

Project:	University Course Scheduler
Team No.:	Team Alpha
Class:	CSE 3310; Spring 2015
Module:	System Requirements Analysis (SRA)
Deliverable:	SRA Document

Version: [1.0]

Date: [5/7/2015]

Contributors:

Arun Kalahasti

arun.kalahasti@mavs.uta.edu

972-408-5600

Eric Katz

ekatz@mavs.uta.edu

214-797-6686

Mason Wilkens

mason.wilkens@mavs.uta.edu

817-996-8592

Peyton Casper

peyton.casper@mavs.uta.edu

469-569-5711

Document Revision History

<i>Version number</i>	<i>Date</i>	<i>Originator</i>	<i>Reason for change</i>	<i>High level description of changes</i>
0.1	3/24/2015	Mason Wilkens	Base document	Document Creation
0.2	3/26/2015	Mason Wilkens	Revision	Added Business Objectives, Added System Objectives, Added Domain Model, Changed Assumptions/Constraints
0.2.1	3/27/2015	Arun Kalahasti	Revision	State Diagrams Added
0.2.2	3/27/2015	Eric Katz	Revision	Introduction and Product Overview, Context Diagram
0.2.3	3/29/2015	Arun Kalahasti	Revision	Sequence Diagrams Added
0.2.4	3/30/2015	Peyton Casper	Revision	Database Model added
0.2.5	3/31/2015	Mason Wilkens	Revision	Changes to delivery and schedule
0.3	3/31/2015	Arun Kalahasti Eric Katz Mason Wilkens Peyton Casper	Revision	Updated Business Objectives
0.3.1	4/5/2015	Arun Kalahasti	Formatting	Fixed alignment issues across requirements and changed indenting for detailed requirements
0.4	5/7/2015	Eric Katz	Addition	Added User Manual
0.4.1	5/7/2015	Eric Katz	Revision	Updated Contributor page to add contact info. Added a Header
1.0	5/7/2015	Arun Kalahasti	Revision	Updated User Manual Screens

Code Revision History

<i>Version number</i>	<i>Date</i>	<i>Originator</i>	<i>Reason for change</i>	<i>High level description of changes</i>
0.1	4/5/2015	Arun Kalahasti	Created application base	Created a login page using a default android template, Created a blank main activity page to be displayed after login, Created Data classes to hold course data
0.1.1	4/7/2015	Arun Kalahasti	Created HTTP server Interface	Login page now uses HTTP Service to check if user login is valid, server always responds true regardless of input. Implemented logic to check if courses conflict
0.2	4/8/2015	Arun Kalahasti Eric Katz	Simplified logic, created initial schedule generation	Updated Logic of section conflicts and Added Spooft Data to test conflicts, Used section conflict logic to create simple schedule generation function
0.3	4/10/2015	Peyton Casper Arun Kalahasti	True server interaction	Server now has ability to receive requests to store a new user's data in database, Login screen updated to match new server requirements and add new users. Server not only authorizes a login if user and password provided by app exist in database
0.4	4/13/2015	Peyton Casper	Server gets data from MyMav	Server can now pull live course information from MyMav to give back to app upon URL based request.
0.4.1	4/14/2015	Arun Kalahasti	Application parses server data	Created raw ability to get user input from main activity page and pass the request to the server. Server response is parsed from JSON to internal classes and displayed to user.
0.4.2	4/20/2015	Arun Kalahasti	Created Setup Schedule page	Setup Schedule page implemented to allow more robust user input for desired courses
0.4.3	4/22/2015	Arun Kalahasti	Created Add Block-Out Times page	Block-Out times can be setup and saved to the application for use during schedule generation
0.5	4/24/2015	Peyton Casper	Server pulls semester data from MyMav	Server updated with ability to find all classes available in selected semesters and group them by department
0.5.1	4/27/2015	Arun Kalahasti	Setup Schedule page updated	Application will request semester data from the server and only allow the user to select a class for schedule generation if it exists in any given semester
0.5.2	4/29/2015	Arun Kalahasti	Select Courses updated	Application will search semester data as user inputs text into the select course input and suggest choices to the user

University Course Scheduler - SRA Document

0.6	5/2/2015	Arun Kalahasti	Functional Schedule display	Fetched course info and user selected block-out times are passed to the schedule factory to generate schedules. The generated schedule is now shown to the user.
0.9	5/5/2015	Eric Katz Arun Kalahasti	UI Updates and HTTP Service updates	All application pages updated for appearance. Created and implemented an application wide theme. HTTP Service now uses a static factory pattern to create new instances of http requests. HTTP service can now post data to web pages
1.0	5/6/2015	Arun Kalahasti Eric Katz Peyton Casper	Bug fixing, UI finalization	Major application and server wide bug fixing, and UI finalization

TABLE OF CONTENTS

1. INTRODUCTION AND PROJECT OVERVIEW

2. OBJECTIVES

2.1 BUSINESS Objectives

2.2 SYSTEM Objectives

3. PROJECT CONTEXT DIAGRAM

4. SYSTEMS REQUIREMENTS

4.1 "User Account" Requirements

4.2 "Login" Requirements

4.3 "Schedule" Requirements

4.4 "Block-Out Time" Requirements

4.5 "Course" Requirements

5. SOFTWARE PROCESSES AND INFRASTRUCTURE

5.1 Hardware and Infrastructure

5.2 UML Diagrams

5.3 Conceptual Data Model - Database

5.4 Screen Shots

5.5 Test Plan

6. ASSUMPTIONS AND CONSTRAINTS

6.1 ASSUMPTIONS

6.2 CONSTRAINTS

6.3 Out of Scope material

7. DELIVERY AND SCHEDULE

8. STAKEHOLDER APPROVAL FORM

9. USER MANUAL

APPENDIX:

1. Introduction and Project Overview

This document outlines implementation specifications for software that will allow users to generate course schedules for The University of Texas at Arlington(UTA). The application will be developed to run on Android version 4.2+. The software has two major components – the client application and the server database. Users will interact with the client application on an Android device. The server database is used by the client application to obtain course data and store user data.

Users will be required to make an account to access the functionality of the software. The client application will allow the user to create, edit, and delete an account to gain access. Once the user has successfully created an account and gained access to the software, they will be able to generate, save, and delete schedules. In generating a schedule, the user will be able to add and remove block-out times, and add and remove courses. Schedules will be generated and presented to the user such that there are no conflicts between the block-out times and courses the user has specified. When a user finds a course schedule they wish to keep, they can save it to their account. The user can verify saved schedules are still valid through the application.

The server will host an SQL database that will store user data including username, password, and saved schedules. The database will also store course information that will be accessed by the client application. The database will obtain course information by scraping information from UTA's website MyMav.

2. Objectives

2.1 BUSINESS OBJECTIVES

The following is a list of business objectives:

Objective 1: User Account Registration: Every account must include the following:

- Username
- Password
- E-mail address

Objective 2: User Login: Users must login to the system with a user/password that match with their respective account.

Objective 3: Schedule: The system should create schedules from a list of selected courses with their respective section.

Objective 4: Edit Schedule: The system should allow the user to add/swap/remove courses from their respective schedule.

Objective 5: Remove User: The system should allow the user to remove their respective account.

Objective 6: Block-Out Time: The system should allow the user to block out a specified amount of time for that schedule.

Objective 7: No Schedule Conflicts: The system must not allowed conflicting courses/block-out time to be added to a single schedule.

Objective 8: User Notifications: The system must notify the user if a course/block-out conflict occurs in their schedule.

Objective 9: Saturday Course: The system must allow the user to add valid saturday courses to their own schedule.

Objective 10: Verify Schedule: The system must verify a stored schedule upon login and mark respective schedules as valid/invalid..

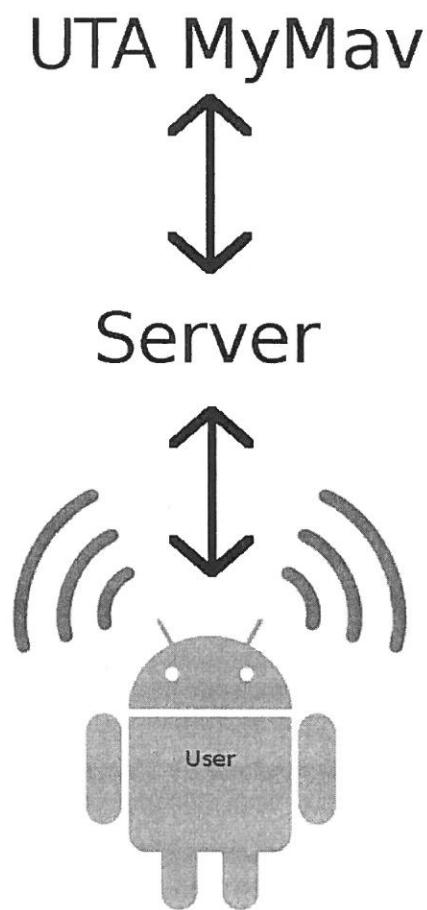
2.2 SYSTEM OBJECTIVES

The following is a list of system objectives:

Objective 1: Runs on an Android device.

Objective 2: Microsoft SQL will be used as the database.

3. Project Context Diagram



4. Systems Requirements

4.1 "USER ACCOUNT" REQUIREMENTS

Requirement Title: (*required)	Create user account
Sequence No: (*required)	1.2
Short description: (*required)	User creates a user account
Detailed Description: (*required)	<ol style="list-style-type: none">1. Application shall get the data from the user interface<ol style="list-style-type: none">1.1. The application shall check data for validity<ol style="list-style-type: none">1.1.1. If the application detects empty username and/or password, the application shall not allow the creation of an account1.1.2. If the application detects invalid characters, the application shall not allow the creation of an account1.1.3. If the application detects another account with the same username, the application shall not allow the creation of an account1.2. If the application detects valid input, the application shall create an account
Pre-Conditions: (optional)	
Post Conditions: (optional)	Application shall auto login account on creation
Other attributes: (optional)	

Requirement Title: (*required)	Delete user account
Sequence No: (*required)	1.3
Short description: (*required)	User deletes an existing account
Detailed Description: (*required)	<ol style="list-style-type: none">1. The application shall allow the user to request deletion of their account.<ol style="list-style-type: none">1.1. The application shall prompt the user to confirm deletion<ol style="list-style-type: none">1.1.1. If the user cancels, the application shall not delete the account1.1.2. If the user confirms, the application shall delete the account
Pre-Conditions: (optional)	The user will be logged into the account before deleting it
Post Conditions: (optional)	The user, after deletion, shall be redirected to the login interface
Other attributes: (optional)	

Requirement Title: (*required)	Edit user
Sequence No: (*required)	1.4
Short description: (*required)	User edits a user account
Detailed Description: (*required)	<ol style="list-style-type: none">1. the application shall receive input from the user interface<ol style="list-style-type: none">1.1. If the user request a password change, the application shall check the data for validity1.1.2. If the application detects invalid characters, the application shall not allow the password change1.1.3. If the application receives valid input, change the password1.2. The application shall not allow the user to change their username
Pre-Conditions: (optional)	The user will be logged into the account before editing it
Post Conditions: (optional)	
Other attributes: (optional)	

4.2 "LOGIN" REQUIREMENTS

Requirement Title: (*required)	Log in
Sequence No: (*required)	1.1
Short description: (*required)	User logs into account
Detailed Description: (*required)	<p>1. Application shall get the data from the user interface</p> <p>1.1. Application shall check the data for validity</p> <p>1.1.1. If the application detects an empty username and/or password, the application shall not allow a login and require a login with valid data</p> <p>1.1.2. If the application detects invalid characters, the application shall not allow a login and require a login with valid data</p> <p>1.1.3. If the application detects valid input, the application shall compare the inputted username and password with those stored in the system.</p> <p>1.1.3.1. If the username and password match a pre-existing user, the application shall display the main application interface</p> <p>1.1.3.2. If either the username or the password do not match a pre-existing user, the application shall not allow a login and require a login with valid data</p>
Pre-Conditions: (optional)	There must be a pre-existing valid user in the system
Post Conditions: (optional)	If user successfully logs in, show main user interface
Other attributes: (optional)	<p>Username and password are stored on device</p> <p>Username shall be a minimum of 4 characters and not exceed 12 characters and shall be comprised of upper and lowercase alphanumeric characters</p> <p>Passwords shall be a minimum of 8 characters comprised of upper and lowercase alphanumeric characters</p>

4.3 "SCHEDULE" REQUIREMENTS

Requirement Title: (*required)	Create New Schedule
Sequence No: (*required)	2.1
Short description: (*required)	User creates a new schedule.
Detailed Description: (*required)	<ol style="list-style-type: none">1. The application shall create a new blank schedule.1.1. The application will prompt the user to add courses and/or block out times.<ol style="list-style-type: none">1.1.1. If the user cancels, the application shall not create a schedule1.1.2. if the user selects either add courses or block out times, the application shall allow the user to create a schedule
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Generate Schedule Choices
Sequence No: (*required)	2.2
Short description: (*required)	User creates schedule(s)
Detailed Description: (*required)	<ol style="list-style-type: none">1. Application shall retrieve set of desired courses1.1. If there are no courses the application should prompt the user to add at least one1.2. Application will retrieve set of course sections for each course1.3. Application shall generate sets of course sections1.3.1. If sets have internal conflicts the application should discard them1.3.2. Application should not create sets with closed courses in them1.4. Application shall present the generated schedules to the user
Pre-Conditions: (optional)	User has entered at least one desired course
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Edit Schedule Choices
Sequence No: (*required)	2.3
Short description: (*required)	User Edits a Schedule in a Set of Schedules
Detailed Description: (*required)	<p>1. Application shall get the selected schedule to be edited from the user interface</p> <p>1.1. Application will show the user detailed information about the selected schedule</p> <p>1.1.1. If the user selects a course the application shall display alternate sections of that course</p> <p>1.1.1. Sections which are closed or show a conflict with another course in the user's schedule should be marked for the user</p> <p>1.2. If the user selects a section which has a conflict with another section(s) the conflicting schedules should be marked for the user</p> <p>1.2.1. If the user selects a section which has no conflicts the new section shall replace the old section of that course</p>
Pre-Conditions: (optional)	A schedule is selected by the user to be edited
Post Conditions: (optional)	The edited schedule will replace the selected schedule
Other attributes: (optional)	

Requirement Title: (*required)	Save Schedule Choices
Sequence No: (*required)	2.4
Short description: (*required)	User saves schedule(s)
Detailed Description: (*required)	<ol style="list-style-type: none">1. Application will get selected schedule(s) from the user interface.<ol style="list-style-type: none">1.1. Application should allow the user to name the schedule1.1.2. Application will check schedule name for a valid name.<ol style="list-style-type: none">1.1.1. If the name is empty, the application shall assign a generated name.1.1.1.2. If the application detects invalid data, the application will not save the schedule and require a valid name.1.1.1.3. If the application detects valid data, the application will save the schedule.
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Remove Schedule
Sequence No: (*required)	2.5
Short description: (*required)	User Removes A Schedule From A Set Of Schedules
Detailed Description: (*required)	<p>1. Application shall get the selected schedule to be removed from the user interface</p> <p>1.1. The user will be prompted for confirmation</p> <p>1.2. If the application detects a confirmation it will proceed to remove the selected schedule from the set of schedules</p> <p>1.3. If the application detects a rejection it will return to the previous screen</p>
Pre-Conditions: (optional)	A schedule is selected by the user to be removed
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Verify Schedule
Sequence No: (*required)	2.6
Short description: (*required)	Application verifies validity of the schedule
Detailed Description: (*required)	<p>1. The application shall get the selected schedule to verify from the user interface</p> <p>1.1. The application shall get the current course-section status of all course-sections in the user's add courses set from the server</p> <p>1.1.1. If all courses are open, The application shall display that all course sections are open and the schedule is valid.</p> <p>1.1.2. If any courses are closed, the application shall display which course sections are closed and mark the schedule as invalid</p>
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

4.4 "BLOCK-OUT TIME" REQUIREMENTS

Requirement Title: (*required)	Add Block Out Time
Sequence No: (*required)	3.1
Short description: (*required)	User Blocks Out A Period Of Time
Detailed Description: (*required)	<ol style="list-style-type: none"> 1. Application shall get the starting and ending times of the block out period from the user interface <ol style="list-style-type: none"> 1.1. Application shall ensure times are valid <ol style="list-style-type: none"> 1.1.1. If the application detects blank times, the application shall refuse the entry and require valid input 1.2. If application detects invalid times, the application shall refuse the entry and require valid input 1.3. If application detects valid times, the application shall compare the input to existing block out times in set of block out times <ol style="list-style-type: none"> 1.3.1. If the application detects a conflict with existing block out times, the application shall display the conflicting time, refuse the entry, and require valid input 1.3.2. If the application does not detect a conflict, it shall add the block out time to the set of block out times
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Edit Block Out Time
Sequence No: (*required)	3.2
Short description: (*required)	User Edits a Block Out Time In The Set of Block Out Times
Detailed Description: (*required)	<p>1. Application shall get the selected block out time to be edited from the user interface</p> <p>1.1. Application will show the user detailed information about the selected block out time</p> <p>1.2. If the user changes either a starting or ending time and attempts to save the information the application shall ensure times are valid</p> <p>1.2.1 If the application detects blank times, the application shall refuse the entry and require valid input</p> <p>1.2.2. If application detects invalid times, the application shall refuse the entry and require valid input</p> <p>1.2.3. If application detects valid times, the application shall compare the input to existing block out times in set of block out times</p> <p>1.3. If the application detects a conflict with existing block out times, the application shall display the conflicting time, refuse the entry, and require valid input</p> <p>1.4. If the application does not detect a conflict, it shall add the block out time to the set of block out times</p>
Pre-Conditions: (optional)	A block out time is selected by the user to be edited
Post Conditions: (optional)	The edited block out time will replace the selected block out time
Other attributes: (optional)	

4.5 "COURSE" REQUIREMENTS

Requirement Title: (*required)	Remove Block Out Time
Sequence No: (*required)	3.3
Short description: (*required)	User Removes A Block Out Time From The Set Of Block Out Times
Detailed Description: (*required)	<ol style="list-style-type: none">1. Application shall get the selected block out time to be removed from the user interface<ol style="list-style-type: none">1.1. The user will be prompted for confirmation1.2. If the application detects a confirmation it shall proceed to remove the block out time schedule from the set of block out times1.2.1. If the application detects a rejection it will return to the previous screen
Pre-Conditions: (optional)	A block out time is selected by the user to be removed
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Add Course
Sequence No: (*required)	4.1
Short description: (*required)	User adds a course to a set of desired courses
Detailed Description: (*required)	<ol style="list-style-type: none">1. Application shall get the course data from the user interface1.1. Application shall check if the data is valid<ul style="list-style-type: none">1.1.1. If the course number or department are empty, the application shall not add a course and require a valid course1.1.2. If the course number and department do not match an existing course the application shall not add a course and require an existing course1.2. If the data matches an existing course, the course shall be added to the set of courses to search.
Pre-Conditions: (optional)	There is a pre-existing schedule
Post Conditions: (optional)	
Other attributes: (optional)	

Requirement Title: (*required)	Remove Course
Sequence No: (*required)	4.2
Short description: (*required)	User removes a course from the search
Detailed Description: (*required)	<ol style="list-style-type: none">1. Application gets the selected course to be removed from the user interface1.1. Application removes the selected course from the set of courses to be searched
Pre-Conditions: (optional)	There is a pre-existing, non-empty schedule
Post Conditions: (optional)	
Other attributes: (optional)	

5. Software Processes and Infrastructure

5.1 HARDWARE AND INFRASTRUCTURE

Requirement Title: (*required)	Runs on an Android Operating System
Sequence No: (*required)	
Short description: (*required)	The app will only officially support an Android version number of 4.2+
Detailed Description: (*required)	1. The app shall support an Android version number of 4.2+
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

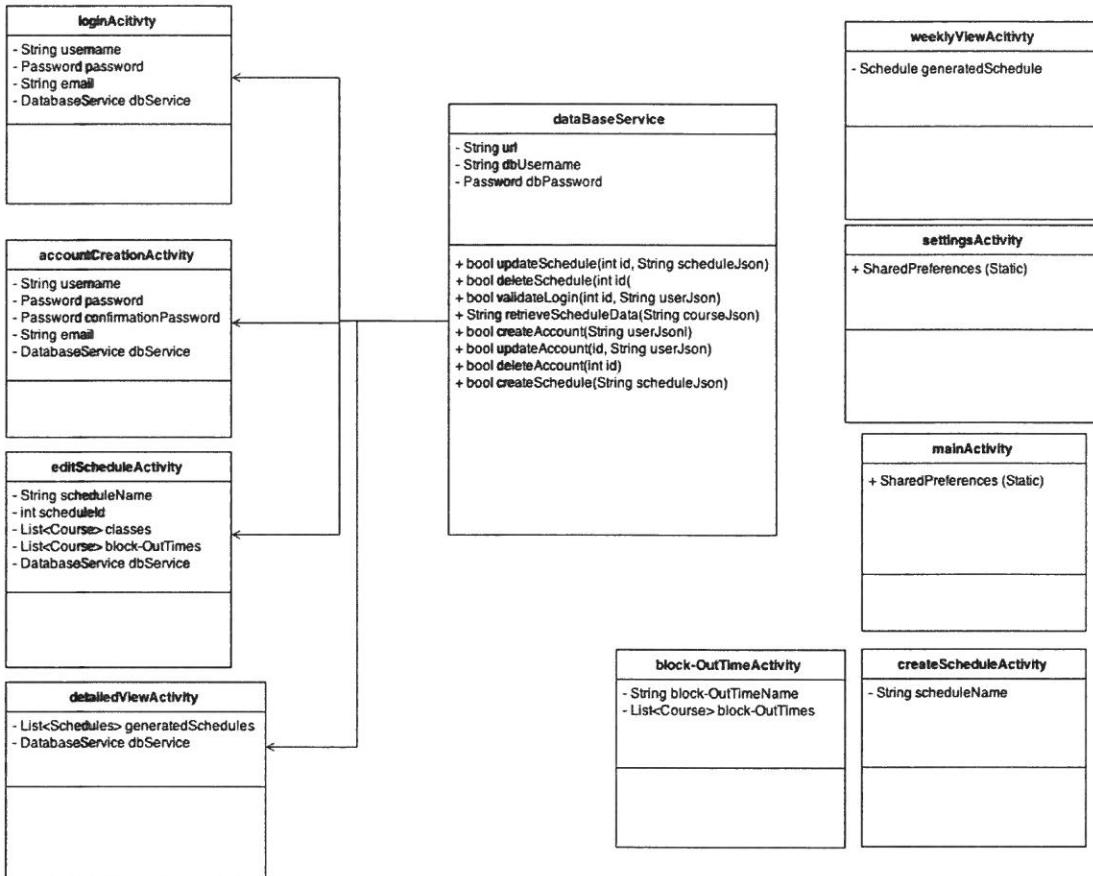
Requirement Title: (*required)	Uses a Server to track and Update user information.
Sequence No: (*required)	
Short description: (*required)	The user will rely on a server based backend that consists of a Database to store user profile information and retrieve, store, and serve data.
Detailed Description: (*required)	<ol style="list-style-type: none">1. The app will communicate with a server to store user information such as account information, and credentials.2. The server will be responsible for collecting and serving schedule validity information back to our app when requested.
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

University Course Scheduler - SRA Document

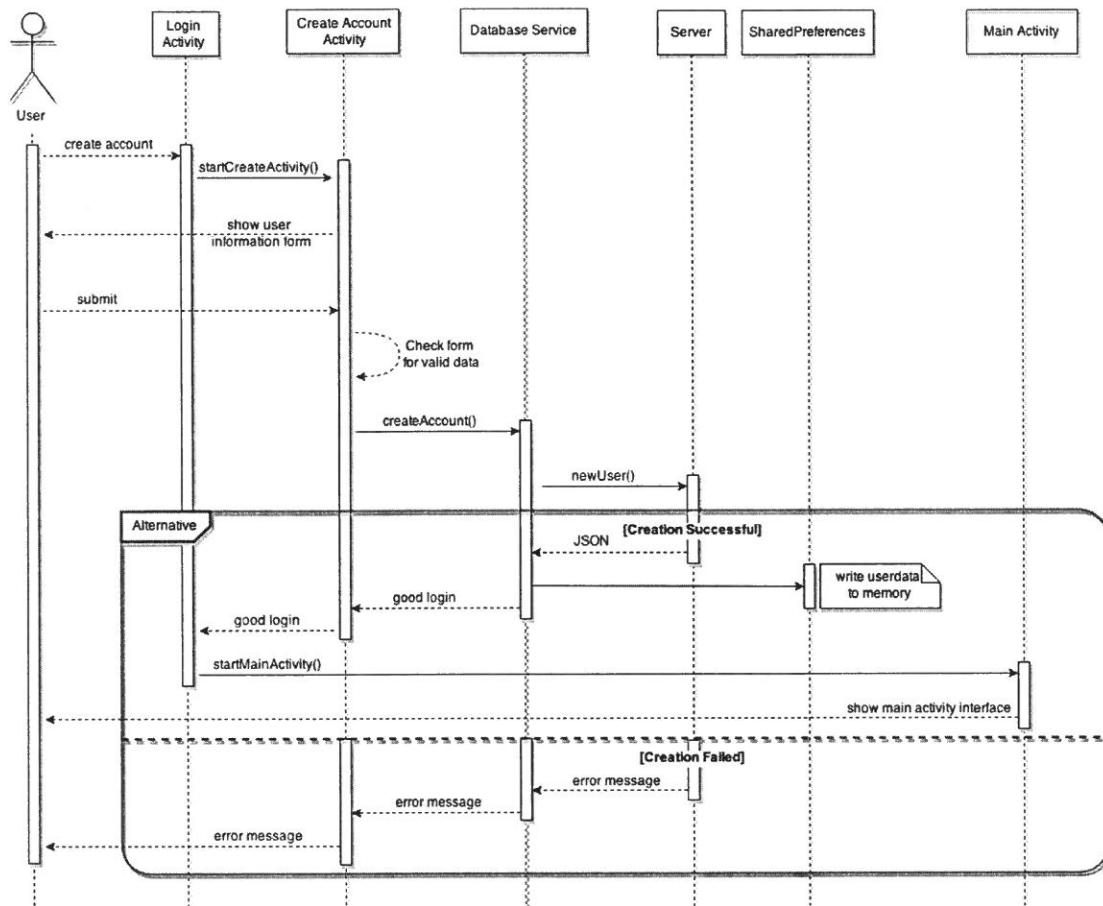
Requirement Title: (*required)	Requires an Internet Connection to Utilize the App
Sequence No: (*required)	
Short description: (*required)	The app will be reliant on an internet connection for various functions within the app.
Detailed Description: (*required)	The app will require an internet connection to verify the users login credentials, validate a user's schedule, modify a user's account information, and to save a schedule.
Pre-Conditions: (optional)	
Post Conditions: (optional)	
Other attributes: (optional)	

5.2 UML DIAGRAMS

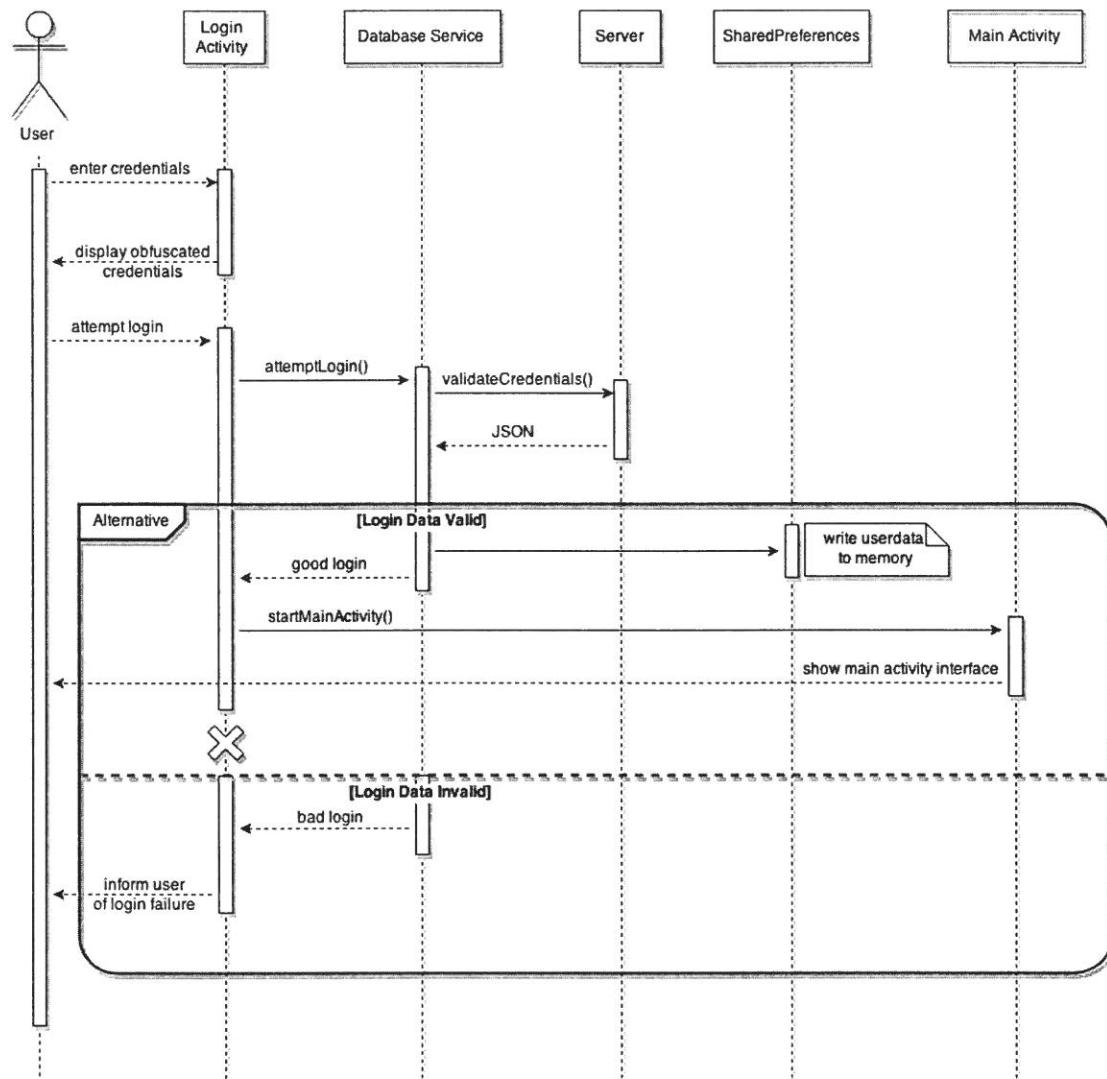
5.2.1 ACTIVITY CLASS DIAGRAM



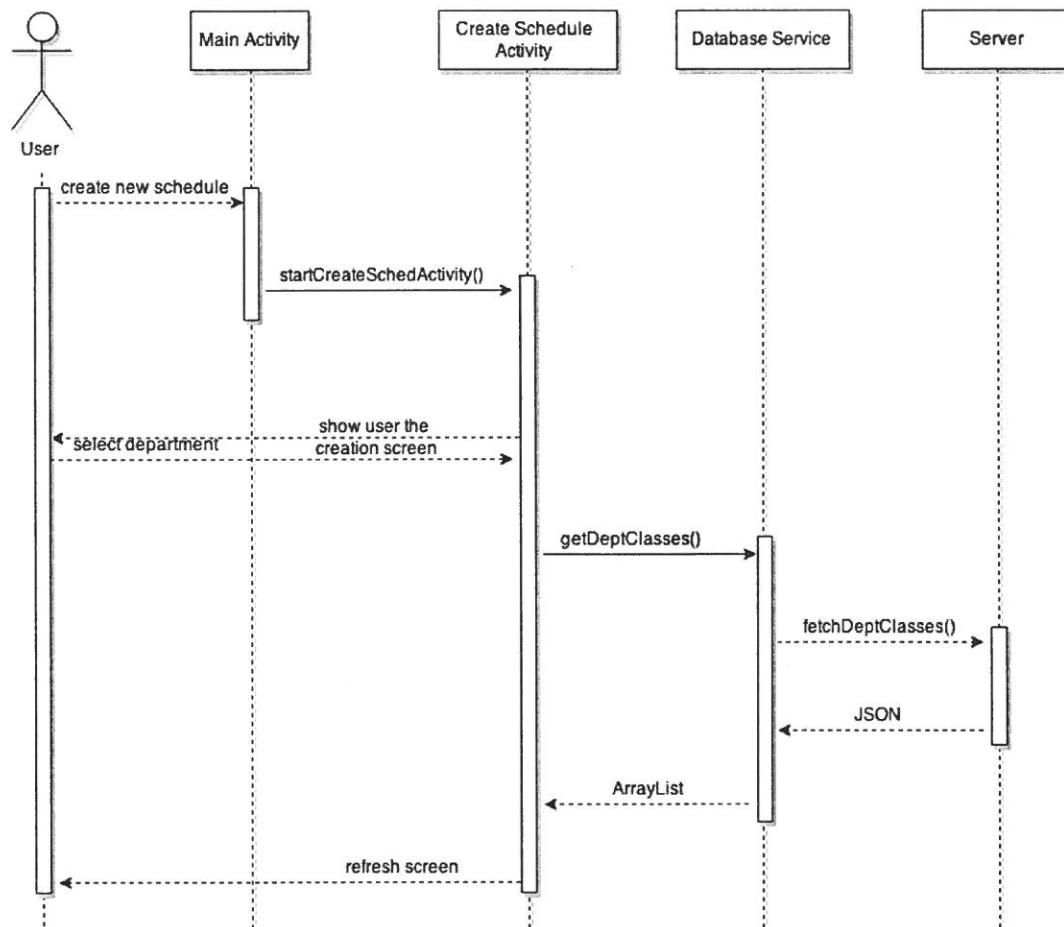
5.2.3 CREATE ACCOUNT SEQUENCE DIAGRAM



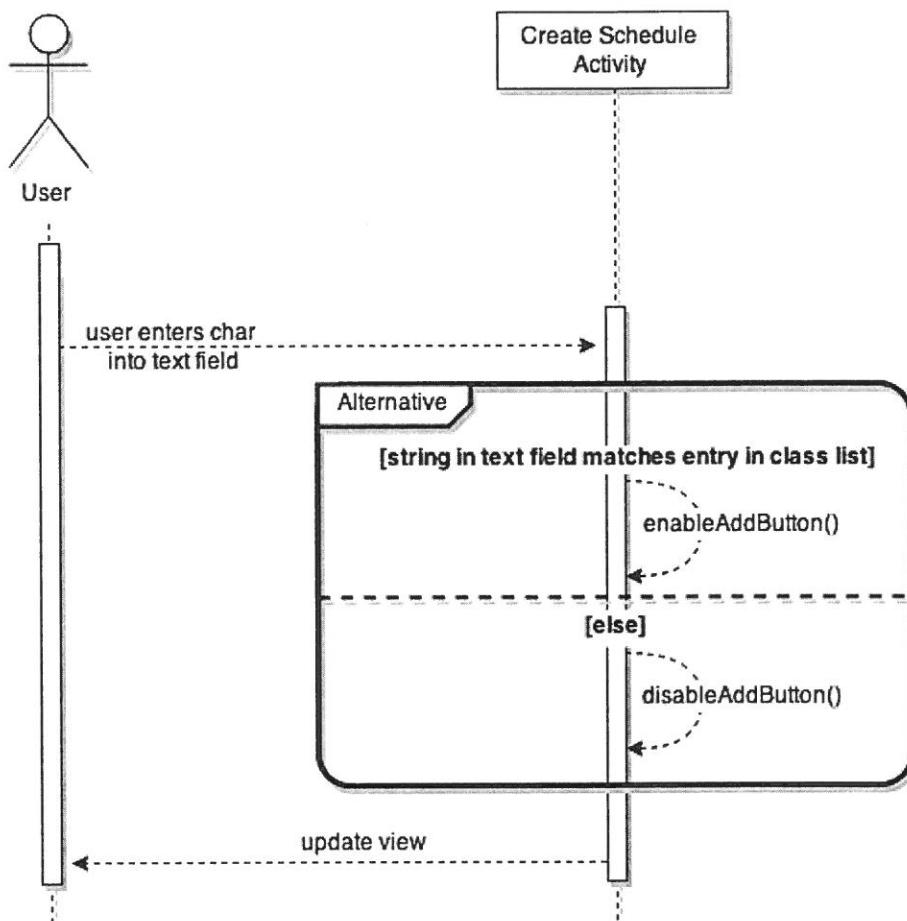
5.2.4 LOGIN SEQUENCE DIAGRAM



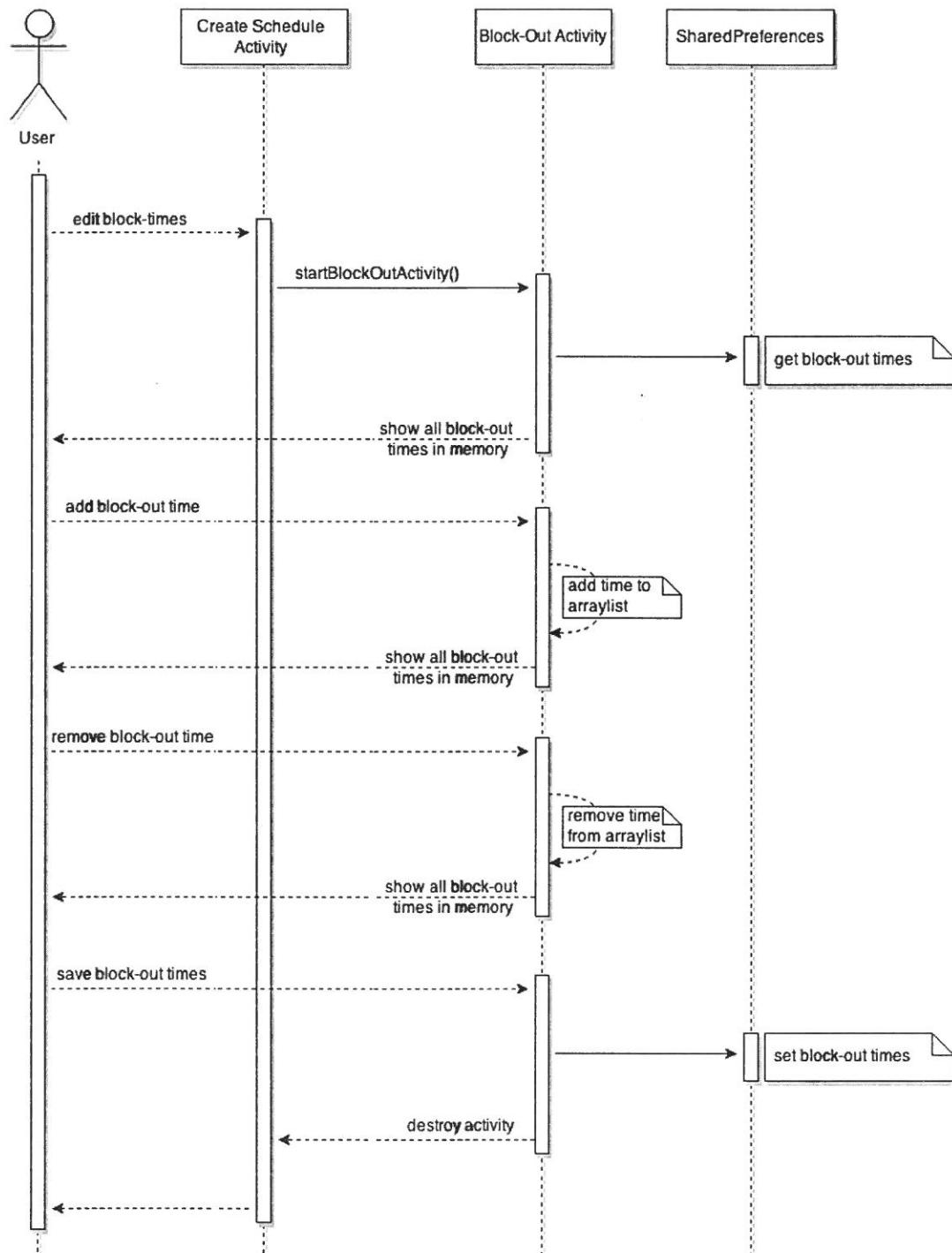
5.2.5 CREATE SCHEDULE SEQUENCE DIAGRAM



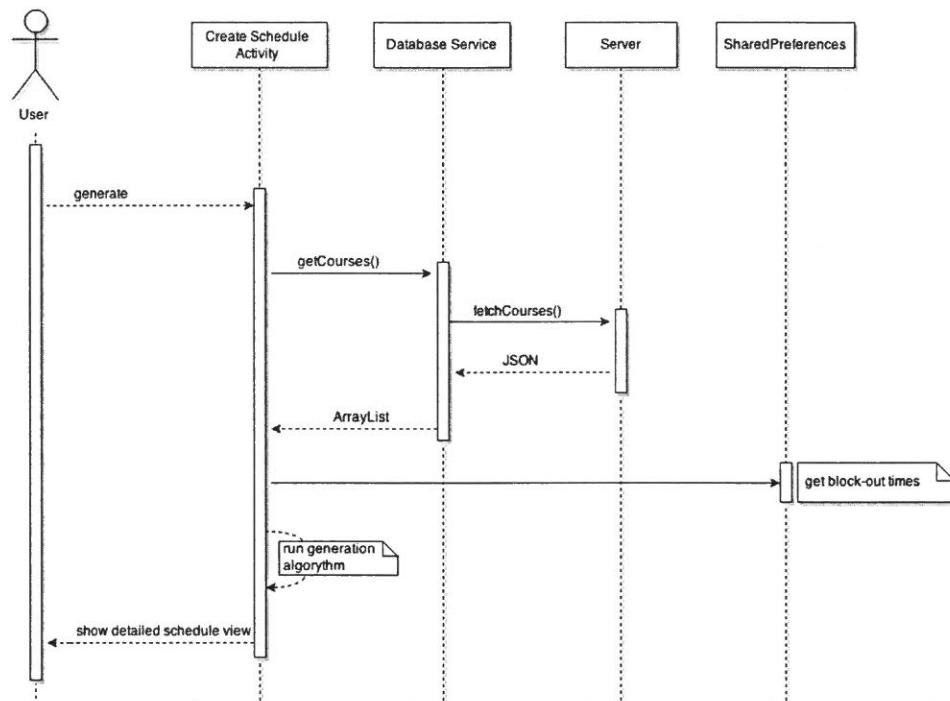
5.2.6 ADD COURSE SEQUENCE DIAGRAM



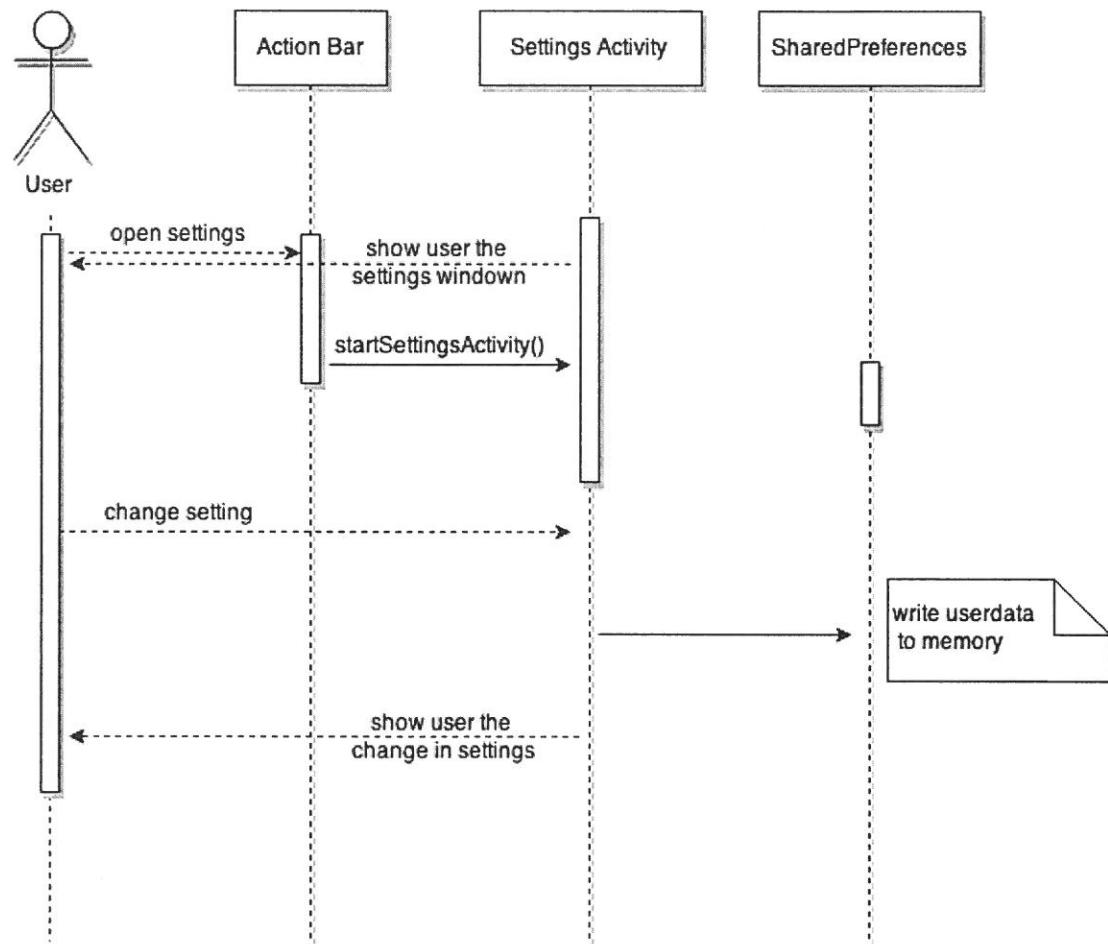
5.2.7 ADD BLOCK-OUT TIME SEQUENCE DIAGRAM



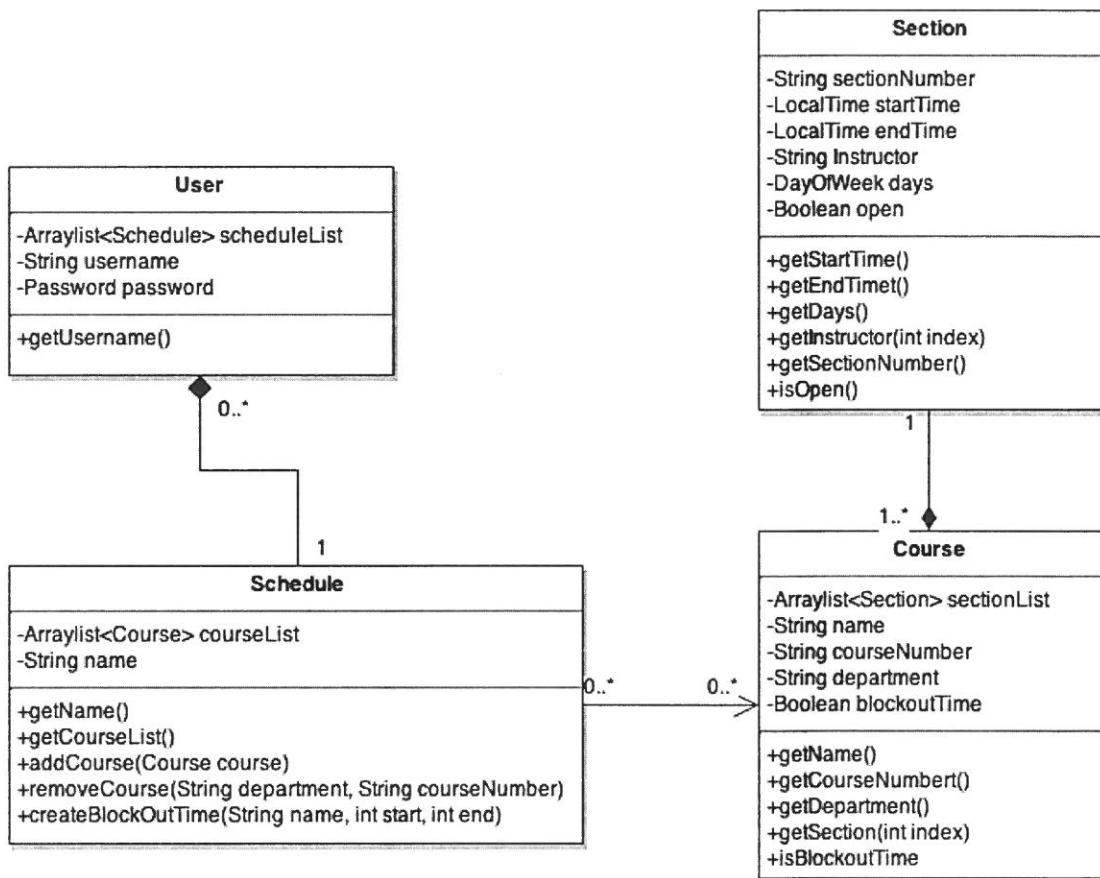
5.2.8 GENERATE SCHEDULE SEQUENCE DIAGRAM



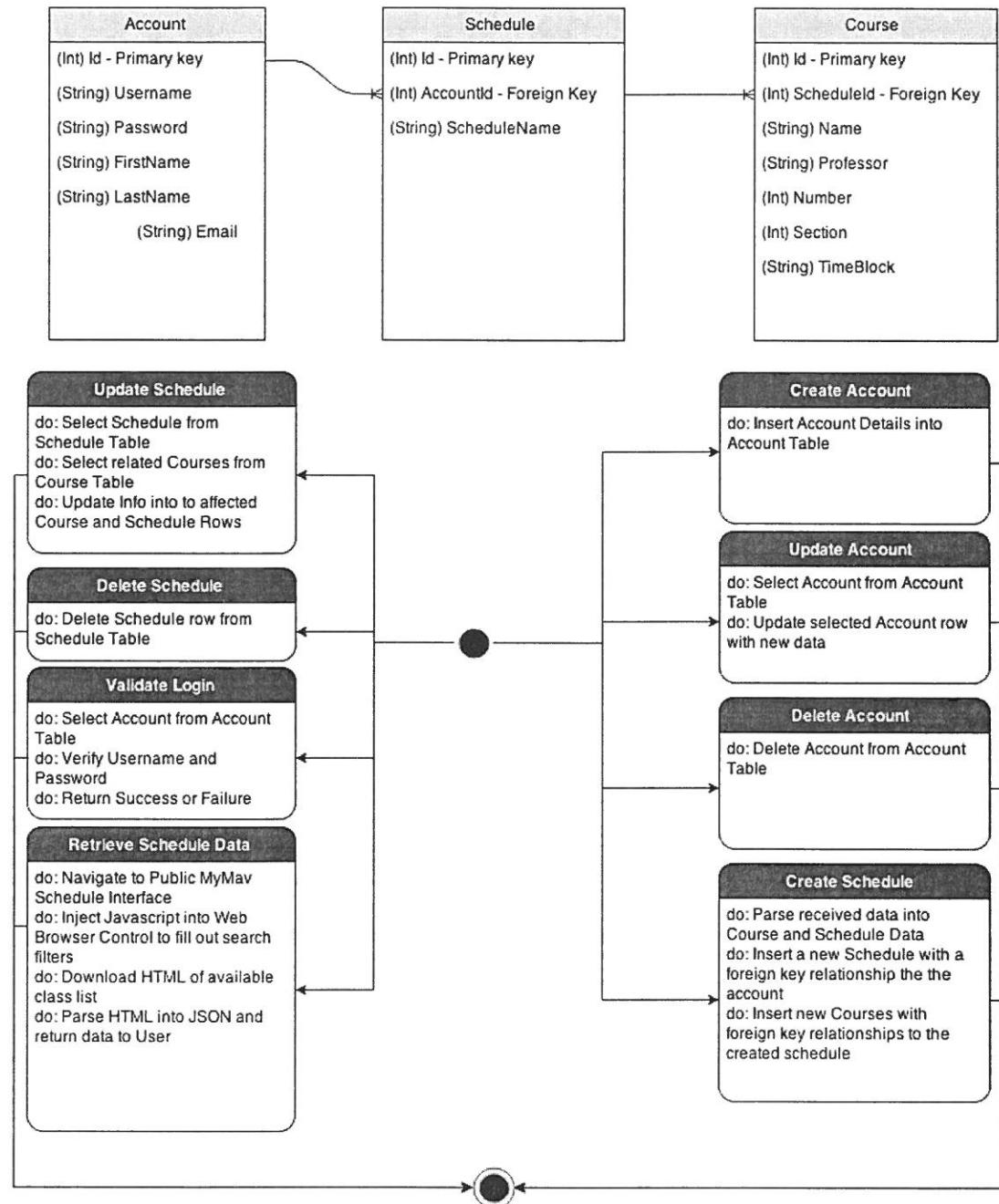
5.2.9 SETTINGS SEQUENCE DIAGRAM



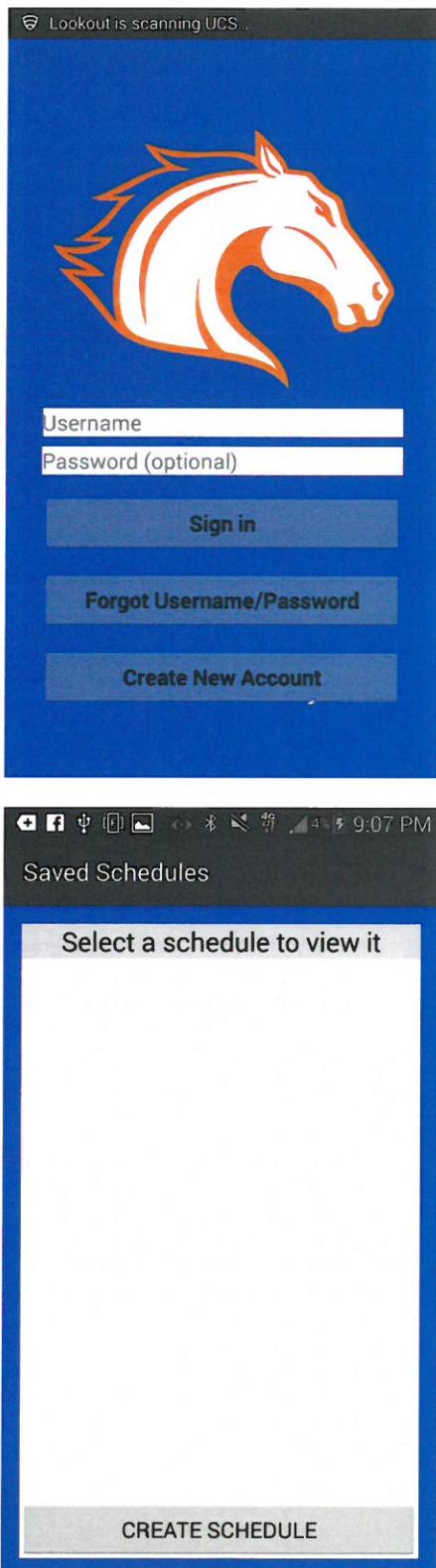
5.9.10 DOMAIN MODEL



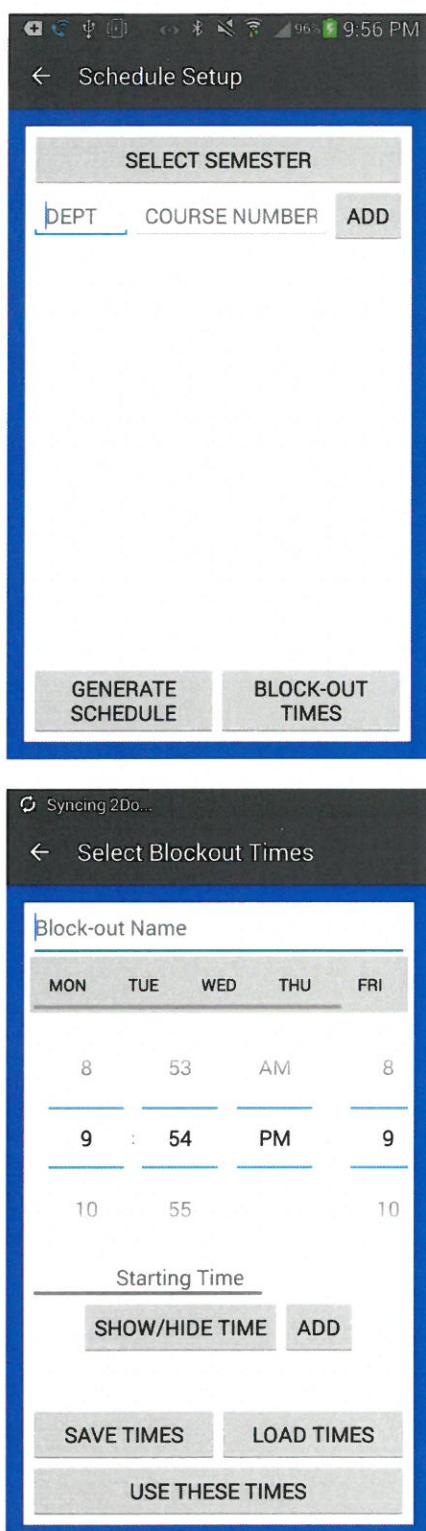
5.3 CONCEPTUAL DATA MODEL - DATABASE



5.4 SCREEN SHOTS



University Course Scheduler - SRA Document



5.5 TEST PLAN

Project: University Course Scheduler

Team No.: Alpha

Class: CSE 3310; Spring 2015

Module: Test Plan

Deliverable: Test Plan Document

Version: [0.9]

Date: [04/23/2015]

Revision History

<i>Version number</i>	<i>Date</i>	<i>Originator</i>	<i>Reason for change</i>	<i>High level description of changes</i>
0.1	04/10/2015	Mason Wilkens	Initial draft	
0.2	04/12/2015	Mason Wilkens	Revision	Added components
0.3	04/17/2015	Mason Wilkens/Eric Katz	Revision	Created test cases 2-5
0.4	04/21/2015	Eric Katz	Revision	Created test cases 6-9
0.5	04/21/2015	Peyton Casper	Revision	Created test cases 10-12
0.6	4/22/2015	Arun Kalahasti	Revision	Created test cases 12-15
0.7	04/22/2015	Mason Wilkens	Revision	Created test cases 17-18
0.8	4/23/2015	Arun Kalahasti/Eric Katz	Revision	Added test cases, re-organized numbering, created links
0.9	4/23/2015	Group	Revision	Final revisions and formatting

TABLE OF CONTENTS

- 1. INTRODUCTION AND PLAN OF APPROACH**
- 2. TEST CASES: “LOGIN AND ENTRY”**
- 3. TEST CASES: “FORGOT PASSWORD”**
- 4. TEST CASES: “CREATE ACCOUNT”**
- 5. TEST CASES: “TIME FORMAT”**
- 6. TEST CASES: “DELETE ACCOUNT”**
- 7. TEST CASES: “MODIFY ACCOUNT SETTINGS”**
- 8. TEST CASES: “CHANGE EMAIL”**
- 9. TEST CASES: “CHANGE PASSWORD”**
- 10. TEST CASES: “ADD BLOCK OUT TIME”**
- 11. TEST CASES: “REMOVE BLOCK OUT TIME”**
- 12. TEST CASES: “SAVE BLOCK OUT TIMES”**
- 13. TEST CASES: “LOAD BLOCK OUT TIMES”**
- 14. TEST CASES: “SERVER”**
- 15. TEST CASES: “PICK SCHEDULE”**
- 16. TEST CASES: “VIEW SCHEDULE”**
- 17. TEST CASES: “GENERATE SCHEDULE”**
- 18. TEST CASES: “VERIFY SCHEDULE”**
- 19. TEST CASES: “ADD COURSE”**
- 20. TEST CASES: “REMOVE COURSE”**

1. Project Overview and Plan of Approach

The application will be developed to run on Android version 4.2+. The software has two major components – the client application and the server database. Users will interact with the client application on an Android device. The server database is used by the client application to obtain and format course data and store user data.

Users will be required to make an account to access the functionality of the software. The client application will allow the user to create, edit, and delete an account to gain access. Once the user has successfully created an account and gained access to the software, they will be able to generate, save, and delete schedules. In generating a schedule, the user will be able to add and remove block-out times, and add and remove courses. Schedules will be generated and presented to the user such that there are no conflicts between the block-out times and courses the user has specified. When a user finds a course schedule they wish to keep, they can save it to their account. The user can verify saved schedules are still valid through the application.

The server will host an SQL database that will store user data including username, password, and saved schedules. The database will also store course information that will be accessed by the client application. The database will obtain course information by scraping information from UTA's website MyMav.

We assume scraped data is valid and up to date (i.e. MyMav). All testing will be done through a combination of Junit testing in Android Studio as well as manual testing on Phone/tablet.

List of Components:

1. Login and Entry Page
2. Forgot Password Page
3. Create Account Page
4. Account Settings
5. Block Out Time Page
6. Course Selection Page
7. Select Schedule Page
8. Detailed Schedule View
9. Server

Testing Tools:

1. Android Studio
2. Genymotion Android Emulator

3. Galaxy Note 3
4. Sony Xperia Tablet Z1

2. Test Cases: “Login and Entry”

Project Name:

UCS

Test Case Name:

Login and Entry

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Login_and_Entry

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Tab into the User and password fields and enter a valid user id/password (a test id should be created and provided to testers)	System should let you in	
TC2	Tab into the User and password fields and enter an invalid user id/password	System should not accept and prevent you from entry	

3. Test Cases: “Forgot Password”

Project Name:

UCS

Test Case Name:

Forgot Password

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Forgot_Password

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Tab into email field and enter valid email	System should change current password to temporary password and send the temporary password to the email. System should notify the user the password has been changed and sent.	
TC2	Tab into email field and enter invalid email	System should not change current password to temporary password and not send a temporary password to the email. System should notify the user the email is invalid.	

4. Test Cases: “Create Account”

Project Name:

UCS

Test Case Name:

Create Account

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Create_Account

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Tab into username, password, and email fields and enter a valid username, password, and email.	System should create a new account.	
TC2	Tab into username, password, and email fields and enter a username that is already taken.	System should not create a new account and notify the user the account name is already taken.	
TC3	Tab into username, password, and email fields and enter an email that is already taken.	System should not create a new account and notify the user this email address is already taken.	
TC4	Tab into username, password, and email fields and enter an invalid username, password, and email.	System should not create a new account and notify the user one of the fields were invalid.	

5. Test Cases: “Time Format”

Project Name:

UCS

Test Case Name:

Time Format

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Time_Format

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Toggle time format to 12-hour time	System should display course and schedule time in 12-hour format.	
TC2	Toggle time format to 24-hour time	System should display course and schedule time in 24-hour format.	

6. Test Cases: “Delete Account”

Project Name:

UCS

Test Case Name:

Delete Account

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Delete_Account

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Select to Delete Account. (a test id should be created and provided to testers)	Account should be Deleted from database.	

7. Test Cases: “Modify Account Settings”

Project Name:

UCS

Test Case Name:

Modify Account

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Modify_Account

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Account settings are changed on account.	New account settings should replace old account settings, old account settings are no longer effect.	

8. Test Cases: “Change Email”

Project Name:

UCS

Test Case Name:

Change Email

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Change_Email

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Input a new valid email address.	Email associated with user account should be changed.	
TC2	Input a new invalid email address.	Email address should be not changed.	

9. Test Cases: “Change Password”

Project Name:

UCS

Test Case Name:

Change Password

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Change_Password

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Attempt to change password to a new valid password.	System should change user password.	
TC2	Attempt to change password to a new invalid password.	System should not change user password.	

10. Test Cases: “Add Block Out Time”

Project Name:

UCS

Test Case Name:

Add Block Out Time

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Add_Block_Out_Time

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Don't specify any days for the block out time to take effect.	System should prevent the user from adding the block out time.	
TC2	Leave block out name empty.	System should prevent the user from adding the block out time.	
TC3	Don't specify a time for the block out time to take effect.	System should prevent the user from adding the block out time.	
TC4	Add block out time with all required fields	System should add the block out time to the list of block out times.	

11. Test Cases: “Remove Block Out Time”

Project Name: UCS
Test Case Name: Remove Block Out Time
Test Case Id: CSE3310/Spring 2015/Team-Alpha/
Remove_Block_Out_Time

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Attempt to remove block out time. (A sample block out time to remove should be provided to testers)	System should remove selected block out time from the block out time list.	

12. Test Cases: “Save Block Out Times”

Project Name: UCS
Test Case Name: Save Block Out Time
Test Case Id: CSE3310/Spring 2015/Team-Alpha/
Save_Block_Out_Time

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Select save block out button when there are block out times in memory.	System should save block out times as JSON to SharedPrefs.	
TC2	Select save button when there are no block out times in memory.	System should not save block out times as JSON to SharedPrefs.	

13. Test Cases: “Load Block Out Times”

Project Name: UCS
Test Case Name: Load Block Out Time
Test Case Id: CSE3310/Spring 2015/Team-Alpha/
Load_Block_Out_Time

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Select block out time name to load. (Provide a sample block out time to load)	System should load block out time with matching name from SharedPrefs.	
TC2	Select to load block out time, but no block out times exist.	System should notify the user there are no block out times to load.	

14. Test Cases: “Server”

Project Name:

UCS

Test Case Name:

Server

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/ Server

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Submit a request to add a schedule.	A new schedule row should be added and associated courses should be created as new rows in the courses table.	
TC2	Submit a request to remove a schedule.	The specified schedule row should be deleted along with any related course rows..	
TC3	Submit a request to create a user.	A new user row should be added to the database.	
TC4	Submit a request to edit a user.	The specified user row should be updated with the new data.	
TC5	Submit a request to delete a user.	The specified user row should be removed from the database.	
TC6	Submit a request to pull class status from MyMav.	The returned data should match the data returned from manually searching MyMav.	

University Course Scheduler - SRA Document

TC7	Submit a request to pull department list from UTA website.	The returned data should match the department list from the UTA website.	
TC8	Pull all courses by department and semester from MyMav.	The returned data should match a manual search on MyMav with the same filters.	

15. Test Cases: “Pick Schedule”

Project Name:

UCS

Test Case Name:

Pick Schedule

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/ Pick_Schedule

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Select a schedule. (Provide a schedule to the tester)	Old selection should be discarded and this schedule should be selected.	

16. Test Cases: “View Schedule”

Project Name:

UCS

Test Case Name:

View Schedule

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/View_Schedule

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Select a schedule to view. (Provide a schedule to the tester)	System should show a detailed view of the selected schedule.	

17. Test Cases: “Generate Schedule”

Project Name:

UCS

Test Case Name:

Generate Schedule

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Generate_Schedule

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Generate with 2 courses, each having only one section each, which do not conflict	System should successfully generate a schedule	
TC2	Generate with 2 courses which are mutually exclusive	System should notify user that no possible schedule is possible	
TC3	Generate with 2 courses, one of which has only one section, while the other will have a section which conflicts with the first course and a section which does not conflict with the other section	System should successfully generate a schedule	

18. Test Cases: “Verify Schedule”

Project Name:

UCS

Test Case Name:

Verify Schedule

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Verify_Schedule

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Verify a schedule where all sections are known to be open	System should inform user all courses are still available for enrollment	
TC2	Verify a schedule where one section has closed	System should inform user which course(s) are not available for enrollment	
TC3	Verify a schedule where all sections are closed	System should inform user which course(s) are not available for enrollment	

19. Test Cases: “Add Course”

Project Name:

UCS

Test Case Name:

Add Course

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Add_Course

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Tab into department and course number fields and enter a valid department and course number	System should add course to non-generated schedule.	
TC2	Tab into department and course number fields and enter an invalid department and course number	System should not add course to non-generated schedule.	

20. Test Cases: “Remove Course”

Project Name:

UCS

Test Case Name:

Remove Course

Test Case Id:

CSE3310/Spring 2015/Team-Alpha/Remove_Course

Test Case No.	Test Case Description	Expected results	Outcome Pass, Fail, Other (comments)
TC1	Attempt to remove a course. (A sample course to remove should be provided to testers)	System should remove course from schedule.	

6. Assumptions and Constraints

6.1 ASSUMPTIONS

The following is a list of assumptions:

- User has a fully functional android device.
- User is 18 or older.
- User knows the department and course number or the course name of the desired course.

6.2 CONSTRAINTS

The following is a list of constraints:

- Copyright laws

6.3 OUT OF SCOPE MATERIAL

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

- Post Project maintenance is not covered

7. Delivery and Schedule

Task/Milestone Description	Anticipated Start Date	Anticipated End Date	Status	Comments
Prepare Requirements and UML diagram	2/3/2015	3/1/2015	Completed	
SRA document (Includes project objectives, Requirements and UML diagrams)	3/1/2015	3/20/2015	In Progress	Deliverable will be the SRA document. All stakeholders agree on the content of the SRA by signing in section 8.
Presentation of SRA	3/20/2015	3/31/2015	Completed	Pushed back to 3/31/2014
Database Design	02/03/15	03/15/2015	Completed	A SQL database will be used.
Test Data Entry	03/10/2015	04/23/2015	In Progress	
Test Plan Delivery	04/03/2015	04/23/2015	In Progress	
External Documentation (i.e. User Manual)				
Final Milestone: project delivery		5/3/2015		

8. Stakeholder Approval Form

Stakeholder Name	Stakeholder Role	Stakeholder Comments	Stakeholder Approval Signature and Date
Joseph Wigner	Development Mgr		
Elise Austin	Project Assistant		
Eric Katz	Developer		
Mason Wilkens	Developer		
Peyton Casper	Developer		
Arun Kalahasti	Developer		

Appendix:

None



University Course Scheduler

USER MANUAL

For Version 1.0

Contents

Application Access.....	2
Chapter 1 - Login	3
Creating an Account.....	4
Forgot/Reset Password.....	5
Signing in	5
Chapter 2 - Saved Schedules	5
Chapter 3 - Schedule Generation	7
Selecting a Semester	8
Adding a Course	9
Removing a Course.....	9
Creating a Schedule.....	10
Chapter 4 - Select Block-Out Times.....	11
Add a Block-Out Time	12
Save Block-Out Times.....	12
Use Block-Out Times	13
Chapter 5 - Detailed Schedule View	14
Viewing a Schedule.....	14
Deleting a Saved Schedule	14
Verifying a Saved Schedule is Valid.....	14
Swap Section	15
Chapter 6 - Settings.....	16
Use Military Times.....	17
Update Password.....	17
Update Email.....	17
Delete Account.....	17
Allow Non-Open Courses.....	17

Application Access

To launch the application, find and tap the icon in your android device. To access the application, you will be required to login to the application. If you do not have an account you must create one to login.

You must have an android device which runs at least android version 4.2, also known as Jelly Bean, to properly run this application.

NOTE: All functionality of the application is dependent on internet access.

NOTE: Screenshots may appear differently in the app, but all functionality will remain the same.

Chapter 1 - Login

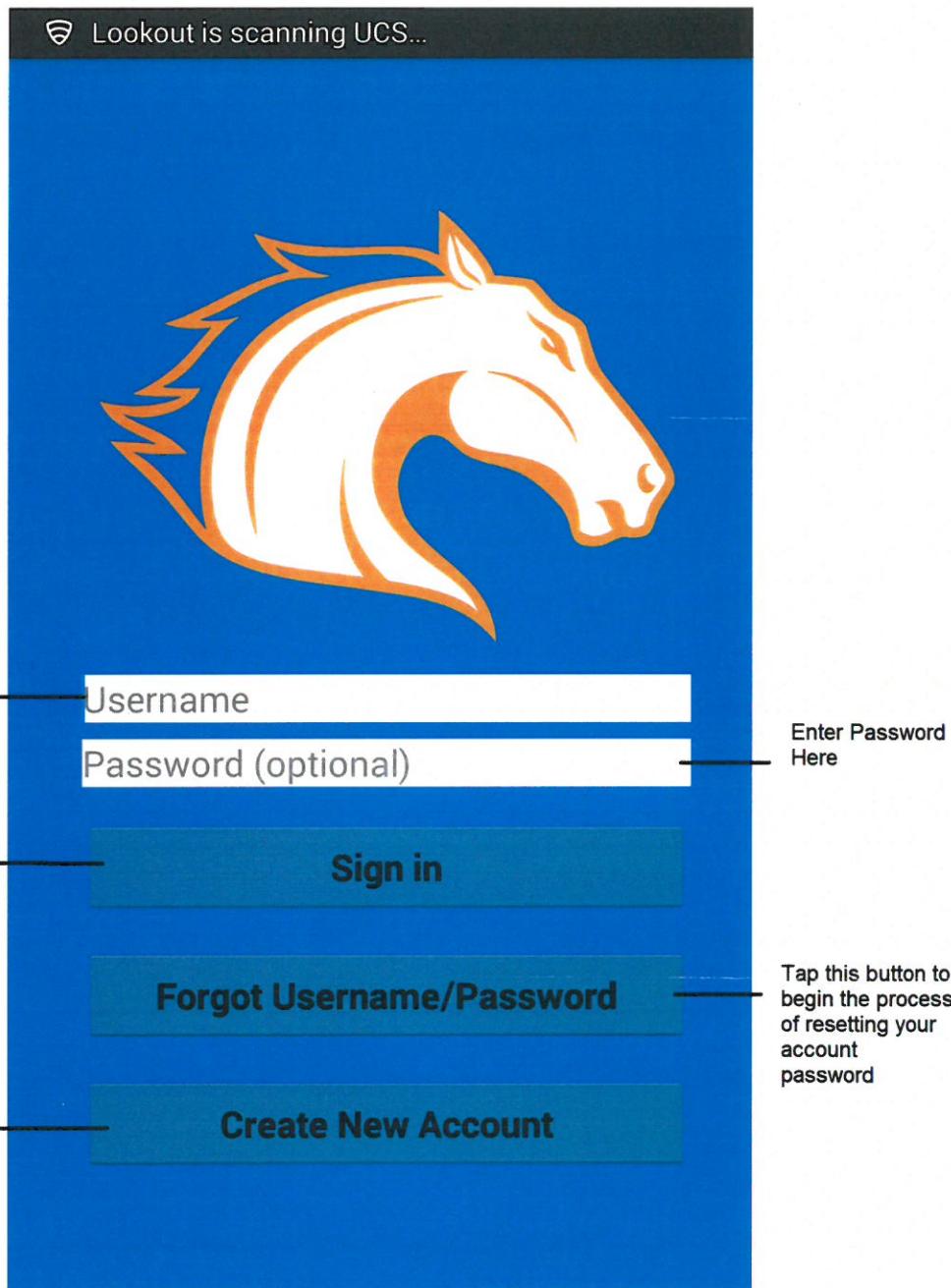


Figure 1-1 Login Page

The login page is the initial page the application displays. From this page, you can **Create an Account**, **Sign in** to an existing account, or if you **Forgot Username/Password** you can reset your password.

Creating an Account

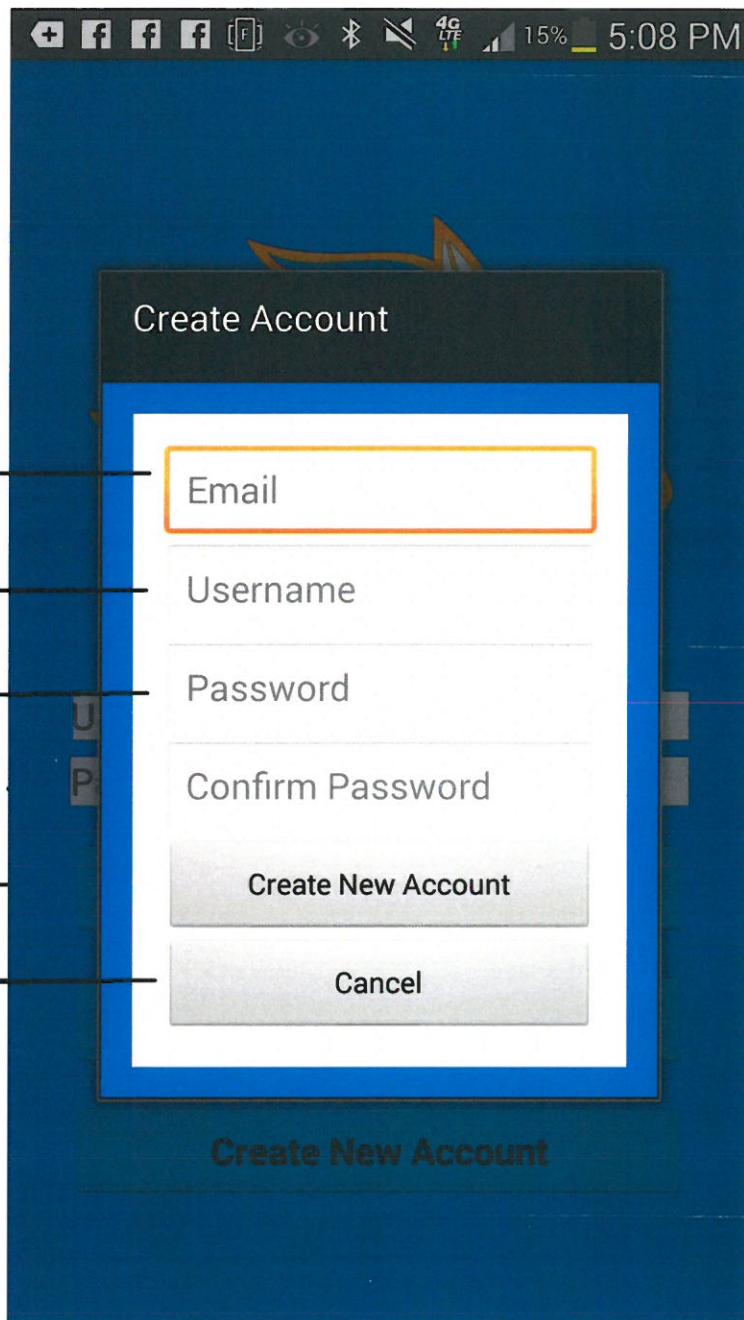


Figure 1-2 Create New Account Popup

Once the application has been launched, from the login page tap on **Create New Account** button to begin creating an account. Enter in an Email, Username, and Password in the corresponding fields and tap **Create New Account**. If at any time you wish to discontinue account creation, tap **Cancel** and you will be taken back to the Login Page.

Forgot/Reset Password

To reset a forgotten password, from the login page, first tap **Forgot Password**. You will be presented with a popup dialog. Input the email address associated with your account, then tap **OK**, and an email will be sent with a new password. If you are in this pop-up prompt to add an email address, and you remember your password or do not wish to have your password reset, tap **Cancel** and you will be taken back to the login page.

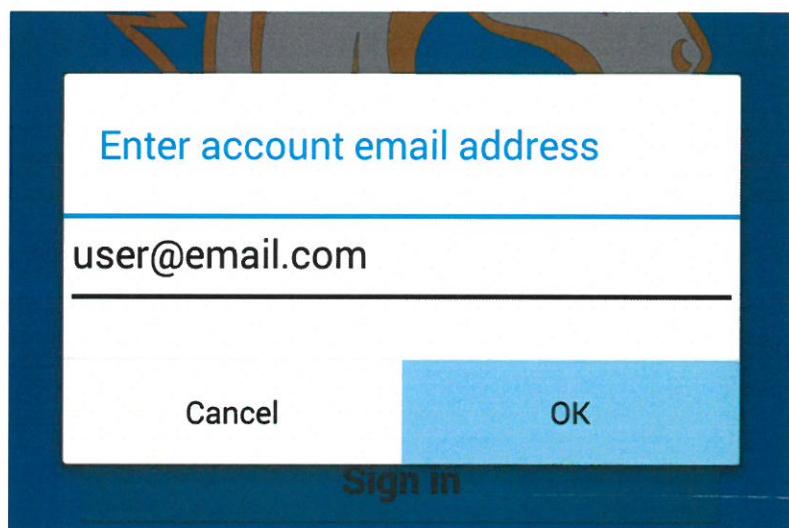


Figure 1-3 Forgot Password Popup

Signing in

Once the application has been launched, and you have creating or already had an existing user account, you can sign in to access the application. Enter your Username and Password into the corresponding fields and tap **Sign in**. Upon successful login, you will be taken to the **Saved Schedules** page.

Chapter 2 - Saved Schedules

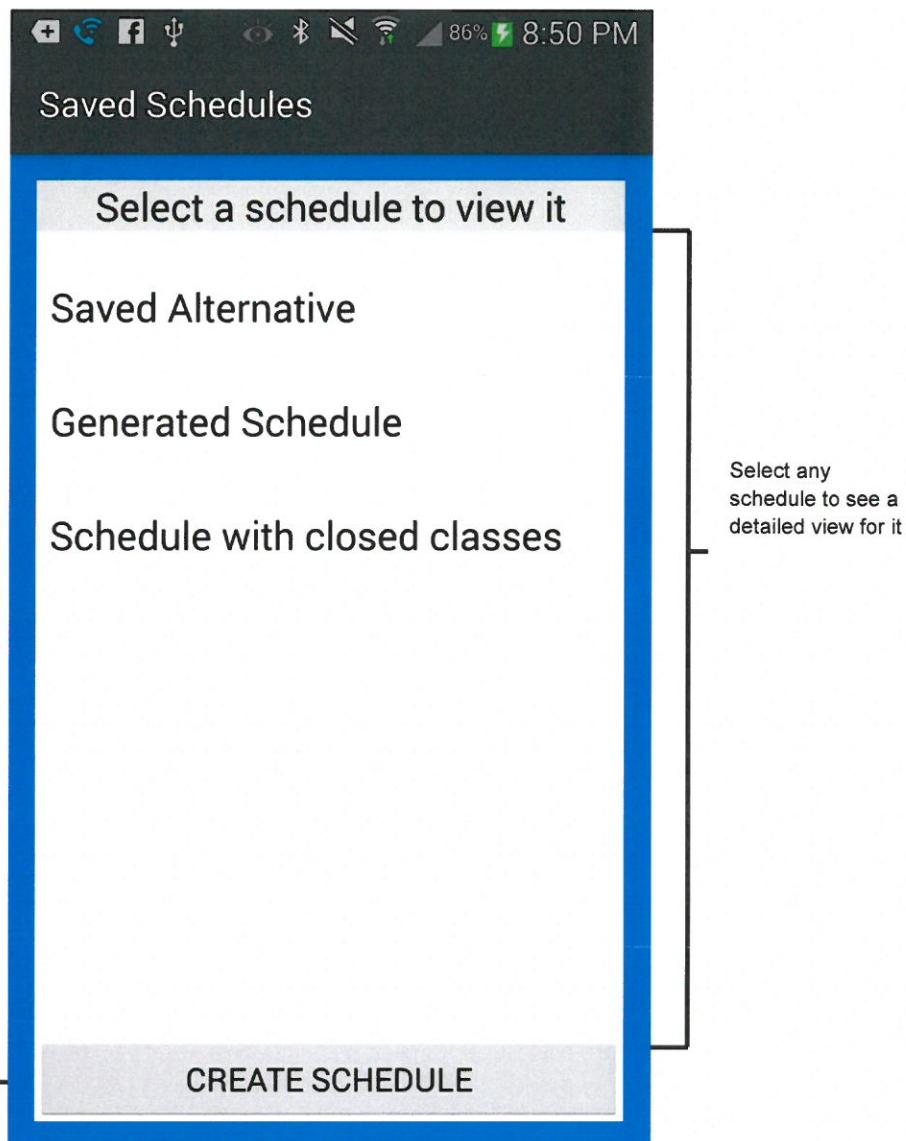


Figure 2-1 Saved Schedules Page

Once you have signed into the application you will be presented with the saved schedule page. Here you can select any schedules you have saved to see a Detailed View of the schedule, along with the associated options, or you can create a new schedule.

NOTE: If you are signing in for the first time after creating an account you will not be shown any schedules in this page until you have created and saved one.

In order to generate a schedule, you will need to be signed in. Once signed in, tap **Create Schedule** in the Saved Schedules page. You will then be able to begin schedule generation.

Chapter 3 - Schedule Generation

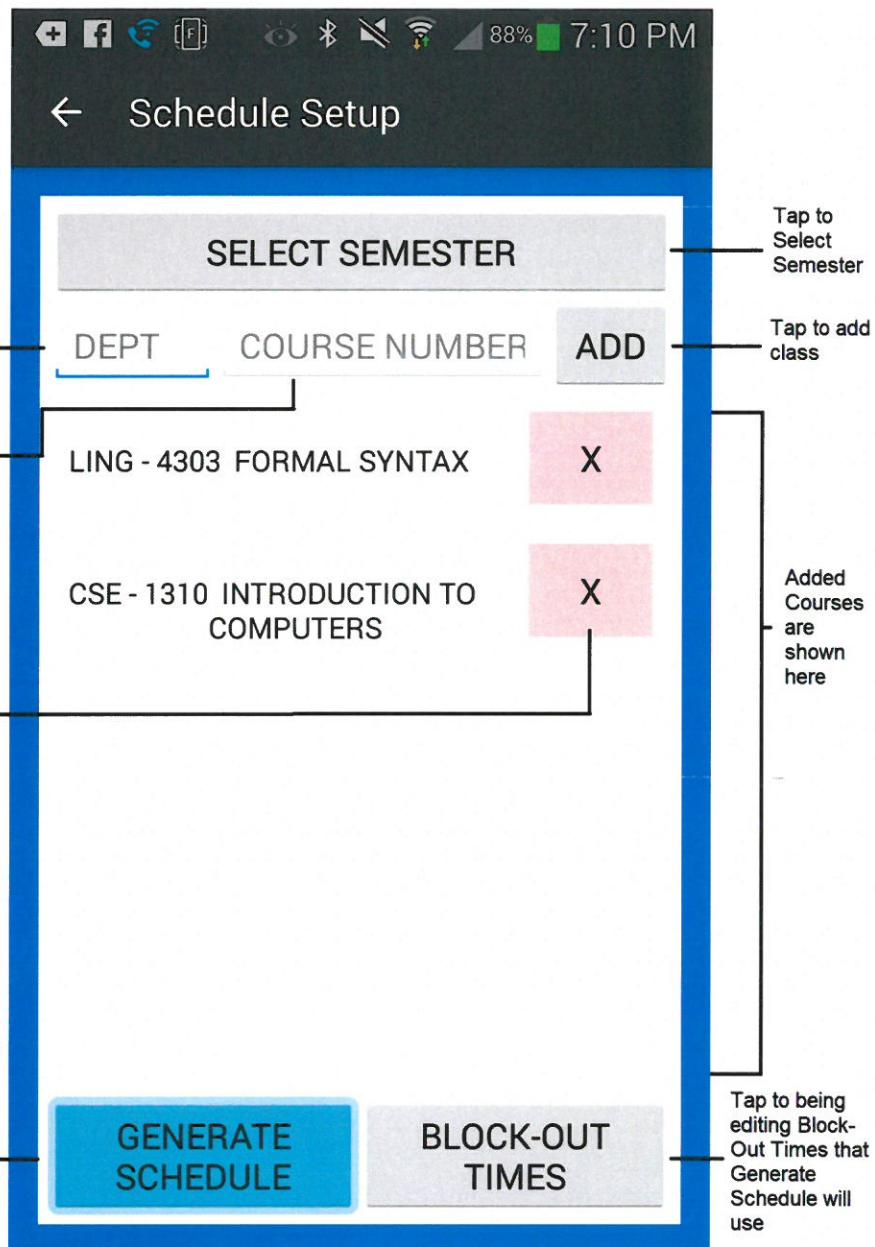


Figure 3-1 Generate Schedule Page

This page allows you to **find, add, and remove courses** to generate schedules with. It will also be where you can generate schedules using the application. In the Saved Schedules page, tap **Generate Schedule** to get here.

NOTE: The first time you open the schedule generation page it will automatically update the classes available. This process may take up to a minute, but does not need to be repeated unless you wish to update the data manually.

Selecting a Semester

This option allows you to choose which semester the schedule will be built for. In the Schedule Setup page, tap **Select Schedule**. Select the semester you want to make a schedule for by tapping one of the provided semesters.

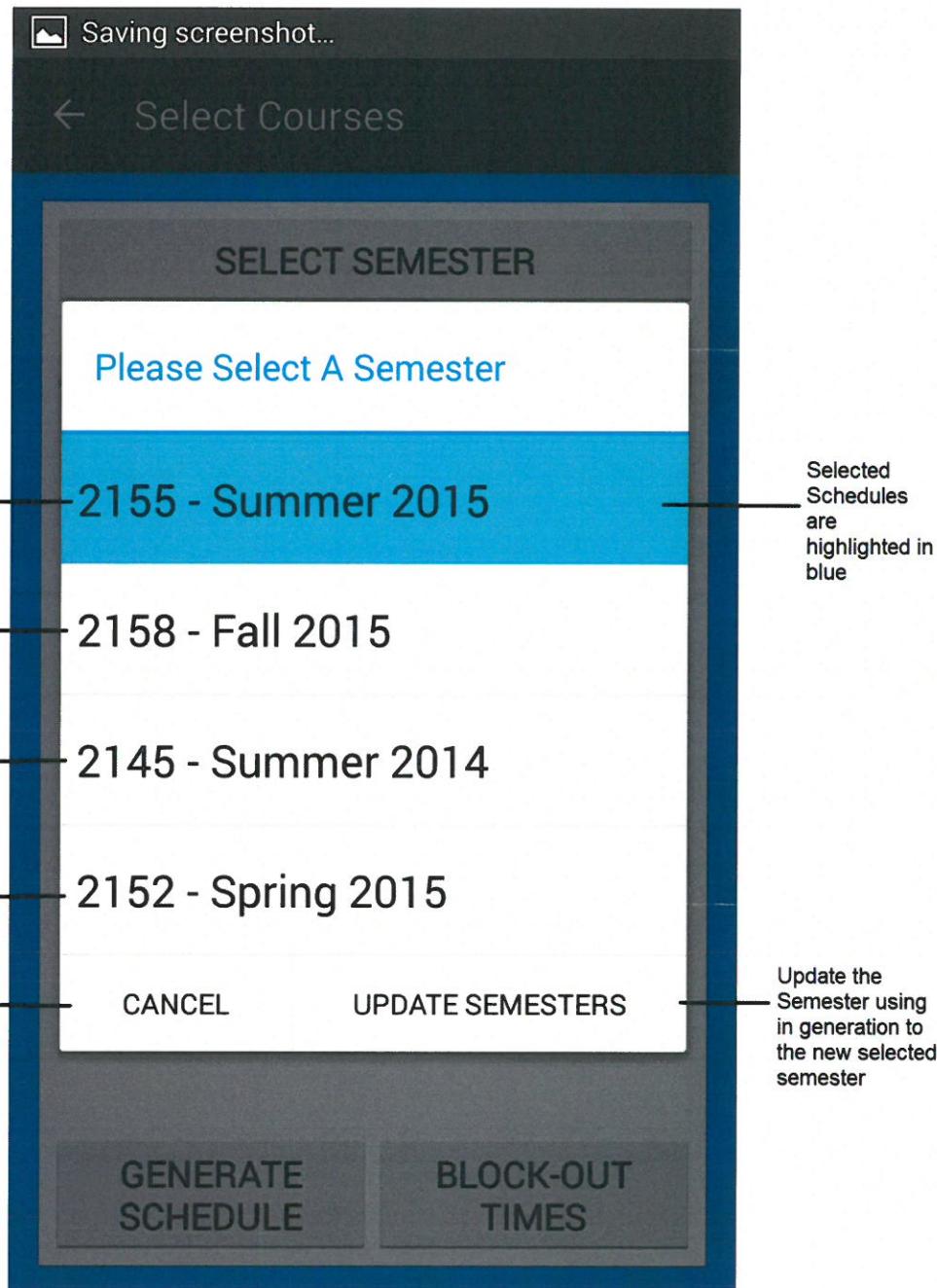


Figure 3-2 Select Semester Popup

Adding a Course

Once in the Schedule Generation page, to add a desired Course to a schedule, enter the department and the course number of the Course you wish to add.

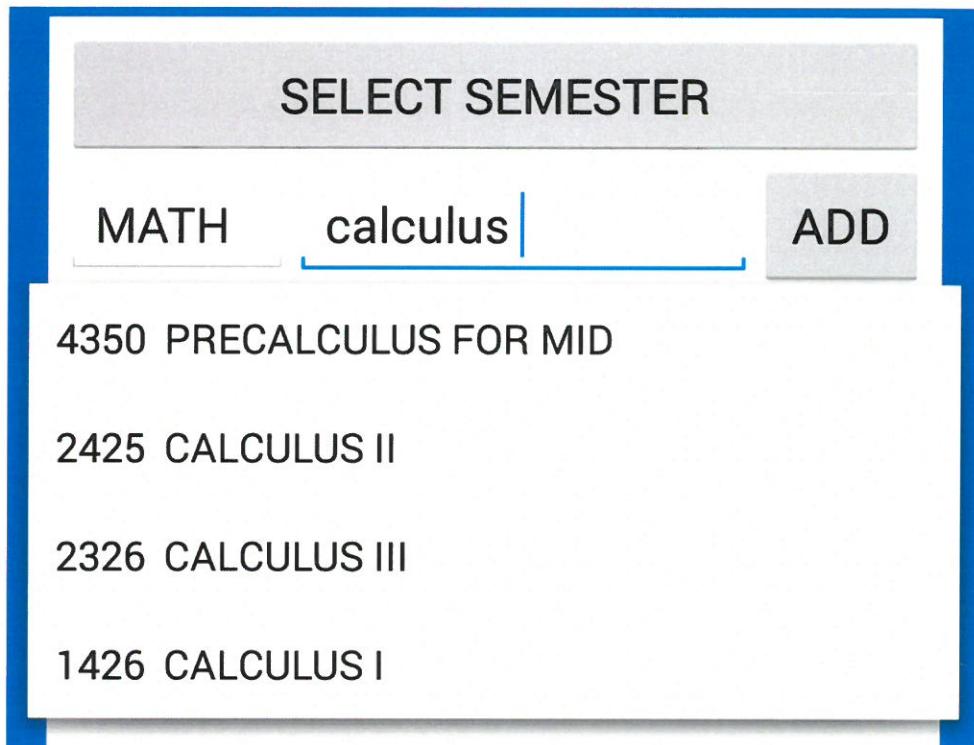


Figure 3-3 Entering A Course by Name

NOTE: A dropdown will appear below the text fields as you type to help you find departments and courses. If you search by name select an item from the dropdown to ensure the application can find the associated course.

Once the department and course number has been selected, tap **add**. This will add the corresponding course to the list of courses and be displayed in a list below. If one of the fields are blank or do not match an existing course, no course will be added to the list.

Removing a Course

This option allows you to remove a course from the generation. In the Schedule Setup page, in order to remove a Course from the list, tap the red box next to the course you want to remove to remove it from the list.

Creating a Schedule

This option allows you to create a Schedule. To create a Schedule, first navigate to Schedule Generation and add at least one class. If there is at least one class in the list, tap **GENERATE SCHEDULE** to attempt to create a schedule. The application will ask you to wait while courses are retrieved.

If a schedule is successfully generated, it will be displayed in a Detailed Schedule View popup similar to the one shown in the Detailed View section.

If a schedule cannot be generated because of conflicts you will be presented with the following dialog where you can either change the courses you have selected to avoid the conflict or ignore the problems.

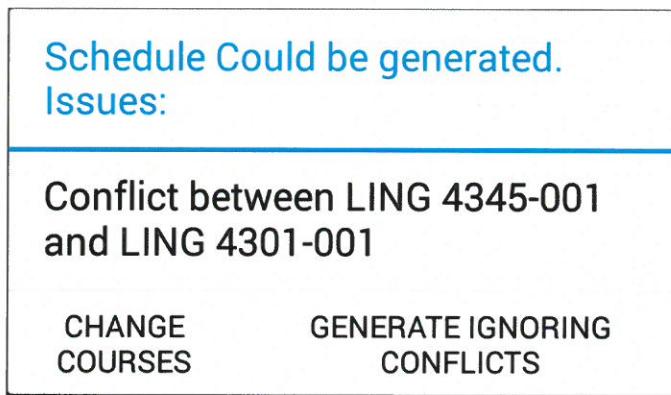


Figure 3-4 Example of a Schedule Conflict Popup

Chapter 4 - Select Block-Out Times

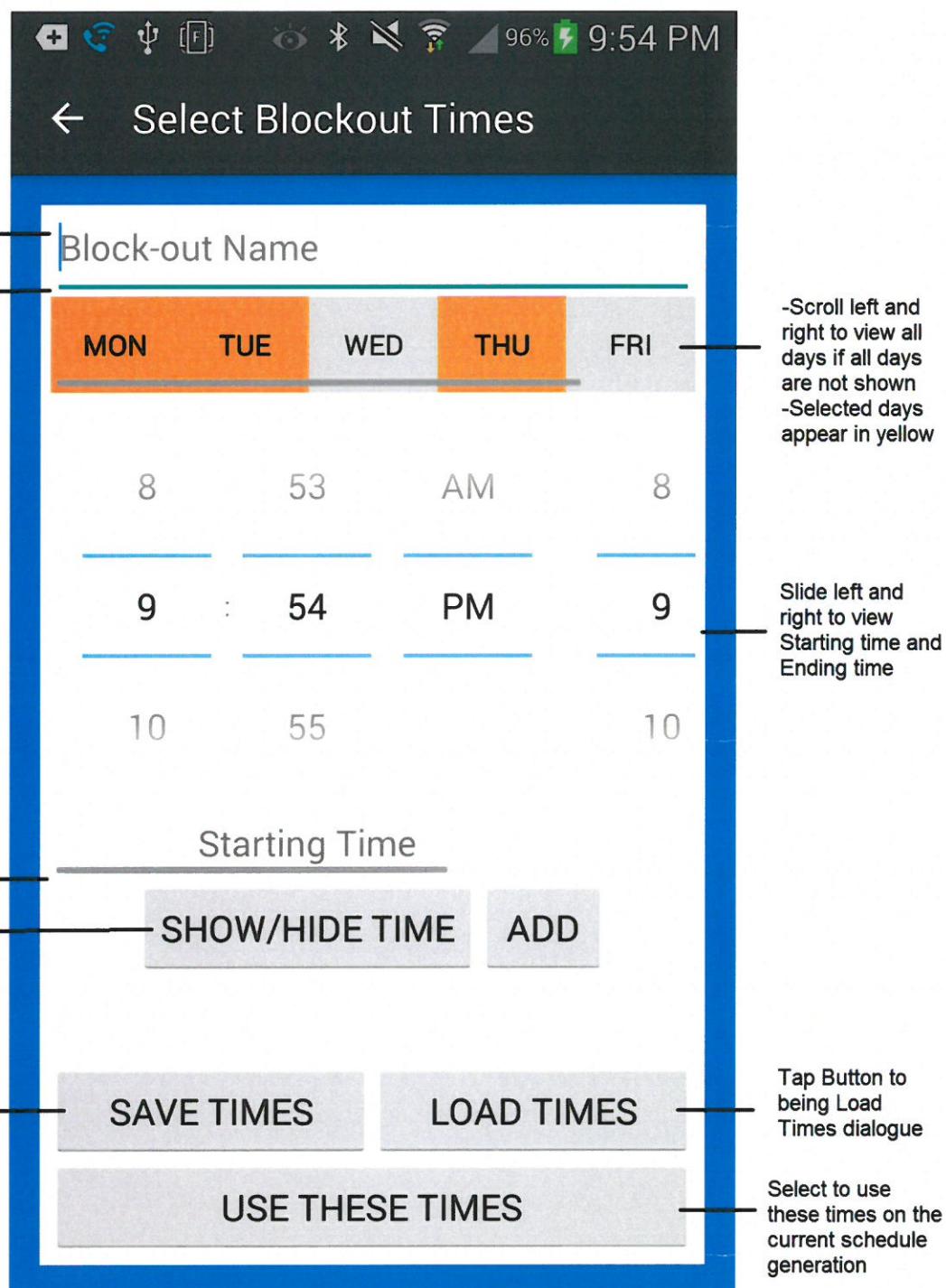


Figure 4-1 Select Block-Out Times Page

This page allows you to **add, remove, save, load, and use block-out times**. In the Schedule Setup page, tap **Block-Out Times** to get to this page.

Add a Block-Out Time

In the Select Block-Out Time page, enter a block-out name, select the days of the week, the starting time, and the ending time for the block-out time you want to create. When all information has been entered, tap **ADD** to add the block-out time to the list. Upon successful add, you should see the block-out time name in the list below the add button.

NOTE: You must set a name and have at least one day selected for the block-out time you wish to add. If you do not, the application will give you a warning and the time will not be added.

Save Block-Out Times

This option allows you to save a list of block-out times to be re-used at a later time. In the Select Block-Out Time page, tap **SAVE TIMES**. Enter a name for the Block-Out Times, then tap **OK**.

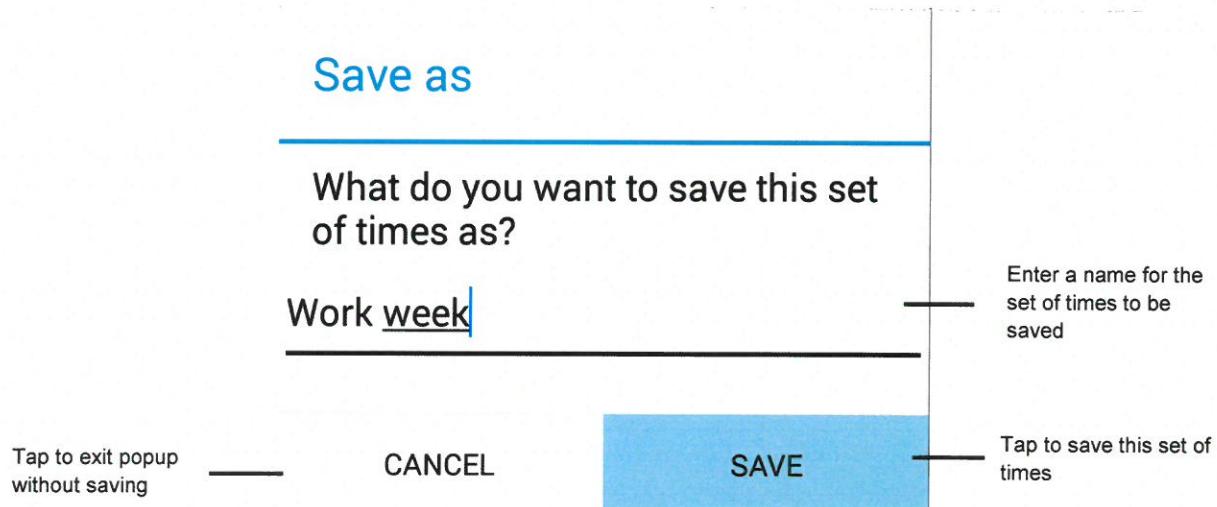


Figure 4-2 Save Block-Out Times Popup

Load Block-Out Times

This option allows you to load previously created and saved block-out times into the Block-Out Times display. In the Select Block-Out Time page, tap **LOAD TIMES**. You will be presented with a pop up similar to the one below where you can select previously created Block-Out Times. When the desired block-out times are selected by check box, tap **LOAD**. If there are times which you no longer wish to be shown you can select the delete button which will remove them.

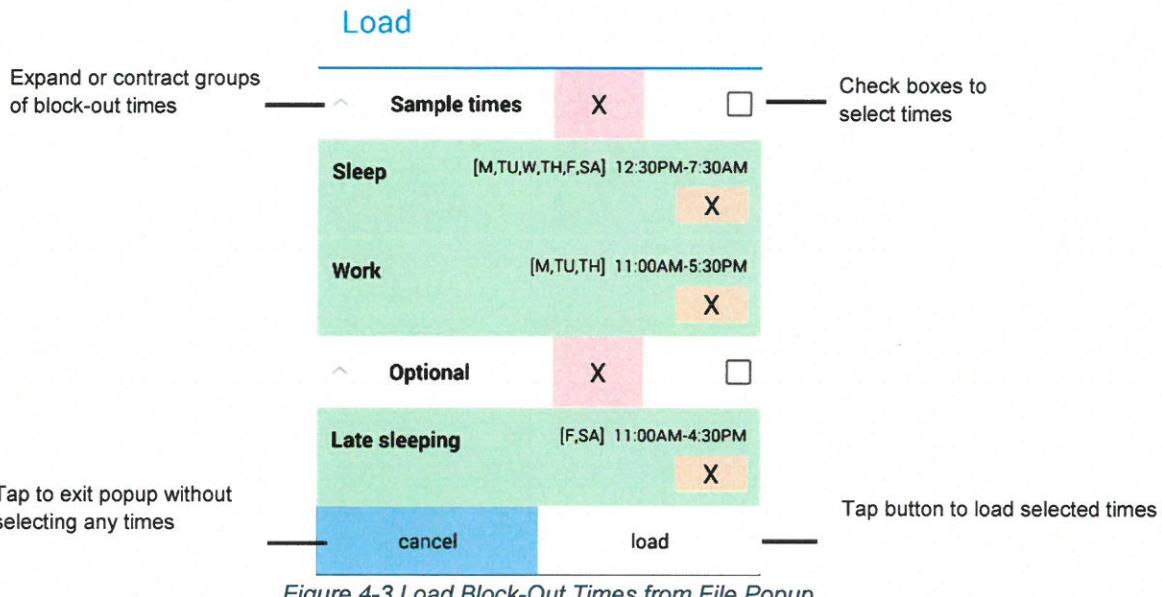


Figure 4-3 Load Block-Out Times from File Popup

Use Block-Out Times

This option allows you to use a list of block-out times in a schedule. This is the only way to set block-out times to have an effect on the schedule generation. To apply the list of block-out times to schedule generation, tap **USE THESE TIMES** in the Select Block-Out Time page.

Chapter 5 - Detailed Schedule View

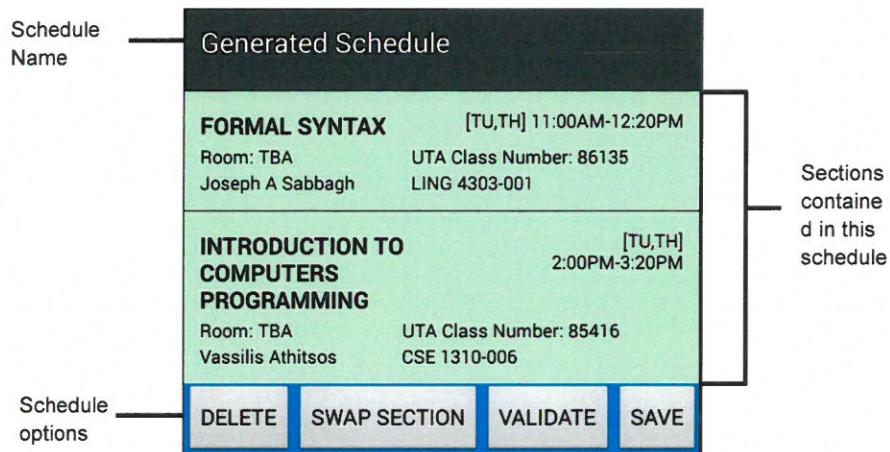


Figure 5-1 Detailed Schedule View

The Detailed Schedule View allows you to view schedules that you have saved. It will also allow you to **Delete**, **Validate**, **Save** the schedule it is displaying. You can also **Swap Sections** in the schedule for alternate sections of a course. Select any schedule on the **Saved Schedules** page or tap **Generate Schedule** on the Schedule Generation page to see a Detailed Schedule View.

Viewing a Schedule

To view a schedule tap a schedule in the saved schedules page or generate a new schedule in the Schedule Generation page.

Deleting a Saved Schedule

This option allows you to delete a saved schedule. To delete a schedule, first you need to View a Saved Schedule. While viewing a schedule you wish to delete, tap **DELETE**. You will then be presented with a dialog to confirm your deletion. Successfully deleting the schedule should remove it from the Saved Schedules Page.

Verifying a Saved Schedule is Valid

This option allows you to delete a saved schedule. To delete a schedule, first you need to View a Saved Schedule. While viewing a schedule you wish to verify, tap **VERIFY**. The application will ask you to wait while it checks statuses, then show a pop up informing you about any changes.

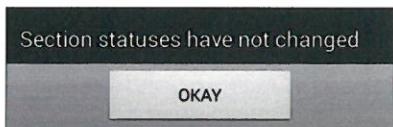


Figure 5-2 Verify Schedule shows no changes



Figure 5-3 Verify Schedule found status changes

Swap Section

This option allows you to swap a saved section. To swap a section, first you need to swap a saved schedule. While viewing a schedule you wish to swap a section in, tap **SWAP SECTIONS**. You will be asked which section you would like to swap out. Once you have selected a section, the application will retrieve all alternate sections for that course, color code them according to status and conflicts with other sections, and finally display them for you to select. If you select any of the options presented the schedule will be updated to show the new selection.

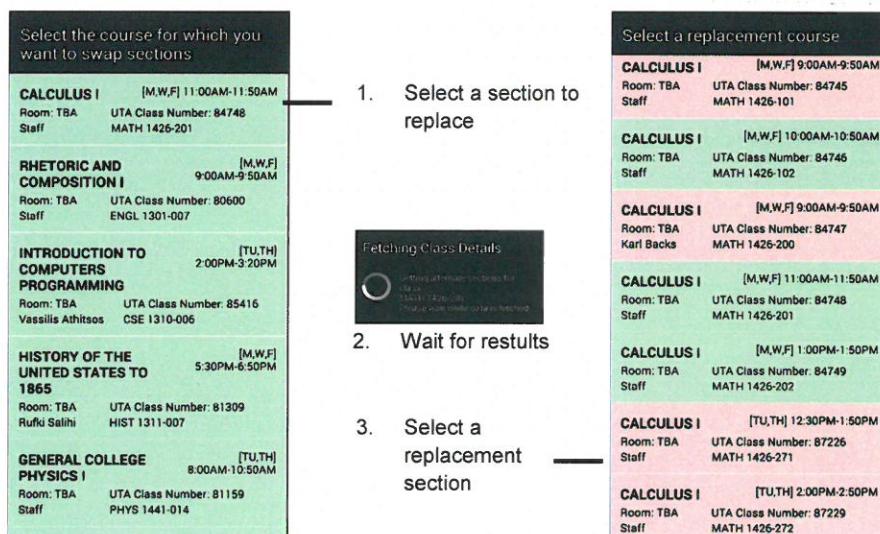


Figure 5-4 Swap Sections sequence

Chapter 6 - Settings

To navigate to the settings page, use the standard android menu button. The location of this

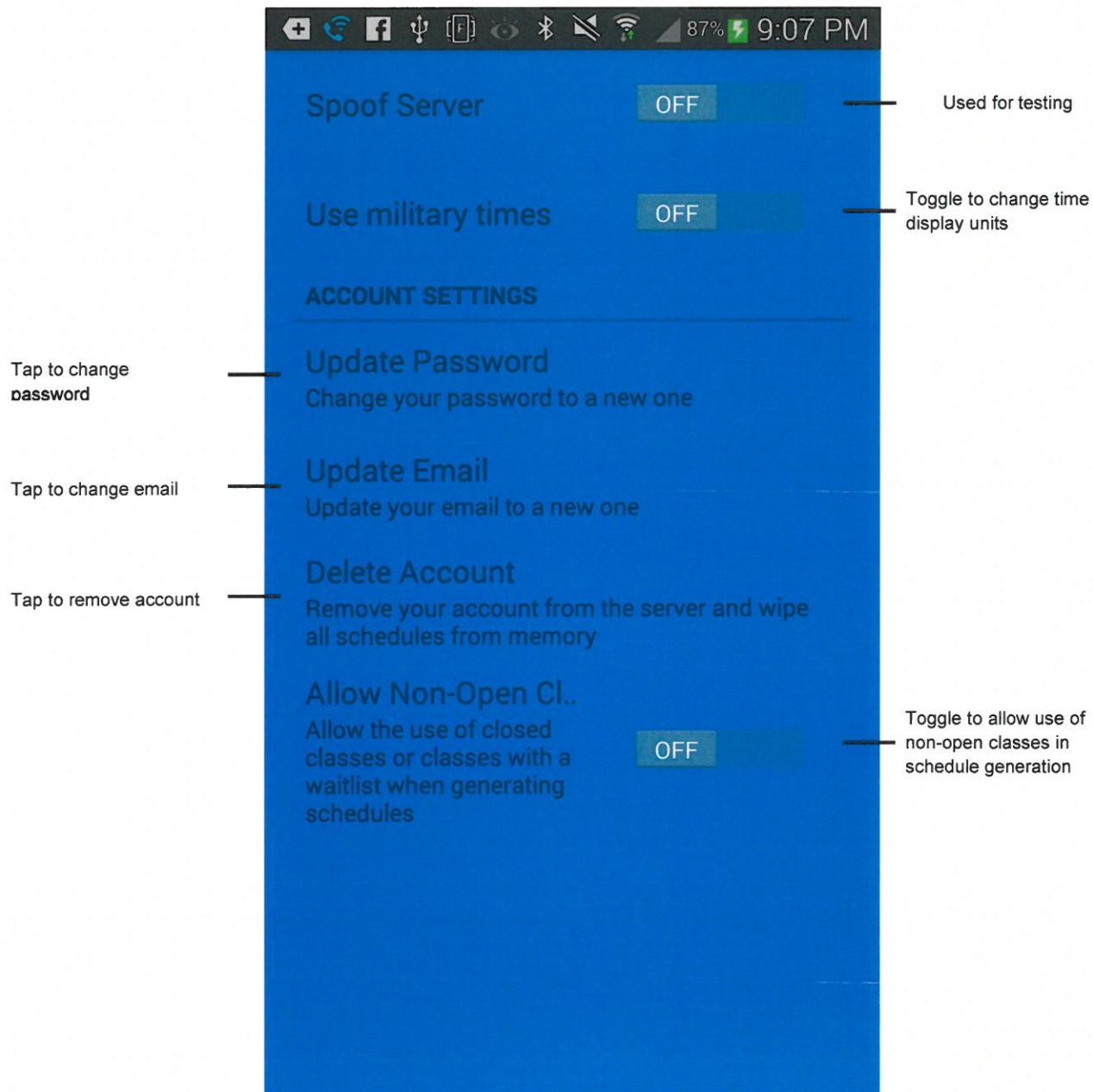


Figure 5-1 Settings Page

button will vary across different android devices, so consult your device manual to find it. If your device does not have a menu button the settings page link will be accessible from the three dots menu option which will appear on the top of every page in this app.

Use Military Times

This option allows you to toggle between using standard 12 hour AM/PM time and 24 hour military time when displaying time units across the application.

Update Password

This option allows you to change an existing password. To update a password, first you need to tap **Update Password**. A field will appear for you to enter a new password. Upon successful update you will receive a notification with the message "Password Successfully Updated".

Update Email

This option allows you to change the email associated with your account. To update email, tap **Update Email**. Enter a new email address and tap **OK** to confirm the changed email.

Delete Account

This option allows you to delete your account. Tapping **Delete Account** prompt you for a confirmation. If you confirm you wish to delete your account it will be removed from the server and all settings and schedules saved to the server will be erased. This cannot be undone, so be sure you want to delete your account before attempting to do so.

Allow Non-Open Courses

This option allows you to permit the usage of closed and waitlisted courses during schedule generation. By default, schedule generation does not include closed or waitlisted courses. Activating **Allow Non-Open Courses** is not recommended, as you will likely end up with a schedule with conflicting classes when you generate schedules.