|  |  |  |  |
| --- | --- | --- | --- |
| Project: | MyMav Mobile | | |
| Team: | Phi | | |
| Class: | CSE 3310; Spring 2015 | | |
| Module: | System Requirements Analysis (SRA) | | |
| Deliverable: | SRA Document | | |
| Version: | [1.0] | Date: | [3/23/2015] |

Contributors:

Greg Wilkinson

1000779713

Ph: 740-507-8340

[gregory.wilkinson@mavs.uta.edu](mailto:gregory.wilkinson@mavs.uta.edu)

Kristina Jefferson

1000685986

Ph: 214-717-8758

[Kristina.jefferson@mavs.uta.edu](mailto:Kristina.jefferson@mavs.uta.edu)

Nick Parker

1000732463

Ph: 214-864-7984

[Nick.parker@mavs.uta.edu](mailto:Nick.parker@mavs.uta.edu)

Jose Caballero

1000683861

Ph: 817-948-6585

[Jose.caballero@mavs.uta.edu](mailto:Jose.caballero@mavs.uta.edu)

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version number | Date | Originator | Reason for change | High level description of changes |
| 1.0 | 03/22/2015 | Greg Wilkinson | Initial draft | Initial write up of document |
| 1.1 | 03/30/2015 | Nick Parker | Section Editing | Added content to Section 6 |
| 1.2 | 3/30/2015 | Jose Caballero | Section Editing | Added content to Section 2 and 3 |
| 1.3 | 05/07/2015 | Jose, Greg | Final Editing |  |

TABLE OF CONTENTS

[1. Introduction and Project Overview 5](#_Toc415514375)

[2. Objectives 5](#_Toc415514376)

[2.1 BUSINESS Objectives 6](#_Toc415514377)

[2.2 SYSTEM Objectives 7](#_Toc415514378)

[3. Project Context Diagram 8](#_Toc415514379)

[4. Systems Requirements 9](#_Toc415514380)

[4.1 “Login” Requirements 9](#_Toc415514381)

[4.2 “Add User” Requirements 10](#_Toc415514382)

[4.3 “Delete User” Requirements 11](#_Toc415514383)

[4.4. “Main Menu” Requirements 12](#_Toc415514384)

[4.5 “Search For Classes” Requirements 13](#_Toc415514385)

[4.6 “Obtain Classes” Requirements 14](#_Toc415514386)

[4.7 “Block Out Time In Schedule” Requirements 15](#_Toc415514387)

[4.8 “Add Class” Requirements 16](#_Toc415514388)

[4.9 “Remove Class” Requirements 17](#_Toc415514389)

[4.10 “Create Schedule” Requirements 18](#_Toc415514390)

[4.11 "View Schedule" Requirements 19](#_Toc415514391)

[4.12 “Delete Schedule” Requirements 20](#_Toc415514392)

[4.13 “Verify Saved Schedule” Requirements 21](#_Toc415514393)

[4.14 “Choose From Multiple Schedules” Requirements 22](#_Toc415514394)

[4.15 "Load Schedule" Requirements 23](#_Toc415514395)

[4.16 "Log Out" Requirements 24](#_Toc415514396)

[5. Software Processes and Infrastructure 25](#_Toc415514397)

[5.1 Hardware and Infrastructure 25](#_Toc415514398)

[5.2 UML Diagrams 26](#_Toc415514399)

[5.3 Conceptual Data Model - Database 30](#_Toc415514400)

[5.4 Screen Shots 30](#_Toc415514401)

[5.5 Test Plan 31](#_Toc415514402)

[6. Assumptions and Constraints 32](#_Toc415514403)

[6.1 ASSUMPTIONS 32](#_Toc415514404)

[6.2 CONSTRAINTS 32](#_Toc415514405)

[6.3 Out of Scope material 32](#_Toc415514406)

[7. Delivery and Schedule 33](#_Toc415514407)

[Appendix: 36](#_Toc415514408)

# 1. Introduction and Project Overview

Our objective is to design and develop an Android application with the functionality to create school schedules from the courses offered at UT Arlington. The application will primarily be used by UT Arlington students.

From the application, students will have the capability to create an account, delete an account, and log in or out of the application. Once logged into the system, students will be able to search for classes, add or drop classes, create/modify/delete schedule, view schedules, and verify schedules.

Proper memory management, battery usage management, and closure of the app is key, since the app should be functional and easy to navigate.

# 2. Objectives

## 2.1 BUSINESS Objectives

The following is a list of business objectives:

**Objective 1**: Member Registration: The user shall provide the following information prior to using the system:

* + First Name, Middle Name {Optional}, Last Name
  + UT Arlington Username and Password

**Objective 2**: Login functionality: The user shall login to the system with a user/password that was established during Member Registration stage.

**Objective 3**: Creating Schedules: The user shall be able to select or create current/new schedule templates.

**Objective 4**: Delete Schedules: The user shall be able to select created schedules and be able to delete that schedule.

**Objective 5**: Add Classes: The user shall able to add a course, from a list of available courses, to a schedule.

**Objective 6**: Delete Classes: The user shall be able to select a course to delete from a schedule.

**Objective 7**: Verify Schedules: The user shall be able to verify a selected schedule to ensure that it is a valid schedule.

**Objective 8:** Block Time: The user shall be able to block out a particular time frame on a schedule.

**Objective 9:** Choose Schedule: The user shall be able to choose a schedule from several schedule if more than one has been created.

**Objective 10:** Remove User: The user shall be able to delete a user from the system.

**Objective 11:** Edit User: The user shall be able to edit the following information:

* + Password

**Objective 12:** Search For Classes: The user shall be able to search for a particular course.

## 2.2 SYSTEM Objectives

The following is a list of system objectives:

**Objective 1**: The system will be Android-based.

**Objective 2**: The system will use local storage on the client's device.

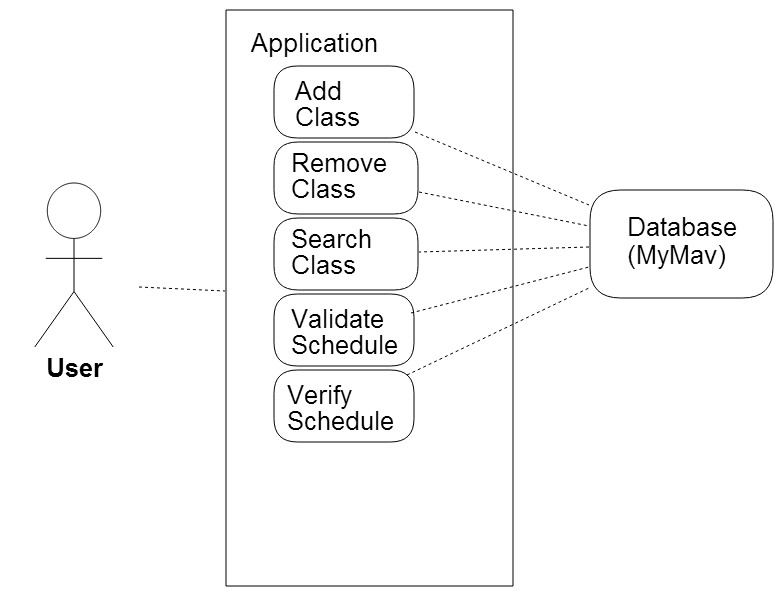
**Objective 3:** The system will run on Android 4.X.X.

**Objective 4:** The system will need an internet connection.

**Objective 5:** The system should not kill a user's phone, battery, or usage plan.

**Objective 6:** The system should connect to the UTA server.

# 3. Project Context Diagram



# 4. Systems Requirements

## 4.1 “Login” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Login |
| Sequence No:  (\*required) | 4.1 |
| Short description:  (\*required) | The user should be able to log into the application |
| Detailed Description:  (\*required) | 1.1 The user should be able to input username and password.  1.1.1 The user should have valid input for username and password.  1.1.2 The application shall check for empty input for username and password.  1.1.3.1 The application shall check for valid characters for the username and password  1.1.3.2 If the application detects invalid characters or an empty input, it shall not let the user log into the application  1.2.1 The user should have a way to initiate the login once username and password have been entered.  1.2.2 System shall compare given username and password to existing user information  1.2.3 If input username and password do not match existing user information, the application shall not let the user log into the application  1.3.1 If the entered username and password match an existing user's information, the application lets the user log in and takes the user to the main interface |
| Pre-Conditions:  (optional) | In order for the user to log into the application, a valid user account shall exist in the system. |
| Post Conditions:  (optional) | If the username and password the user inputs match user information that exists in the application's database, the application shall take the user to the main interface. |
| Other attributes:  (optional) | The username and password should be at least 8 alpha numeric characters long. |

## 4.2 “Add User” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Add User |
| Sequence No:  (\*required) | 4.2 |
| Short description:  (\*required) | The user should be able to create a user account. |
| Detailed Description:  (\*required) | 1. The user should provide his or her first and last name, as well as a unique username, UTA email address, and unique password.  1.1 If the user enters their credentials and are valid, the application shall create a new user account. |
| Pre-Conditions:  (optional) | The user should be connected to the internet.  The user should be a current student at UT Arlington. |
| Post Conditions:  (optional) | The application should create a user profile and log the user into the application. |
| Other attributes:  (optional) |  |

## 4.3 “Delete User” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Delete user |
| Sequence No:  (\*required) | 4.3 |
| Short description:  (\*required) | The application shall delete the user’s account if it is prompted by the user. |
| Detailed Description:  (\*required) | 1. The user shall be able to remove his or her user profile from the system.  1.1 If the user prompts the application to remove the user account, the application shall delete all saved schedules and the user’s account profile from the user’s local storage.  1.2 The application shall also remove the user's login data from the application's database. |
| Pre-Conditions:  (optional) | The user should have an existing user account.  The user should be logged into the application with their account. |
| Post Conditions:  (optional) | The application shall remove the user from the user interface and not let him log back into that account. |
| Other attributes:  (optional) |  |

## 4.4. “Main Menu” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Main Menu |
| Sequence No:  (\*required) | 4.4 |
| Short description:  (\*required) | The application should have a main menu where the user should be able to navigate through the application and choose from a list of options. |
| Detailed Description:  (\*required) | 1. After a successful login, the user should be presented with a main menu that should display the options available to the user.  1.1 The options on the menu should have a home screen, schedule management, class management, search class, a set time constraint and a profile settings screen.  1.1.1 The Main Menu should also have an option for the user to log out. |
| Pre-Conditions:  (optional) | User should be logged into the application.  User should be connected to the internet. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) |  |

## 4.5 “Search For Classes” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Search For Classes |
| Sequence No:  (\*required) | 4.5 |
| Short description:  (\*required) | The application should have a search interface, where the user should search for classes to add to the schedule. |
| Detailed Description:  (\*required) | 1. The user should be able to enter a course department, a course number, or a class ID number to search for a class.  1.1 The user should be able to enter a valid three or four letter department code and a valid four digit course number, and a five digit class ID number.  1.1.1 After the user sends a search request, the application should obtain the courses from UT Arlington's website  1.1.2 The course information retrieved by the application should match the department code, course number, and/or the class ID number that the user inputs when searching for a class  1.1.3 The application should list the class sections that match the user's search request and display them for the user. |
| Pre-Conditions:  (optional) | User should be logged into the application and on the search class screen.  User should be connected to the internet. |
| Post Conditions:  (optional) | The course information should be displayed to the user and the application shall give the user the option to add a course to his or her schedule. |
| Other attributes:  (optional) |  |

## 4.6 “Obtain Classes” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Obtain Classes |
| Sequence No:  (\*required) | 4.6 |
| Short description:  (\*required) | The application shall obtain a list of courses from the web. |
| Detailed Description:  (\*required) | 1. The application shall retrieve UTA course information.  1.1 The application should communicate with the UT Arlington servers in order to obtain a list of courses that the user is able to take.  1.1.1 The list of classes that the application obtains from the UT Arlington MyMav interface shall match the department code and course number the user enters if the department code, and course number combination is a valid course offered at UT Arlington during the semester chosen by the user.  1.1.1.1 If the department code and course number entered by the user is not a valid course, or is not a course that is being offered at the time the user searches, the application should notify the user that the course is not a valid search request. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should be connected to the internet. |
| Post Conditions:  (optional) | The course information should be displayed to the user and the application shall give the user the option to add a course to his or her schedule |
| Other attributes:  (optional) |  |

## 4.7 “Block Out Time In Schedule” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Block Out Time In Schedule |
| Sequence No:  (\*required) | 4.7 |
| Short description:  (\*required) | The application shall include the option that lets the user block out a time in the user’s schedule. |
| Detailed Description:  (\*required) | 1. If there is a certain time during the day that the user cannot or does not want to have class, the user should select a time frame to block out from his or her schedule.  1.1 The user should select the days and times that he or she wants to block from his or her schedule.  1.1.1 The application shall save the designated time block.  1.1.1.1 If the user has one or more schedule saved in the application, the application should check the course time for each class listed in each schedule, and verify that the course time does not conflict with the time block defined by the user. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should be connected to the internet. |
| Post Conditions:  (optional) | The application should not be able to add classes to the user’s schedule if the classes conflict with the blocked out time defined by the user. |
| Other attributes:  (optional) | The application should give the user the option to block out time from Monday through Saturday and from 12:00AM to 11:59PM.  Sunday should not be included as an option since UT Arlington does not offer courses on Sundays. |

## 4.8 “Add Class” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Add Class |
| Sequence No:  (\*required) | 4.8 |
| Short description:  (\*required) | The user should be able to add a class to his or her schedule. |
| Detailed Description:  (\*required) | 1. The user should search for a class before the user is able to add a class.  1.1 The user should provide the class department code and the course number.  1.1.1 The application should verify that the user entered a valid department code and a valid course number.  1.1.2 If the department number or the course number is not valid, the application should not show a list of classes.  1.1.2.1 The application should notify the user that the course the user searched for is not a valid course.  1.1.3 If the department number and course number are valid, the application should display a list of classes of the specified course  1.2 The user should select a class from the list of courses  1.2.1 If the user has created more than one schedule, the user should select which schedule to add the class  1.2.2 The application shall add the selected class to the user’s chosen schedule |
| Pre-Conditions:  (optional) | The list of classes shall be obtained from the web before the user is able to add a class.  The user should be connected to the internet.  The user should be logged into the application. |
| Post Conditions:  (optional) | The application shall add the class to the user’s chosen schedule. |
| Other attributes:  (optional) |  |

## 4.9 “Remove Class” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Remove Class |
| Sequence No:  (\*required) | 4.9 |
| Short description:  (\*required) | The application shall have the option to remove classes from a schedule. The user should be able to choose which class to remove from a schedule. |
| Detailed Description:  (\*required) | 1. The application should show a list of the user’s schedules.  1.1 The schedules in the list should be able to be modified.  1.1.1 The user should choose a schedule to modify.  1.1.2 After the user chooses a schedule to modify, the application should list the classes in the chosen schedule.  1.1.3 The user should select the class he or she wants to remove.  1.1.3.1 After the user selects the class to remove, the application shall delete the selected class from the user’s schedule. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should have at least one class added to a schedule. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) |  |

## 4.10 “Create Schedule” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Create Schedule |
| Sequence No:  (\*required) | 4.10 |
| Short description:  (\*required) | The application shall have the option to create a schedule. |
| Detailed Description:  (\*required) | 1. The application shall be able to create a new schedule to list the user’s classes.  1.1 The schedules that the application creates can either be created as an empty schedule, or it can be created after choosing a class to add to a schedule.  1.1.1 If the user chooses to create a schedule without first adding a class, the application shall create a new, empty schedule for the user.  1.1.2 If the user decides to add a class to a new, not yet created schedule, the user should have the option to create a new schedule.  1.1.2.1 The application shall create a new schedule, and the class the user wants to add should be added to the newly created schedule. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should be connected to the internet. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) | The schedules should have names so that the user can differentiate between different schedules if more than one schedule exists. |

## 4.11 "View Schedule" Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | View Schedule |
| Sequence No:  (\*required) | 4.11 |
| Short description:  (\*required) | The user should be able view the list of schedules and the classes listed in each schedule. |
| Detailed Description:  (\*required) | 1. The application should display a list of the created schedules  1.1 The user should choose a schedule to view  1.1.1 The application should display the classes listed under the chosen schedule |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should be connected to the internet. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) |  |

## 4.12 “Delete Schedule” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Delete Schedule |
| Sequence No:  (\*required) | 4.12 |
| Short description:  (\*required) | The user should be able to delete a schedule |
| Detailed Description:  (\*required) | 1. The application should list the schedules the user has created.  1.1 The user should choose which schedule to delete.  1.1.1 If the schedule has classes stored in it, the application shall remove all of the classes, and delete the schedule  1.1.2 If the chosen schedule contains no classes, the application shall delete empty schedule. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should have a saved schedule. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) |  |

## 4.13 “Verify Saved Schedule” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Verify Saved Schedules |
| Sequence No:  (\*required) | 4.13 |
| Short description:  (\*required) | The application shall verify if the user’s schedules are valid by checking for time conflicts. |
| Detailed Description:  (\*required) | 1. Application should list the saved schedules  1.1 The user should have a way to tell the application to verify the schedule(s).  1.1.1 The application shall check for any time conflicts that may exist in a schedule.  1.1.1.1 If there are time conflicts between the classes listed in a schedule, the application shall not verify the schedule and should notify the user that the schedule is not valid.  1.1.2 The application should also check if there are available seats in each class.  1.1.2.1 If there are no available seats, the application shall not verify the schedule and should notify the user that the schedule is not valid.  1.1.3.1 The application should check if there are any conflicts with the time the user blocks out.  1.1.3.2 If there are any conflicts with the blocked out time, the application should not verify the schedule and should notify the user that the schedule is not valid  1.2 If there are no time conflicts or full classes, the application should verify that the schedule(s) is/are valid. |
| Pre-Conditions:  (optional) | User must be logged into the application.  User must have a saved schedule  User must be connected to the internet |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) |  |

## 4.14 “Choose From Multiple Schedules” Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Choose From Multiple Schedules |
| Sequence No:  (\*required) | 4.14 |
| Short description:  (\*required) | The user shall be able to choose a schedule as his or her final schedule for an upcoming semester. |
| Detailed Description:  (\*required) | 1. The application should list the schedules the user has created.  1.1 The user should choose which schedule he or she wants to be the final schedule.  1.2 The application should assign the chosen schedule as the final schedule. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should have a saved schedule.  The schedules shall be verified before the user can choose a final schedule. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) | The application should have something that indicates that the chosen schedule is the user’s final schedule. |

## 4.15 "Load Schedule" Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Load Schedule |
| Sequence No:  (\*required) | 4.15 |
| Short description:  (\*required) | The application shall be able to load the schedules the user has saved. |
| Detailed Description:  (\*required) | 1. Once the user has logged into the application, the application shall check for saved schedules  1.1 If there are saved schedules, the application shall load the schedules from the local storage to the application’s user interface. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should be connected to the internet.  The user should have saved schedules to load. |
| Post Conditions:  (optional) |  |
| Other attributes:  (optional) |  |

## 4.16 "Log Out" Requirements

|  |  |
| --- | --- |
| Requirement Title:  (\*required) | Log Out |
| Sequence No:  (\*required) | 4.16 |
| Short description:  (\*required) | The user should be able to log out of the application. |
| Detailed Description:  (\*required) | 1. The user should have a way to tell the application to log out of his or her account.  1.1 Once the user tells the application to log out, the application should not let the user access his or her MyMav mobile account until the user logs into the application again. |
| Pre-Conditions:  (optional) | The user should be logged into the application.  The user should be connected to the internet. |
| Post Conditions:  (optional) | Once the user logs out of the application, the application should display the Login interface. |
| Other attributes:  (optional) |  |

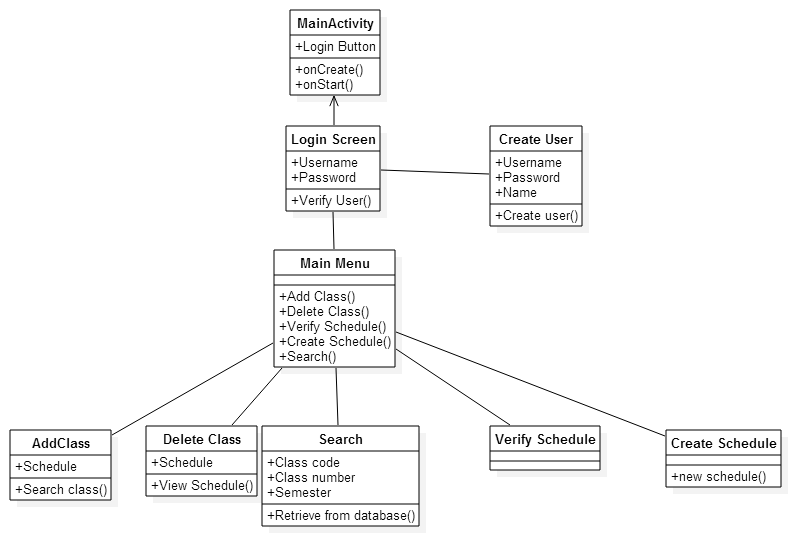
# 5. Software Processes and Infrastructure

## 5.1 Hardware and Infrastructure

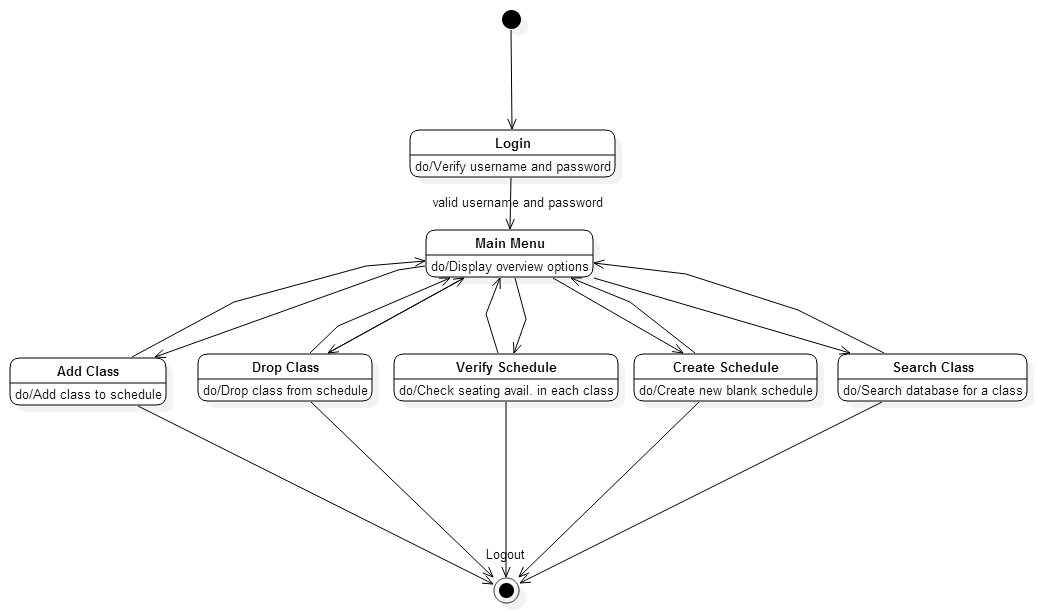
* Application will run on Android 5.0 (exact version will be decided at a later date)
* User's phone must be running Android 4.0 or a later version
* User's phone must have enough storage space for the application, and any schedules which they will create through the application's use
* User's phone will contact a UTA server in order to query for class schedule information

## 5.2 UML Diagrams

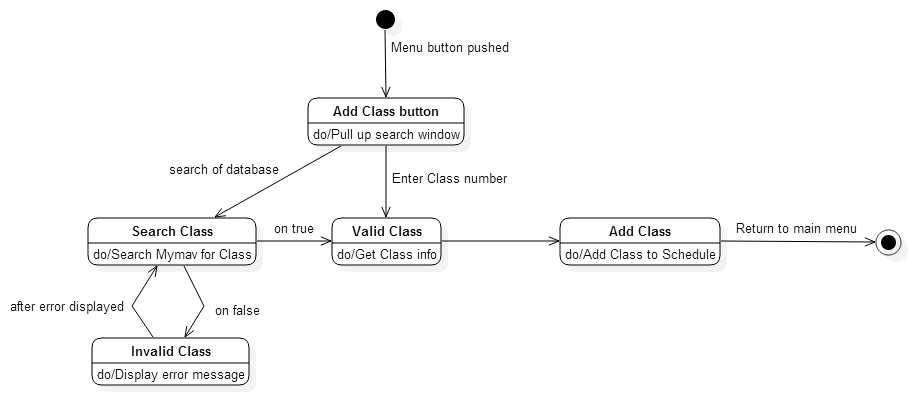
Main Application



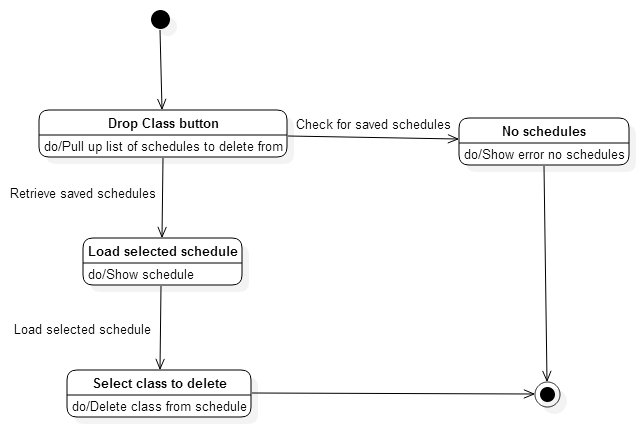
Big Picture



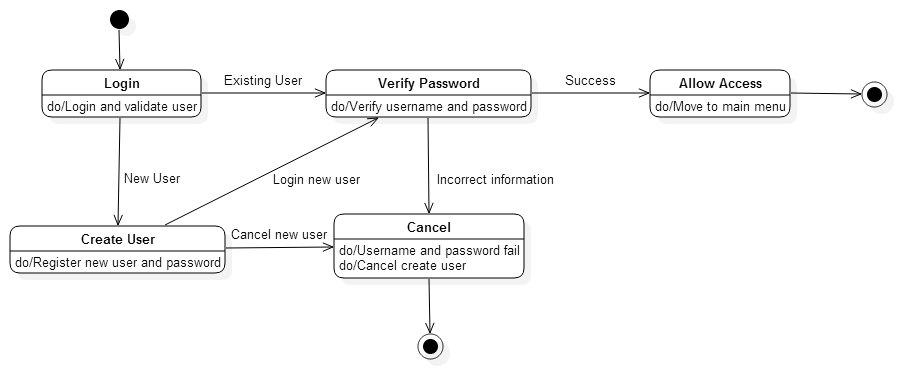
Add Class



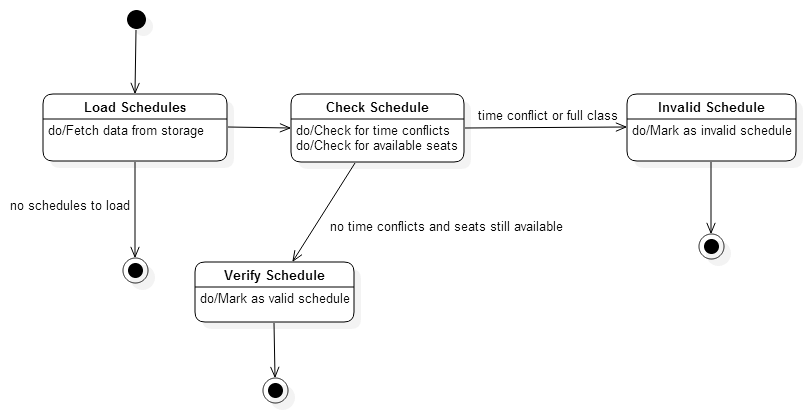
Drop Class



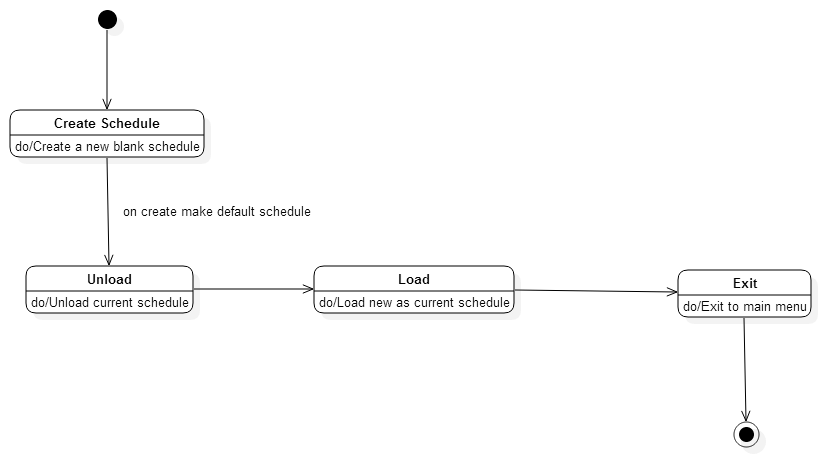
Login



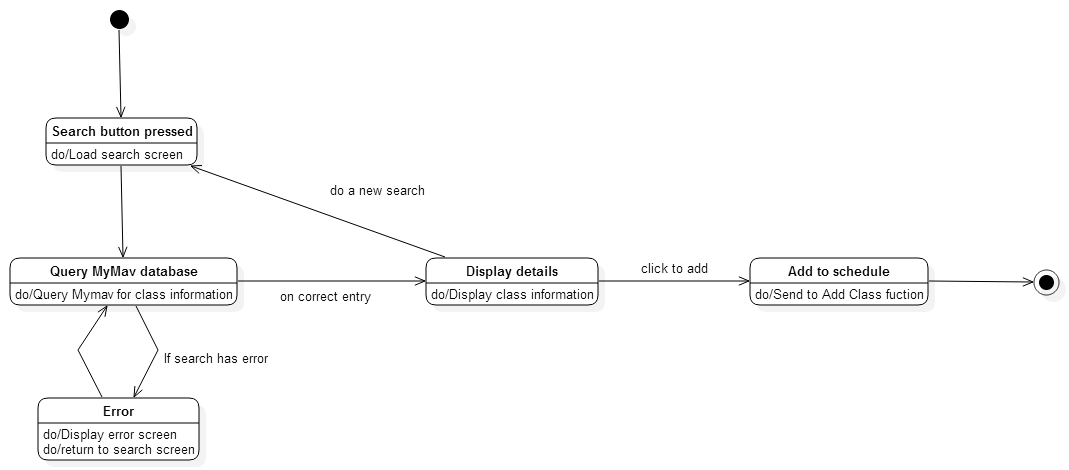
Verify Saved Schedules



Create Schedule



Search



## 5.3 Conceptual Data Model - Database

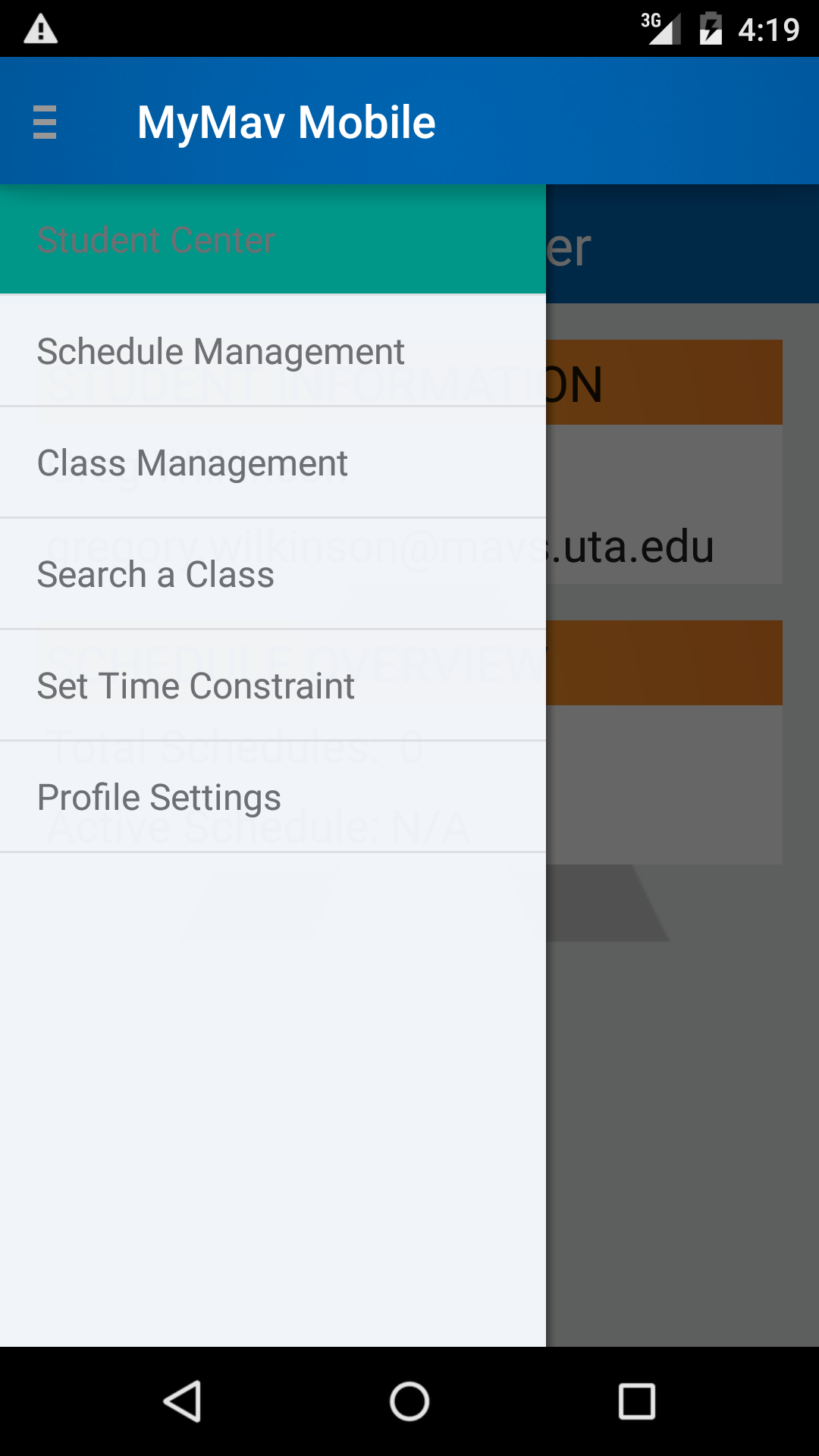
* User login data will be stored locally on the device
* Schedules associated with a user will be stored locally
* Class information will be retrieved from the MyMav database

## 5.4 Screen Shots

## C:\Users\mynam_000\AppData\Local\Microsoft\Windows\INetCache\Content.Word\device-2015-03-31-034117.pngC:\Users\mynam_000\AppData\Local\Microsoft\Windows\INetCache\Content.Word\login screen.png

Original Final

## C:\Users\mynam_000\AppData\Local\Microsoft\Windows\INetCache\Content.Word\device-2015-03-31-033721.png



Original Final

## 5.5 Test Plan

* A test plan will be provided at a later stage of the project

# 6. Assumptions and Constraints

## 6.1 ASSUMPTIONS

The following is a list of assumptions:

* Ignore collecting money from external advertisement and general accounting
* Ignore compliance issues
* Ignore network issues
* All user phones have Wi-Fi capability
* All valid users already have valid MyMav account credentials
* All users have unlimited data transfer capabilities (they are not limited by their phone plan)

## 6.2 CONSTRAINTS

The following is a list of constraints:

* Team has limited/no experience in interfacing with a web server within a program
* Team has limited/no Android development experience
* Time to work on the project is limited due to weather, school, and other outside obligations

## 6.3 Out of Scope material

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

* Post Project maintenance is not covered
* Distribution of the application is not covered
* Legal issues resulting from the use of UTA assets are not covered

# 7. Delivery and Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task/Milestone Description | Anticipated Start Date | Anticipated End Date | Status | Comments |
| Prepare Requirements | 2/10/2015 | 2/26/2015 | Completed |  |
| UML diagram | 3/16/2015 | 4/2/2015 | In Progress |  |
| SRA document (Includes project objectives, Requirements and UML diagrams) | 3/16/2015 | 4/2/2015 | 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 | 3/26/2015 | 3/31/2015 |  |  |
| Have a testable application | 4/10/2015 |  |  |  |
| Test Plan Delivery |  | 4/23/2015 |  |  |
| Write Login class | 3/21/2015 | 4/05/2015 | In Progress |  |
| Write AddUser class | 4/01/2015 | 4/21/2015 |  |  |
| Write DeleteUser class | 4/01/2015 | 4/21/2015 |  |  |
| Write MainMenu class | 3/28/2015 | 4/30/2015 | In Progress |  |
| Write SearchClass class | 4/05/2015 | 4/20/2015 |  |  |
| Write ObtainClass class | 4/05/2015 | 4/20/2015 |  |  |
| Write BlockTime class | 4/05/2015 | 4/10/2015 |  |  |
| Write AddClass class | 4/10/2015 | 4/20/2015 |  |  |
| Write RemoveClass class | 4/10/2015 | 4/25/2015 |  |  |
| Write CreateSchedule class | 4/15/2015 | 4/25/2015 |  |  |
| Write ViewSchedule class | 4/15/2015 | 4/25/2015 |  |  |
| Write DeleteSchedule class | 4/15/2015 | 4/25/2015 |  |  |
| Write VerifySchedule class | 4/15/2015 | 4/30/2015 |  |  |
| Write MultSchedule class | 4/20/2015 | 4/30/2015 |  |  |
| Write LoadSchedule class | 4/20/2015 | 4/30/2015 |  |  |
| Write LogOut class | 4/20/2015 | 5/1/2015 |  |  |
| External Documentation (i.e. User Manual) | 4/25/2015 | 5/3/2015 |  |  |
| 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 Manager |  |  |
| Elise Austin | Project Assistant |  |  |
| Greg Wilkinson | Developer |  |  |
| Kristina Jefferson | Developer |  |  |
| Jose Caballero | Developer |  |  |
| Nick Parker | Developer |  |  |

# Appendix:

None