

## Table of Contents

1. INTRODUCTION, PROJECT OVERVIEW AND READ ME .....	4
2. OBJECTIVES .....	6
2.1 BUSINESS OBJECTIVES.....	6
2.2 SYSTEM OBJECTIVES. ....	6
3. PROJECT CONTEXT DIAGRAM .....	7
4. SYSTEM REQUIREMENTS .....	8
4.1 ADD USER.....	8
4.2 LOG IN .....	9
4.3 CREATE SCHEDULE .....	9
4.4 ADD COURSE .....	11
4.5 VERIFY SCHEDULE .....	12
4.6 SAVE SCHEDULE.....	13
4.7 SELECT SCHEDULE .....	14
4.8 MODIFY SCHEDULE .....	15
4.9 REMOVE COURSE.....	16
4.10 LOGOUT .....	17
4.11 EDIT/DELETE USER .....	18
5. SOFTWARE PROCESSES AND INFRASTRUCTURE.....	21
5.1 HARDWARE AND INFRASTRUCTURE .....	21
5.2 UML DIAGRAM .....	22
5.2.1 STATE DIAGRAMS.....	22
LOGIN .....	23
ADD/EDIT/DELETE USER .....	24
ADD COURSE.....	25
REMOVE COURSE .....	26
CREATE/DELETE/MODIFY SCHEDULE .....	27

## **Table of Contents (Continued)**

<b>5.3 TEST PLAN...</b>	<b>28</b>
<b>6. ASSUMPTIONS AND CONSTRAINTS .....</b>	<b>35</b>
<b>6.1 ASSUMPTIONS.....</b>	<b>35</b>
<b>6.2 CONSTRAINTS.....</b>	<b>35</b>
<b>6.3 OUT OF SCOPE MATERIAL.....</b>	<b>35</b>
<b>7. USER MANUAL .....</b>	<b>36</b>
<b>8. DELIVERY AND SCHEDULE .....</b>	<b>45</b>
<b>9. STAKEHOLDER APPROVAL FORM .....</b>	<b>46</b>

## **1. INTRODUCTION, PROJECT OVERVIEW AND README**

**Mostafa Galib**

Email: mostafa.galib@mavs.uta.edu

Phone: 503-928-9537

**Matthew Zielke**

Email: matthew.zielke@mavs.uta.edu

Phone: 817-583-4537

**Hani Jouya**

Email: hani.jouya@mavs.uta.edu

Phone: 817-663-3609

**Readme:**

**To log in, please enter username as “admin” and password as “admin” (both without quotations).**

An Android application will be developed that will help create a class schedule. The application shall have the following minimum functionalities:

1. Login
  - User authentication (username/password)
  - Add user
  - Edit user
  - Remove user
2. Obtain open classes from web
3. Create class schedule
  - Class schedule should have no conflicts--no classes may overlap
  - User should be notified if classes conflict
  - Remember, the university now has Saturday classes
4. Users should be able to block out time on a schedule
  - Classes must be arranged around any blocked out time
  - If a set of classes conflicts with the some blocked out time, user should be notified
5. Users should be able to store a class schedule
  - Stored schedules must be reverified to make sure all classes are still available
6. More than one schedule may work. Users should be given several schedules to choose from.
7. Phone resource management: the app should not kill a user's phone, battery, or usage plan

## **2. OBJECTIVES**

### **2.1 BUSINESS OBJECTIVES**

**Objective 1:** Add User: First-time users must be able to create an account. The user should provide the following information:

- Full Name
- Email Address
- User name
- Password

**Objective 2:** Login functionality: User must be able to login to the account using user name and password.

**Objective 3:** Search functionality: User must be able to look up for courses.

**Objective 4:** Add/Remove Course functionalities: User must be able to add/remove courses.

**Objective 5:** Create/Save/Verify/Modify Schedule functionalities: User must be able to create, save, verify, and modify schedule.

**Objective 6:** Logout functionality: User must be able to logout of the account anytime the user wants to.

### **2.2 SYSTEM OBJECTIVES**

The following is a list of system objectives:

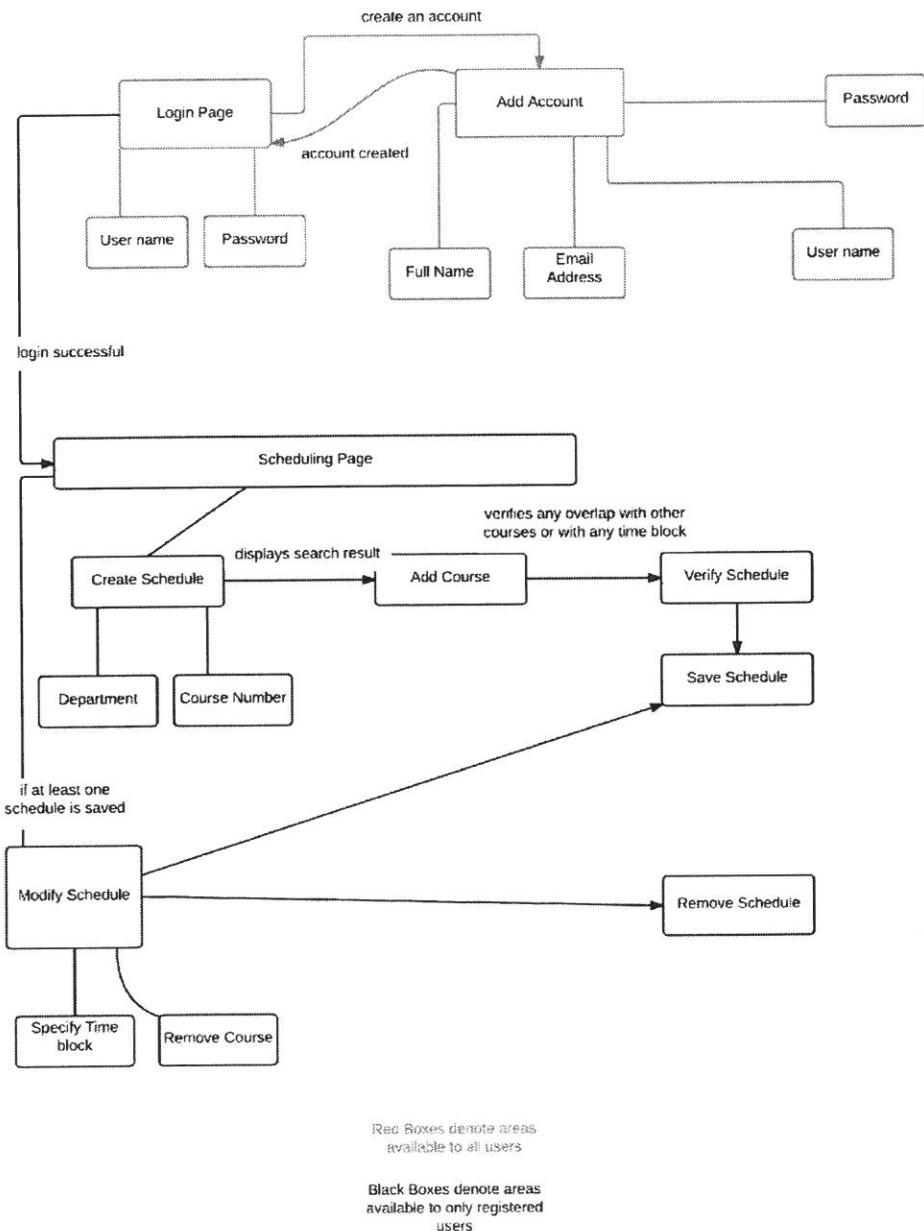
**Objective 1:** System will be Android-based.

**Objective 2:** System will run on Android 2.33 or higher for greatest compatibility.

**Objective 3:** MySQL will be used for database purposes.

**Objective 4:** System will retrieve course data from the UTA website

### 3. PROJECT CONTEXT DIAGRAM



## 4. REQUIREMENTS

<b>Requirement Title:</b> (*required)	Add User
<b>Sequence No:</b> (*required)	1
<b>Short description:</b> (*required)	Allows the user to add a user in the application
<b>Detailed Description:</b> (*required)	<p>1. The application shall show all saved users as a list as well as an add user button. If no users are added yet it will just show the add user button.</p> <p>1.1 When the user taps on the add user button, the application shall show a new screen with text fields for user to write in a user name and two identical passwords to add a user.</p> <p>1.2 Application will check the data entered by user for validity.</p> <p>1.2.1. If the application detects invalid user name and/or password, or an already existing user name, then the application shall not allow a user to be added.</p> <p>1.2.2. If the application detects valid input, it shall compare both passwords entered to be sure they are identical.</p> <p>1.2.2.1. If the passwords are not identical then an error message will be shown to the user and the application will return to the add user screen.</p> <p>1.2.2.2 If the passwords are identical then the user will be added</p>
<b>Pre-Conditions:</b> (optional)	

<b>Post Conditions:</b>  (optional)	
<b>Other attributes:</b>  (optional)	<ul style="list-style-type: none"> <li>- User name and password are stored on device.</li> <li>- User name shall be a minimum of 4 characters and not exceed 12 characters and shall be comprised of upper and lower case alphanumeric characters.</li> <li>- Passwords shall be a minimum of 8 characters upper and lower case alphanumeric characters.</li> </ul>

<b>Requirement Title:</b>  (*required)	Log in
<b>Sequence No:</b>  (*required)	2
<b>Short description:</b>  (*required)	User logs in
<b>Detailed Description:</b>  (*required)	<p>2. Application gets the data from the user interface</p> <p>2.1 Application shall check the data for validity</p> <p>2.1.1 If the application detects an empty user name and/or password, the application shall not allow a login and require a login with valid data</p> <p>2.1.2 If the application detects invalid characters, the application shall now allow a login and require a login with valid data</p> <p>2.1.3 If the application detects valid input, the application shall compare the inputted user name and password with those stored in the system</p> <p>2.1.3.1 If the user name and password match a pre-existing user,</p>

	the application shall display the main application interface
	2.1.3.2 If either the user name or password do not match a pre-existing user, the application shall not allow a login and require a login with valid data
<b>Pre-Conditions:</b>  (optional)	- There must be a pre-existing valid user in the system
<b>Post Conditions:</b>  (optional)	
<b>Other attributes:</b>  (optional)	<ul style="list-style-type: none"> <li>- User name and password are stored on device.</li> <li>- User name shall be a minimum of 4 characters and not exceed 12 characters and shall be comprised of upper and lower case alphanumeric characters.</li> <li>- Passwords shall be a minimum of 8 characters upper and lower case alphanumeric characters.</li> </ul>

<b>Requirement Title:</b>  (*required)	Create a Schedule
<b>Sequence No:</b>  (*required)	3
<b>Short description:</b>  (*required)	Allows the user to create a schedule.
<b>Detailed Description:</b>  (*required)	<p>3. Application shall provide a button to create a new schedule.</p> <p>3.1. This button shall navigate the user to a new screen to create a schedule.</p> <p>3.1.1. Application shall provide a department text field and a course number text field for course selection.</p>

	3.1.1.1. Application shall check the data for validity
	<p>If the application detects an empty course number or department, the application shall not allow course selection and display an error message.</p> <p>If the application detects an invalid input, the application shall not allow course selection and display an error message</p> <p>If the application detects valid input, it shall compare the input with the courses contained in the database.</p> <p>If the input does not match a course in the database, the application shall display a message that the course does not exist.</p> <p>If the input matches a course in the database, the application shall display the courses to the user.</p> <p>3.1.2. The page shall provide an option to create a new time block to prevent any classes from showing up within that time range.</p> <p>3.1.2.1. If the user adds a time block that overlaps the range of an existing time block, the application shall not add the time block and display a message stating that you cannot overlap time blocks.</p>
<b>Pre-Conditions:</b> (optional)	- There must be a network connection to get information from the database.
<b>Post Conditions:</b> (optional)	
<b>Other attributes:</b> (optional)	<ul style="list-style-type: none"> <li>- Course number shall be a 4 digit integer.</li> <li>- Department shall be a minimum of 2 alphabetic characters and shall not exceed 4 alphabetic characters.</li> </ul>

<b>Requirement Title:</b> (*required)	Add a course
<b>Sequence No:</b> (*required)	4
<b>Short description:</b> (*required)	Allows the user to add a course.
<b>Detailed Description:</b> (*required)	<p>4. Application shall show the list of all of the available courses.</p> <p>4.1. Application shall allow the user to select one of those courses.</p> <p>4.1.2. Application shall provide an “Add Course” button which the user may click on to add the course.</p> <p>4.2. Application shall then verify if the user has already met the prerequisite of the course.</p> <p>4.2.1. If the user has already met the prerequisite of the course, or if there is no prerequisite at all, then the application shall allow the course to be added into the schedule.</p> <p>4.2.2. If the user has not met the prerequisite of the course, if any, then the application shall display an error message and shall not allow the course to be added into the schedule.</p> <p>4.3. If the user selects cancel button, the application shall return to the course selection screen.</p>
<b>Pre-Conditions:</b> (optional)	-There must be a network connection to get information from the database.
<b>Post Conditions:</b> (optional)	- The course will be added into the schedule.

<b>Requirement Title:</b> (*required)	Verify a Schedule
<b>Sequence No:</b> (*required)	5.
<b>Short description:</b> (*required)	The application verifies a schedule.
<b>Detailed Description:</b> (*required)	<p>5. When saving a schedule, an application shall verify whether the proposed schedule is valid or not.</p> <p>5.1. The application shall not allow any classes that overlap within time blocks.</p> <p>5.2 The application shall not allow time blocks that overlap other time blocks.</p> <p>5.2.1. The application shall not allow a time block to be saved with an end time prior to a start time.</p>
<b>Pre-Conditions:</b> (optional)	- The user must be creating/modifying a schedule.
<b>Post Conditions:</b> (optional)	
<b>Other attributes:</b> (optional)	

<b>Requirement Title:</b> (*required)	Save a Schedule
<b>Sequence No:</b> (*required)	6
<b>Short description:</b> (*required)	Allows the user to save a schedule.
<b>Detailed Description:</b> (*required)	<p>6. The application shall provide a button to save a schedule.</p> <p>6.1. The application shall commit changes to the database.</p> <p>6.1.1. The application shall display a message stating that the changes were saved.</p>
<b>Pre-Conditions:</b> (optional)	
<b>Post Conditions:</b> (optional)	- The schedule will be added to the database.
<b>Other attributes</b> (optional)	

<b>Requirement Title:</b> (*required)	Select Schedule
<b>Sequence No:</b> (*required)	7
<b>Short description:</b> (*required)	Application allows the user to select a schedule from a list of saved schedules.
<b>Detailed Description:</b> (*required)	<p>7. The application shall show a list of saved schedules.</p> <p>7.1. The application shall allow the user to select a schedule he/she wishes to view/edit by clicking on one of them.</p> <p>7.2. When a schedule is selected, the application shall go to that schedule's modify screen.</p>
<b>Pre-Conditions:</b> (optional)	- There must already be verified saved schedules.
<b>Post Conditions:</b> (optional)	
<b>Other attributes:</b> (optional)	

<b>Requirement Title:</b> (*required)	Modify a Schedule
<b>Sequence No:</b> (*required)	8
<b>Short description:</b> (*required)	Application allows the user to modify a schedule.
<b>Detailed Description:</b> (*required)	<p>8. The application shall provide a button in the user interface to “Edit/Modify” schedule.</p> <p>8.1. The application shall navigate the user to a page that allows him/her to make changes to the schedule.</p> <p>8.2. The application shall provide options for the user to set, specifying the schedule.</p> <p>8.2.1. The application shall provide an option to create a new time block to prevent any classes from showing up within that time range.</p> <p>8.2.2. The application shall display a list of the classes within the schedule.</p> <p>8.2.3. The application shall give options to save or remove the current schedule after changes are made.</p>
<b>Pre-Conditions:</b> (optional)	- There must be a schedule saved to the database so it can be modified.
<b>Post Conditions:</b> (optional)	
<b>Other attributes:</b> (optional)	

<b>Requirement Title:</b> (*required)	Remove a course
<b>Sequence No:</b> (*required)	9
<b>Short description:</b> (*required)	Allows the user to remove a course.
<b>Detailed Description:</b> (*required)	<p>9. Application shall allow the user to view the list of the added courses.</p> <p>9.1. Application shall provide users check-boxes to select one or multiple added courses.</p> <p>9.2. Application shall provide a “remove” button for the users to confirm if they would like to remove the selected course/courses.</p> <p>9.2.1. If the user confirms to remove the selected course/courses, then the application shall remove the courses and shall update the list of the added courses.</p>
<b>Pre-Conditions:</b> (optional)	- There must be a network connection to get information from the database.
<b>Post Conditions:</b> (optional)	- The application will display a success message, if the course is successfully removed.
<b>Other attributes:</b> (optional)	

<b>Requirement Title:</b> (*required)	Logout
<b>Sequence No:</b> (*required)	10
<b>Short description:</b> (*required)	Allows the user to log out of the application.
<b>Detailed Description:</b> (*required)	<p>10. Application shall provide a logout button at the top right corner of the screen.</p> <p>10.1. If the logout button is pressed, the application shall pop up a confirmation dialog to ask the user if he/she really wants to log out of the application.</p> <p>10.1.1. If the user presses OK button, then the application shall log out and take the user back into the login screen.</p> <p>10.1.2. If the user presses cancel button, then the application shall stay on the current screen.</p>
<b>Pre-Conditions:</b> (optional)	<ul style="list-style-type: none"> <li>- There must be a network connection.</li> <li>- The user must be logged into the application.</li> </ul>
<b>Post Conditions:</b> (optional)	<ul style="list-style-type: none"> <li>- The application shall take the user back to the login menu, if the user opts to log out.</li> </ul>
<b>Other attributes:</b> (optional)	

<b>Requirement Title:</b> (*required)	Edit/Delete User
<b>Sequence No:</b> (*required)	11
<b>Short description:</b> (*required)	Allows the user to edit or delete a user in the application
<b>Detailed Description:</b> (*required)	<p>11. The application shall show a list of all users.</p> <p>11.1. The user shall select a user to edit/delete by clicking on one of them.</p> <p>11.1.1. The application shall prompt the user to enter the password for the user selected.</p> <p>11.1.1.1. If the password entered does not match the password stored for selected user, an error message shall be shown and the application shall return to the edit/delete user screen.</p> <p>11.1.1.2. If the password entered matches the password stored for the selected user, the application shall show the details for the selected user (user name and password) as well as a button to save edits and a button to delete this user.</p> <p>11.2. If the user taps the save button, the application shall check the data for validity and run through the same checks in the Add User requirement.</p> <p>11.3. If the user taps the delete button, the application shall display a confirmation dialog asking the user if he/she would really like to delete the selected user.</p> <p>11.3.1. If the user taps the No button, the application shall return to the edit/delete screen for the selected user.</p> <p>11.3.2. If the user taps the Yes button, the application shall remove the selected user from the system and return to the list of users.</p> <p>11.4. If there are no more saved users, the application shall</p>

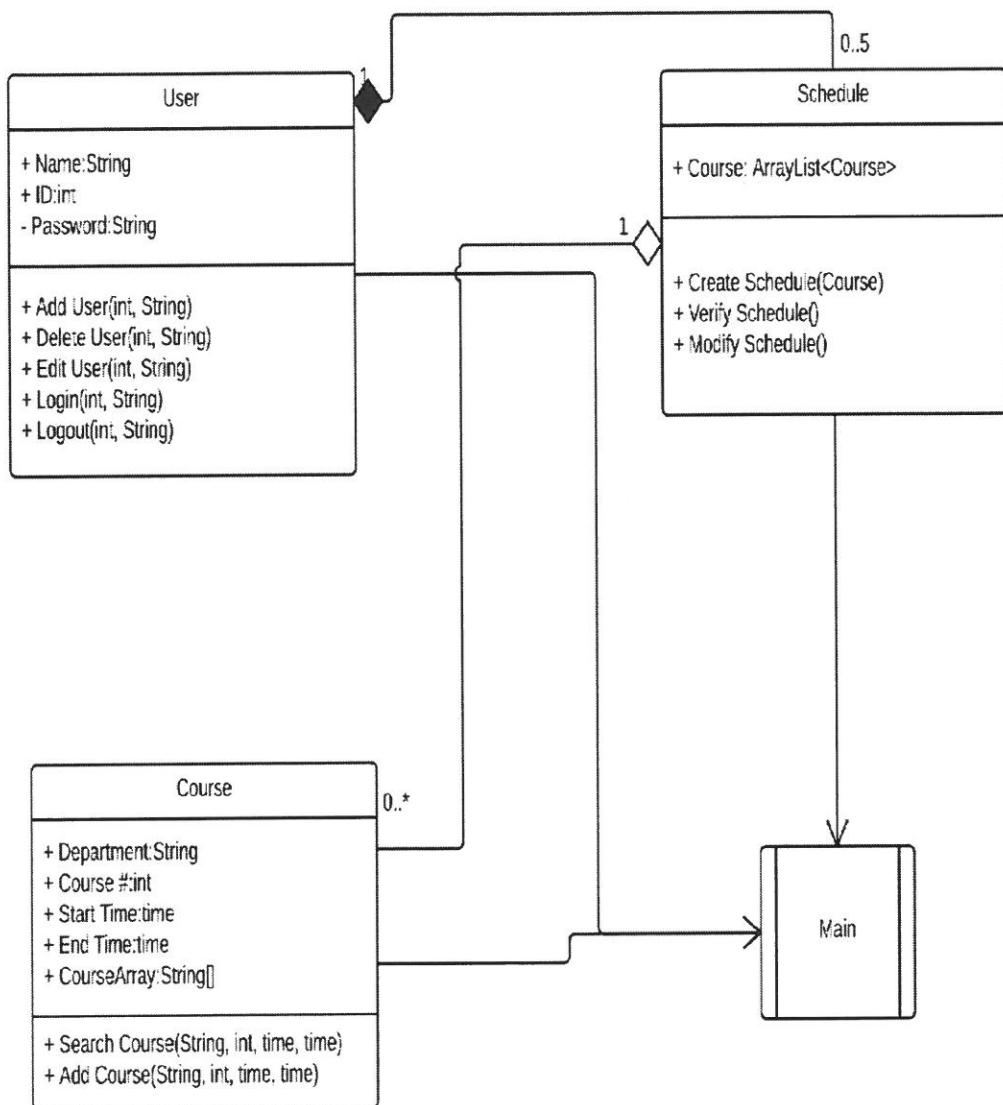
	return to the main screen.
<b>Pre-Conditions:</b>  (optional)	- There must already be users in the system.
<b>Post Conditions:</b>  (optional)	- The application must not display the deleted user.
<b>Other attributes:</b>  (optional)	

## **5. SOFTWARE PROCESSES AND INFRASTRUCTURE**

### **5.1 HARDWARE AND INFRASTRUCTURE**

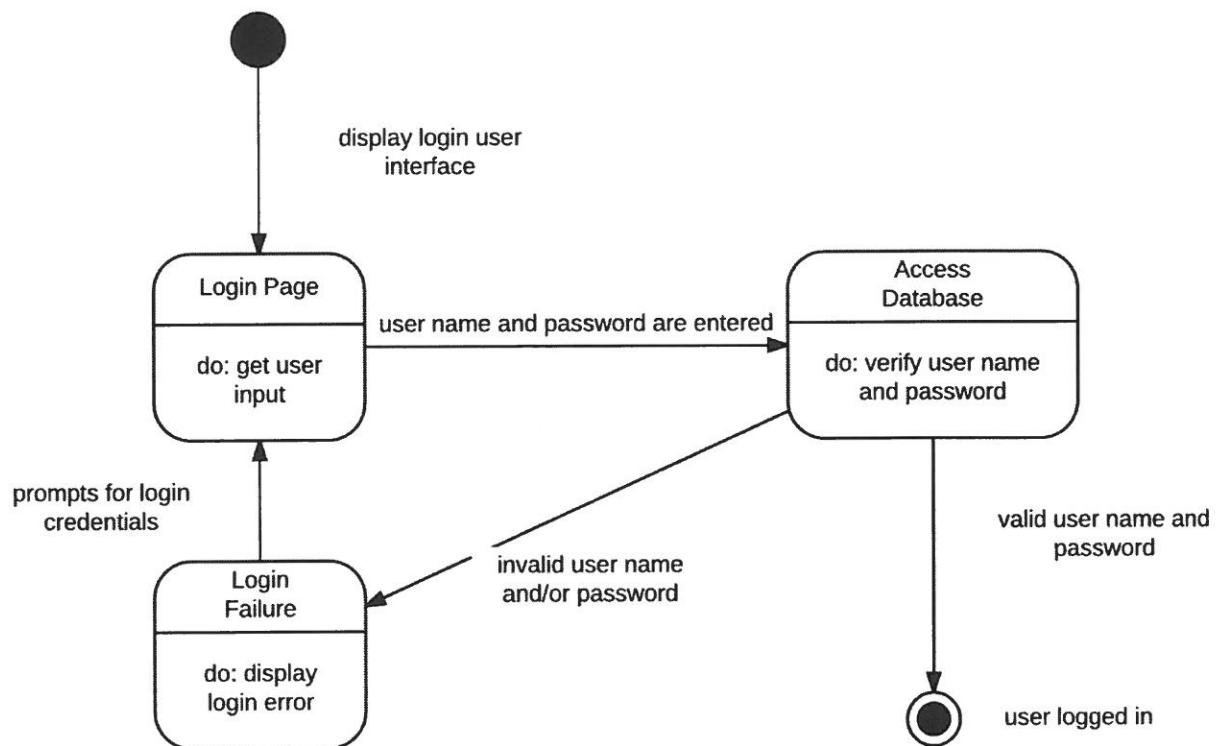
The application is compatible with Android-based hardware architecture

## 5.2 UML DIAGRAM

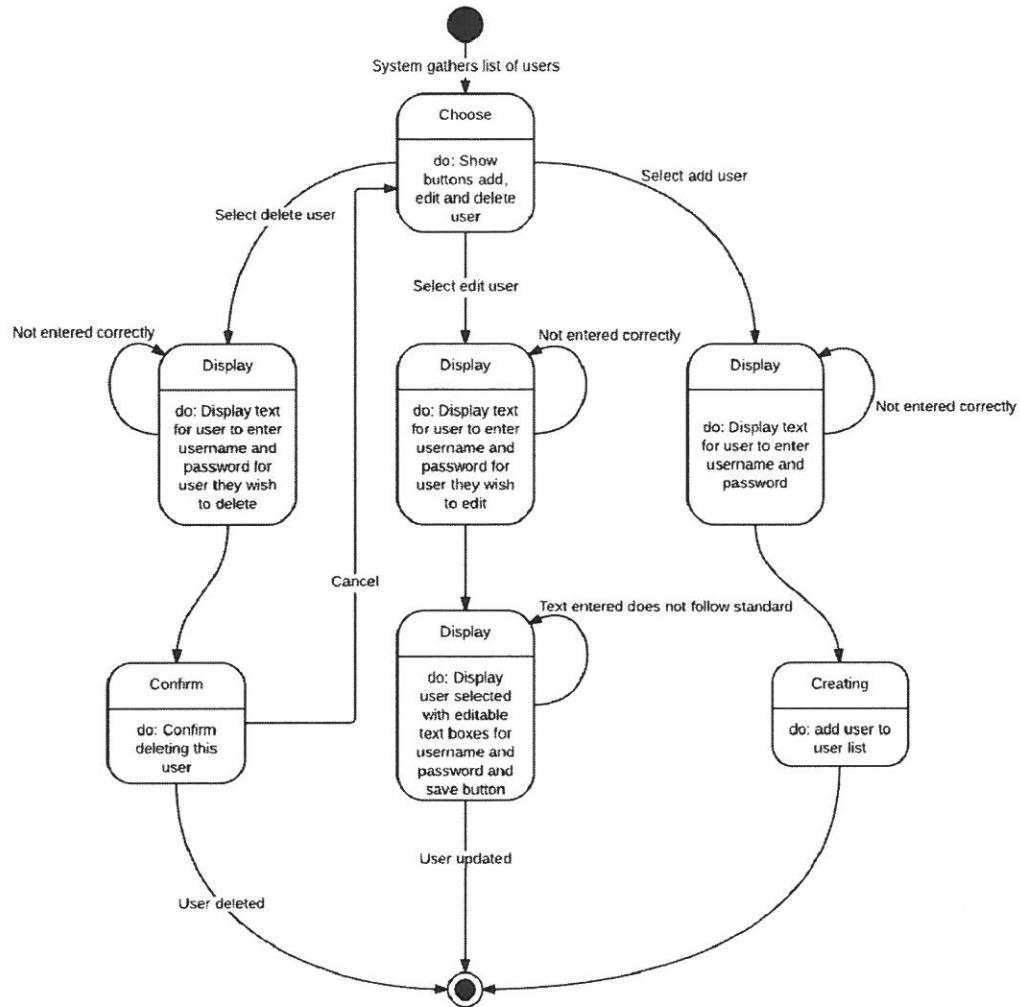


### 5.2.1 STATE DIAGRAMS

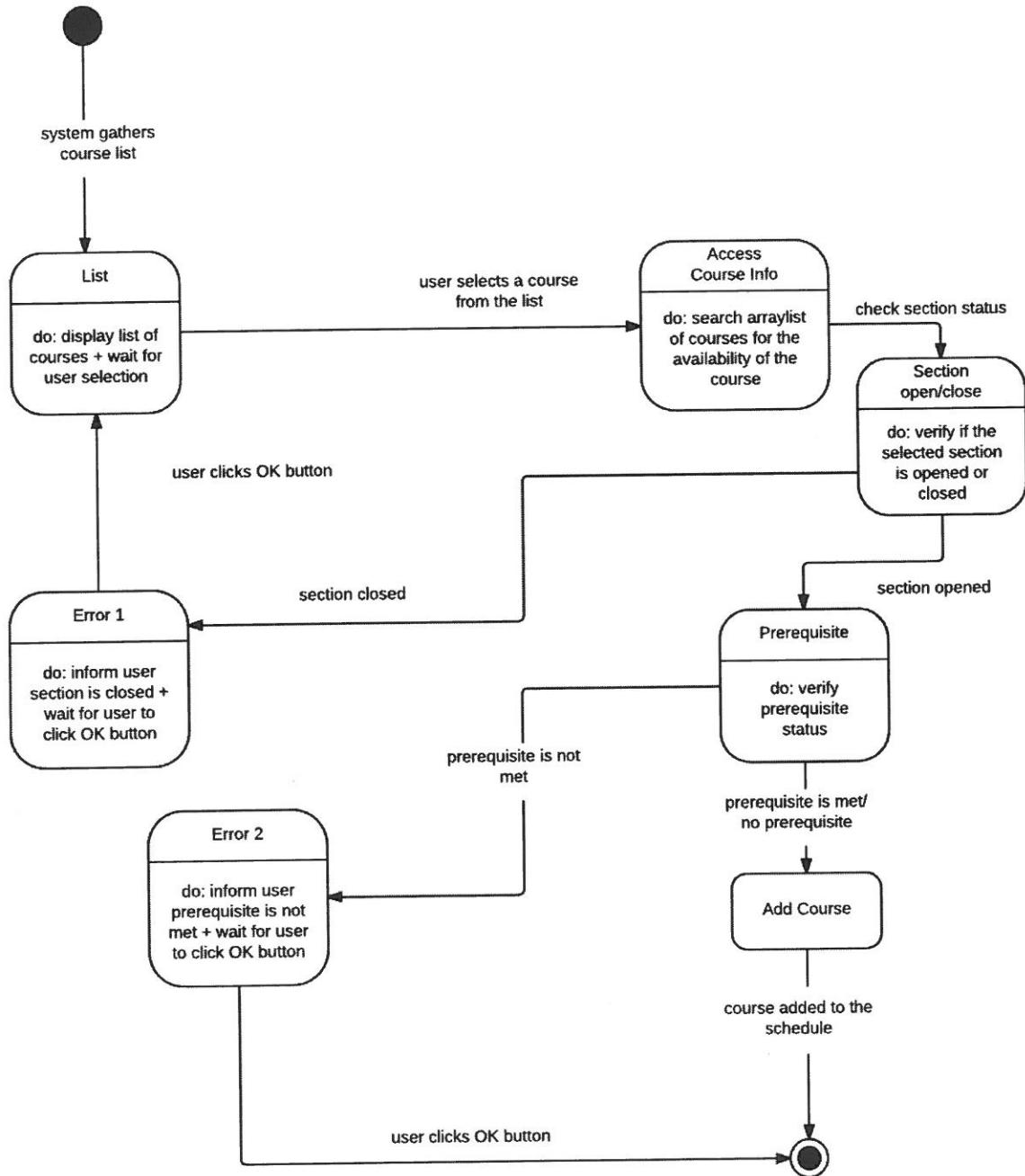
#### LOGIN



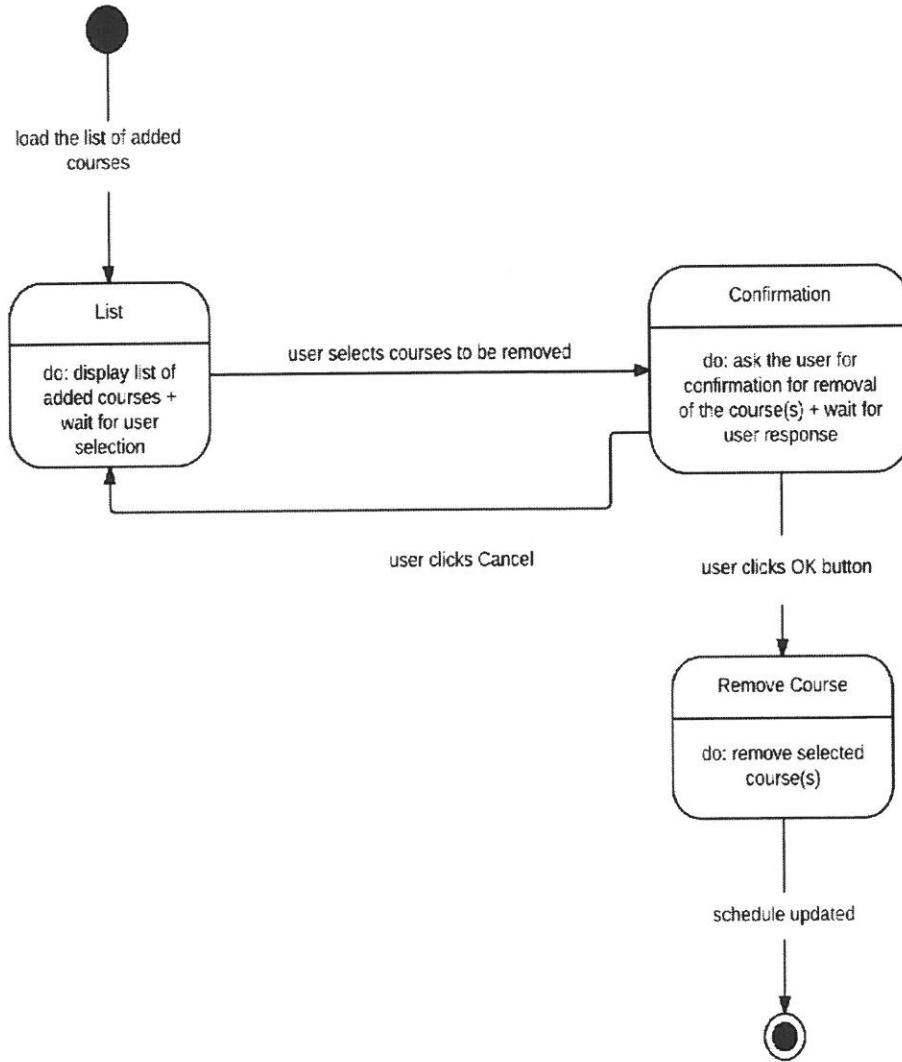
## ADD/EDIT/DELETE USER



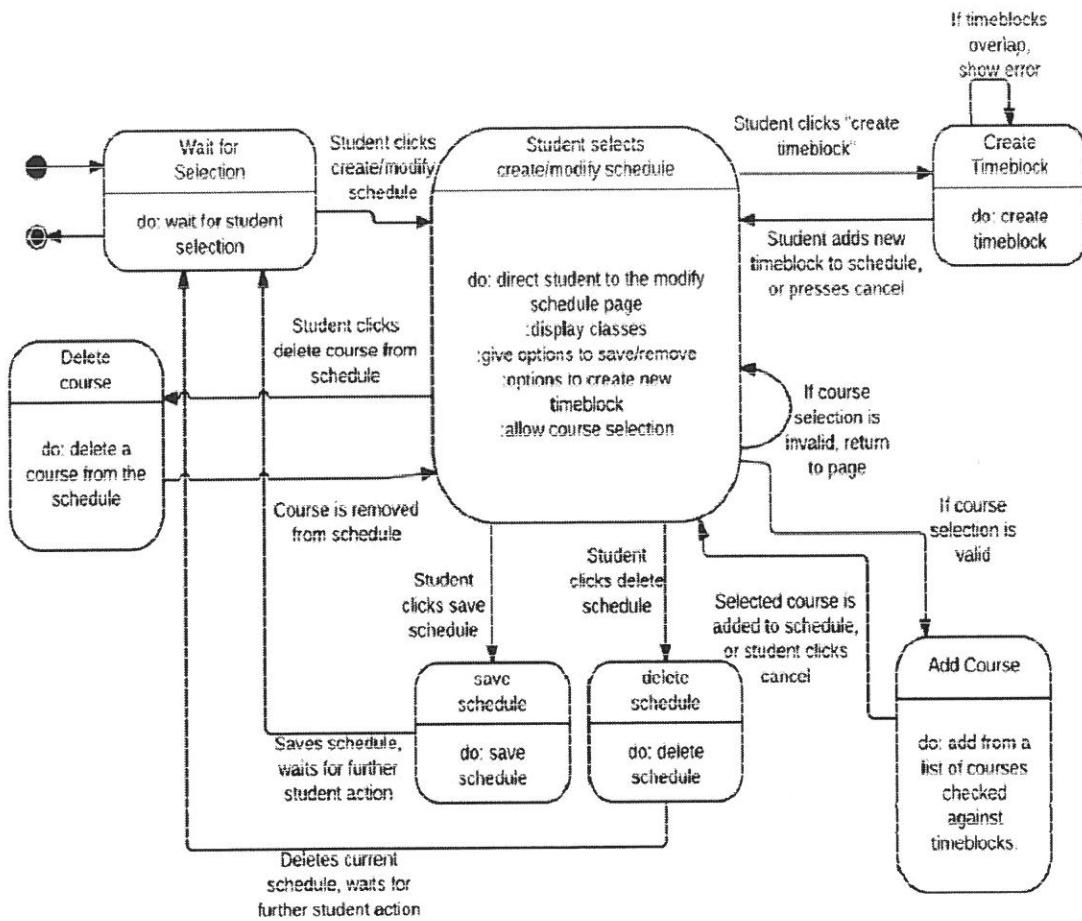
## ADD COURSE



## REMOVE COURSE



## CREATE /DELETE/MODIFY SCHED



## 5.3 TEST PLAN

### 1. INTRODUCTION AND PLAN OF APPROACH

#### PROJECT OVERVIEW

An Android application will be developed that will help create a class schedule. The application shall have the following minimum functionalities:

1. Login
  - User authentication (username/password)
  - Add user
  - Edit user
  - Remove user
2. Obtain open classes from web
3. Create class schedule
  - Class schedule should have no conflicts--no classes may overlap
  - User should be notified if classes conflict
    - Remember, the university now has Saturday classes
4. Users should be able to block out time on a schedule
  - Classes must be arranged around any blocked out time
  - If a set of classes conflicts with the some blocked out time, user should be notified
5. Users should be able to store a class schedule
  - Stored schedules must be reverified to make sure all classes are still available
6. More than one schedule may work. Users should be given several schedules to choose from.
7. Phone resource management: the app should not kill a user's phone, battery, or usage plan

#### **TEST CASE COMPONENTS TO BE COVERED**

1. Add User
2. Login
3. Create Schedule
4. Add Course
5. Verify Schedule
6. Save Schedule
7. Select Schedule
8. Modify Schedule
9. Remove Course
10. Logout
11. Edit/Delete User

### **TEST CASES: LOGIN**

<b>Test Case No.</b>	<b>Test Case Description</b>	<b>Expected Results</b>	<b>Outcome (Pass/Fail/Other Comments)</b>
TC1	User clicks on the login button without entering user name and/or password	System should display a login error and remind the user that the user name and password field cannot be left empty	Fail
TC2	User enters invalid user name and/or password	System should display a login error and ask the user to enter valid user name and password	Fail
TC3	User enters valid user name and password	System should allow the user to log in	Pass

### **TEST CASES: CREATE SCHEDULE**

<b>Test Case No.</b>	<b>Test Case Description</b>	<b>Expected Results</b>	<b>Outcome (Pass/Fail/Other Comments)</b>
TC1	User clicks on the “Create Schedule” button.	System should save a new schedule to the database and take the user to the modify schedule screen for the newly created schedule.	Pass

## **TEST CASES: VERIFY SCHEDULE**

<b>Test Case No.</b>	<b>Test Case Description</b>	<b>Expected Results</b>	<b>Outcome (Pass/Fail/Other Comments)</b>
TC1	User selects a schedule and clicks on the “save schedule” button with valid course information	System should save the schedule to the database	Pass
TC2	User clicks on “save schedule” button with a course that overlaps a created time block.	System should not save the schedule to the database, and display an error message stating that “Selected class overlaps a blocked time”	Fail
TC3	User clicks on “save schedule” button with time blocks that overlap other time blocks.	System should not save the schedule to the database, and display an error message stating that “Multiple time blocks are overlapping”	Fail
TC4	User clicks on “save schedule” button with time blocks that have an end time prior to a start time.	System should not save the schedule to the database, and display an error message stating that “Time blocks cannot have a start time prior to an endtime”	Fail

## TEST CASES: MODIFY SCHEDULE

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User selects a schedule and clicks on the “Modify schedule” button	System should navigate to the “modify schedule” page	Pass
TC2	User clicks on the “delete schedule” button.	System should remove the selected schedule from the database	Pass
TC3	User clicks on the “Save Schedule” button.	System should save the schedule to the database	Pass
TC4	User clicks “create Timeblock” button and enters in a valid start time and valid end time.	System should add the timeBlock to the schedule and return a success message.	Pass
TC5	User clicks on “create Timeblock” button and enters in an end time prior to a start time	System should not add the timeBlock to the schedule and return an error message stating that “End times may not occur prior to start times”.	Fail
TC6	User clicks on “Add Course” button with a valid department name and course number.	System should return a list of courses that do not occur within any added timeBlocks for selection.	Pass
TC7	User clicks on the desired course from the list returned from the “Add Course” button.	System should add that course to the schedule.	Pass
TC8	User clicks “Delete course” button next to a course within the schedule.	System should remove that course from the schedule.	Pass

## TEST CASES: SAVE SCHEDULE

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User selects a schedule and clicks on the “save schedule” button	System should save the schedule to the database	Pass

## TEST CASES: REMOVE COURSE

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User selects course(s) to be removed from the list of added courses	System should successfully remove the courses from the list of added courses	Pass

## TEST CASES: LOGOUT

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User clicks on “logout” button	System should log the user out of it	Pass

## TEST CASES: ADD USER

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User enters a username that does not meet requirements for user being added	System should display an error and remind user the requirements for username	Fail
TC2	User enters password twice for verification for user being added and passwords do not match	System should display an error letting the user know the passwords did not match	Fail
TC3	User enters username and both passwords correctly for user being added	System will add new user to the database	Pass

## TEST CASES: EDIT USER

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User selects edit user and enters a new username that does not meet requirements	System should display an error and remind user the requirements for username	Fail
TC2	User selects edit user and enters a new password twice for verification and passwords do not match	System should display an error letting the user know the passwords did not match	Fail
TC3	User selects edit user and enters username and both passwords correctly	System will edit user in the database	Pass

## TEST CASES: DELETE USER

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User selects delete user and enters wrong password for verification before user is deleted	System should display an error letting user know the password is wrong	Fail
TC2	User selects delete user and enters password correctly	System will delete user from the database	Pass

## TEST CASES: SELECT SCHEDULE

Test Case No.	Test Case Description	Expected Results	Outcome (Pass/Fail/Other Comments)
TC1	User selects a schedule from list of schedules	System will load schedule for user to manipulate as needed	Pass

## **6. Assumptions and Constraints**

### **6.1 ASSUMPTIONS**

The following is a list of assumptions:

- Assume the user has web access
- Assume the user uses Android 2.33 or higher
- The application will not be released to the market
- Not a large number of users will be using this application
- All users are UTA students
- Schedules are stored locally in the device

### **6.2 CONSTRAINTS**

The following is a list of constraints:

- Team lacks Android development skills
- Time is limited, so debugging might be limited
- System will be developed with Android studio
- Difficult to coordinate schedules with team members

### **6.3 Out of Scope Material**

The following is a list of “out of scope” material:

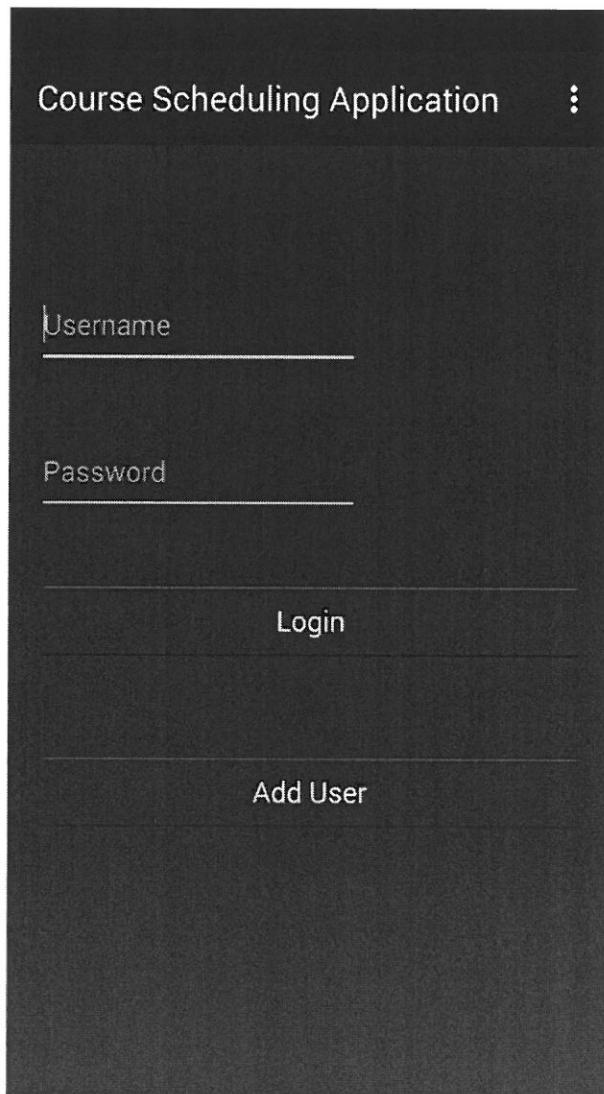
- Post Project maintenance is not covered
- Budget is not covered
- No risk-assessment

## 7. USER MANUAL

This is the start screen of our application and is where the user will initially be able to create a new user to add to the database or login. There are two text boxes where an existing user can input their username and password and a button labeled “Add User” where the user can click on to go to the next screen to add a new user.

### Login

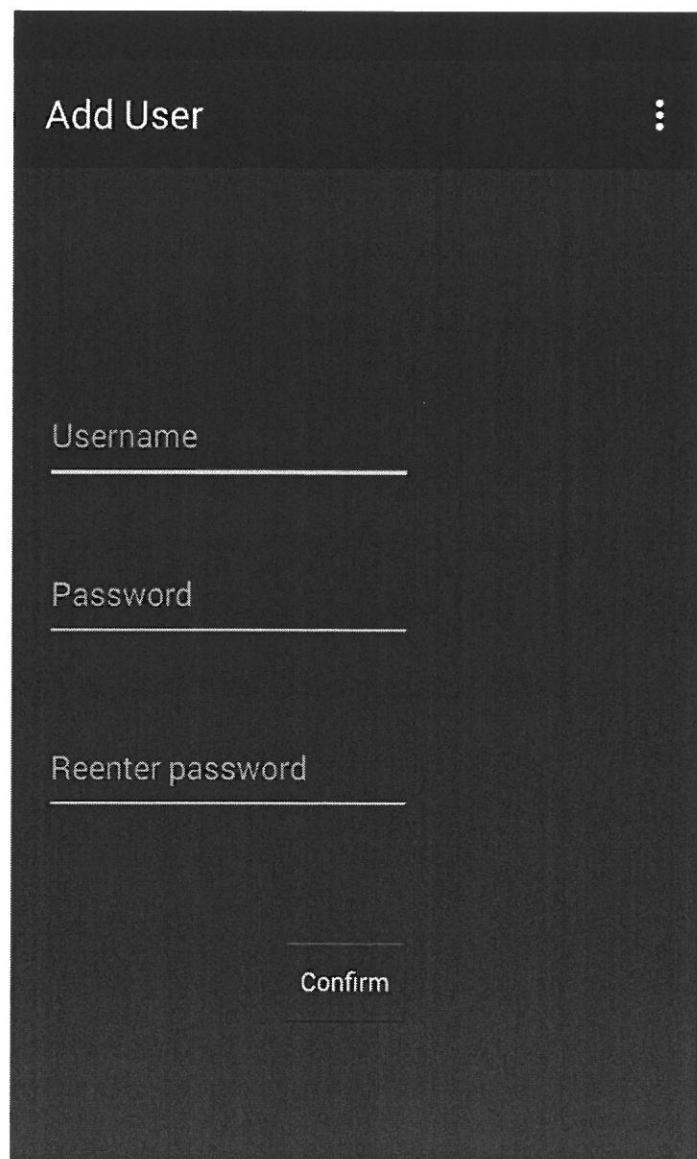
- 1) If the user is an existing user and has previously created a user in the system input your username and password then click login
- 2) If the user is NOT an existing user and wishes to access the system you must first create a username and password in the system. Click add user from this screen



From this screen a user is able to add a new user in the system for access to create and save schedules. They must enter their MyMav username and password then click confirm to add this to the system.

### Add User

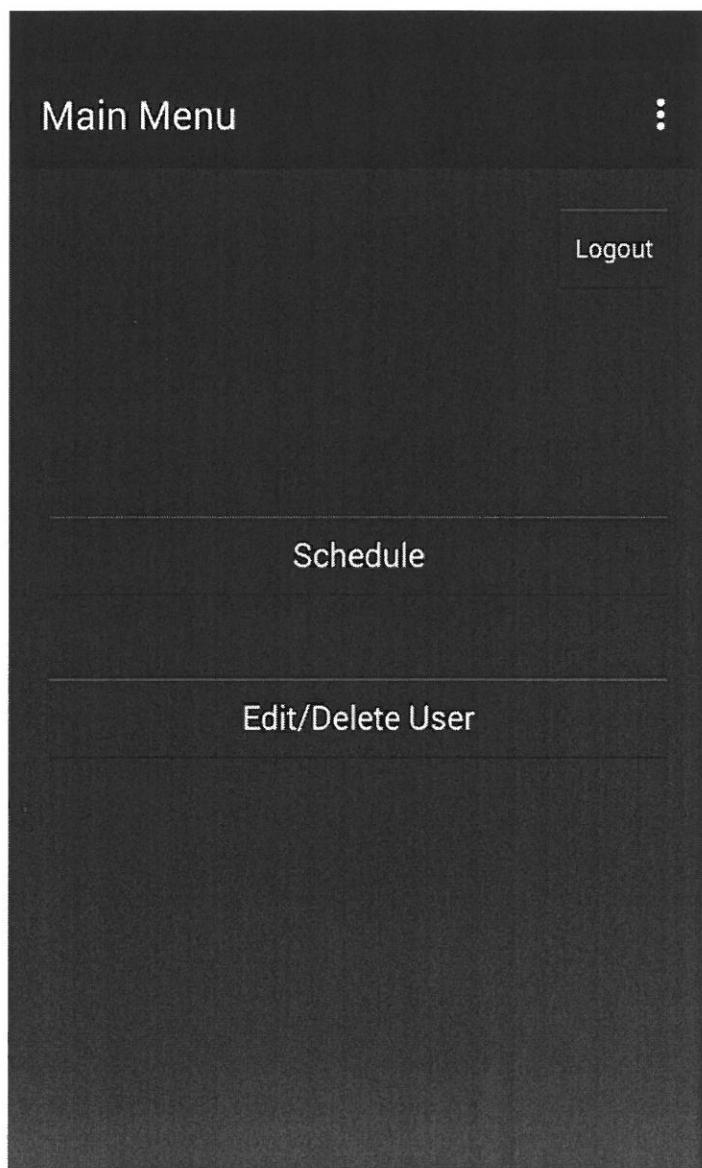
- 1) Enter your MyMav username and password here. The password must be entered twice to ensure it is entered correctly
- 2) Click confirm when done



This is the main menu that will show once a user is logged on. From here the user has the options to view their saved schedules and on subsequent pages modify and/or delete these schedules or to edit/delete their user saved information in the system. It is also possible to logout, this will take them back to the login page where they will have to login again to access their schedules.

### Main Menu

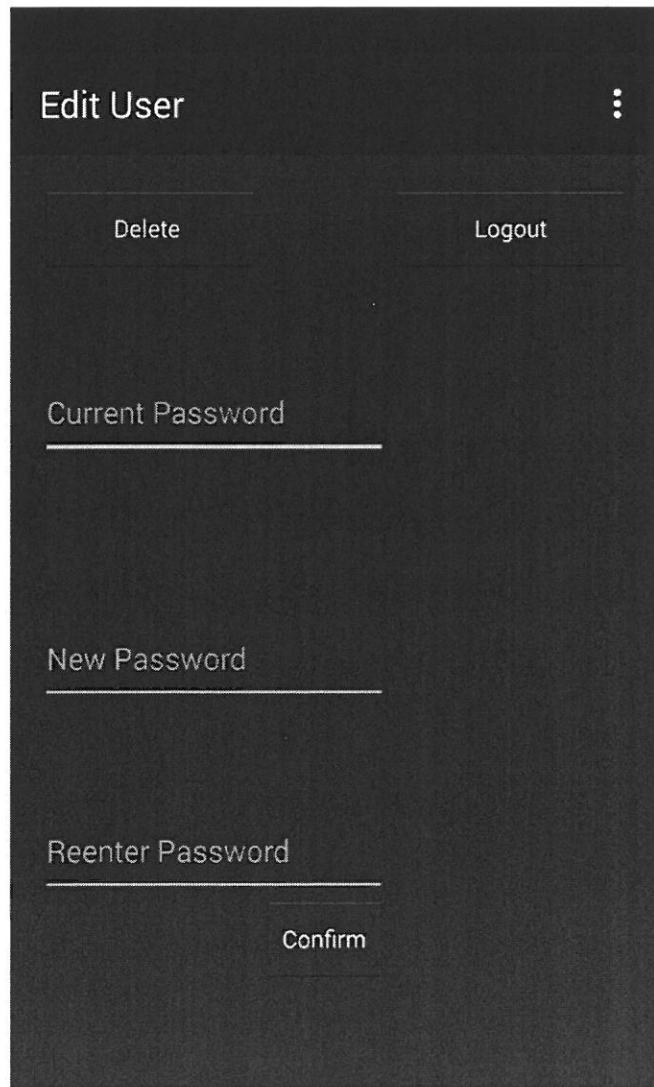
- 1) If you would like to view/edit or create a schedule click on the schedule button which will take you to the schedule interface
- 2) If you would like to edit/delete your user in the system click on edit/delete user which will take you to the edit user interface
- 3) Also if you would like to logout click the logout button which will take you back to the main menu



From this screen the user can edit their password. This is done in case the user's MyMav password is ever changed then they will be able to also change it here to prevent any errors. The user can also delete their profile entirely from the system by clicking delete. As well as being able to logout from this screen.

### Edit/Delete User

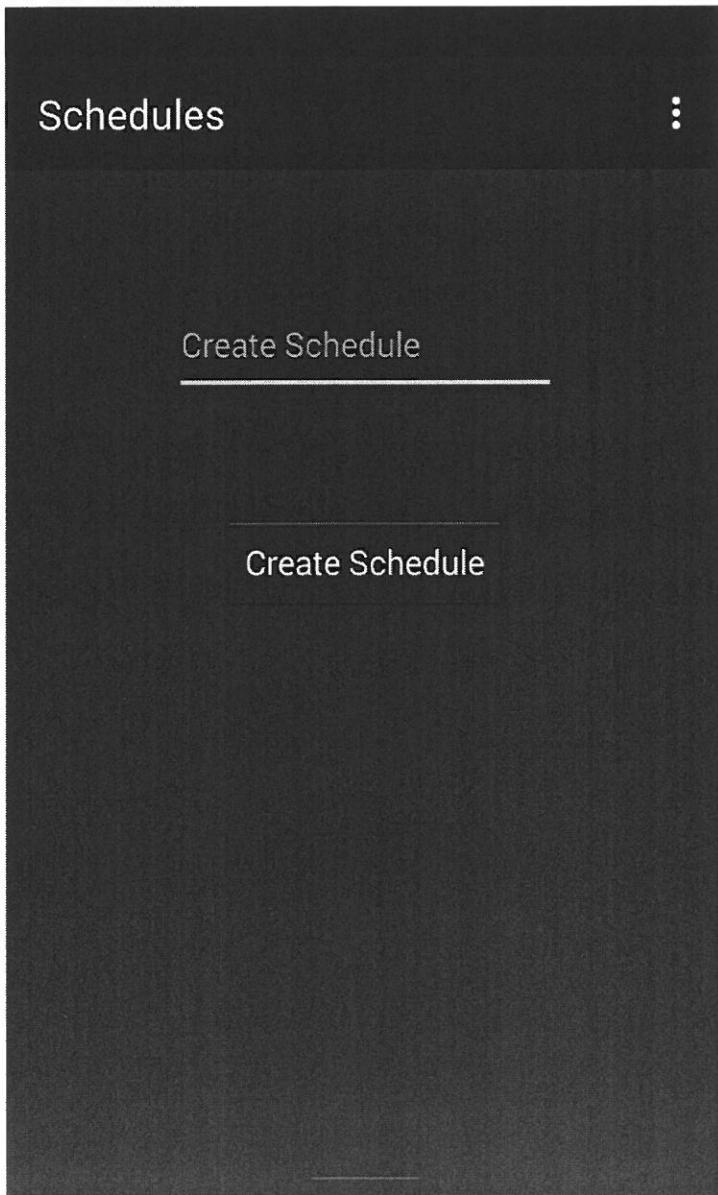
- 1) If you would like to change your password. Enter your current password followed by the new password you would like twice then click confirm to make the change
- 2) If you would like to delete your user from the system click the delete button to do this, this will also delete any saved schedules you might have
- 3) You can also logout from this screen as before which takes you back to the login screen



This screen will show all of the users currently saved schedules as well as giving them the option to create a new one. If the user would like to view and/or edit a current schedule they can click on it and be taken to the modify screen for that schedule. If the user would like to add a new schedule they must first give it a name then click create schedule and then they will be taken to the modify screen for that new schedule.

#### Schedule

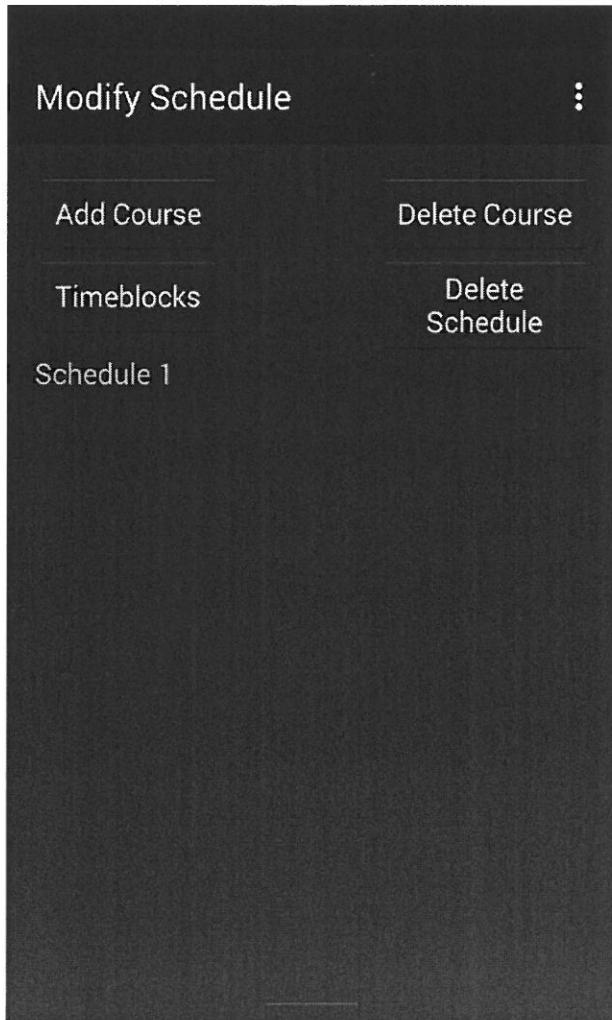
- 1) If you would like to create a new schedule you must first give the schedule a name and then click create schedule, this will take you to a page where you can add courses, time blocks and other functionality for this new schedule
- 2) If you would like to edit/view a schedule that you have previously created they will be shown here on this screen. You can click on the one you want to view/edit and will be taken to a page where you can modify this schedule or just view the courses



Here a user can do many things with their schedule. They can add a new course, delete a current course, add a time block or delete the entire schedule. This is the main screen for a particular schedule. The current courses will be shown in a list here.

### Modify Schedule

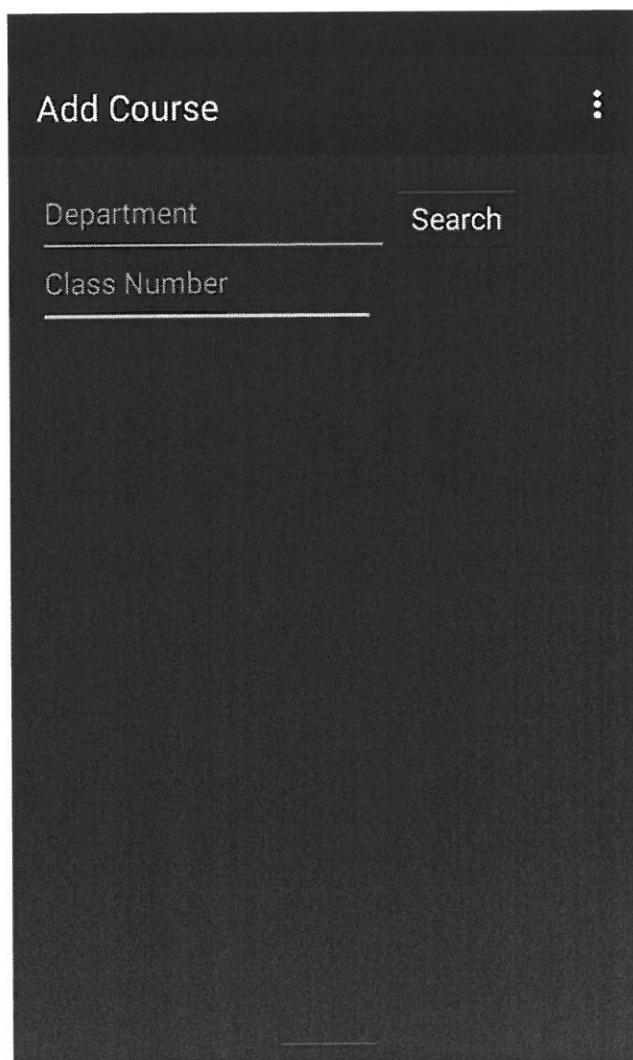
- 1) You have many options from this screen, all of a particular schedules courses will be shown here for you to view, if you would like to change this schedule in any way here are your options
- 2) Add course: click on the add course button and it will take you to a screen where you can search for a new course to add
- 3) Timeblock: click on the timeblocks button to add a new timeblock to your schedule. This is a period of time you can specify that you do not want to have any courses in for a particular schedule
- 4) Delete course: if you would like to delete a previously added course from a schedule click on this button where it will take you to a screen where you can select which course to delete
- 5) Delete schedule: if you would like to delete the current schedule all together click on this button to do so. You will be taken back to the schedule screen



This is the screen where a user can search for a course they would like to add to their schedule. When they enter in the department and class number the system will search for a course matching this information and show them what it finds where the user can subsequently choose from the resulting list what course they would like to add.

### Add Course

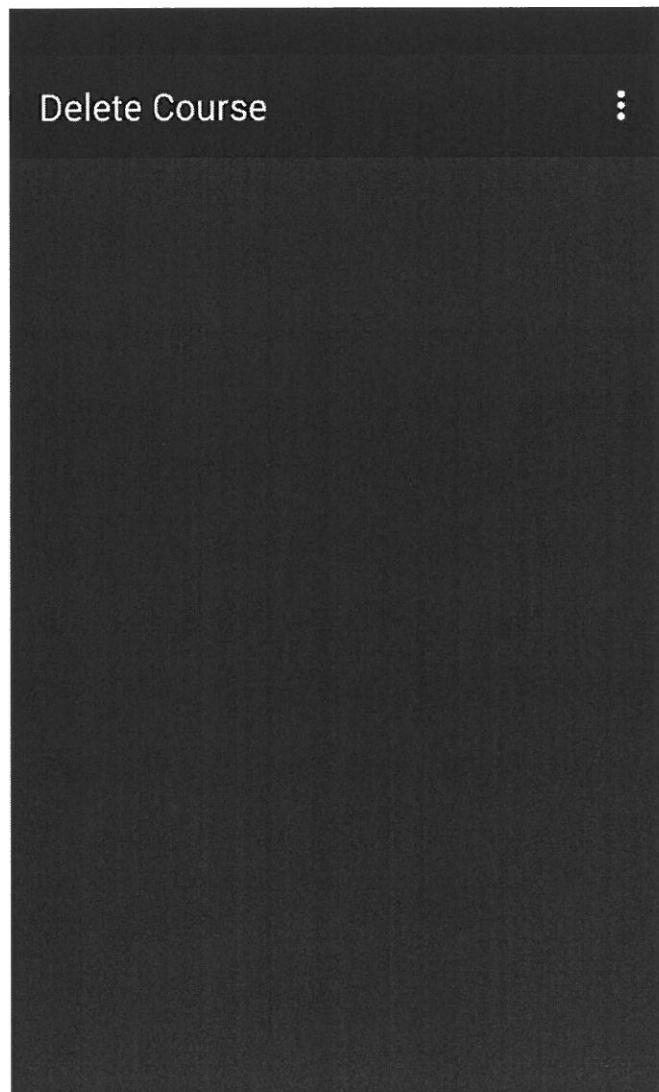
- 1) To search for a course, you must enter in the department and class number for the course you are searching for
- 2) Enter the department in with the department initials. i.e. Computer Science will be CSE, Electrical Engineering will be EE, etc.
- 3) Also enter the class number that you are searching for. Double check for typos then click search



This screen will populate all of the courses for the current schedule here. The user has the option to click on a particular course and it will be deleted.

#### Delete Course

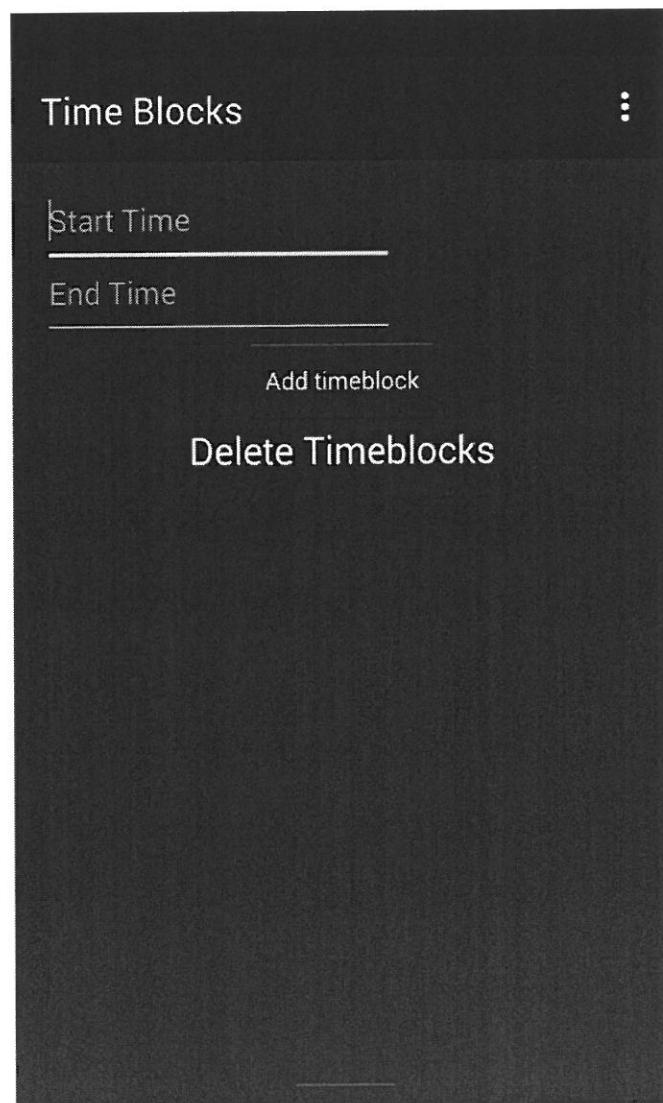
- 1) From here you will be shown all of the courses that have been previously added to a particular course
- 2) To delete a course tap on it and it will be removed
- 3) To go back click the back button on your device



This screen gives the user the ability to add and/or delete timeblocks to their schedule. When a timeblock is added, any courses that are searched for that fall within this time will be excluded from the list shown to the user to add. This screen will also list all current timeblocks and a timeblock can be deleted by clicking on it from this list.

### Timeblocks

- 1) Here you can add a timeblock to your schedule.
- 2) Enter the start and end time (in 24-hour format) for this timeblock then click add timeblock.
- 3) All of your current timeblocks will be shown here as well.
- 4) If you would like to delete a timeblock click on it in this list and it will be removed.



## 8. DELIVERY AND SCHEDULE

Task/Milestone Description	Anticipated Start Date	Anticipated End Date	Status	Comments
Prepare Requirements	2/20/2015	2/26/2015	Completed	
SRA document (Includes project objectives, Requirements and UML diagrams)	3/27/2015	3/31/2015	Completed	
Presentation of SRA	3/31/2015	3/31/2015	Completed	Document consisting of requirements, UML diagrams, and assumptions and constraints
GUI Design and Navigation	4/3/2015	4/10/2015	Completed	
Functional Implementation	4/11/2015	4/20/2015	Completed	Start implementing the requirements of the application
Test Data Entry	4/28/2015	4/29/2015	Completed	
Debugging	4/30/2015	5/4/2015	Completed	Time will be reduced if there are less bugs to deal with/more time is required to develop functionalities
Final: Project Delivery	5/5/2015	5/7/2015	Completed	



## **8. Stakeholder Approval Form**

Stakeholder Name	Stakeholder Role	Stakeholder Comments	Stakeholder Approval Signature and Date
Joseph Wigner	Development Mgr		
Elise Austin	Project Assistant		
Hani Jouya	Developer		
Matthew Zielke	Developer		
Mostafa Galib	Developer		