Transcript'.

- x. The “Add to Transcript” section has error handling features that keep the user from adding duplicate classes and also makes sure that a valid grade is provided. Each error handling consists of appropriate error messages that alert the user to their mistake and provides guidance of how to correct that mistake.

The top row of screenshots shows error handling for duplicate courses and invalid grades. The left screenshot shows an error message: 'The Course you Selected is Already in Your Transcript Use the Update Transcript Below'. The right screenshot shows an error message: 'Grand must be a Single Character (A, B, C, D, F) Lower case is acceptable'. The bottom row of screenshots shows error handling for invalid grades and courses not found in the transcript. The left screenshot shows an error message: 'Grand must be a Single Character (A, B, C, D, F) Lower case is acceptable'. The right screenshot shows an error message: 'The Course was not found in Your Transcript Try using the Add To Transcript button Above'.

<p>xi. The Final Section of the application allows the user to view their transcript (View Transcript) providing a list of the courses they have taken while also providing their calculated GPA and the amount of credit hours they have accumulated. If the user has not added any classes to their transcript then an empty list is displayed and their GPA and Credits are displayed as 0.0 and 0</p>	
<p>xii. Other screens and maintenance functionality of this application include an About Screen that provides a general overview of the application and a delete profile function that will clear the data from the application allowing the user to start over. Both of these screens are found in the application menu of the Main Screen (top right corner)</p>	

b) Mobile Design Adherence

i. Original Design Wireframes vs Final Screens

As shown in the table below the Final design stays true to the original wireframe layouts that were outlined in the design document

CSU

LOGO

TITLE



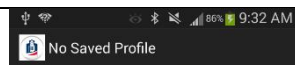
WELCOME, MESSAGE
PLEASE CREATE
PROFILE

CSU

LOGO

START

TITLE



Welcome to Cougar Advisor!
Please create a new profile



CSU Logo

TITLE

FIRST NAME

LAST NAME

LEVEL

☐ UNDERGRADUATE
 ☐ GRADUATE

CLASSIFICATION

TRACK

FRESHMAN

APPLIED

MAJOR

COMPUTER SCIENCE

STARTING TERM

STARTING YEAR

FALL

2014

CREATE PROFILE

Create Profile

First Name

Last Name

Level

☒ Undergraduate
 ☐ Graduate

Classification

Track

Freshman

Applied

Major

Computer Science

Starting Term

Starting Year

Fall

2014

CREATE PROFILE

CSU
Logo

TITLE

DELETE
PROFILE

ABOUT


WELCOME, [USER]

CPSC COURSES

GENERAL ED. COURSES

REQUIRED TRACK COURSES

TRANSCRIPT

 Menu

DELETE PROFILE

ABOUT

PROFILE SAVED.

Welcome, Ortiz Green

CPSC Courses

General Ed. Courses

Required Track Courses

Add To Transcript

View Transcript

CSU
Logo

TITLE

GO TO MENU


[UNDERGRADUATE/GRADUATE CPSC COURSES]

SELECT COURSE TO SEE DETAILS AND DESCRIPTION

CPSC COURSE

CPSC COURSE

CPSC COURSE

 Computer Science Clas...

GO TO MENU

Graduate CPSC Courses

Select a Course to see Details and Description

CPSC 5115
Algorithm Analysis and Design

CPSC 5125
Computer Graphics

CPSC 5135
Programming Languages

CPSC 5138

CSU
Logo

TITLE

GO TO MENU

ESSENTIAL SKILLS

INSTITUTIONAL OPTIONS

HUMANITIES/FINE ARTS/ETHICS

SCIENCE/MATH/TECHNOLOGY

SOCIAL SCIENCES

 General Ed. Courses

GO TO MENU

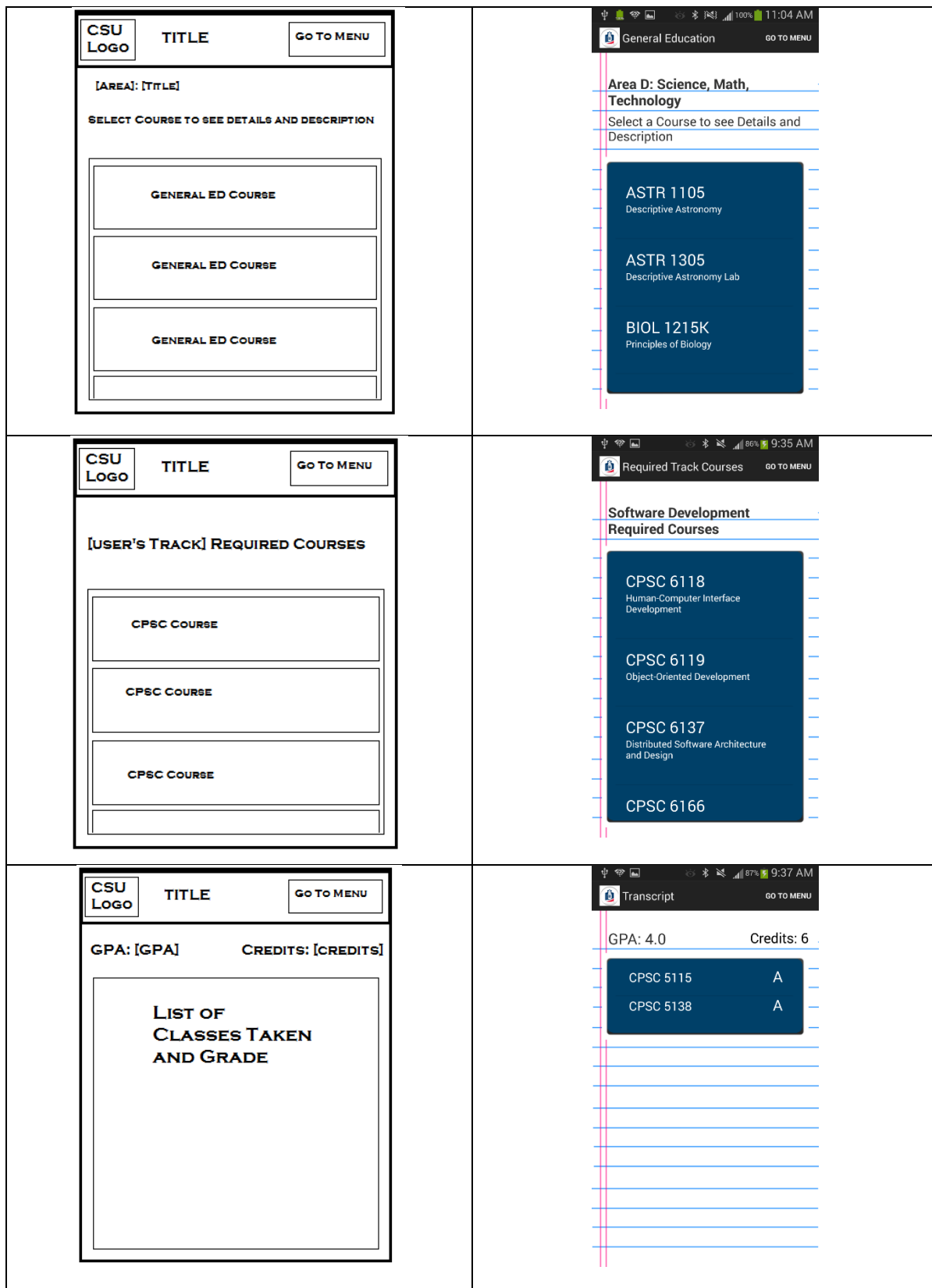
Essential Skills
Required Hours: 9

Institutional Options
Required Hours: 4

Humanities/Fine Arts/Ethics
Required Hours: 6

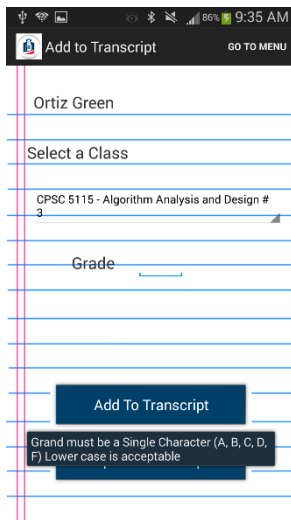
Science/Math/Technology
Required Hours: 11

Social Sciences
Required Hours: 12



ii. Additional Screens not included in Design

One additional screen was added that was missed in the original design document and that is the “Add to Transcript” Screen shown in the screen shot below.



iii. Background and Color Scheme

The background and Colors for all screens, buttons, text, and lists were configured to match all screens from the original design document. This was accomplished by creating images with the selected colors and/or design and using them as the background for all lists, screens, and buttons.

c) Mobile Requirements Adherence

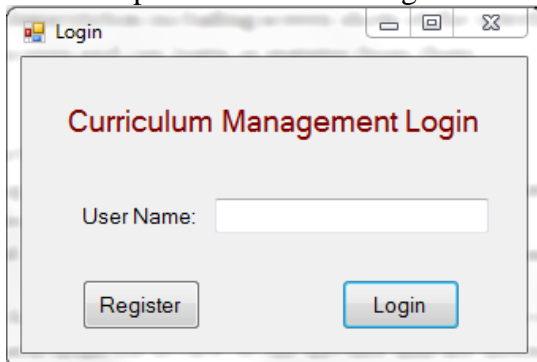
Requirement	Met by implementation of
2. Functional Requirements c. Mobile Application i. Allows users to see curriculum requirements ii. Allows users to see courses they have taken	i. “Required Track Course” and “General Ed. Courses” allow the user to see their selected track’s required Courses (Undergraduate and Graduate) as well as the General Education Course required (Undergraduate Only) ii. “Add to Transcript” and “View Transcript” allowing the user to add taken courses to their transcript as well as view their transcript which lists all courses they have completed.
3. Performance Requirements a. Scalability <ul style="list-style-type: none"> i. System should be able to handle multiple departments to spread school wide ii. System should be able to handle up to 10,000 users and advisors b. Workload <ul style="list-style-type: none"> i. System should be able to keep work queued during peak work load if backed up ii. System should use transactions to allow rollback in case of overload crashing 	a. Scalability <ul style="list-style-type: none"> i. Systems is scalable to multiple departments as all functionality would work as is and to expand to other departments only additional database entries would be required ii. N/A as mobile version is an application that is self-contained on users mobile device b. N/A to Mobile Application c. N/A as Mobile Application requirements were revised and Mobile App is now a native app and not a mobile browser application

<p>c. Platform</p> <ul style="list-style-type: none"> i. Desktop application should support windows platform ii. Web application should be supported by all browsers iii. Mobile application should be supported by all mobile device browsers <p>d. Response Time</p> <ul style="list-style-type: none"> i. Users should not see an impact due to workload ii. Database connections should be kept short and avoid excess information iii. 95% of all responses should remain under 8 seconds 	<p>d. N/A to Mobile Application</p>
<p>4. Functional Requirements</p> <ul style="list-style-type: none"> a. User must have access to database from desktop, web, and mobile application b. User must have profile created and advisor assigned c. User must be able to choose a curriculum for Computer Science d. User must be able to add classes to track their progress in the curriculum e. User must be able submit schedule of classes for advisors approval f. Advisor must be able to approve/adjust students classes request 	<ul style="list-style-type: none"> a. Database is built into the mobile application via sqlite so user will always have access to data. b. Requirement is met during the profile create process. (No advisor is assigned as mobile application is only for student to view curriculum and Transcript information) c. Requirement is met during the profile create process with the Track dropdown list d. Requirement is met by the add to transcript functionality e. N/A for Mobile Application f. N/A as Mobile Application is for student use only.
<p>5. Interface Requirements</p> <ul style="list-style-type: none"> a. Interface must provide methods for all functional requirements b. Interface must be flexible and dynamic c. Interface must provide fluid design to transition between features and functions 	<ul style="list-style-type: none"> a. Met by the successful implementation of the “Required Track Courses”, “General Ed. Courses”, and “View Transcript” sections of the application b. Interface displays data as it pertains to the Track the user selected c. Interface is fluid and transitions seamlessly as a typical mobile application would.

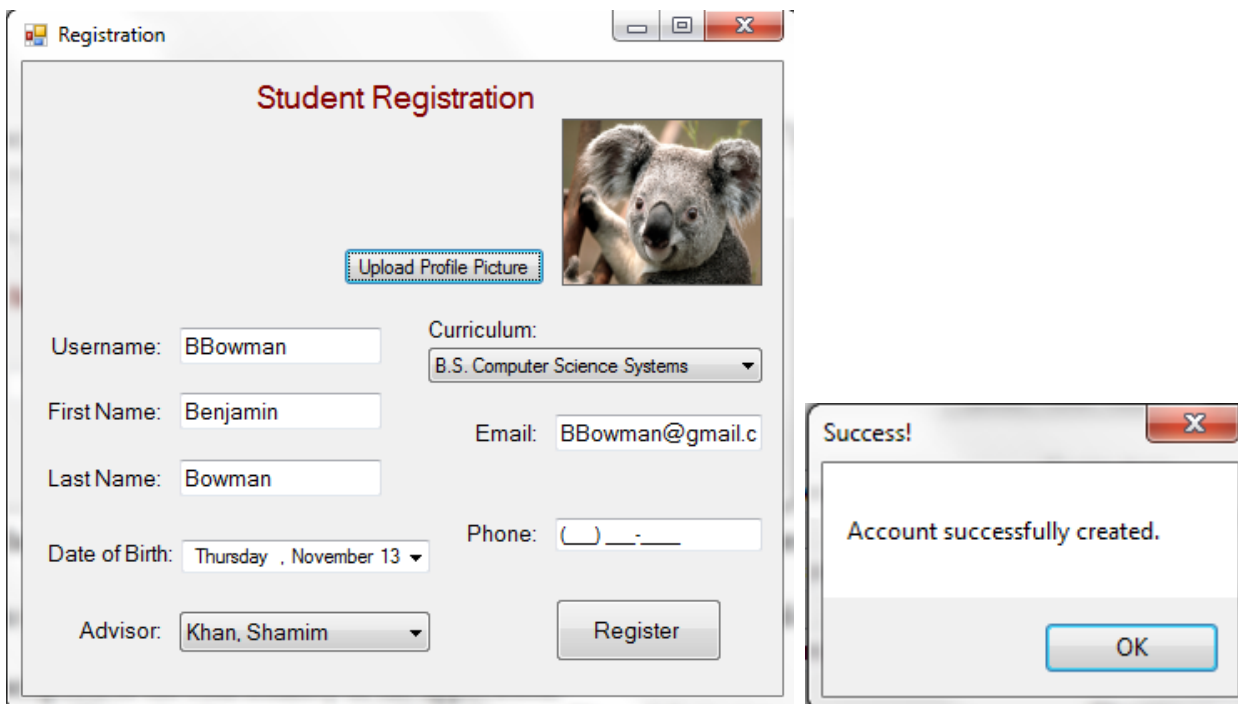
3. Desktop Implementation

a) Implementation

Users are presented with the login screen and can login or register from there.



When clicking the register button users can create their profile and upload a picture.



When logging in users are presented with their curriculum and currently registered classes. This is also accessible through curriculum -> view curriculum.

User Details

File Profile **Curriculum**

Current Curriculum

B.S. Computer Science Systems

■ - Indicates class has been registered

CRN	Name	Pre-Requisites	Credits
ENGL 1101	English Composition 1		3
ENGL 1102	English Composition 2	ENGL 1101	3
MATH 1113	Pre-Calculus	Math 1111	4
COMM 1110	Communication		3
CPSC 1301	Computer Science 1		3
CPSC 1301L	Computer Science 1 Lab		1
CPSC 1302	Computer Science 2	CPSC 1301	3
CPSC 2105	Computer Organization	CPSC 1301	3
CPSC 2108	Data Structures	CPSC 1302	3
MATH 1131	Calculus with Analytic Geometry 1	MATH 1113	4
CPSC 3121	Assembly Language Programming 1	CPSC 2105	3
CPSC 3125	Operating Systems	CPSC 2108	3
CPSC 3131	Database Systems 1	CPSC 1302	3
CPSC 3165	Professionalism In Computing		3

Once logged in user can logout or exit to be returned to the login screen. User can select profile -> edit to be presented with the following profile edit screen; clicking cancel will refresh all the information while clicking update will commit it to the database.

User Details

File Profile **Curriculum**

Edit Profile


Username:

First Name:

Last Name:

Date of Birth:

Advisor: Khan, Shamim

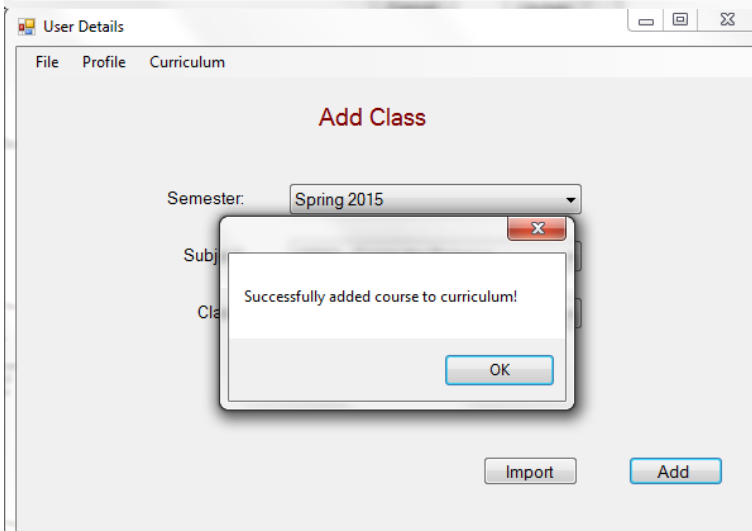
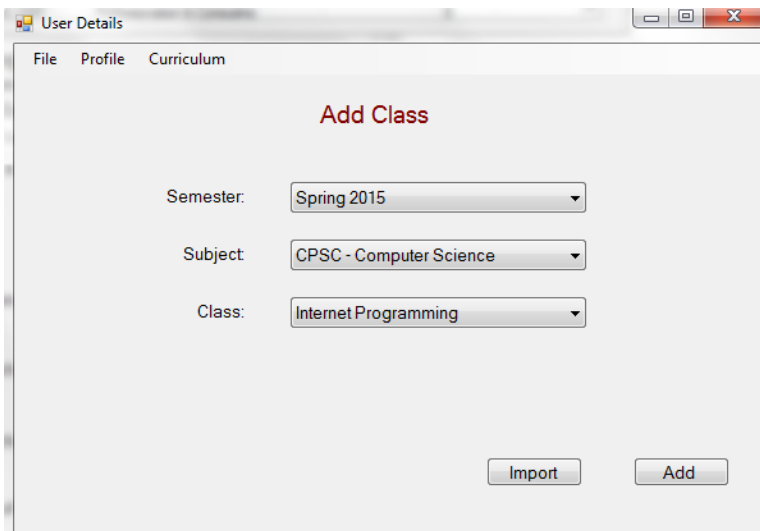


Curriculum:

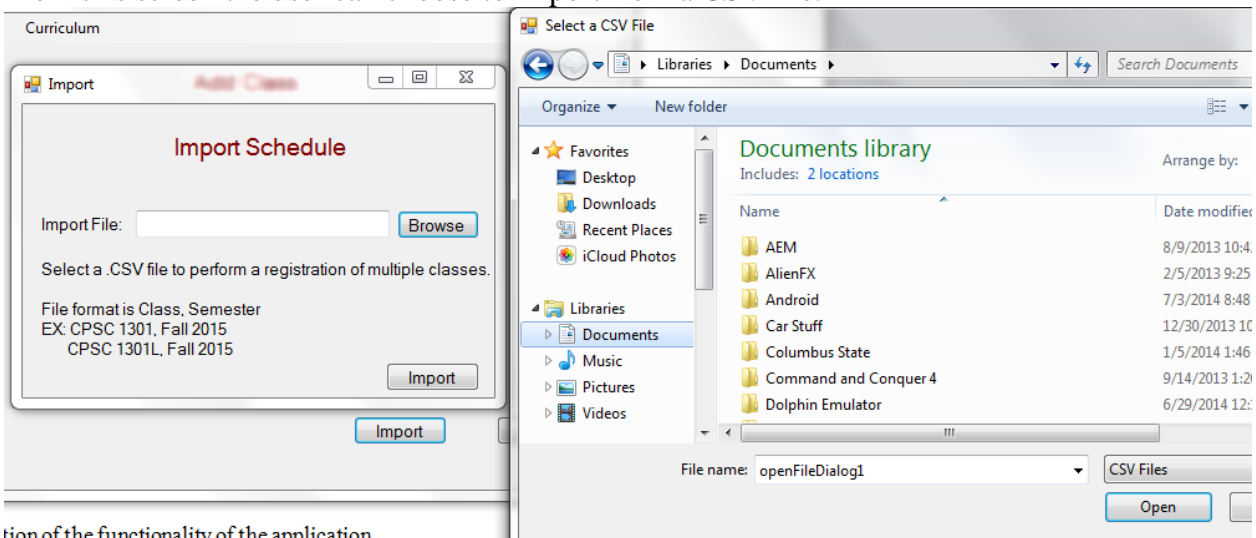
Email:

Phone:

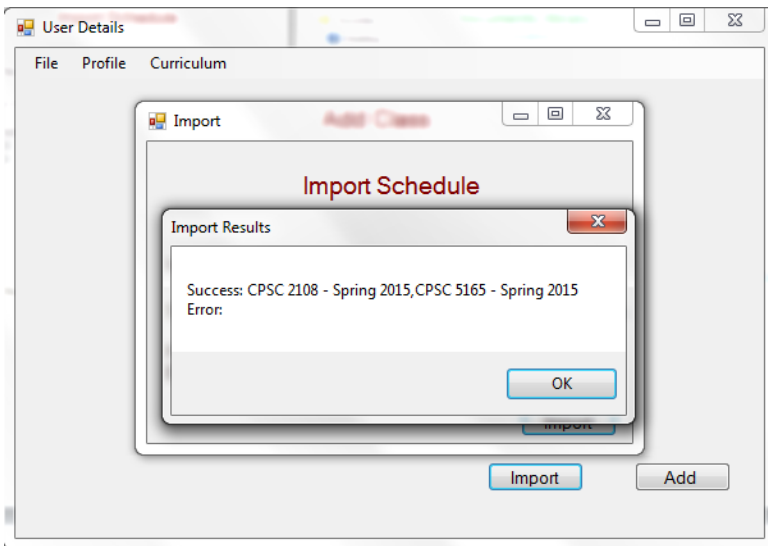
The curriculum menu gives the user the ability to add, drop, view and submit their semester of classes.



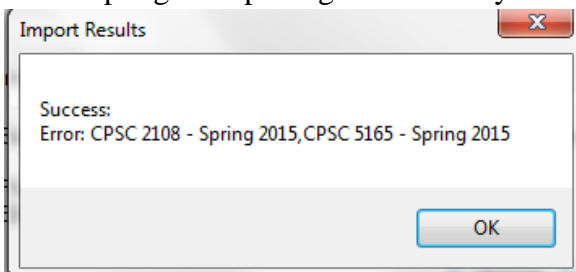
From this screen the user can choose to import from a CSV file.



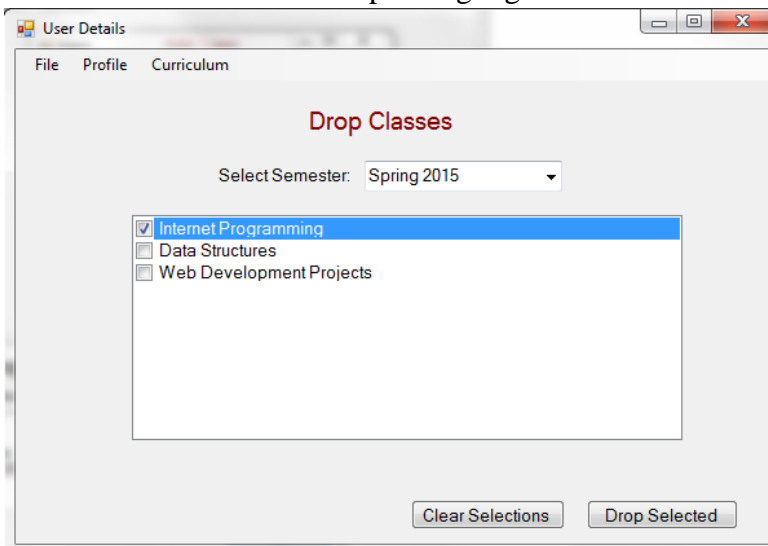
tion of the functionality of the application.



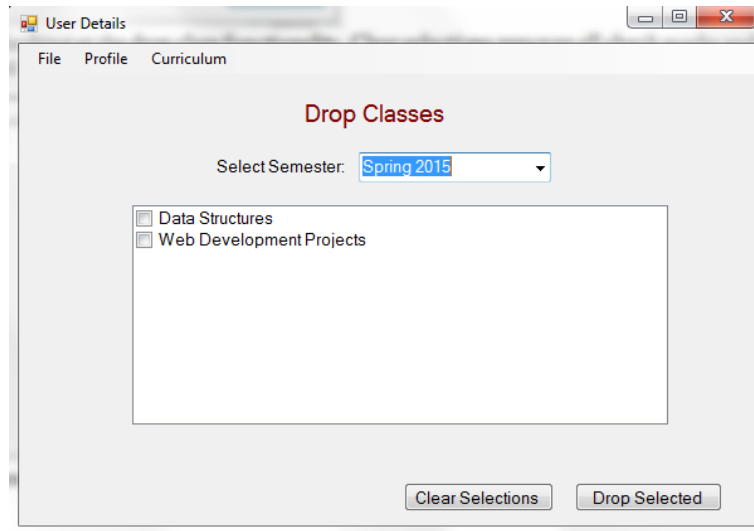
If attempting to import again since they are already registered the following error is displayed.



When looking at the drop class functionality. Clear selections removes all check marks and drop selected will remove the classes from the pending registration table.

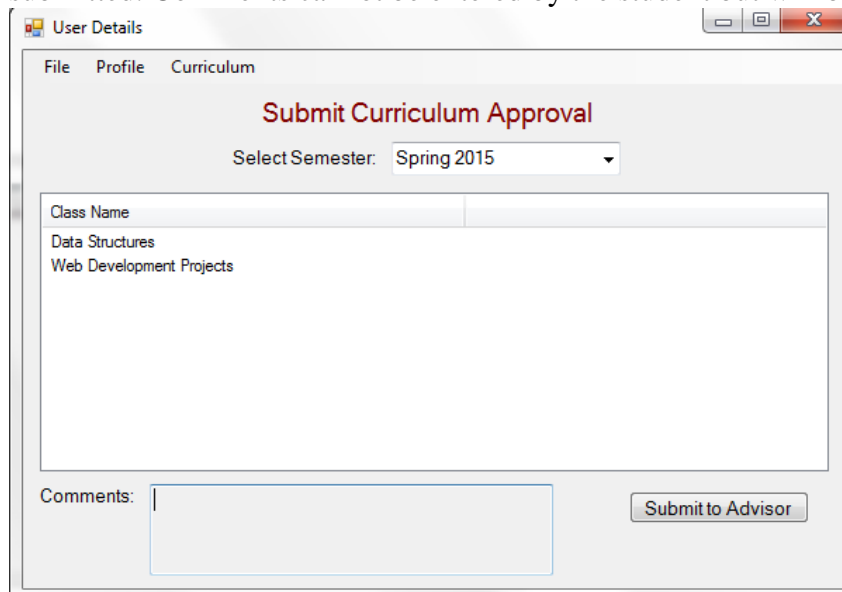


If successful the screen will refresh and when selected the intended semester the class should be gone.



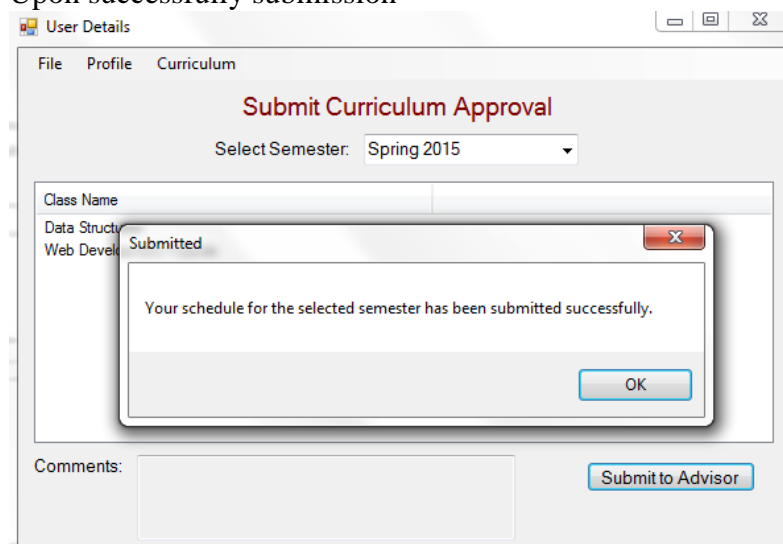
The screenshot shows a window titled "User Details" with a tabbed interface. The "Curriculum" tab is active, displaying a section titled "Drop Classes". Below the title, there is a "Select Semester:" label followed by a dropdown menu showing "Spring 2015". Underneath, there is a list of classes with checkboxes: "Data Structures" and "Web Development Projects". At the bottom of the window, there are two buttons: "Clear Selections" and "Drop Selected".

When submitting the curriculum semester you must select a semester to display the classes the below to be submitted. Comments cannot be entered by the student but will be display if returned by the advisor.



The screenshot shows a window titled "User Details" with a tabbed interface. The "Curriculum" tab is active, displaying a section titled "Submit Curriculum Approval". Below the title, there is a "Select Semester:" label followed by a dropdown menu showing "Spring 2015". Underneath, there is a list of classes with checkboxes: "Data Structures" and "Web Development Projects". At the bottom of the window, there is a "Comments:" label followed by a text input field and a "Submit to Advisor" button.

Upon successfully submission



The screenshot shows the same "Submit Curriculum Approval" window as before, but with a success message dialog box overlaid. The dialog box is titled "Submitted" and contains the text: "Your schedule for the selected semester has been submitted successfully." Below the text is an "OK" button. The background window remains visible, showing the "Submit Curriculum Approval" section with the "Spring 2015" semester selected and the "Data Structures" and "Web Development Projects" classes listed.

When the advisor logs in to the system they are displayed with the approval screen

User Details

File Advisor

Approve Student Schedule

Select a student: Bowman, Benjamin Semester: Spring 2015

Class
Data Structures
Web Development Projects

Comments: (Only if not approved)

Return to student Approve

User Details

File Advisor

Approve Student Schedule

Select a student: Bowman, Benjamin Semester: Spring 2015

Class
Data Structures
Web Development Projects

Comments: (Only if not approved) This is a test

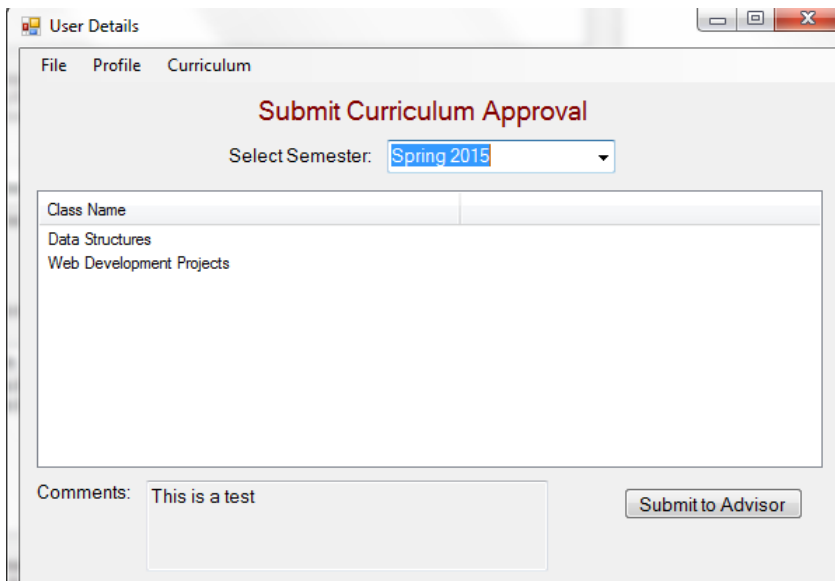
Return to student Approve

Success!

Schedule was successfully returned to the student.

OK

The student will now see the curriculum returned with notes and will have to resubmit. Which will clear out the comments.



User Details

File Profile Curriculum

Submit Curriculum Approval

Select Semester: Spring 2015

Class Name

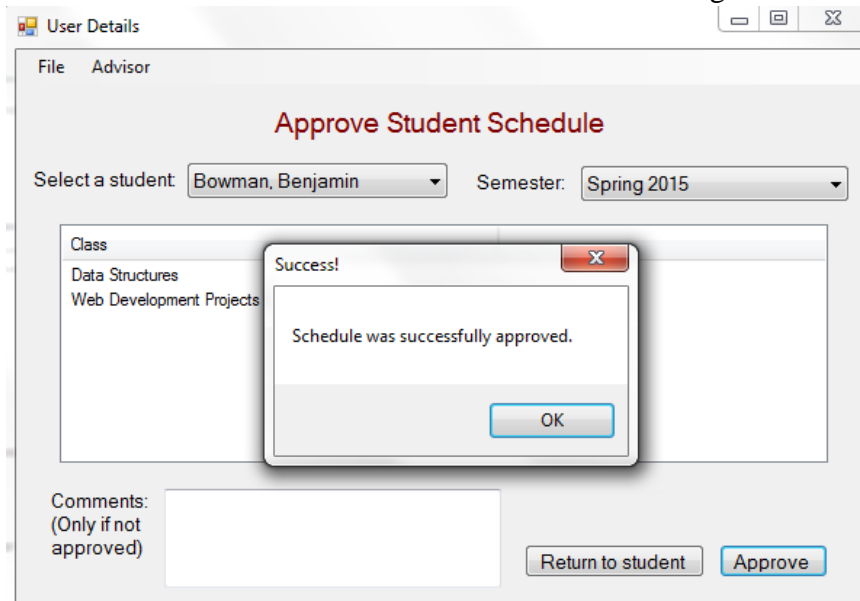
Data Structures

Web Development Projects

Comments: This is a test

Submit to Advisor

Now the user has corrected his issues or made his changes the advisor will accept.



User Details

File Advisor

Approve Student Schedule

Select a student: Bowman, Benjamin Semester: Spring 2015

Class

Data Structures

Web Development Projects

Success!

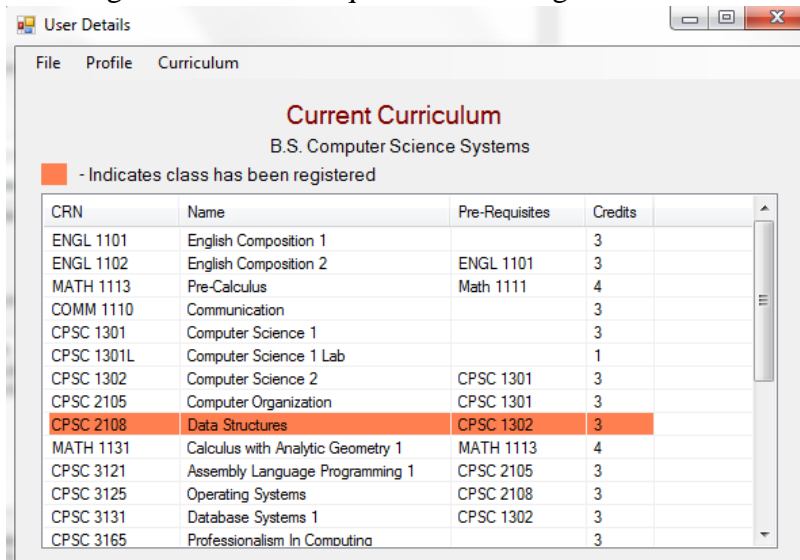
Schedule was successfully approved.

OK

Comments: (Only if not approved)

Return to student Approve

If the registered class is required for the degree it will be on the students curriculum view when logging back in.



User Details

File Profile Curriculum

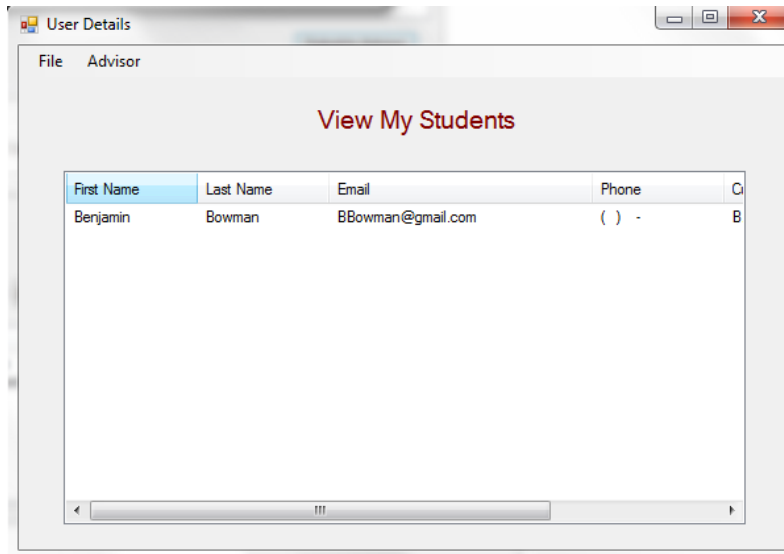
Current Curriculum

B.S. Computer Science Systems

■ - Indicates class has been registered

CRN	Name	Pre-Requisites	Credits
ENGL 1101	English Composition 1		3
ENGL 1102	English Composition 2	ENGL 1101	3
MATH 1113	Pre-Calculus	Math 1111	4
COMM 1110	Communication		3
CPSC 1301	Computer Science 1		3
CPSC 1301L	Computer Science 1 Lab		1
CPSC 1302	Computer Science 2	CPSC 1301	3
CPSC 2105	Computer Organization	CPSC 1301	3
CPSC 2108	Data Structures	CPSC 1302	3
MATH 1131	Calculus with Analytic Geometry 1	MATH 1113	4
CPSC 3121	Assembly Language Programming 1	CPSC 2105	3
CPSC 3125	Operating Systems	CPSC 2108	3
CPSC 3131	Database Systems 1	CPSC 1302	3
CPSC 3165	Professionalism In Computing		3

Advisors can choose the advisor->view students



b) **A detailed description of the functionality of the application.**

Users can create a profile, uploading a picture and edit their profile and their picture.

Comma Separated Value (CSV) files can be imported to bulk add classes.

Students can pull up their registered curriculum and see a highlighted row for classes they are registered for.

Students can view required classes for their curriculum, drop and add classes individually.

Students can finally submit their curriculum for review to the advisor and see notes upon return.

Advisors can send a schedule back to the student with notes or approve the curriculum which will highlight on the students schedule.

Advisors can see students that are assigned to them.

c) **How the implementation adheres to the previously submitted design.**

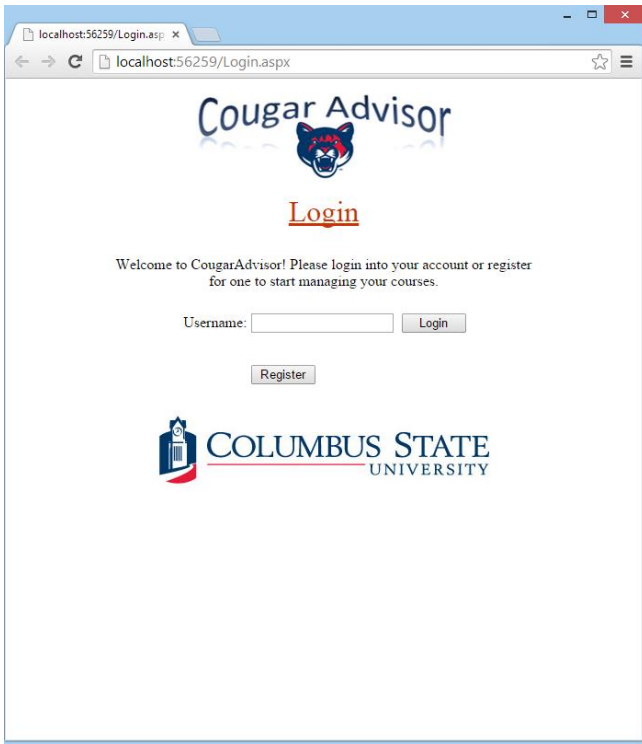
The design screenshots submitted were used with minor modifications, utilizing the standard controls stated in the design and following a pattern of displaying information. Buttons, Grids, Text Boxes, File Dialog, and Message boxes were used for direct manipulation

d) **How the implementation meets the previously submitted requirements definition.**

Requirements were to provide functionality stated above. The requirements were implemented through stored procedure execution from the database. The detailed description of functionality covers the requirements definition

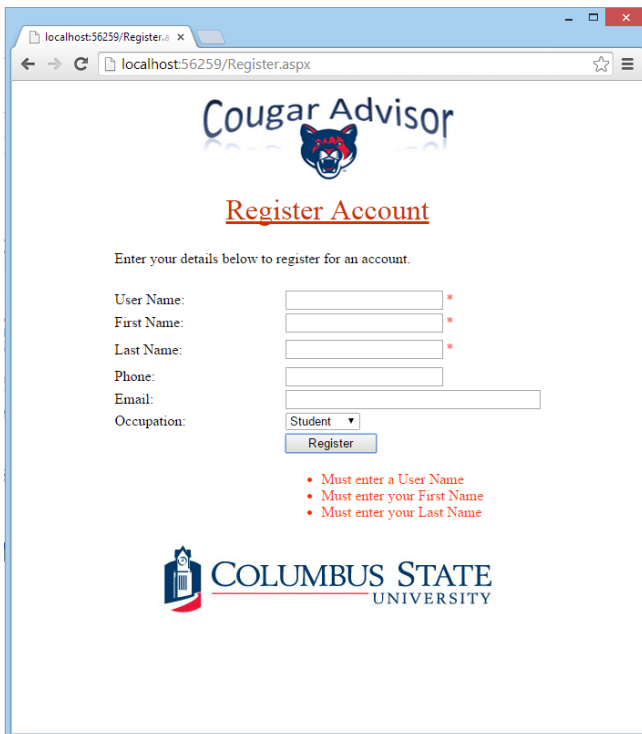
4. Web Implementation

a) A detailed description of the implementation including screen shots of the interfaces



Login.aspx

The first page they will see is the login page. Here they can register for a new account or login into their account. If they type in an incorrect username, they will receive an error.



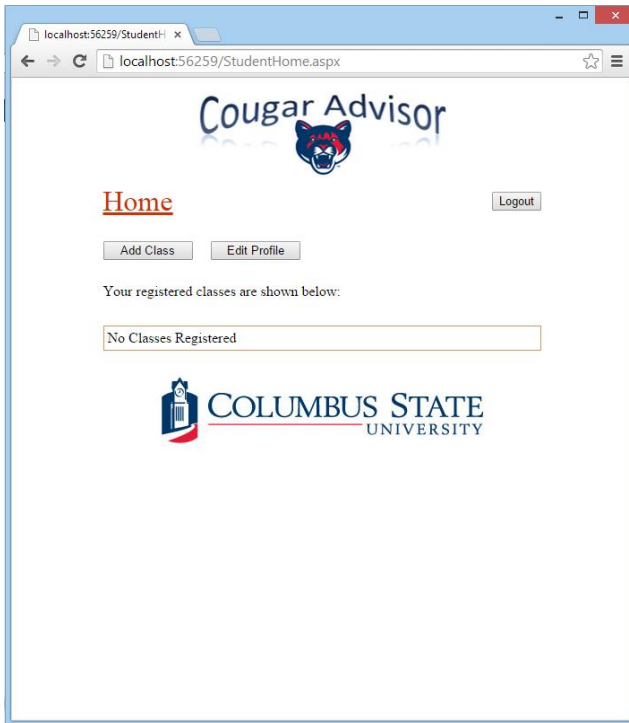
Register.aspx

This page appears when the user clicks on the register button in the *Login.aspx* page. The user can create an account by typing their details.

User Name, First Name and Last Name are all required fields so if they do not enter these details, they cannot proceed.

There are two types of accounts; student and instructor.

The application features will be different depending on what account the user creates.



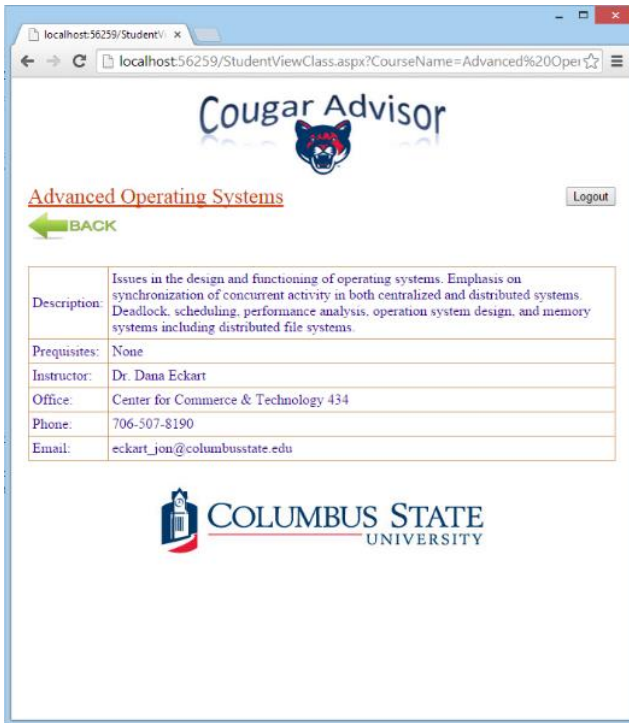
StudentHome.aspx (Without classes)

Student accounts will see this home page when they login. Here they can see the classes they registered for and they can also add more classes, edit their profile and logout.



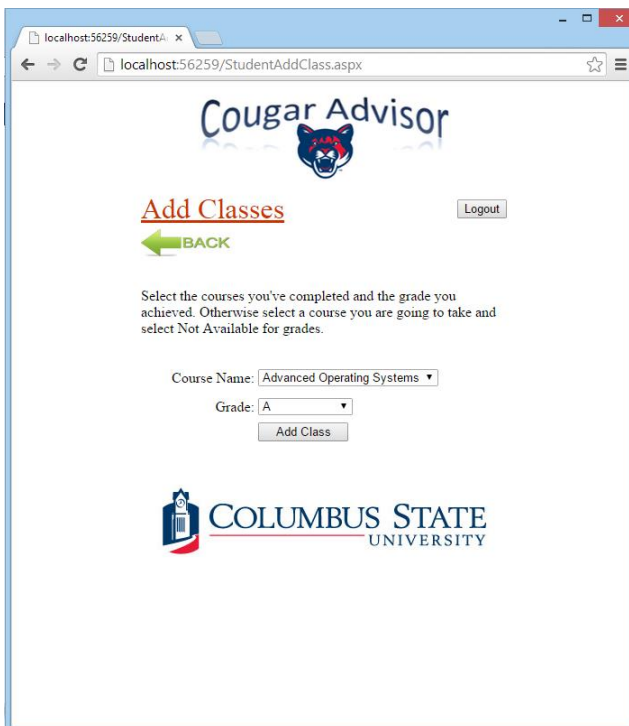
StudentHome.aspx (With classes)

The student's registered classes are shown in a GridView. They can click on the trash icon to drop that class and they can click on the magnifying glass icon to get more details on the course.



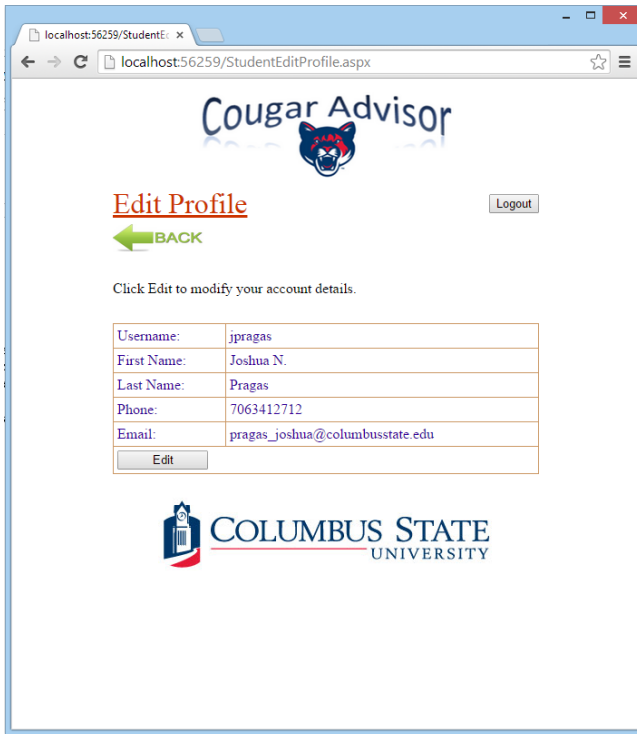
StudentViewClass.aspx

When students click on the magnifying class they are redirected to this page that contains details for that specific course. They can return to the home page by clicking the “Cougar Advisor” banner or the green back button.



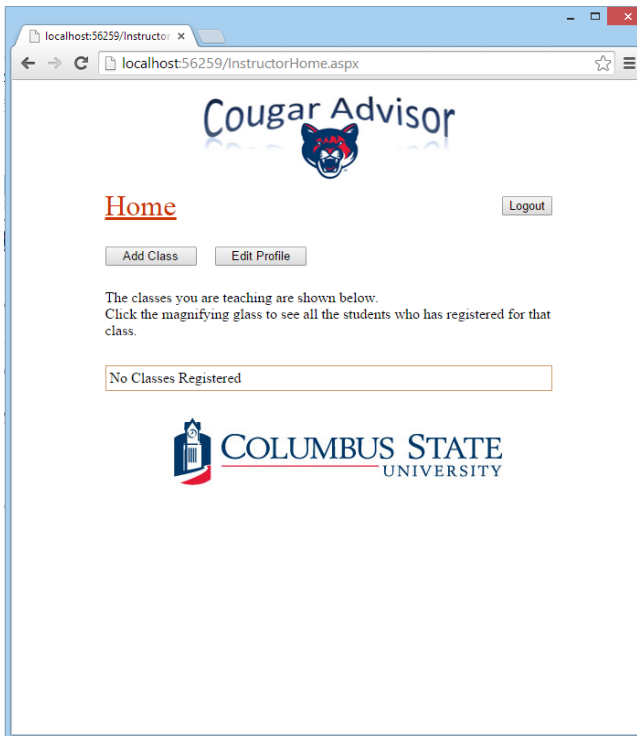
StudentAddClass.aspx

Students can add classes by selecting a course from the drop down list. They can also select their grade if they have already completed their class. They can return to the home page by clicking the “Cougar Advisor” banner or the green back button.



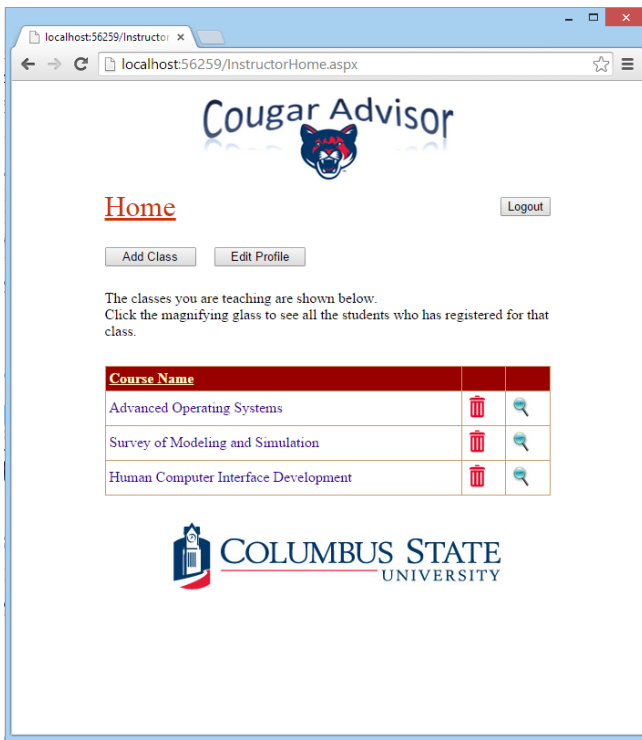
StudentEditProfile.aspx

Students can edit their profile by clicking on the Edit button. They can return to the home page by clicking the “Cougar Advisor” banner or the green back button.



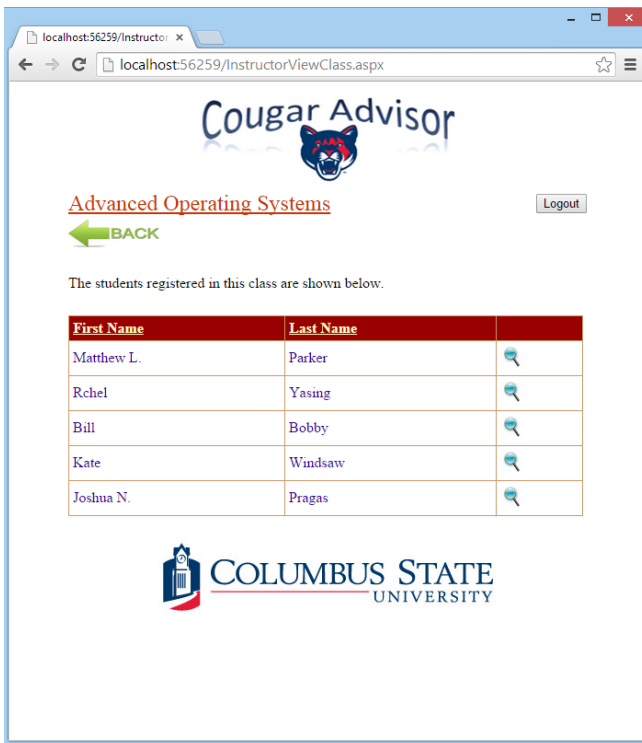
InstructorHome.aspx (Without classes)

Instructor accounts will see this home page when they login. Here they can see the classes they are teaching and they can also add more classes, edit their profile and logout.



InstructorHome.aspx (With classes)

The instructor's registered classes are shown in a GridView. They can click on the trash icon to delete that class and they can click on the magnifying glass icon to view which students are taking that class.



InstructorViewClass.aspx

The instructor will be redirected to this page when they click on the magnifying glass icon in the *InstructorHome.aspx* page. Here they can see a list of students taking that class and they can get their details by clicking on the magnifying glass icon. They can return to the home page by clicking the "Cougar Advisor" banner or the green back button.

localhost:56259/InstructorViewStudent.aspx

Cougar Advisor

[Student Details](#) [Logout](#)

[← BACK](#)

First Name:	Matthew L.
Last Name:	Parker
Phone:	7062689715
Email:	parker_mathew@columbusstate.edu

This student is currently registered in the following courses:

Course Name	Grade
Advanced Operating Systems	C
Survey of Modeling and Simulation	A
Professionalism In Computing	Not Available
Artificial Intelligence	A
Robotics Programming I	C

COLUMBUS STATE UNIVERSITY

InstructorViewStudent.aspx

The instructor will be redirected to this page when they click on the magnifying glass icon in the *InstructorViewClass.aspx* page. Here they can see the details for a specific student such as their name phone number, email and the classes they are taking along with their grades.

They can return to the home page by clicking the “Cougar Advisor” banner or return to the *InstructorViewClass.aspx* page by clicking the green back button.

localhost:56259/InstructorAddClass.aspx

Cougar Advisor

[Add Classes](#) [Logout](#)

[← BACK](#)

Select the courses you are teaching below.

Course Name:

[Add Class](#)

COLUMBUS STATE UNIVERSITY

InstructorAddClass.aspx

Instructors can add the classes they will be teaching by selecting a course from the drop down list.

They can return to the home page by clicking the “Cougar Advisor” banner or the green back button.



InstructorEditProfile.aspx

Instructors can edit their profile by clicking on the Edit button. They can return to the home page by clicking the “Cougar Advisor” banner or the green back button.

b) A detailed description of the functionality of the application.

Users can create their account – student/instructor.

When a student logs in they can add classes or edit their profile.

They can drop their classes and view detailed descriptions for that specific class.

When an instructor logs in they can add classes they are teaching and edit their profile.

They can delete any classes they are teaching.

They can view which students are taking that class and view the student’s details.

The ‘Cougar Advisor’ logo will always redirect the user to their home page.

Excluding the login, register and home pages there will always be a green back arrow button to allow users to return to the previous page

c) How the implementation adheres to the previously submitted design.

The design has changed quite a bit in areas of how it functions and its interface. The interface now includes metaphors such as the magnifying glass icon and trash icon instead of ‘Delete’ and ‘View’ buttons. The CSU banner is now always at the bottom and the ‘Cougar Advisor’ logo is located at the top. There is also a green back arrow button for users to return to the previous page. With regards to the functionality, users cannot upload a picture to their profile and students cannot upload a list of their classes. They can only add them one at a time. Instructors can only view who’s taking their class and see the student’s details but not approve them.

d) How the implementation meets the previously submitted requirements definition.

Functional Requirements	
<ul style="list-style-type: none"> i. Allows users to create/edit their profile ii. Allows users to see and edit curriculum requirements iii. Allows users to add classes they have taken, individually iv. Allows users to submit their schedule to the advisor for approval or adjustments v. Allows advisors to manage their students vi. Allows advisors to approve or adjust students submitted schedules 	<ul style="list-style-type: none"> i. Students and instructors can create an account and edit their profile if they wish. ii. Students can see what prerequisites are required to take the course. iii. Students are allowed to add classes they have completed or are going to take. iv. Instructors can see what classes students are taking. v. Instructors can see who are taking their class and view their student details. vi. Instructors can only view and read students schedules.
Performance Requirements	
<ul style="list-style-type: none"> a. Scalability <ul style="list-style-type: none"> i. System should be able to handle multiple departments to spread school wide ii. System should be able to handle up to 10,000 users and advisors b. Workload <ul style="list-style-type: none"> i. System should be able to keep work queued during peak work load if backed up ii. System should use transactions to allow rollback in case of overload crashing c. Platform <ul style="list-style-type: none"> i. Web application should be supported by all browsers d. Response Time <ul style="list-style-type: none"> i. Users should not see an impact due to workload ii. Database connections should be kept short and avoid excess information iii. 95% of all responses should remain under 8 seconds 	<ul style="list-style-type: none"> a. Scalability <ul style="list-style-type: none"> i. Currently only Computer Science courses are in the list of courses available. More classes can be included in the database if required. ii. If the database table allows unlimited rows for new data then it should be able to handle it. b. Workload <ul style="list-style-type: none"> i. N/A ii. N/A c. Platform <ul style="list-style-type: none"> i. It currently works on Mozilla Firefox and Google chrome. d. Response Time <ul style="list-style-type: none"> i. There can be a small delay when logging in to your account. ii. There are few columns in each table and only the necessary information is displayed. iii. Nothing takes more than 8 seconds to respond.
Interface Requirements	

<ul style="list-style-type: none"> a. Interface must provide methods for all functional requirements b. Interface must be flexible and dynamic c. Interface must provide fluid design to transition between features and functions 	<ul style="list-style-type: none"> a. Interface provides methods for most of the functional requirements. b. It works and is s viewable in popular web browsers such as Firefox and Chrome. c. There are small instructions on each page to help the user. The interface uses nice images, colors and buttons to make it a visually pleasing interface to work with.
---	---