Mobile Student Lookup

Compiled Information from 371

12/2/2011

Mark Vitale, Chris Gropp, Brandon Knight, and Ann Say

Campus Mail: 1200

Contents

Executive Summary

We are producing a mobile application for iPhone and iPod Touch (iOS) for the clients Tim Ekl and Eric Stokes. The features of this application will include, but not be limited to: searching for a student's schedule by username, searching by classroom, searching by separate terms, overlaying schedules (comparing more than one schedule), and syncing with the Calendar app on the iPhone. The user needs this application to be easy to use, easy to navigate, and easy to read. Having an application on the iPhone will drastically improve the process of looking up schedules on the iPhone. In the future, we may extend this project to iPad, or even Android platforms.

Introduction

Currently to view their schedule on the iPhone or iPod Touch, students have to manually navigate through the Safari mobile web browser to the Rose-Hulman website, and then to the registrar's office, logging in with their Kerberos password several times before they can access their schedule. We want to simplify this experience for mobile devices, and make it easier and quicker for students to access this data. We will be designing a mobile application on iOS for iPhone, which will allow students to access all available schedule data from the registrar's office website. The application will be capable of storing the password, so students will not have to type it in more than once. Students should be able to easily look up their own schedules, easily find a classmate's schedule, and view all of the members of their classes.

Client Background

We have two clients for Mobile Student lookup, Erik Stokes and Tim Ekl. Both are former undergraduate students at Rose-Hulman. Tim Ekl is currently a graduate student at Rose-Hulman for Engineering Management, he spends a lot of his time managing his proposed projects in the CS Labs. Erik Stokes works for a software development company based out of Colorado. Both of the clients have very good technical skills and vast knowledge about what kind of systems and software the proposed software would use. The clients will maintain and update the system at the conclusion of the junior project series. (Ekl & Stokes, 2011)

Current System

In order for students today to check their class schedules on their mobile phones, they have to go to the Schedule Lookup page on Banner Web, in the same manner as they would on a computer. First, they would have to log in using their usernames and Kerberos passwords. Once they reach the main page, there are three search boxes they could use: search by a student's ID number or username, by room number, or by course ID. They can search for this information from different school years and terms as well. Also on the main page, the users can select the layout of their search query. By default, the schedule has a table listing class information - class ID number, name, instructor, credits, students enrolled, term schedule, and time of class final – and another table showing the times, class periods, and days the classes meet. Additional layouts include the class information table and finals table, the class information table only, and a course matrix. The course matrix allows students to type in various class

IDs and several valid schedule possibilities are generated based on the information provided. Lastly, under the search functions are instructions regarding how to use search functions, information about registration process, and various dates pertaining to registration.

When a user has searched by his username, his schedule is shown, in the layout that he had indicated in the previous page. Several items are hyperlinks, meaning they lead to a new page when clicked. These hyperlinks are the instructor names and course IDs. Selecting the instructor pulls up the instructor's schedule. Selecting the course ID goes to a new page, which lists all the students enrolled in that class. The columns of the table include the person's username, full name, campus mailbox number, declared major, class, advisor, and email address. Clicking on the username pulls up the respective person's schedule. The advisor's name can be selected as well to display his or her schedule.

Searching by a room number returns a result similar to the class information table and class times table for a student, but instead, it displays all classes that meet in that specific room. Inputting an asterisk in the search query displays the list of all classes, and the times they all meet.

The third search method, searching by course ID, allows the user to type in the course ID, and all sections of that class will be returned, as well as the times and places those classes meet.

The interface is easily navigable on a computer. However, on a mobile device that has a smaller screen, the website is extremely difficult to manage. The tables are too small to read without zooming in. Additionally, the hyperlinks are too small and close together, so the user's fingers can't accurately select the links; either the wrong one would be selected, or no link will be selected.

Alternatives and Competition

There are hardly any alternatives available to mobile phone users in terms of browsing a student's schedule. Currently, there is no application available on the App Store that provides a schedule lookup for Rose-Hulman students. The only way a student can look up the schedule is through the mobile browser. Possible competition for this project would be other students who designed their own product similar to this one.

User Profile

Students would most likely be c (Ekl & Stokes, 2011)oncerned with the speed and ease of accessing their own schedule, as they would be mainly using the app to find where there next class is while they're on their way to classes. They would probably be fairly comfortable using the iPhone and its apps.

Professors would use the program mostly to look at rosters for their classes. They would probably be concerned with the information being formatted in an easy-to-read way. They may or may not be comfortable with the iPhone.

Advisors would use the program to check on the classes of their advisees. They would probably be concerned with finding pages of students easily. They would be as comfortable with the iPhone as other professors.

User Needs

A survey was completed by forty-seven members of the student and faculty, who were asked for features and functionality that they would like to see in a Schedule Lookup Page app. Over half of the people who responded noted that the layout of the schedule lookup page needs improvement, especially on the iPhone or iPod Touch screen. Twelve people complained that logging in every time was annoying and that an app for the schedule lookup page should store that information. Other common suggestions were the ability to compare or overlay schedules, and bookmarking your own schedules and those of your friends. A few people surveyed said they would like to see more information on a person's page, such as their major and campus mailbox number, and some asked for the ability to search by a person's name, rather than username, or by a department. One faculty member also noted that he would like the ability to see a list of all the classes a student has ever taken.

Most common suggestions, such as a layout redesign and saving logins, are core requirements for this project. A few, such as integrating the study buddy tool, Angel, and class registration; showing maps of campus; or showing professors office hours, are out of the scope of the project. Others, such as displaying extra information on a student's page and rotating the interface when the device is rotated, would be of low importance if they were included at all.

Product Perspective

The mobile student lookup application will be a native application for iOS devices (devices which run the mobile operating system distributed by Apple). This application will allow anyone with Rose-Hulman Network access (students or faculty) to access all schedule lookup tools provided by the Registrar's student lookup website. These tools include search by username (partial or whole), search by class ID, search by room number, and the ability to search through all past terms.

The student schedule system we are building would be a native mobile application. This is meant to replace the current web based system. The current system's data will not be able to be directly accessed, however. If we want to utilize the current system in any way, we will have to scrape the data from the website. For our mobile application, we will take the data that we scrape, and have a User Interface (UI) that is a skinned version of the web based one. We want to really simplify the UI, and make it intuitive. We were thinking of having only one search bar for user names. The current system and the application will have no interaction beyond simple data scraping. The mobile application will heavily rely on the registrar's student lookup page, but the registrar's page does not care that the mobile app exists.

Feature Listing

Below is a listing of the features for the mobile student lookup app. The priorities have been assigned based on input from the client, while effort, risk, and stability have been estimated from previous experience developing apps for the iOS platform.

1. Users will be able to look up a schedule for the current quarter by username.

Status	Accepted
Priority/benefit	Critical
Effort	Medium
Risk	Medium
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Required to satisfy basic project requirements

2. Credentials will be stored on first entry and will only be requested again if credentials are incorrect.

Status	Accepted
Priority/benefit	Critical
Effort	Low
Risk	Low
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Required to satisfy basic project requirements

3. Users will be able to view the roster of a class based on a class section.

Status	Accepted
Priority/benefit	Critical
Effort	Medium
Risk	Low
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Required to satisfy basic project requirements

4. All users will have instant access to their own schedule with a click of a button.

Status	Accepted
Priority/benefit	Critical
Effort	Low
Risk	Low
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	This will reduce the amount of time needed to use

	the app when a user is trying to find their next class

5. Users will be able to view schedule information for all quarters supported by schedule lookup page

Status	Accepted
Priority/benefit	High
Effort	Medium
Risk	Medium
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Allows users to view historical or future information without needing a computer. Could be useful when seeing if a friend has already taken a class.

6. Look up schedule information based on class section number.

Status	Accepted
Priority/benefit	High
Effort	Medium
Risk	Medium
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Useful if attempting to view how many sections are currently offered or various professors teaching a certain class

7. Lookup a user's contact information.

Status	Accepted
Priority/benefit	High
Effort	Medium
Risk	Medium
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Often when looking up the schedule of others, the user intends to contact that user in some fashion.

8. Open a new email in the native email client addressed to another student or faculty when clicking on any email address in the app.

Status	Accepted
Priority/benefit	High
Effort	Medium
Risk	Low
Stability	High

Target Release	Version 1.0
Assigned to	Team
Reason	The ability to quickly compose an email to a student or faculty member from within the app would streamline even communication with others

9. Lookup a schedule based on room number.

Status	Accepted
Priority/benefit	Low
Effort	Medium
Risk	Low
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	Students and teachers alike often wonder whether a certain room has class during a specific period to determine whether or not that room would be a useful meeting location.

10. Sync current schedule with calendar app simplistically.

Status	Accepted
Priority/benefit	Low
Effort	High
Risk	High
Stability	Medium
Target Release	Version 1.0
Assigned to	Team
Reason	This would provide a simple way to get schedule information into a user’s calendar which will sync automatically to any backend calendar already used on the device.

11. Provide a layover of various schedules to help determine common breaks for meeting times.

Status	Accepted
Priority/benefit	Low
Effort	High
Risk	High
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	This functionality is not present in the current schedule lookup page, but it is frequently the goal of users looking up schedules to identify common breaks for meetings. The app should at minimum support 4 users in layover view.

12. Access and maintain list of favorite users to quickly get up-to-date information on commonly viewed schedules.

Status	Accepted
Priority/benefit	Low
Effort	Low
Risk	Low
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	The majority of users don't access a wide variety of schedules on a regular basis, but want to visit a few schedules quickly and frequently.

13. Send any schedule information to another person via email.

Status	Accepted
Priority/benefit	Low
Effort	High
Risk	High
Stability	Medium
Target Release	Version 1.0
Assigned to	Team
Reason	The intention of the user may be to look up a specific schedule in order to share it with group members or other interested parties. This would simplify that process.

14. Begin a phone call to an individual when clicking on any phone number in the app.

Status	Accepted
Priority/benefit	Low
Effort	Low
Risk	Low
Stability	High
Target Release	Version 1.0
Assigned to	Team
Reason	The ability to quickly call an individual from within the app would streamline communication with others

Constraints

Several constraints on the project were discovered through conversations with the client. They are listed below.

1. The mobile application developed will be an iOS native app on iPhone and iPod Touch devices.
2. The development will be done using Xcode 4.3 and targeting iOS 5.

3. The app will follow all of the constraints discussed in Apple’s Human Interface Guidelines.
4. The language used in the development of this app will be Objective-C
5. The project is operating with no budget at all.
6. The project must be completed by May, 2012 at which point all development will cease.
7. The capabilities of this software will be limited to the functionality of the web system, except where otherwise specified.

Use Cases

Global Pre Conditions

We are assuming that all users of the application will have a Rose-Hulman network account, if they do not, they will not be able to access any of the application's features. We also assume that:

1. The application is already running.
2. Except where specifically stated, the user's credentials are stored in the application settings page.
3. The user has entered a valid username or password. But if they haven't:
    a. The user will be denied access to the server functionalities.
    b. The user will view a warning from the system.
    c. The user will have to enter a username and password before proceeding.

Global Alternate Flow

If the server is not available, or if the user has entered an incorrect username or password, we would need a separate flow of events to describe the actions necessary in the respective situation.

1. If the username or password is incorrect when attempting to access the server:
    a. Prompt the user to go to the settings page in order to enter the correct username and password
    b. If the user does not have a Rose account, or cannot remember their password, our application will not handle these situations, because IAIT currently has webpages devoted to these activities that are beyond our control.
2. If the server cannot be reached:
    a. Display a warning to the user, then cancel the current action. To avoid errors, or something malfunctioning, we will simply display blank schedules if necessary.
        i. Nothing will be displayed
3. If the user cancels a current action:
    a. If the action is loading server information, the user will not be able to see the information and will be taken back to the previous screen.

1. Mobile Student Lookup Use Case: Invalid or No Credentials

Revision History			
Date	Issue	Description	Author
9/30/11	1.0	Initial Creation of “First Time Use” use case.	Brandon Knight
10/02/11	1.1	Revisions and word editing	Ann Say
10/3/11	1.2	Revisions and editing.	Brandon Knight
10/5/11	1.3	Changing name to “Invalid or No Credentials,” generalizing the use case	Mark Vitale

Relevant Feature(s): 2

Brief Description: This is the use case for the sequence of actions a user would have to perform whenever there are no credentials registered with the app or the credentials that are registered are invalid. This includes the first use of the application before the user has entered valid credentials.

Basic Flow:

1. If the app does not have any credentials registered or, when trying to query the schedule lookup page, the stored credentials fail, the application defaults to the “Settings” page, where the user must enter his or her username and password for his or her Rose-Hulman account.
2. The server verifies the credentials, and the first step is repeated until the credentials are found to be valid.
3. If this is the first use of the application, the user’s schedule is immediately loaded and displayed. In the case that previous credentials failed, the user is taken to the page they were trying to access when the previous credentials were refused.

Pre-conditions: We are going to assume that the application is running

Post conditions: The user will be able to see the most updated version of their personal schedule in the “My Schedule” tab of the application. The user will also be able to access any other part of the application without re-entering their password.

2. Mobile Student Lookup Use Case: View Own Schedule

Revision History			
Date	Issue	Description	Author
9/30/11	1.0	Initial Creation of "View Own Schedule" use case	Brandon Knight
10/3/2011	1.1	Revisions and editing, clarifying.	Brandon Knight

Relevant Feature(s): 4

Brief Description: This use case describes the actions a user would take to view his/her own schedule. The actions are simple, in that there is a separate section of the application that will directly retrieve the user's schedule based on his/her username.

Basic Flow: The basic flow begins when the user opens the application. The application will perform the following upon opening:

1. Retrieve the user's stored username and Kerberos password from the device's memory.
2. Automatically use the username and password to attempt to access the server
3. Use the username on server to find user's schedule and retrieve it
4. Display the updated schedule in a tabular format that orders the classes by time, earliest to latest.
5. The user is then free to view his or her schedule in the "My Schedule" tab, and perform other application features.

If the user has the application open, he or she is able to click a 'refresh' button on the screen, which will perform all of the above actions, therefore updating the schedule in the "My Schedule" tab.

Post conditions: The user views his/her most recent schedule through the "My Schedule" tab.

3. Mobile Student Lookup Use Case: Search by Username

Revision History			
Date	Issue	Description	Author
10/02/11	1.0	Initial Creation of "Search by Username" use case	Ann Say
10/03/11	1.1	Editing.	Brandon Knight
10/5/11	1.1	Added an alternate flow handling exact and partial matches	Mark Vitale

Relevant Feature(s): 1

Brief Description: This use case describes the sequence of actions the user and the system takes when the user decides he or she would like to search by a person's username in the mobile application.

Basic Flow: This use case will begin when the user selects the "Search Bar."

1. The user will select the "username" option.
2. The user will type a username that he or she wants to search.
3. The app will automatically use the stored Kerberos username and password to navigate to server and pull off all available information available from a "username search" on the website.
4. If the user typed in an exact match for an individual's username, that individual's information is displayed directly. If the user typed in a partial username, follow the alternative flow of events under "If the user only typed in part of a username."
5. The application will display this information to the user formatted for the small screen.

Alternative Flow of Events:

If the user only typed in part of a username:

- Return a list of usernames that match the current search entry. The user should be able to scroll through the list and by selecting a username, they are taken to that person's schedule. Go to step 5 in the basic flow.

If there are no usernames that match the user's query:

- Display a message saying no people with that username were discovered.

Post conditions: The user views the schedule for the username entered.

4. Mobile Student Lookup Use Case: Search by Classroom

Revision History			
Date	Issue	Description	Author
9/30/11	1.0	Initial Creation of search by classroom use case	Brandon Knight
10/5/11	1.1	Added an alternate flow handling exact and partial matches	Mark Vitale

Relevant Feature(s): 9

Brief Description: This use case describes the sequence of actions the user and the system take when the user decides he/she would like to search by classroom in the mobile application.

Basic Flow: We are going to assume that the user has the application open already, so the basic flow begins when the user goes to the search tab of the application.

1. The user will select the search bar
2. The user will select the ‘class room’ option
3. The user will type a class number that they want to search
4. The app will automatically use the stored Kerberos username and password to navigate to server and pull off all available information available from a ‘classroom search’ on the website.
5. If the user typed in an exact match for a classroom number, that classroom’s information is displayed directly. If the user typed in a partial classroom number, follow the alternative flow of events under “If the user only typed in part of a classroom number.”
6. The application will display this information to the user formatted for the small screen.

Alternative Flow of Events:

If the user only typed in part of a classroom number:

- Return a list of classrooms that match the current search entry. The user should be able to scroll through the list and by selecting a classroom, they are taken to that classroom’s schedule. Go to step 6 in the basic flow.

Post conditions: The user views the information for the classroom entered.

5. Mobile Student Lookup Use Case: Search by Course ID

Revision History			
Date	Issue	Description	Author
9/30/11	1.0	Initial Creation of search by course ID use case	Brandon Knight
10/5/11	1.1	Added an alternate flow handling exact and partial matches	Mark Vitale

Relevant Feature(s): 6

Brief Description: This use case describes the sequence of actions the user and the system take when the user decides he/she would like to search by course ID in the mobile application.

Basic Flow: We are going to assume that the user has the application open already, so the basic flow begins when the user goes to the search tab of the application.

1. The user will select the search bar
2. The user will select the 'course ID' option
3. The user will type a course ID number that they want to search
4. The app will automatically use the stored Kerberos username and password to navigate to server and pull off all available information available from a 'course ID search' on the website.
5. If the user typed in an exact match for a course ID, that course's information is displayed directly. If the user typed in a partial course ID, follow the alternative flow of events under "If the user only typed in part of a course ID."
6. The application will display this information to the user formatted for the small screen.

Alternative Flow of Events:

If the user only typed in part of a Course ID:

- Return a list of classrooms that match the current search entry. The user should be able to scroll through the list and by selecting a course ID, they are taken to that course's schedule. Go to step 6 in the basic flow.

Post conditions: The user views the information for the course ID entered.

6. Mobile Student Lookup Use Case: Future or Past Schedule Search

Revision History			
Date	Issue	Description	Author
10/2/11	1.0	Initial Creation of “Future or Past Schedule Search” use case.	Mark Vitale
10/3/11	1.1	Updated this use case.	Brandon Knight

Relevant Feature(s): 5

Brief Description: This use case describes the sequence of actions the user and the system take when the user is trying to find previous or future schedules for an individual.

Basic Flow: We are going to assume that the user is looking at the individual’s current schedule.

1. The user will press a button that indicates more options.
2. The app will check the schedule lookup page to view the available quarters for query (whether past or future).
3. The user will select one of these quarters from a list.
4. The app will find the selected individual’s schedule from the selected quarter.
5. The app will display that schedule to the user in the same format as all other schedules.

Alternative Flow of Events:

If the individual was not enrolled at Rose-Hulman during the selected quarter:

- Display a warning.
- Go back to the current schedule for the individual.

Pre conditions: We are assuming that the user is currently viewing an individual’s schedule in the current quarter.

Post conditions: The user views an individual’s past or future schedule information or is brought back to the user’s current schedule information.

7. Mobile Student Lookup Use Case: Overlaying Schedules

Revision History			
Date	Issue	Description	Author
10/02/11	1.0	Initial Creation of "Overlaying Schedules" use case.	Ann Say
10/03/11	1.1	Updated and edited Alternate Flow.	Brandon Knight
10/5/11	1.2	Added more steps to clarify the basic flow	Mark Vitale

Relevant Feature(s): 11

Brief Description: This use case describes the actions a user would take to overlay two or more schedules in the mobile application.

Basic Flow: The basic flow begins when the user reaches the first screen of the application, which would show their schedule.

1. A user can search for a schedule.
2. Once the schedule is viewed, the user chooses an option to overlay the schedule.
3. The user then searches for other schedules and repeats as necessary until all desired schedules are selected for overlay.
4. The user is then taken to the overlay screen where periods in which all schedules are free are made available.
5. The user, while viewing the overlay, can click a button to add more schedules to the current overlay. If the user wants to add more schedules, he or she is taken back to basic flow step 3 and all proceeds from there as usual.

Alternative Flow of Events:

The user may remove any number of schedules from the overlay while viewing the overlay

The user has tried to overlay too many schedules:

- Display a warning saying that no more new schedules can be placed in overlay.
- The most recent schedule that was attempted to be placed in the overlay (and was one too many) will not be added to the overlay.

The user wants to clear the overlay:

- An option in the overlay viewing window is the ability to clear the overlay. All schedules would be removed.

The user attempts to overlay the exact same schedule multiple times:

- No checks are made if the schedules are identical. Both schedules are placed in overlay mode.

Post conditions: Schedules are placed in the overlay view.

8. Mobile Student Lookup Use Case: Find Contact Information

Revision History			
Date	Issue	Description	Author
10/2/11	1.0	Initial Creation of “Find Contact Information” use case.	Mark Vitale
10/3/11	1.1	Edited.	Brandon Knight

Relevant Feature(s): 7

Brief Description: This use case describes the sequence of actions the user and the system take when the user is trying to find information on an individual at Rose-Hulman, whether student or faculty.

Basic Flow: We are going to assume that the user has the application open already, so the basic flow begins when the user goes to the search tab of the application.

1. The user will select the search bar.
2. The user will select the username option.
3. The user will type the username of the individual whose contact information they want.
4. The app will automatically use the stored Kerberos username and password to navigate to server and pull off all available information available from a ‘username’ on the website.
5. The user will then see a list of contact information for the individual.
    a. If the individual is a student, the information available will be username, email, advisor name, campus mail, and a link to the individual’s schedule.
    b. If the individual is a professor, the information available will be username, email, campus mail, office room number, office phone number, and a link to the individual’s schedule.

Alternative Flow of Events:

If the username entered in step 3 does not directly map to a single username (such as searching by a portion of the username):

- the list of potential matches will be listed
    - If there are no potential matches, the user will see an empty list and need to go back to basic flow step 3.
- The user will select the individual they desire information about.
- Move to basic flow step 5.

Post conditions: The user views the information for a specific user.

9. Mobile Student Lookup Use Case: E-mail a User

Revision History			
Date	Issue	Description	Author
10/2/11	1.0	Initial Creation of “E-mail a User” use case	Mark Vitale
10/3/11	1.1	Updated and edited Pre-conditions, Alternate Flows.	Brandon Knight

Relevant Feature(s): 8

Brief Description: This use case describes the sequence of actions the user and the system take when the user is trying to email a user.

Basic Flow:

1. The user will scroll to the individual’s email address in the list of contact information.
2. The user will select this item in the list.
3. The app will then pull up the native iOS mail client and have a blank new email ready to be composed, with the selected individual’s email address in the “to:” field.

Alternate Flow of Events: It is possible that the user of the app has not set up any email accounts on their iOS device. In this case, selecting the individual’s email address will navigate to the iOS native screen for creating a mail client or importing accounts.

Pre-conditions: We are going to assume that the user has an individual’s contact information already pulled up. For more information on how to get to this state, see the “Find Contact Information” use case. We also are assuming that the user has navigated to an individual’s contact information

Post conditions: The user either has a blank email addressed to the desired individual ready to be composed or the user is still on the contact information page if no email accounts have been set up in the native mail app.

10. Mobile Student Lookup Use Case: E-mail a Schedule

Revision History			
Date	Issue	Description	Author
10/02/11	1.0	Initial Creation of E-mail Schedule Use Case	Katie Greenwald
10/02/11	1.1	Added the alternate flow information	Ann Say
10/03/11	1.2	Edited, removed preconditions and alternate flow updated.	Brandon Knight

Relevant Feature(s): 13

Brief Description: User creates an email including the schedule they selected in the native iPhone e-mail app.

Basic Flow:

1. Use case begins when user chooses to send a selected schedule as an e-mail.
2. The system takes the user to the e-mail app.
3. The system opens a partially-composed e-mail containing the selected schedule as a simple text list.

Alternate Flow: If the user has not set up a mail client, or does not have any e-mail accounts programmed onto their iOS device, then selecting the e-mail will navigate to a the iOS native screen for creating a mail client or importing accounts.

Pre-conditions:

1. User has selected to e-mail a schedule.

Post conditions:

User is taken to the e-mail app with an e-mail containing the schedule they selected.

11. Mobile Student Lookup Use Case: Add to Favorites

Revision History			
Date	Issue	Description	Author
10/02/11	1.0	Initial Creation of "Add to Favorites " use case	Ann Say
10/03/11	1.1	Updated and Alternate Flow edited.	Brandon Knight

Relevant Feature(s): 12

Brief Description: The user adds a specific person, such as a student or a professor, to a list, allowing him or her to quickly view his or her schedules.

Basic Flow:

1. The user starts the application and is taken to his or her schedule.
2. The user selects the search bar and searches by username.
3. The user sees the desired person's schedule.
4. On the viewing screen, there is an option called "Add to Favorites" in which the person can select to add the user to the list.

Alternative Flow of Events: Server is not up, or it crashes after the person had already logged into the system, or the person is already in the user's favorites list

If the person is already on the user's favorites list:

- Do not add the user to the favorites list again.

Pre-conditions: The user has already used the application before, so the user's username and password are already stored.

Post conditions: When the user views his or her favorites list, the new person is in the list.

12. Mobile Student Lookup Use Case: View Favorites

Revision History			
Date	Issue	Description	Author
10/02/11	1.0	Initial Creation of “View Favorites” use case.	Ann Say
10/03/11	1.1	Updated and edited a few sections.	Brandon Knight

Relevant Feature(s): 12

Brief Description: This use case describes the sequence of actions the user and the system take when the user wants to view his or her favorites list.

Basic Flow: The basic flow begins when the user is at the main page of the application and currently views his or her schedule.

1. The user will select the ‘view favorites’ option.
2. The user goes to a list of names of users they have saved in their favorites list.
3. Clicking on their user names will show that individual’s schedule.

Alternative Flow of Events: The server may not be live, or the user may have an empty favorites list.

If the server cannot be reached:

- Display the most recent favorites list of the user.

If the user has an empty favorites list:

- A message is displayed saying that the user has no favorites added yet.

Post conditions: The user is viewing his or her favorites list.

13. Mobile Student Lookup Use Case: Sync Schedule

Revision History			
Date	Issue	Description	Author
10/02/11	1.0	Initial creation of "Sync Schedule" Use Case	Katie Greenwald
10/03/11	1.1	Edited several segments.	Brandon Knight

Relevant Feature(s): 10

Brief Description: User chooses the option to sync their schedule from one particular quarter with their iPhone Calendar. Their schedule is automatically added for all the days that class is in session for that quarter.

Basic Flow:

1. Use case begins when user selects to add their schedule to his or her calendar.
2. The system shows the quarters that the user is able to sync with his or her calendar.
3. The user selects a quarter to sync with their calendar. [Alternative Flow: The user does not want to sync a quarter with their calendar].
4. The system adds instances of the users classes from the quarter selected to all the days on which there is class in that quarter.

Pre-conditions:

1. User has working login information.
2. The Schedule Lookup Page is functional and can be accessed.
3. User has selected to add their schedule to Calendar.

Post conditions:

User's schedule for quarter selected is added to iPhone Calendar. If the user chooses to cancel, the system takes him or her back to the page they were on previously.

14. Mobile Student Lookup Use Case: View Class Roster

Revision History			
Date	Issue	Description	Author
10/07/11	1.0	Initial creation of “View Class Roster“ Use Case	Mark Vitale

Relevant Feature(s): 3

Brief Description: After a user has navigated to a course’s information page the user can view that course’s roster. Each of the individuals listed in the courses roster can then be selected for further information.

Basic Flow:

1. Use case begins when user is on the information page for a course.
2. The user presses the button on this page named “view roster.”
3. The system queries the database for the class roster and presents the user with a list of names of members of the course. The list of names on this page is clickable and will take the user to the individual information page for the name he or she selected.

Pre-conditions:

User has already navigated to a course information page.

Post conditions:

User views the roster for the class of interest.

15.Mobile Student Lookup Use Case: Call a User

Revision History			
Date	Issue	Description	Author
10/17/11	1.0	Initial Creation of “Call a User” use case	Mark Vitale

Relevant Feature(s): 14

Brief Description: This use case describes the sequence of actions the user and the system take when the user is trying to call a user.

Basic Flow:

1. The user will scroll to the individual’s phone number in the list of contact information.
2. The user will select this item in the list.
3. The app will then pull up the start a phone call to the selected phone number.

Alternate Flow of Events: It is possible that the user is using an iPod Touch or some other iOS device that does not have phone call functionality. In this case, selecting the phone number will have no reaction.

Pre-conditions: We are going to assume that the user has an individual’s contact information already pulled up. For more information on how to get to this state, see the “Find Contact Information” use case.

Post conditions: The user is in a phone call and the application loses focus to the native phone application.

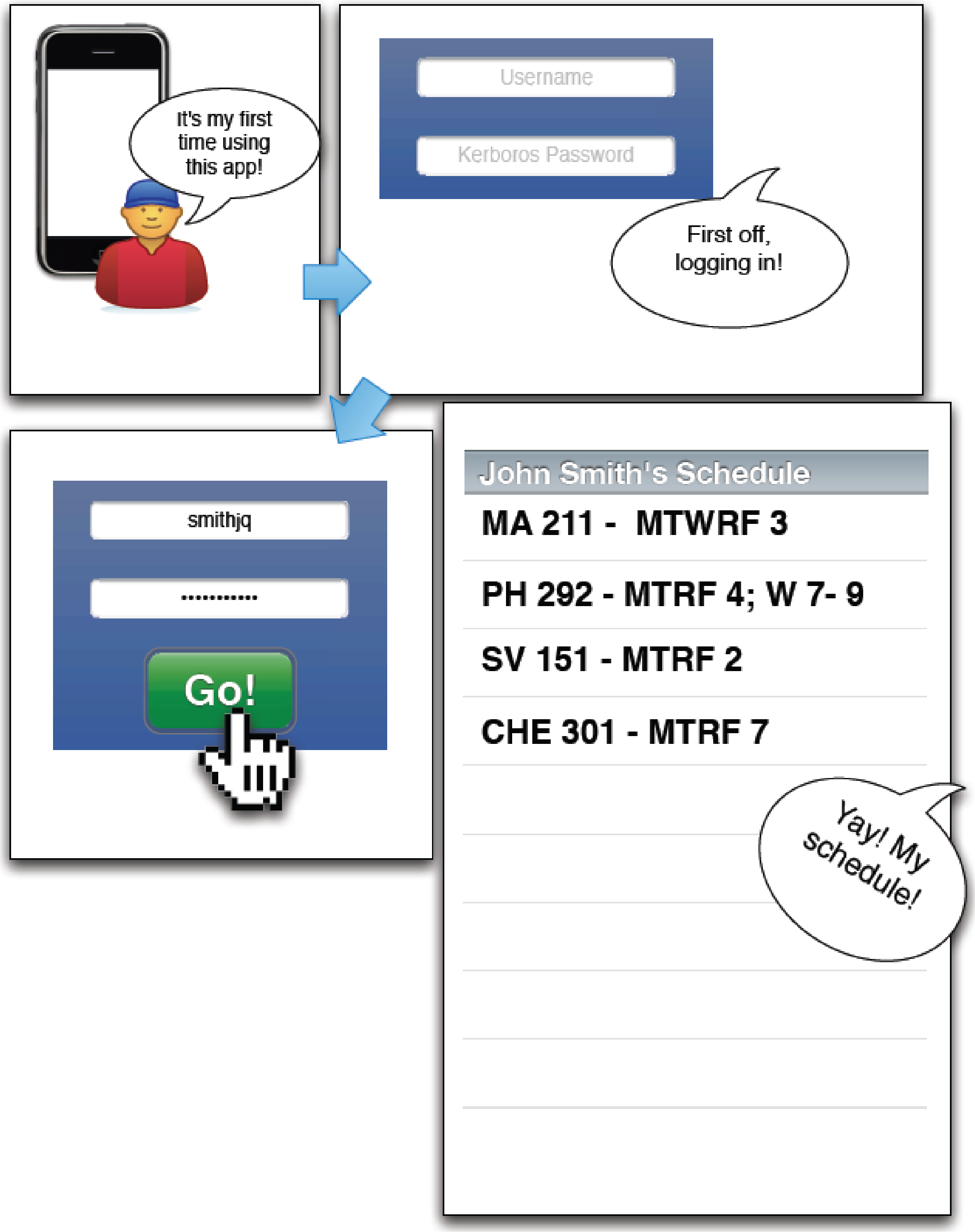
Use Case to Feature Mapping

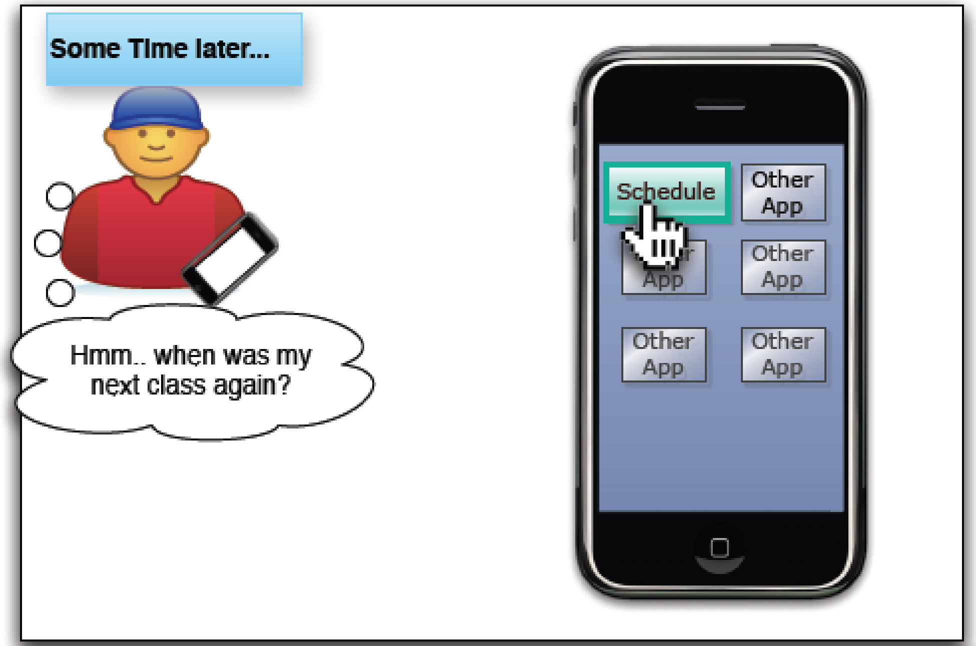
This information is also provided in each use case under “Relevant Feature(s)” but is provided in table form here for convenience.

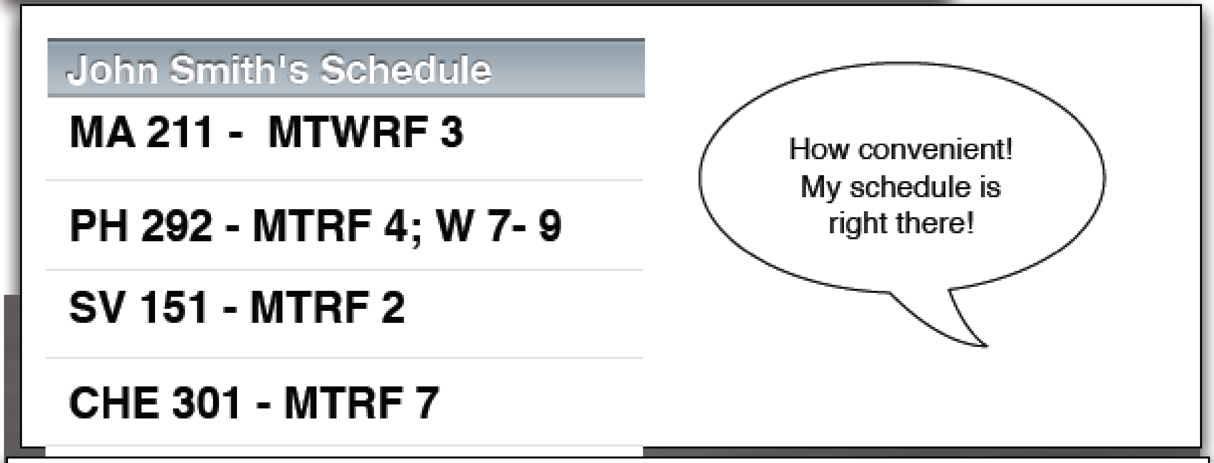
Use Case	Feature
1	2
2	4
3	1
4	9
5	6
6	5
7	11
8	7
9	8
10	13
11	12
12	12
13	10
14	3
15	14

Storyboards

First Time Use







John Smith's Schedule

MA 211 - MTWRF 3

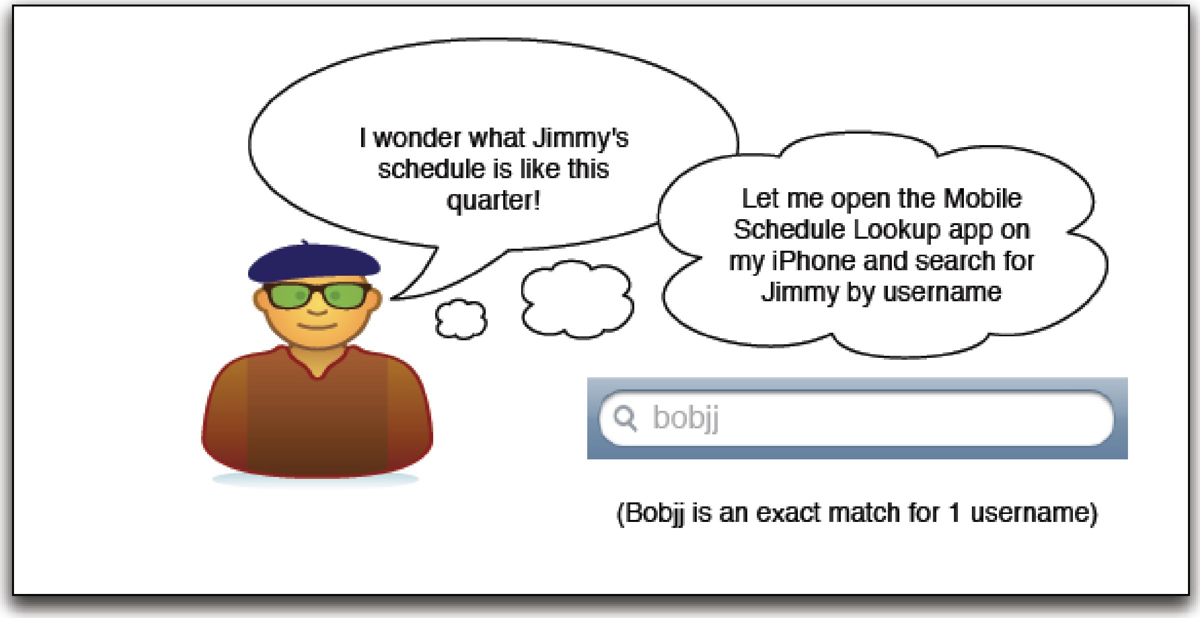
PH 292 - MTRF 4; W 7- 9

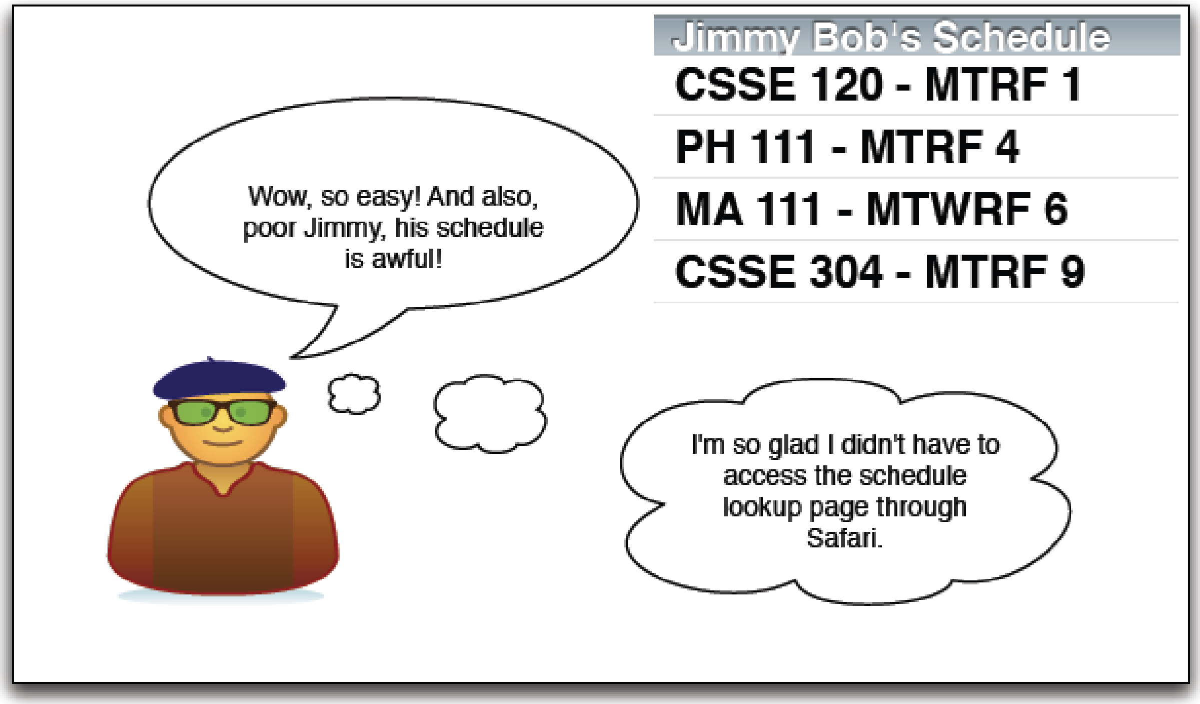
SV 151 - MTRF 2

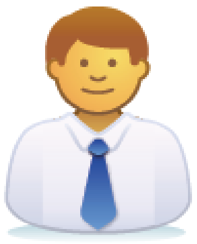
CHE 301 - MTRF 7

Once a user has entered his or her password, the app saves the password, so the person would not have to re-enter it, as long as the password remains valid.

Looking Up a Friend

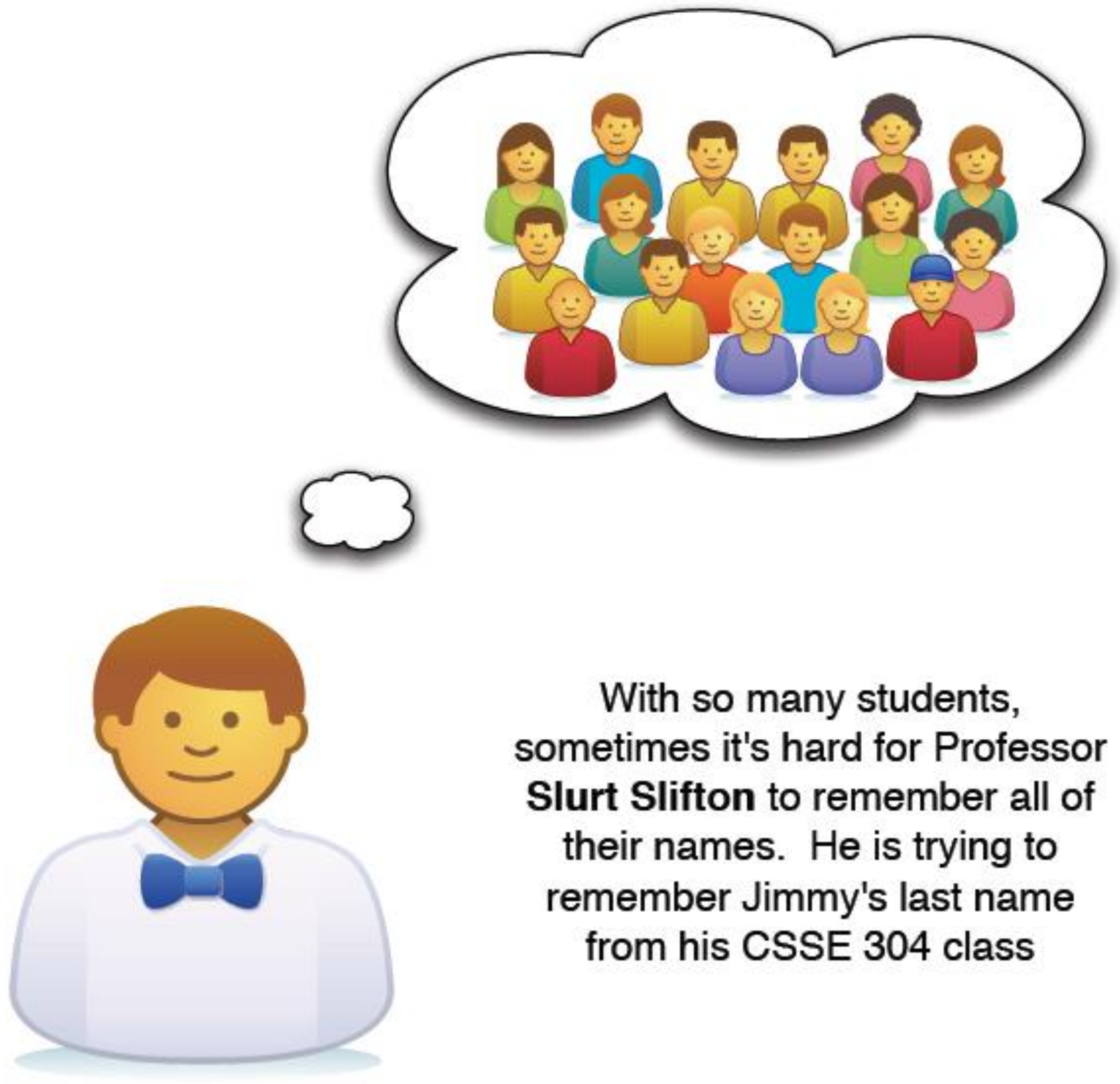

I wonder what Jimmy's schedule is like this quarter!
Let me open the Mobile Schedule Lookup app on my iPhone and search for Jimmy by username
bobjj
(Bobjj is an exact match for 1 username)


Jimmy Bob's Schedule
CSSE 120 - MTRF 1
PH 111 - MTRF 4
MA 111 - MTWRF 6
CSSE 304 - MTRF 9
Wow, so easy! And also, poor Jimmy, his schedule is awful!
I'm so glad I didn't have to access the schedule lookup page through Safari.

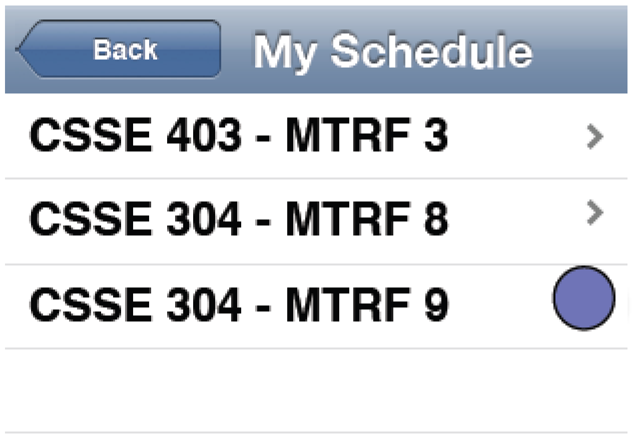


Steve wants to look up his friend Jimmy's schedule while on the go. The mobile schedule lookup app makes it easy to access the information he wants in the timeframe he needs.

Class Roster and Sending Email



With so many students, sometimes it's hard for Professor **Slurt Slifton** to remember all of their names. He is trying to remember Jimmy's last name from his CSSE 304 class

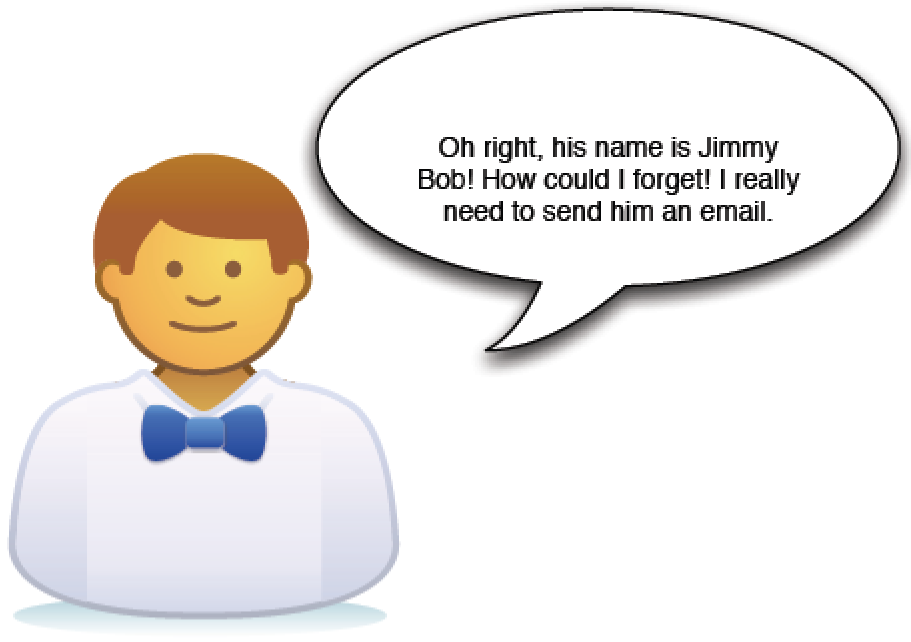


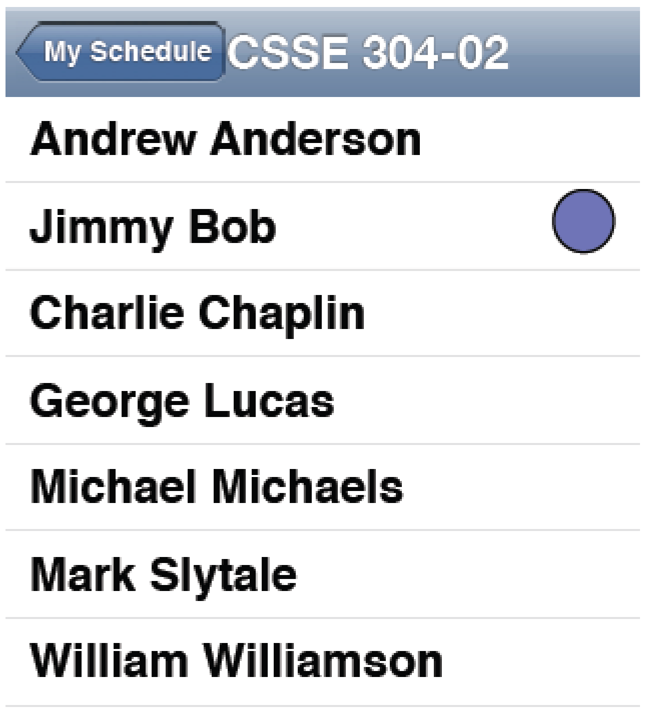
My Schedule

CSSE 403 - MTRF 3

CSSE 304 - MTRF 8

CSSE 304 - MTRF 9


Oh right, his name is Jimmy Bob! How could I forget! I really need to send him an email.


My Schedule CSSE 304-02

Andrew Anderson

Jimmy Bob ●

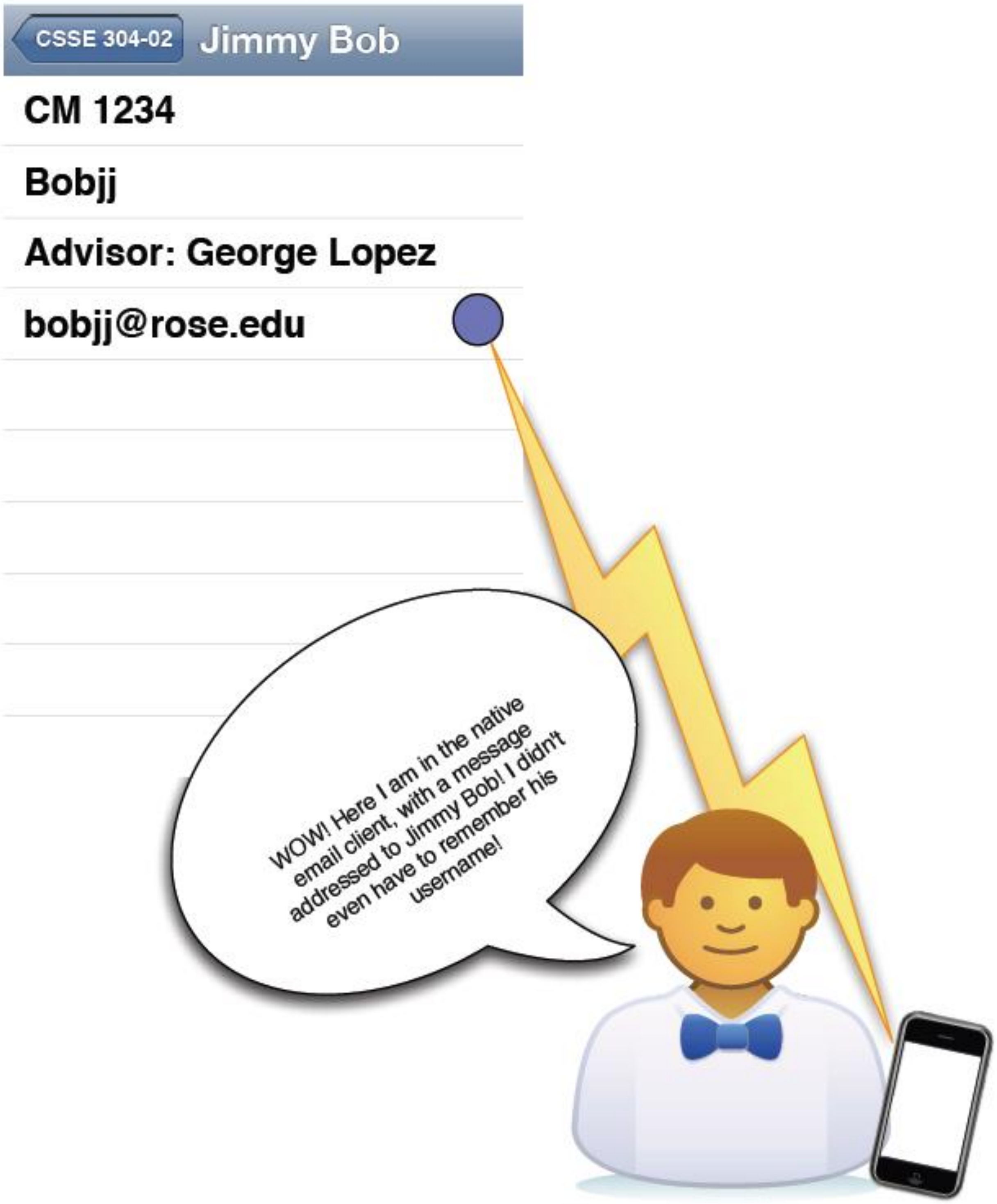
Charlie Chaplin

George Lucas

Michael Michaels

Mark Slytale

William Williamson


WOW! Here I am in the native email client, with a message addressed to Jimmy Bob! I didn't even have to remember his username!

CSSE 304-02 Jimmy Bob

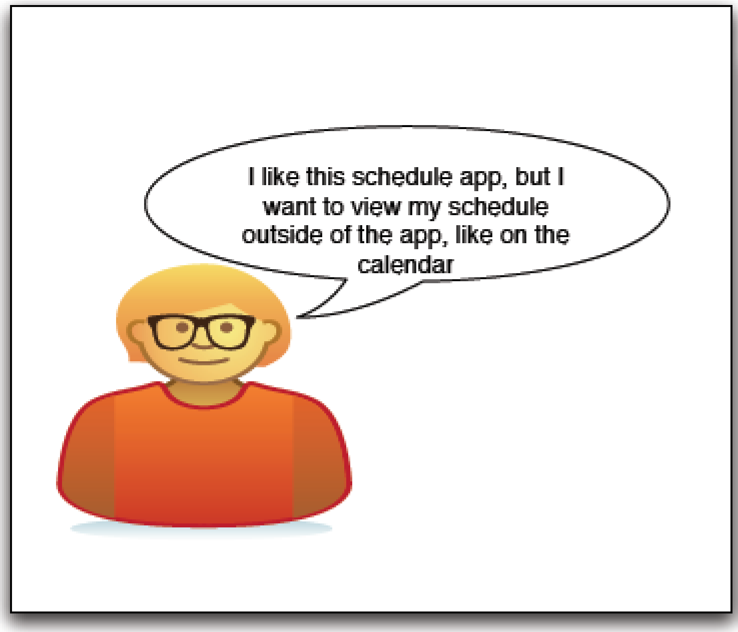
CM 1234

Bobjj

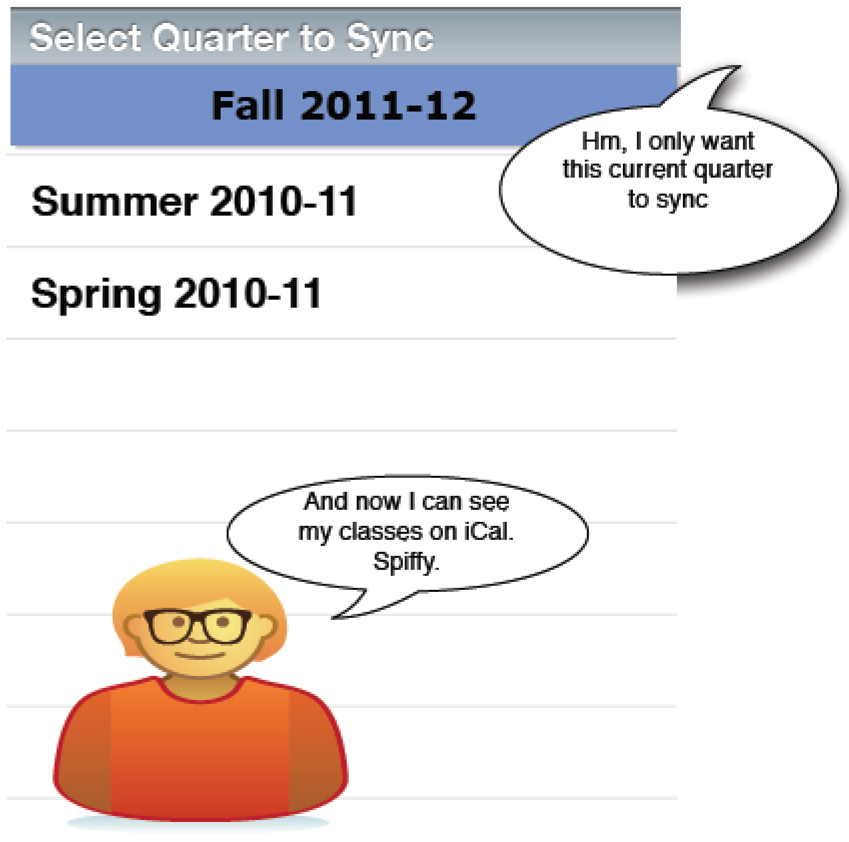
Advisor: George Lopez

bobjj@rose.edu

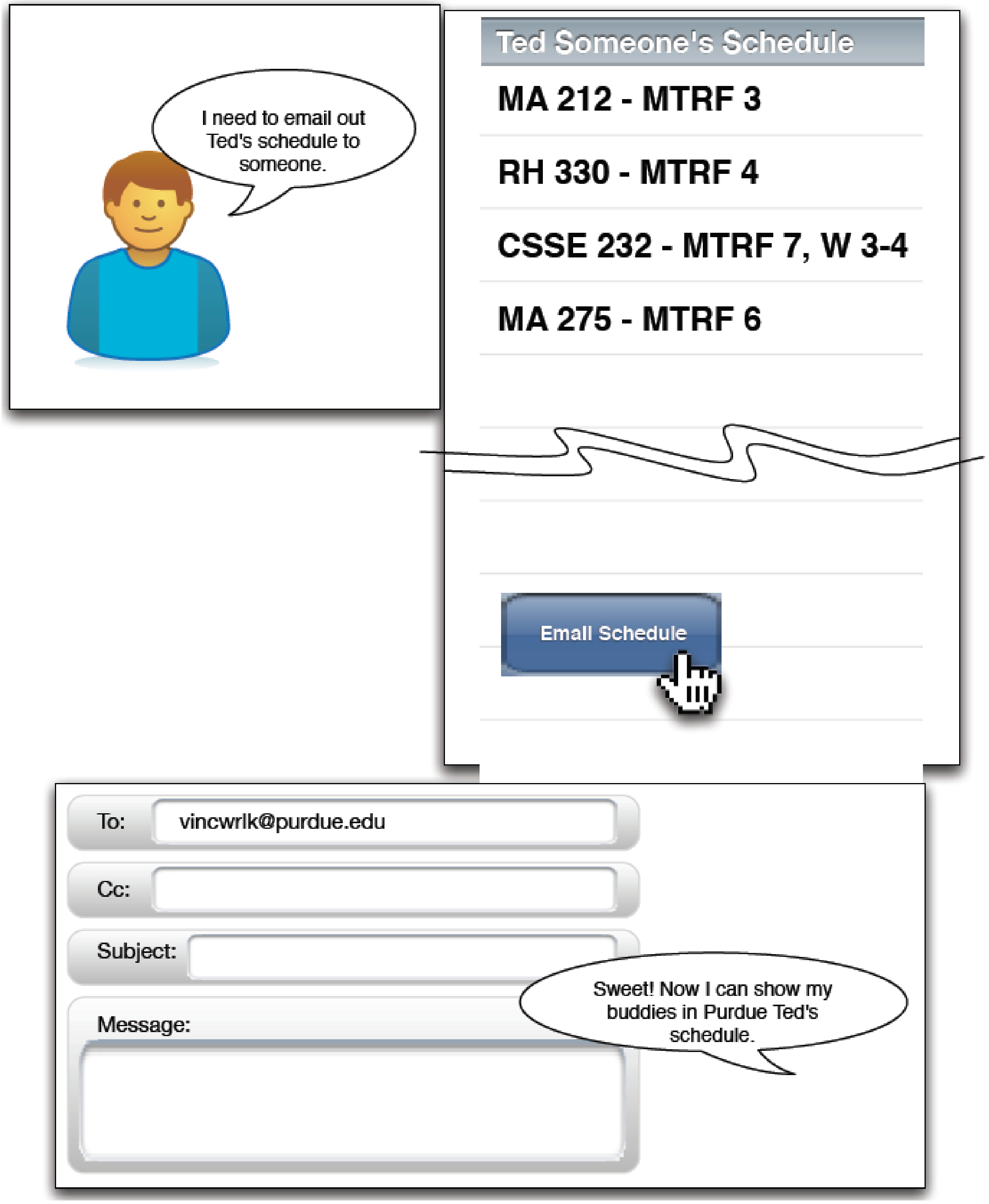
Calendar Sync


I like this schedule app, but I want to view my schedule outside of the app, like on the calendar

Oh look, there's a button for that.
Add to Calendar

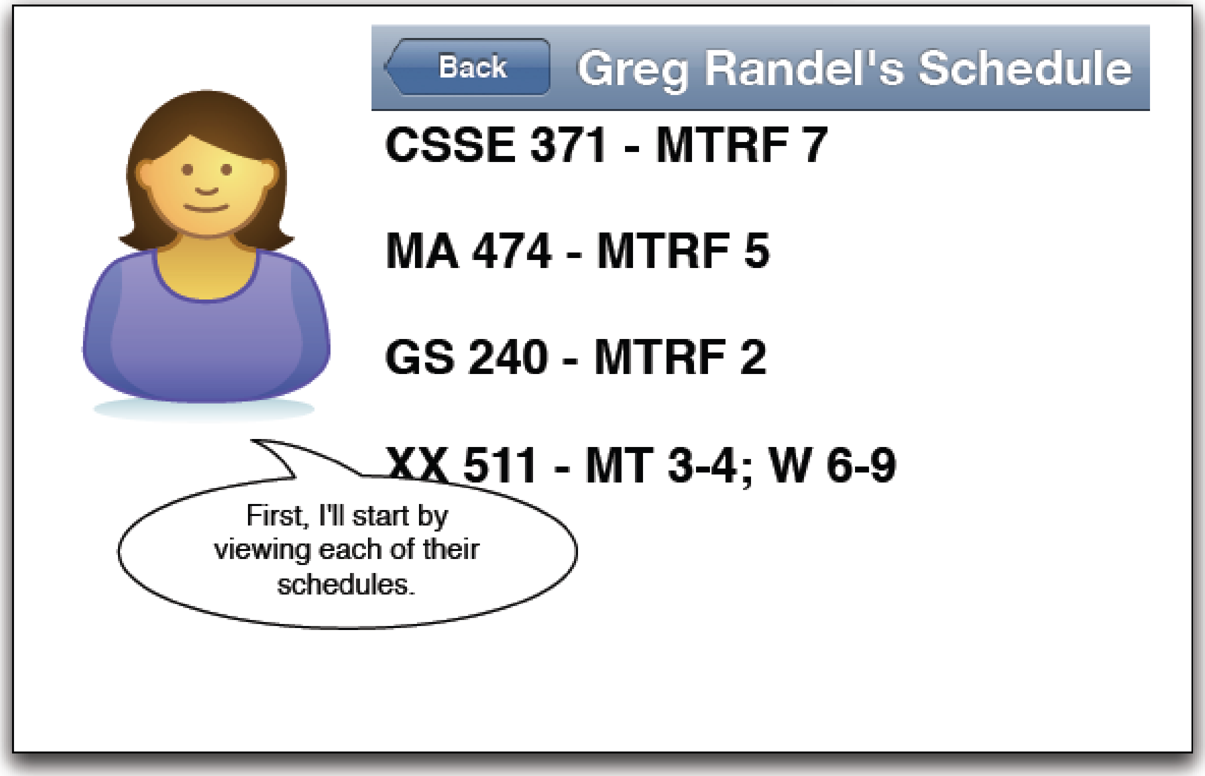

Select Quarter to Sync
Fall 2011-12
Summer 2010-11
Spring 2010-11
Hm, I only want this current quarter to sync
And now I can see my classes on iCal. Spiffy.

Email Schedule



Overlay Schedules

We have a group project, but my team members and I have different schedules
When should we meet?



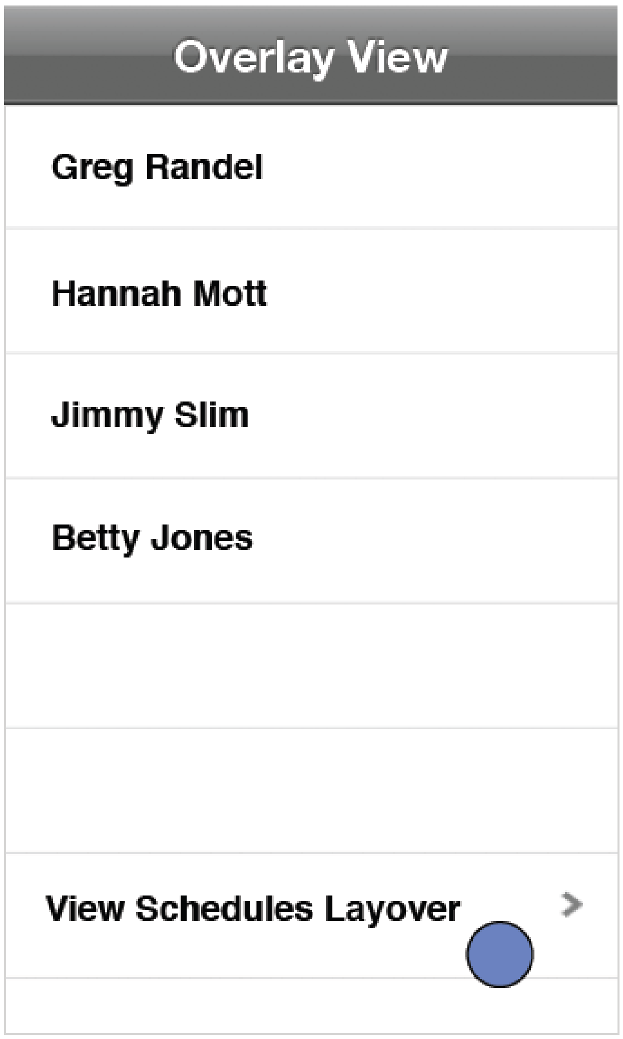
CSSE 371 - MTRF 7

MA 474 - MTRF 5

GS 240 - MTRF 2

XX 511 - MT 3-4; W 6-9

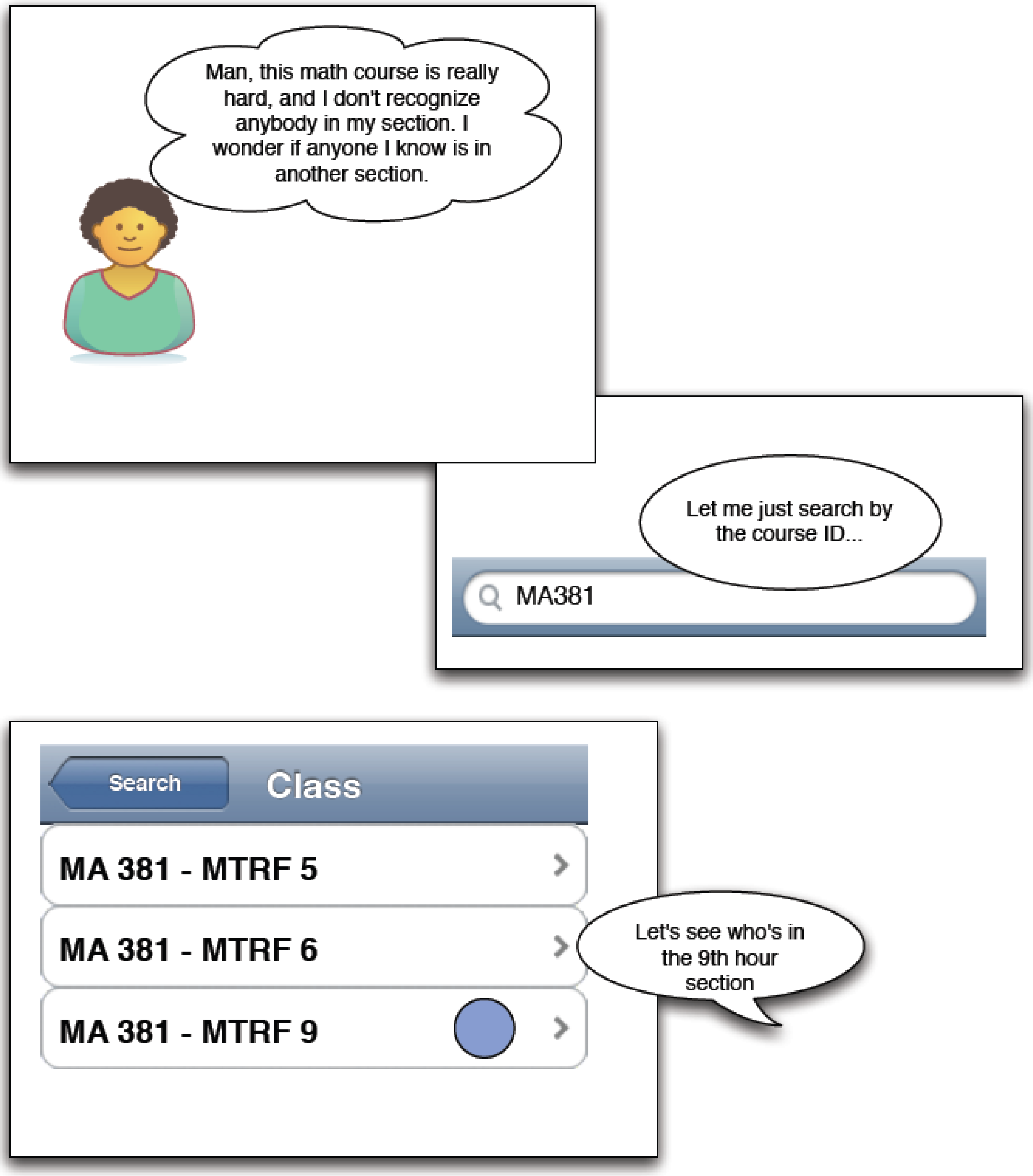
Add to overlay
Then I'll each one to the overlay.

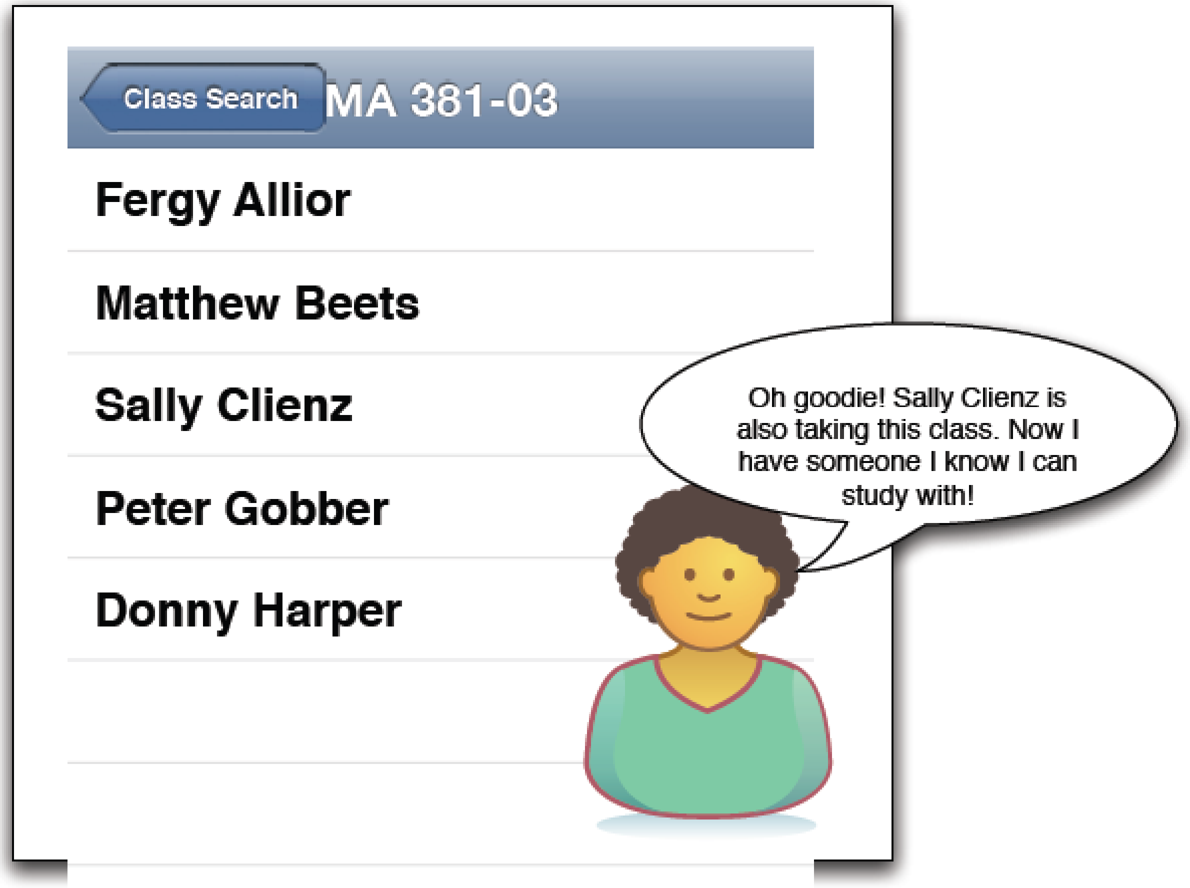

Overlay View
Greg Randel
Hannah Mott
Jimmy Slim
Betty Jones
View Schedules Layover

Monday	Tuesday	Wednesday	Thursday	Friday
randeldg	randeldg jonesba		randeldg jonesba	jonesba
randeldg motthk	randeldg motthk jonesba		randeldg motthk jonesba	motthk jonesba
slimjm motthk	motthk	slimjm	motthk	slimjm motthk
randeldg slimjm jonesba	randeldg jonesba	motthk slimjm	randeldg jonesba	randeldg slimjm jonesba
		motthk		
randeldg motthk	randeldg slimjm motthk	motthk slimjm	randeldg slimjm motthk	randeldg slimjm
randeldg slimjm	randeldg slimjm	slimjm	randeldg slimjm	randeldg slimjm
slimjm jonesba	slimjm jonesba	slimjm	slimjm jonesba	slimjm jonesba
		slimjm		

Look's like everyone is free 5th hour except for Wednesday. Why don't we meet then?
Sounds good

Search by course ID


Man, this math course is really hard, and I don't recognize anybody in my section. I wonder if anyone I know is in another section.
Let me just search by the course ID...
MA381
Search
Class
MA 381 - MTRF 5
MA 381 - MTRF 6
MA 381 - MTRF 9
Let's see who's in the 9th hour section



Fergy Allior

Matthew Beets

Sally Clienz

Peter Gobber

Donny Harper

Oh goodie! Sally Clienz is also taking this class. Now I have someone I know I can study with!

Project Background and Functionality Not In Use Cases

The mobile student lookup application will be a native application for iOS devices**Invalid source specified.**. This application will allow anyone with Rose-Hulman Network access (students or faculty) to access various schedule lookup tools provided by the Registrar's student lookup website**Invalid source specified.**. These tools include search by username (partial or whole), search by class ID, search by room number, and the ability to search through all past terms. Based on the fact that this application is essentially reorganizing and presenting existing information in a new way, almost all of our functionality is directly related to how users will interact with the system. The only functionality that is not directly represented in the use cases is the fact that the application will have to take the source from the html representation of the information available on the schedule lookup page and convert that information into a directly accessible object for the application to use.

Usability Requirements

- 95% of iOS users should prefer the native schedule lookup application to the mobile web browser option.
- Of users familiar with iPhone or iPod Touch apps, 95% of users should be able to use the application within 30 minutes of first opening it.
- Using the native application should be faster than using the schedule lookup webpage through the mobile browser.
- Following the Human Interface Guidelines published by Apple (iOS Developer Library Human Interface Guidelines, 2011) will allow this application to behave just as users will expect an iOS application to behave.

Performance Requirements

- The application will not crash after receiving a memory warning from the iOS operating system during normal use of the application. Normal use is defined as any of the use cases previously stated in Milestone 2.
- A user should be able to complete any of the use cases previously defined within 60 seconds, excluding network or server access times which are not monitored or maintained by this project.

Reliability Requirements

- The application should be functional as long as the existing schedule lookup web page is active.
- There will be no critical bugs present in the system. Critical bugs are defined as bugs that cause any use cases that relate to features marked as critical priority to not function as expected.
- Less than 25% of bugs will be defined as significant bugs. Significant bugs are bugs that cause any use cases that relate to features marked as high priority to not function as expected.
- No less than 75% of bugs will be defined as minor bugs. Minor bugs are defined as any bugs not marked as critical or major bugs.
- There will be no more than 5 bugs per thousand lines of code.

Supportability Requirements

- Long term support will be handled entirely by the clients. Potential support issues include updating the application in the case that the current schedule lookup page changes and updating the app for new iOS releases.
- The application will be designed according to Apple development standards in order to maximize modifiability.
- Object oriented design styles will be used where appropriate to increase understanding.
- Important or non-trivial methods will be commented to increase understanding of the code for future development.
- Commit messages to the Git repository will be verbose and accurately describe what the programmer aimed to accomplish with the code and be written in the present tense. Then, if a future developer is confused by certain methods, he or she can view the messages from the original creation of those methods to better understand the code.
- Communications with the current schedule lookup server will be abstracted as much as possible to allow the application to be updated in the case the schedule lookup page is changed.

Hardware and Software Interfaces

Our iPhone application will not be created to use any hardware resources explicitly. The operating system will interface with our application and may devote hardware resources (RAM) to our application. Our application may also interface with the native iPhone Calendar and Mail applications. Our application will be limited to the resolution of the iPhone screen, but will be able to respond to the 'touch' feature.

Documentation, Installation, Legal and Licensing Requirements

We do not have any legal requirements for our iPhone application other than that it must comply with all federal and local laws and regulations, and must also comply with Apple Inc.'s Legal and Licensing Requirements. The Mobile Student lookup application will be open source with code and documentation committed on GitHub.com. Installation will be handled by the client, as applications are installed to iOS devices through the App Store, and the client is handling the submission of the app to the store.**Invalid source specified.**

Design Constraints

Our iPhone application must comply with Apple Inc.'s Human Interface Guidelines (iOS Developer Library Human Interface Guidelines, 2011), which in short declare that our users should notice a negligible difference between our application and native iOS applications. We will design the iPhone application using Objective C in Xcode 4.2 for iOS 5. Our iPhone application will be physically limited by the constraints laid forth by the iPhone's operating system, display, and other hardware and software.

User Interfaces

The prototypical user interfaces are electronic and interactive. They have been submitted separately.

Coding Standards

Most of the coding standards used for this application will adhere to the Apple coding standards for Cocoa**Invalid source specified.**, which is the API for the iOS. Some examples of guidelines in the standards handbook that we would follow include: not having “do”, adverbs, or adjectives before the verb part of method names, starting method names with lowercase, using const to create constants for floating point values, not naming items with a name that begins with two underscores (these are reserved for objects created directed by Apple), and how to design initializers.

A variance from the coding standards we decided to do is use spaces instead of tabs for indentation.

Change Control

We will use Github for our version control during the coding process. Github is an website for software development projects and has a version control system to manage code. Github will track bugs marked by users and also features requested by others. The version control system will handle version conflicts if and when they arise. Eventually, the milestones and other written documents created in the development of the application will be uploaded to a wiki on Github; Gituhb requires free projects hosted on the site to be open-source, so we will provide our documentation to others to fulfill the spirit of open-source.

If we would receive requests from others, their requests will be marked as a bug at a certain location in the code, whichever part of the code that is most relevant to their inquiry. Then, the developers would notice the bug and decide amongst themselves and with the client(s) whether to fulfill the request or deny it. Their decision would be influenced in part by the level of sophistication and detail provided in the request; if the developers and/or client cannot understand exactly what the request is asking for, it cannot be implemented. A majority vote will decide whether to address this bug or feature. To prevent a single person or group of people from vetoing every request, each person has a limit of five rejects allowed for every ten request received.

Test Cases

Invalid or no credentials use case

Test Case ID	Scenario	Description	Condition: Username and password are correct	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	V	User is logged in and is viewing his/her own schedule
2	1	Basic Flow: username or password is incorrect	I	V	User is prompted to re-enter username and password
3	2	Alternate Flow: No Network Connection	N/A	I	User views error stating that network connection required for this app.

View Own Schedule

Test Case ID	Scenario	Description	Condition: User is currently on "My Schedule" tab	Condition: Valid credentials are stored in the app	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	V	V	User views his/her own schedule in a tabular format
2	2	Alternate Flow: username or password is incorrect	N/A	I	V	User is prompted to re-enter username and password
3	3	Alternate Flow: No Network Connection	N/A	N/A	I	User views error stating that network connection required for this app.

Search By Username

Test Case ID	Scenario	Description	Condition: User typed in complete, correct username	Condition: Valid credentials are stored in the app	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	V	V	User views an information page on the user with the specified username.
2	2	Alternate Flow: User typed an incomplete or partial username	I	V	V	User views a list of usernames that match the current search entry. Selecting any one will take the user to the information page on the user with the selected username. If no matches, an empty list will be displayed.
3	3	Alternate Flow: username or password is incorrect	N/A	I	V	User is prompted to re-enter username and password
4	4	Alternate Flow: No Network Connection	N/A	N/A	I	User views error stating that network connection required for this app.

Search By Classroom

Test Case ID	Scenario	Description	Condition: User typed in complete, correct classroom number	Condition: Valid credentials are stored in the app	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	V	V	User views an information page on the classroom with the specified classroom number.
2	2	Alternate Flow: User typed an incomplete or partial classroom number	I	V	V	User views a list of classroom numbers that match the current search entry. Selecting any one will take the user to the information page on the classroom with the selected classroom number. If no matches, an empty list will be displayed.
3	3	Alternate Flow: username or password is incorrect	N/A	I	V	User is prompted to re-enter username and password
4	4	Alternate Flow: No Network Connection	N/A	N/A	I	User views error stating that network connection required for this app.

Search By Course ID

Test Case ID	Scenario	Description	Condition: User typed in complete, correct username	Condition: Valid credentials are stored in the app	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	V	V	User views an information page on the course with the specified course ID.
2	2	Alternate Flow: User typed an incomplete or partial username	I	V	V	User views a list of course IDs that match the current search entry. Selecting any one will take the user to the information page on the course with the selected course ID. If no matches, an empty list will be displayed.
3	3	Alternate Flow: username or password is incorrect	N/A	I	V	User is prompted to re-enter username and password
4	4	Alternate Flow: No Network Connection	N/A	N/A	I	User views error stating that network connection required for this app.

Search Future or Past Schedule

Test Case ID	Scenario	Description	Condition: User selects valid quarter from individual's information page	Condition: Valid credentials are stored in the app	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	V	V	User views the schedule of the individual from the selected quarter
2	2	Alternate Flow: User selected an invalid quarter	I	V	V	User will see a warning that the individual was not enrolled during that quarter and will view the current schedule for the individual
3	3	Alternate Flow: username or password is incorrect	N/A	I	V	User is prompted to re-enter username and password
4	4	Alternate Flow: No Network Connection	N/A	N/A	I	User views error stating that network connection required for this app.

Overlaying Schedules

Test Case ID	Scenario	Description	Condition: User is trying to overlay an allowed number of schedules	Condition: User wants to clear the overlay	Condition: User adds the same schedule multiple times	Condition: Valid credentials are stored in the app	Condition: Network connection available	Expected Result
1	1	Basic Flow: everything is normal	V	I	I	V	V	User views selected schedules in the overlay view.
2	2	Alternate Flow: User has tried to overlay too many schedules	I	I	I	V	V	User sees a warning that no more schedules can be placed in the overlay, the most recent attempt to add a schedule will be unsuccessful.
3	3	Alternate Flow: User wants to clear the overlay	V	V	I	V	V	All schedules are removed from the overlay. User is returned to the empty list of schedules in the overlay.
4	4	Alternate Flow:	V	I	V	V	V	No checks are made, the user will see the same schedule appear multiple times in the overlay.
5	5	Alternate Flow: username or password is incorrect	N/A	N/A	N/A	I	V	User is prompted to re-enter username and password.
6	6	Alternate Flow: No Network Connection	N/A	N/A	N/A	N/A	I	User views error stating that network connection required for this app.

Find Contact Information Use Case

Test Case ID	Scenario	Description	Condition: Valid credentials are stored in app	Condition: Network connection available	Condition: Has search bar selected	Condition: Selects "username" option	Condition: Types full username.	Condition:	Expected Result
1	1	Basic Flow: user selects search bar	V	V	V	V	N/A		User will be able to search
2	1	Basic Flow: username or password is incorrect	I	V	N/A	N/A	N/A		User is prompted to re-enter username and password
3	2	Alternate Flow: types only partial username	N/A	I	V	V	I		The application should list available users who match the partial username. (if any).

E-mail User Use Case

Test Case ID	Scenario	Description	Condition: User is viewing contact information	Condition: Network connection available	Condition: User has native e-mail app set up.		Expected Result
1	1	Basic Flow: User navigates to individuals contact information.	V	V	N/A		User will be able to go to users e-mail through touch.
2	1	Basic Flow: username or password is incorrect	N/A	V	N/A		User is prompted to re-enter username and password
3	2	Alternate Flow: No e-mail account set up	N/A	V	V		The OS will prompt the user to create a e-mail account.

E-mail a Schedule Use Case

Test Case ID	Scenario	Description	Condition: User is viewing a schedule	Condition: Network connection available	Condition: User has native e-mail app set up.		Expected Result
1	1	Basic Flow: User navigates to any viewable schedule	V	V	N/A		User will be able to click the “e-mail schedule” button.
2	1	Basic Flow: username or password is incorrect	N/A	V	N/A		User is prompted to re-enter username and password
3	2	Alternate Flow: No e-mail account set up	N/A	V	V		The OS will prompt the user to create a e-mail account.

Add to Favorites Use Case

Test Case ID	Scenario	Description	Condition: User has searched for a user	Condition: Network connection available	Expected Result
1	1	Basic Flow: User navigates to individuals contact information.	V	V	User will be able to hit "add to favorites"
2	1	Basic Flow: username or password is incorrect	N/A	V	User is prompted to re-enter username and password
3	2	Alternate Flow: User is already in the favorites	V	V	The application will not add the user to the favorites list.
4	2	Alternate Flow: No internet or network connection	N/A	I	The application will display a toast message and will not display any information that requires network.

View Favorites

Test Case ID	Scenario	Description	Condition: User is any page.	Condition: Network connection available		Expected Result
1	1	Basic Flow: User presses “My favorites tab”	V	V		User will view his or her favorites.
2	1	Basic Flow: username or password is incorrect	N/A	V		User is prompted to re-enter username and password
3	2	Alternate Flow: No internet connection	N/A	I		The application will display a toast message and will not display any information that requires network.

Sync Schedule Use Case

Test Case ID	Scenario	Description	Condition: User is viewing his or her schedule.	Condition: Network connection available		Expected Result
1	1	Basic Flow: User chooses to add his or her schedule to the calendar application.	V	V		The application will add the classes of the current term to the schedule.
2	1	Basic Flow: username or password is incorrect	N/A	V		User is prompted to re-enter username and password
3	2	Alternate Flow: No internet connection	N/A	I		The application will display a toast message and will not display any information that requires network.
4	2	Alternate Flow: User chooses a different term to sync.	V	V		The application will add the classes to the schedule.

View Class Roster Use Case

Test Case ID	Scenario	Description	Condition: User is viewing course information page.	Condition: Network connection available		Expected Result
1	1	Basic Flow: User navigates to course roster	V	V		User will view the course roster.
2	1	Basic Flow: username or password is incorrect	N/A	V		User is prompted to re-enter username and password
3	2	Alternate Flow: No internet connection	N/A	I		The application will display a toast message and will not display any information that requires network.

Executive Summary

A useable prototype was developed for the Mobile Student Lookup application as a vector PDF. This prototype was tested with various users, who compared this proposed interface against the current web system. After reviewing user feedback, we made changes to the interface that would address these issues. We also wrote up the Interaction Architecture, which describes the different types of screens the user will encore while using the application.

Introduction

We have done extensive design to our prototype, through gathering requirements, designing the interface to best suit the iPhone environment, and determining all the various use cases the user can be in during use of the application. But even with this, the prototype is hardly useful until others have tested it and tried out the features; even though the developers may think it is easy to use, that decision lies ultimately with the user. Before testing could be done on the users, various paperwork had to be created, including Consult Forms, Pre-test and Post-test Questionnaires, and Test Procedures. Once these were made, users from various demographics were chosen to test the two systems – the currently used web system, and our iPhone application. The test participants were students ranging from freshman up to graduate students, and included males and females, and people familiar and unfamiliar with the iPhone. Each test participant was given the same briefing before, during, and after the testing. Using the results from the usability tests, a revised version of the interface was designed to be more user-friendly and easier to use.

Project Background

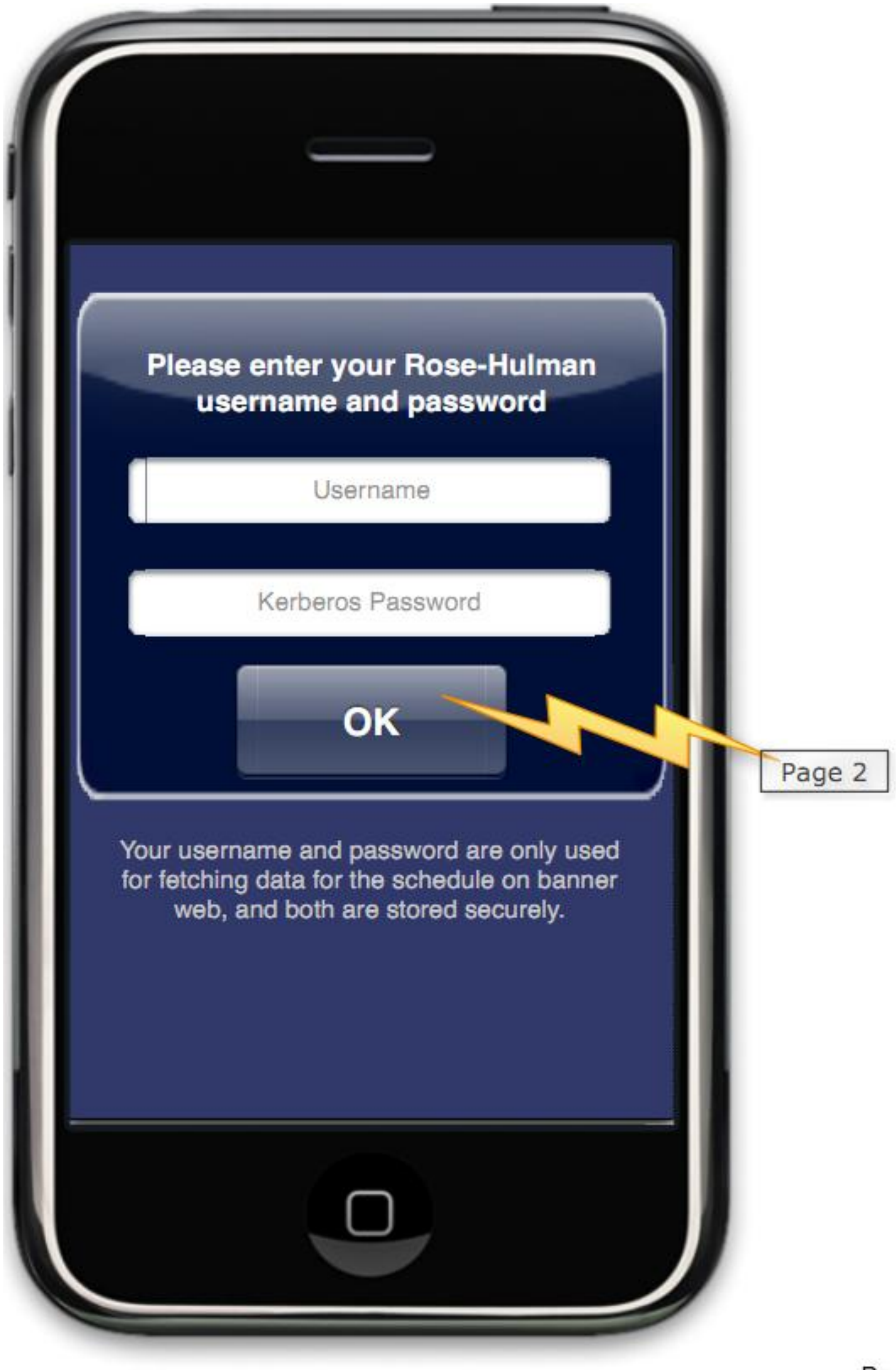
The Mobile Student Lookup Application is a mobile application developed for the iPhone, iPod Touch, and iPad hardware systems. Our clients for this project are Tim Ekl and Eric Stokes, both of which are Rose-Hulman Graduates. Features in this application were determined from the current web system in place, and will not lose any functionality from the functions of the current system, but will not include the class scheduling system. These features include: searching for a student's schedule by username, searching by room number, searching by class number, and overlaying schedules. The application needs to be user friendly, intuitive, and act as an improvement over using the current system on any of our application's supported platforms.

Usability Report

The usability report is stored as an HTML file on the computer located in the usability lab.

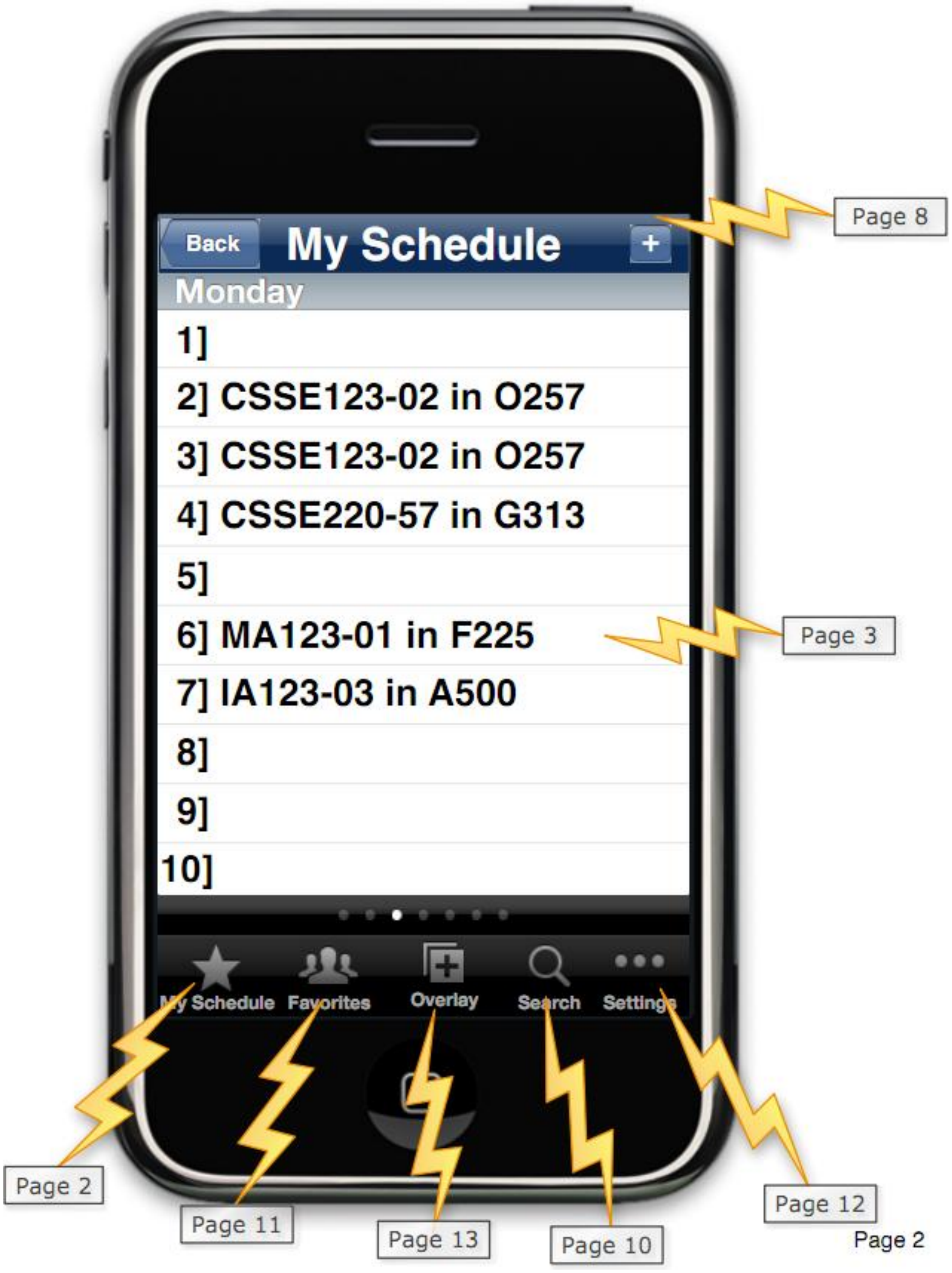
Interaction Architecture

The interaction architecture plan is on the following pages. As these interface architecture drawings are being inserted into this document from a standalone document, conflicting page numbers will exist on these pages. To find a page pertaining to the entire document, look for the page number below the footer bar. For the page numbers referred to by the actual interaction architecture drawings, look at the page number above the footer bar, as those numbers are local to the actual interaction architecture document.



Your username and password are only used for fetching data for the schedule on banner web, and both are stored securely.

Page 1


Back

My Schedule

+

Monday

1]

2] CSSE123-02 in O257

3] CSSE123-02 in O257

4] CSSE220-57 in G313

5]

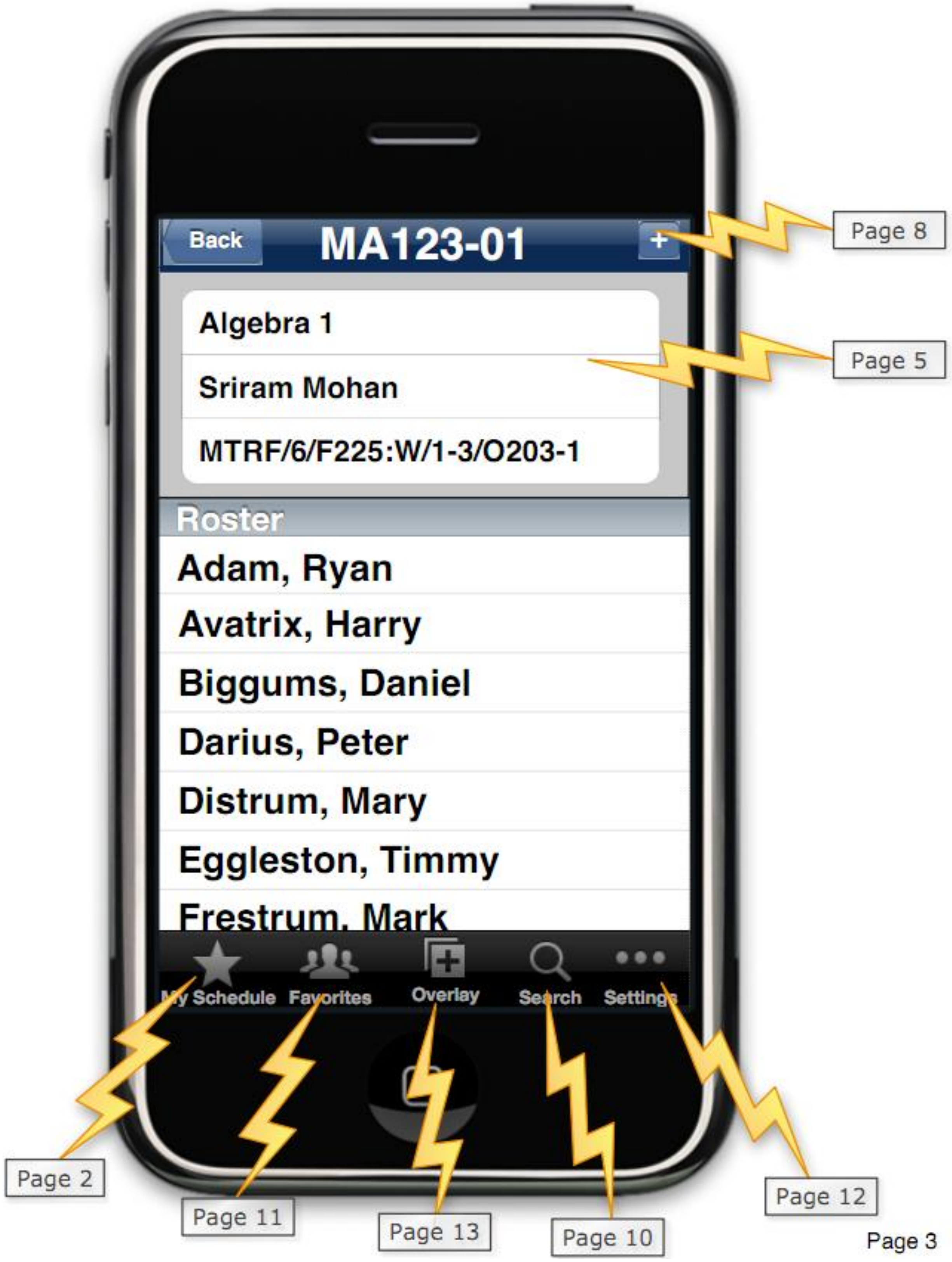
6] MA123-01 in F225

7] IA123-03 in A500

8]

9]

10]



Algebra 1

Sriram Mohan

MTRF/6/F225:W/1-3/O203-1

Roster

Adam, Ryan

Avatrix, Harry

Biggums, Daniel

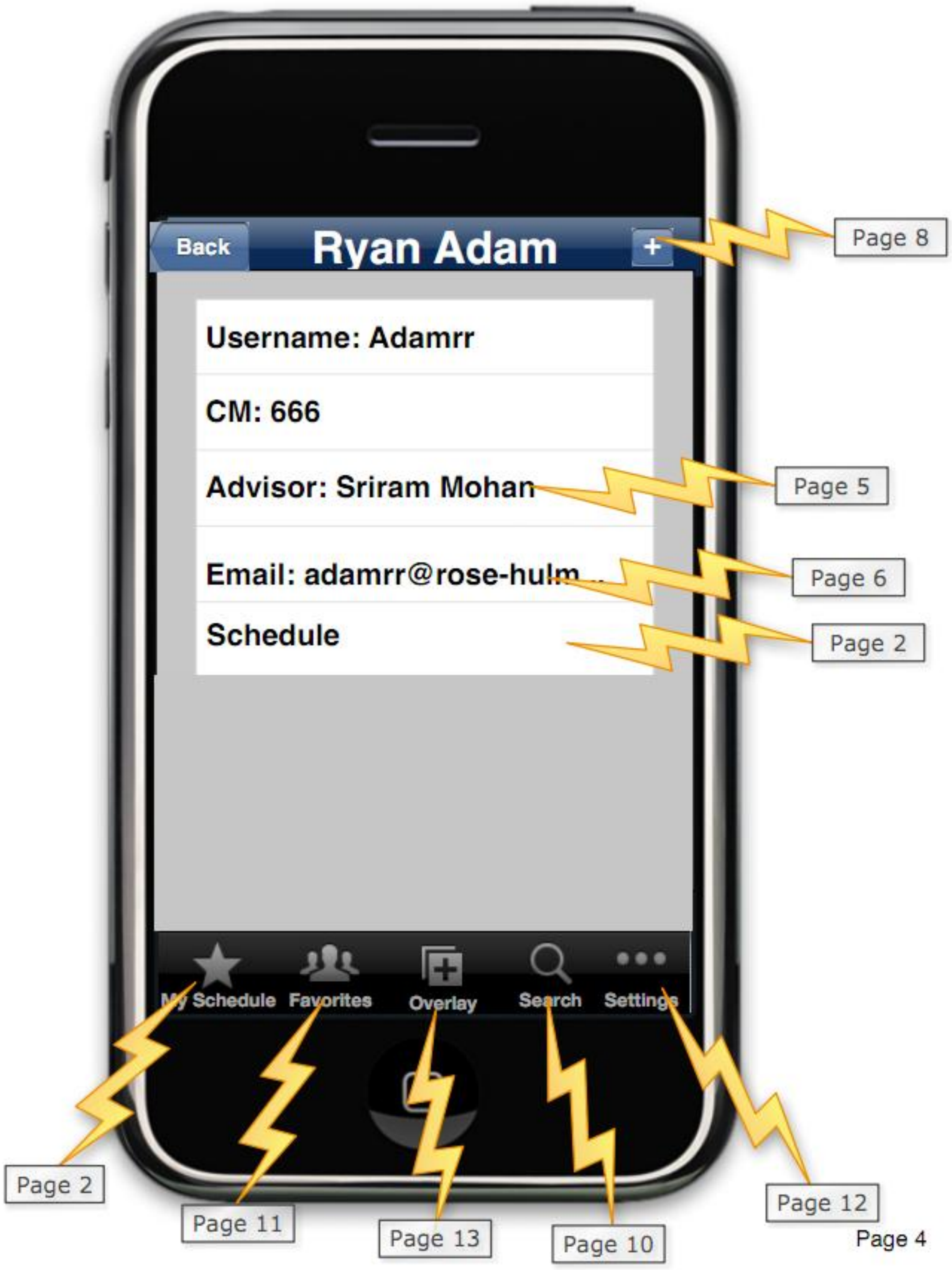
Darius, Peter

Distrum, Mary

Eggleston, Timmy

Frestrum, Mark

Page 3



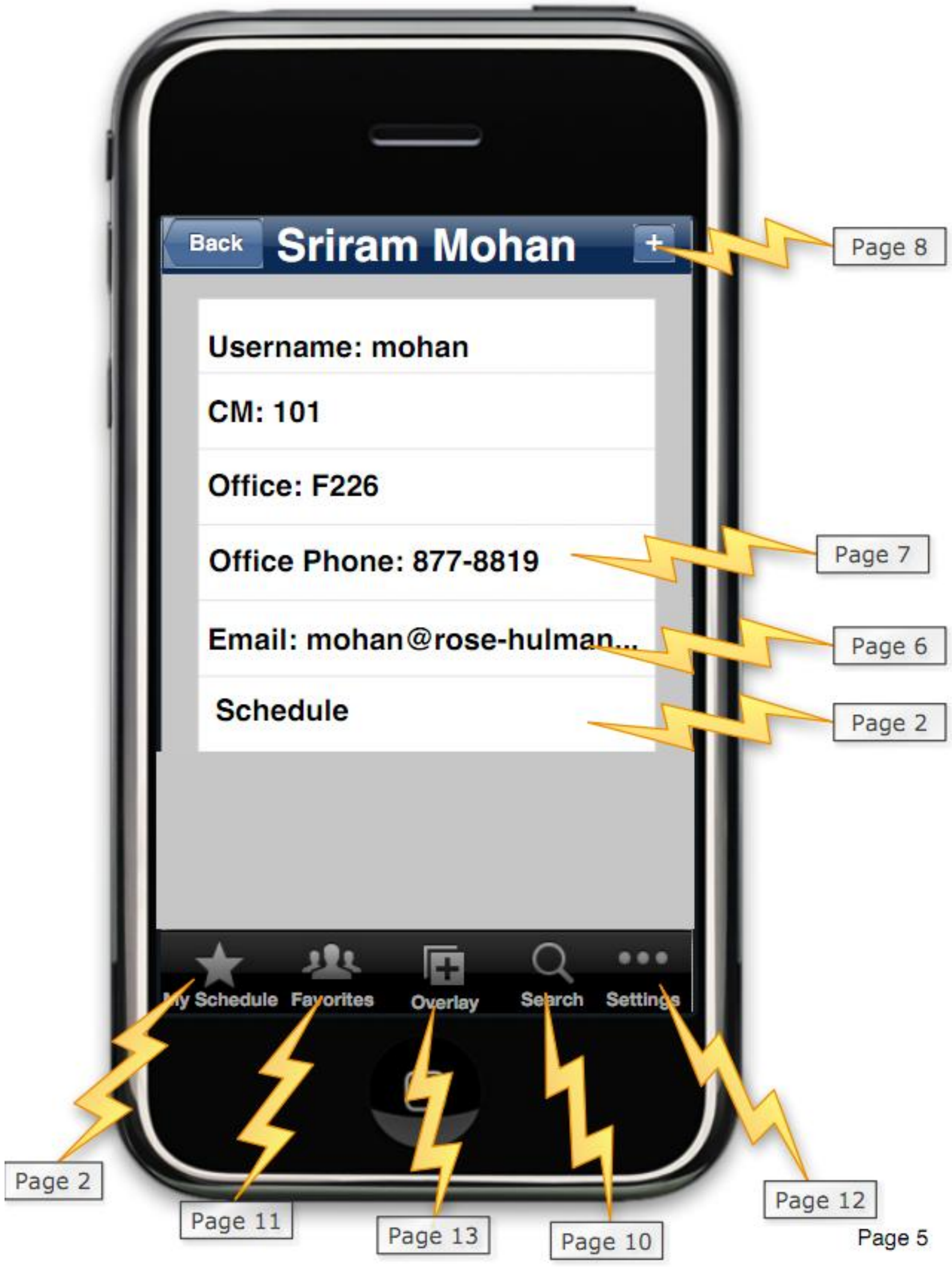
Username: Adamrr

CM: 666

Advisor: Sriram Mohan

Email: adamrr@rose-hulm...

Schedule


Back

Sriram Mohan

+

Username: mohan

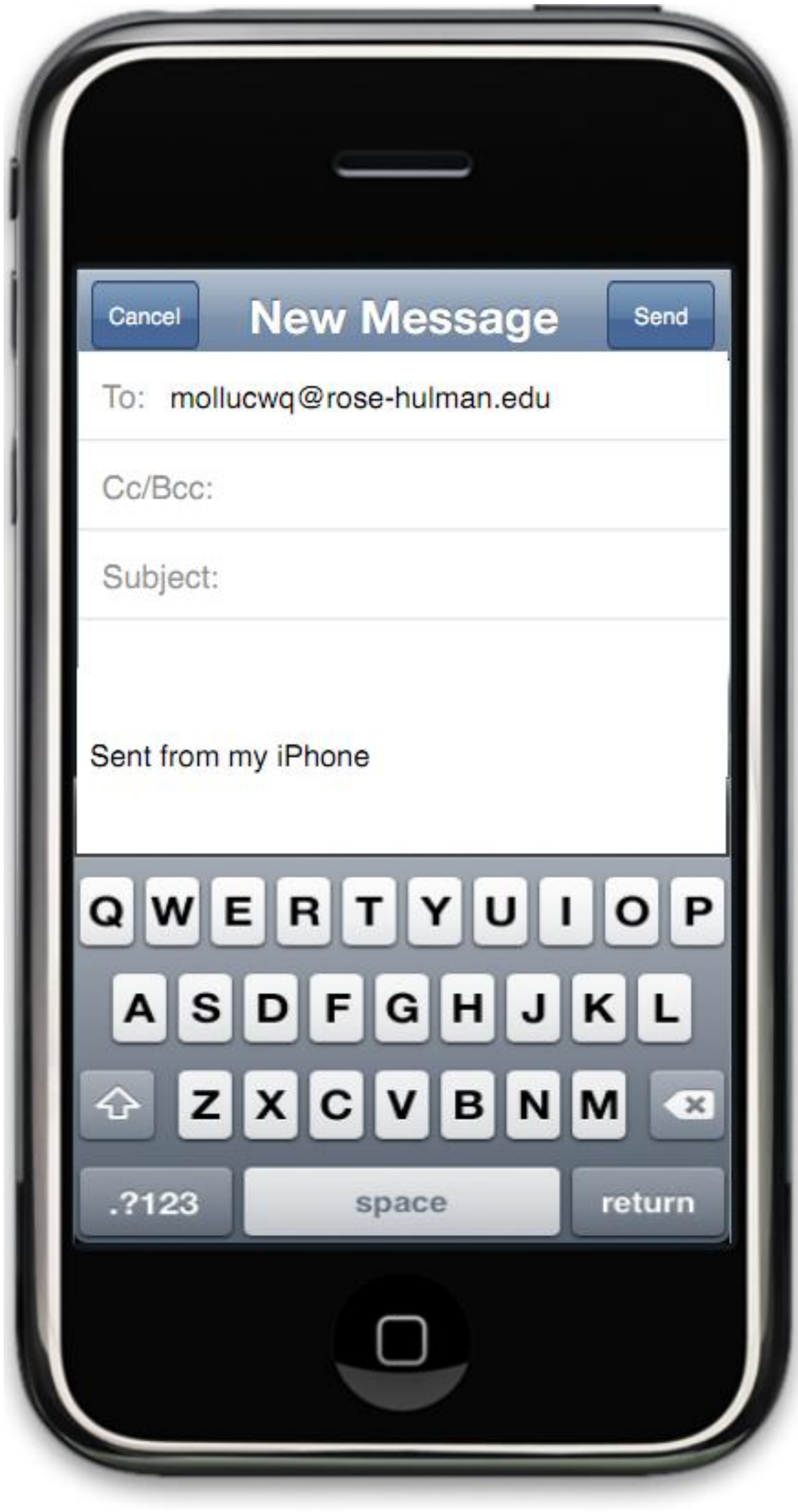
CM: 101

Office: F226

Office Phone: 877-8819

Email: mohan@rose-hulman...

Schedule



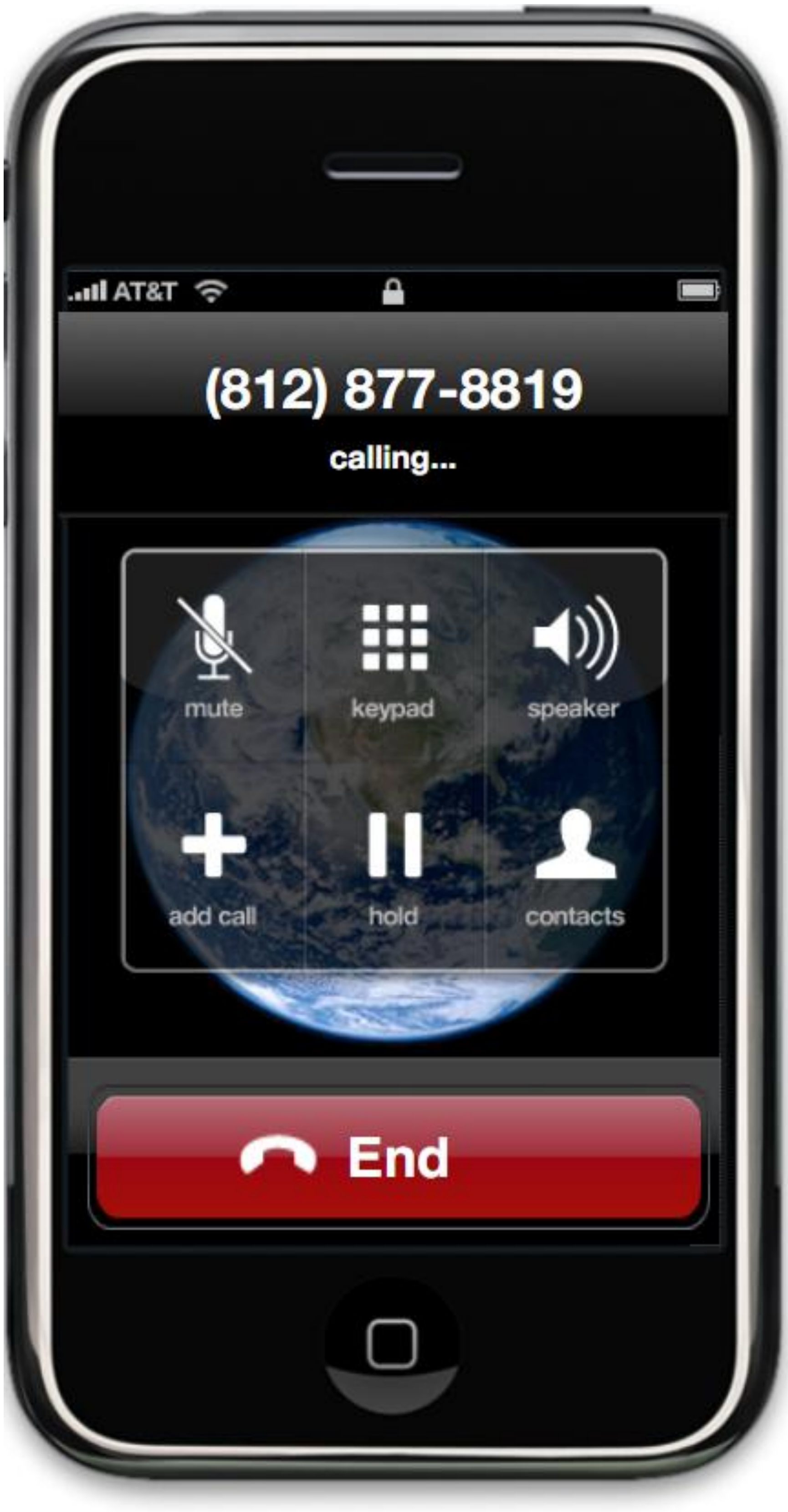
To: mollucwq@rose-hulman.edu

Cc/Bcc:

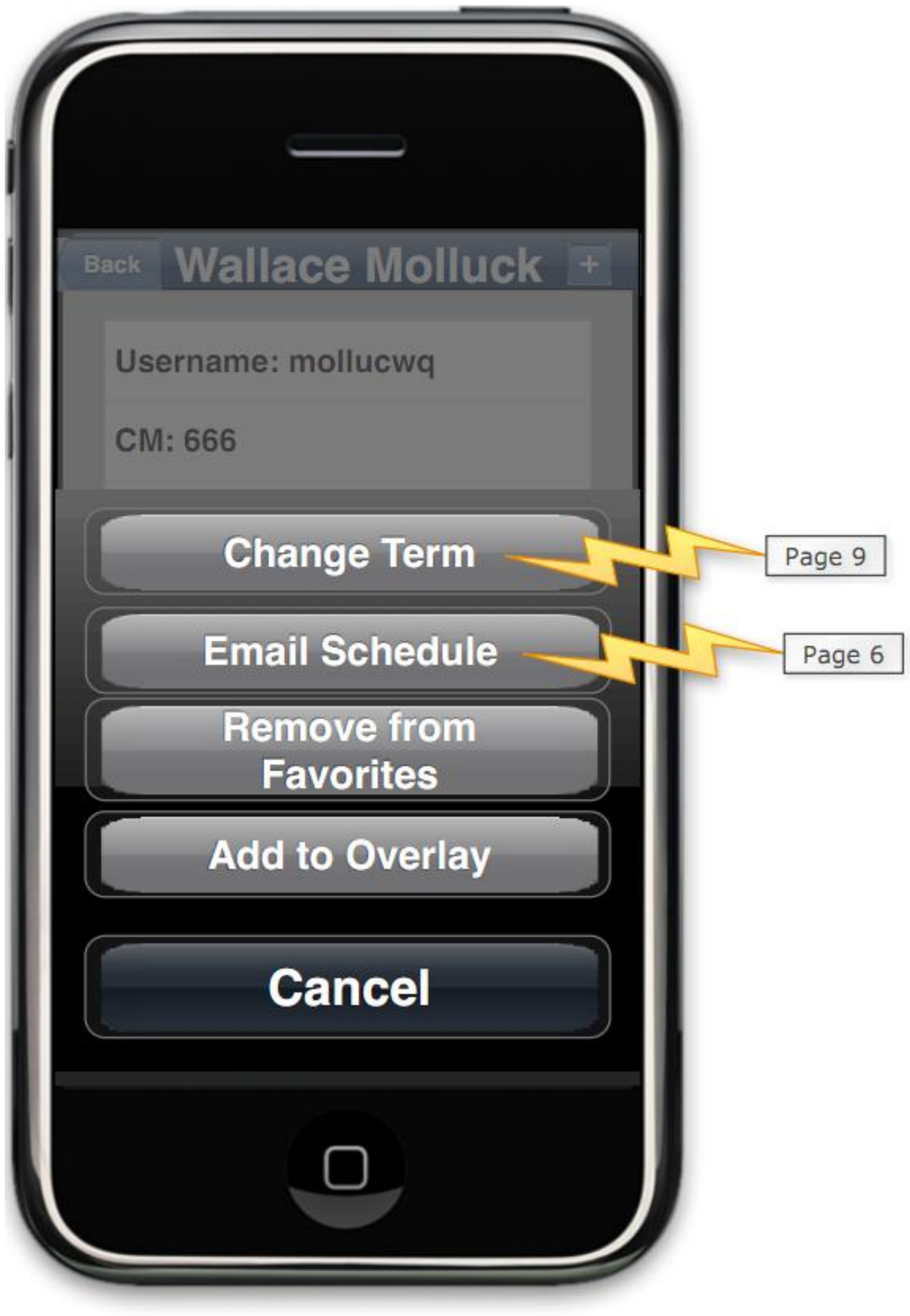
Subject:

Sent from my iPhone

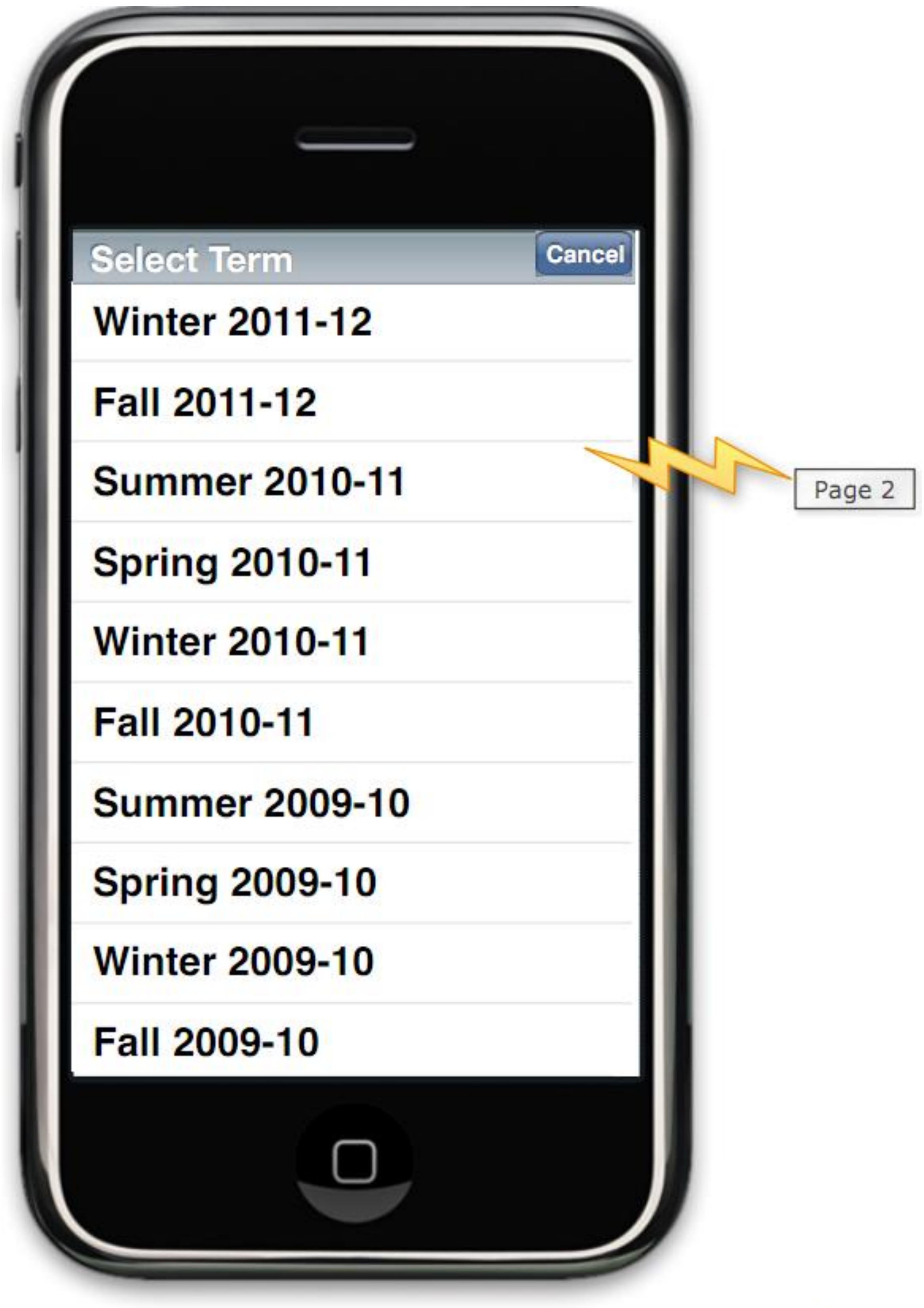
Page 6


AT&T
(812) 877-8819
calling...
mute
keypad
speaker
add call
hold
contacts
End

Page 7


Back
Wallace Molluck
+
Username: mollucwq
CM: 666
Change Term
Page 9
Email Schedule
Page 6
Remove from Favorites
Add to Overlay
Cancel

Page 8



Select Term

Cancel

Winter 2011-12

Fall 2011-12

Summer 2010-11

Spring 2010-11

Winter 2010-11

Fall 2010-11

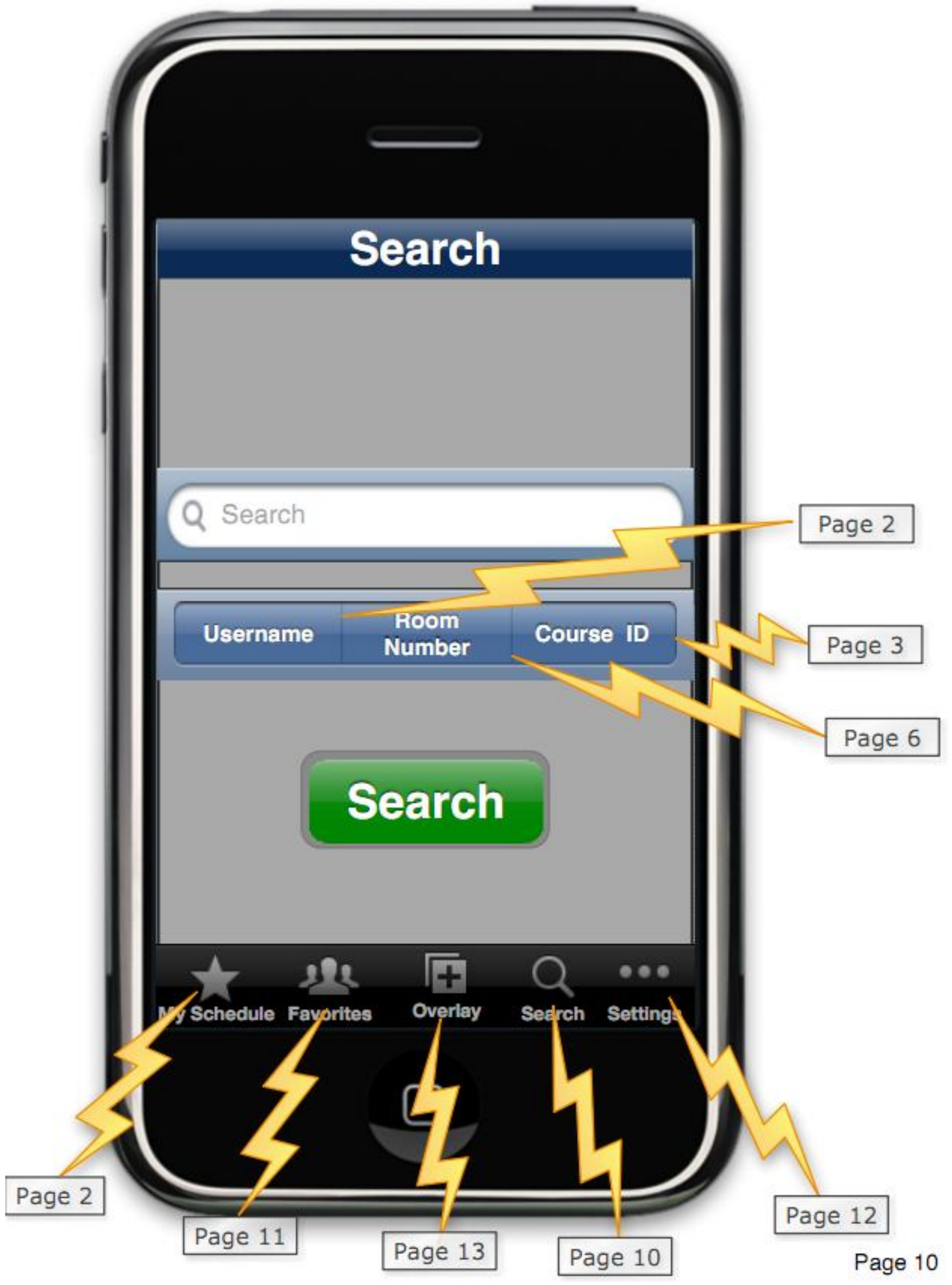
Summer 2009-10

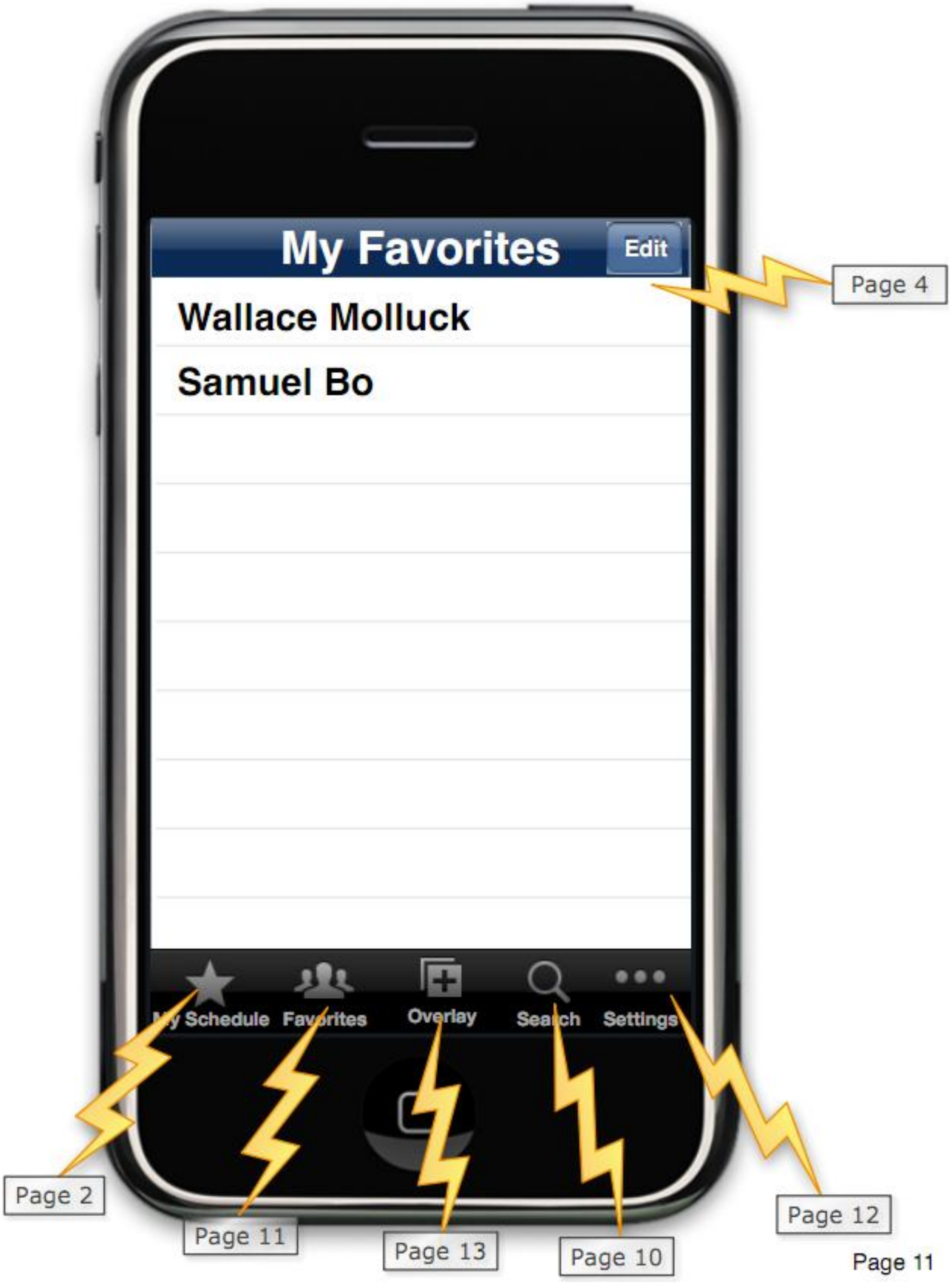
Spring 2009-10

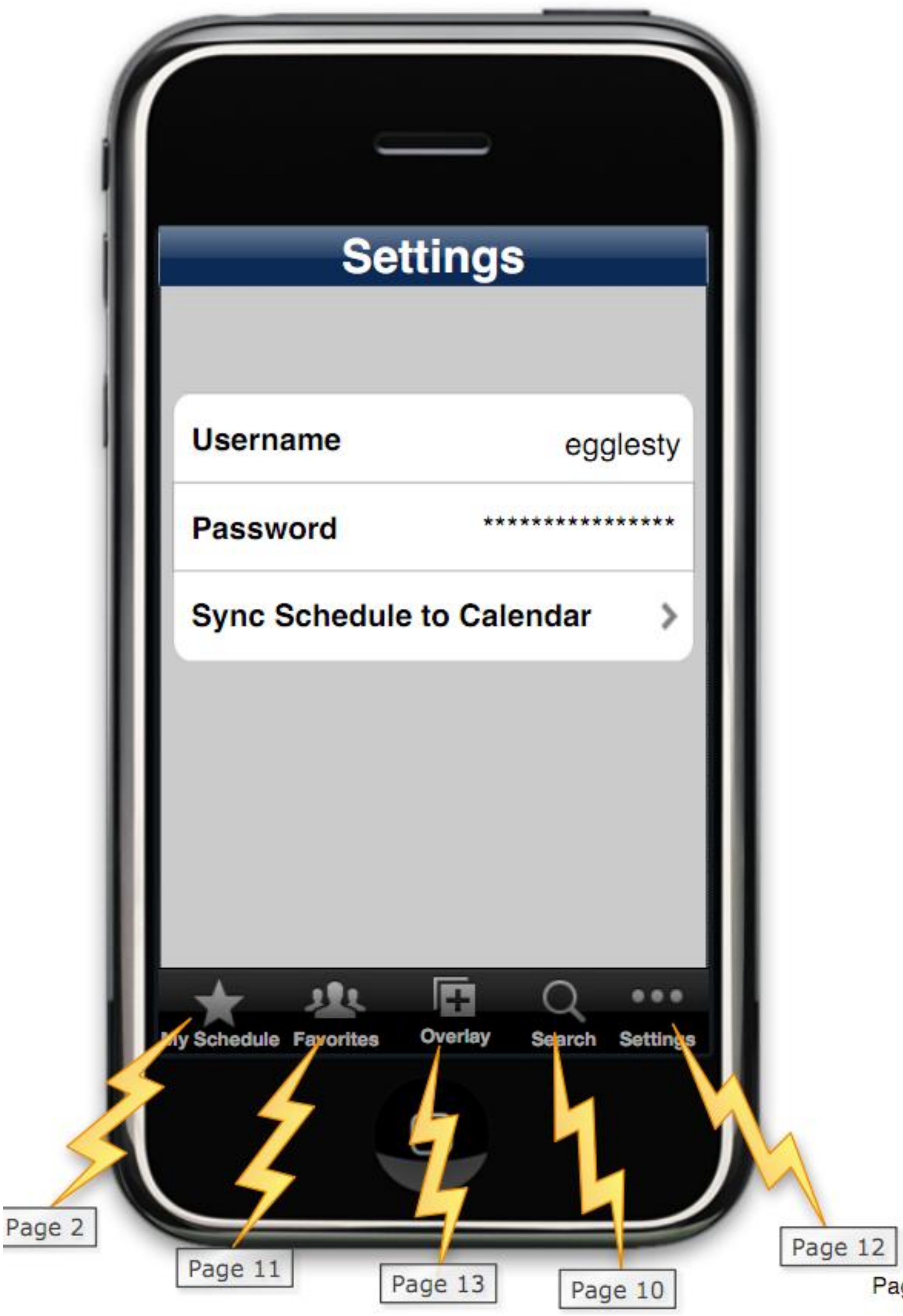
Winter 2009-10

Fall 2009-10

Page 9


Search
Search
Username
Room Number
Course ID
Search
My Schedule
Favorites
Overlay
Search
Settings
Page 2
Page 3
Page 6
Page 2
Page 11
Page 13
Page 10
Page 12
Page 10

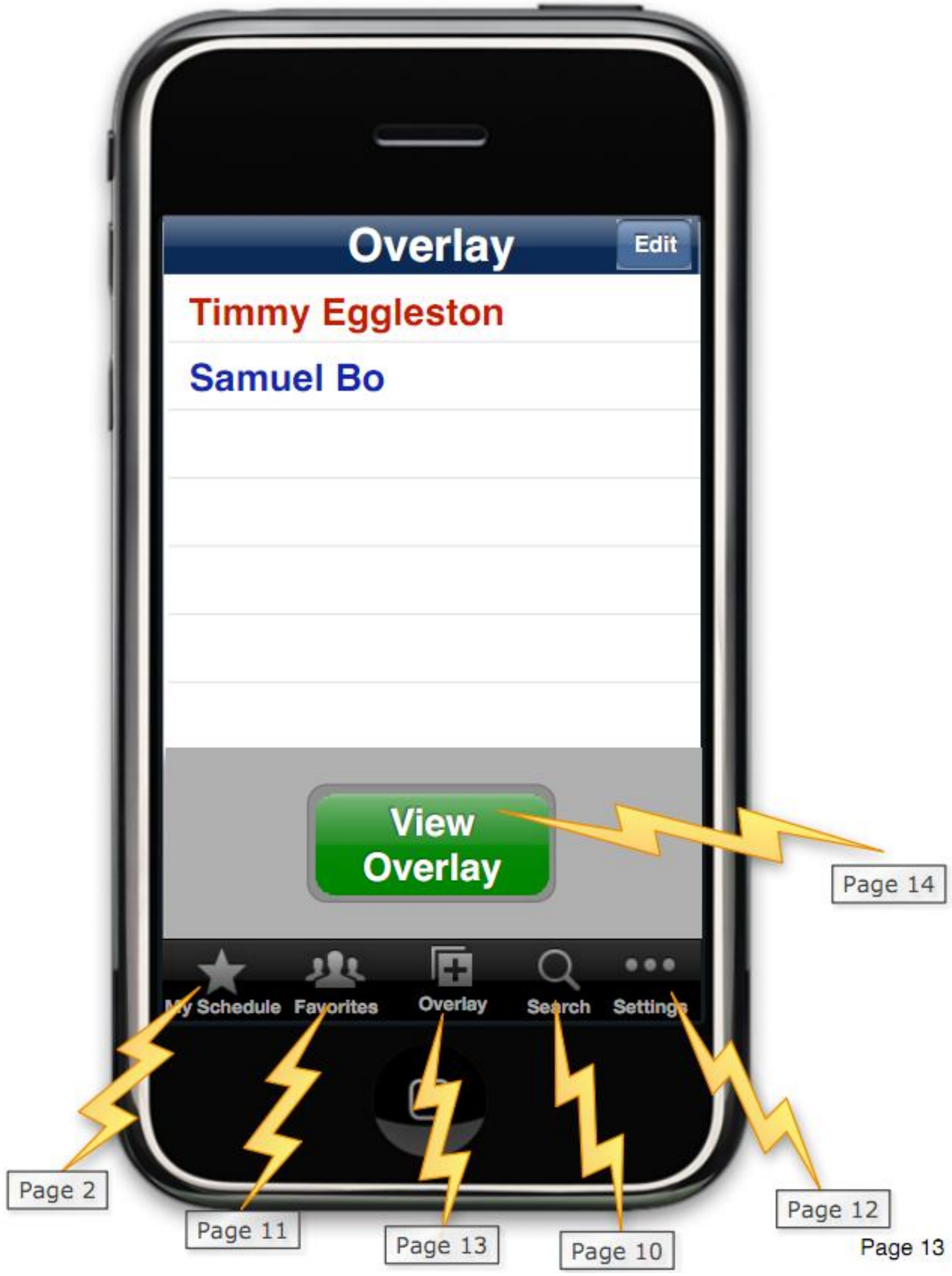

My Favorites
Edit
Wallace Molluck
Samuel Bo
My Schedule
Favorites
Overlay
Search
Settings
Page 4
Page 2
Page 11
Page 13
Page 10
Page 12
Page 11

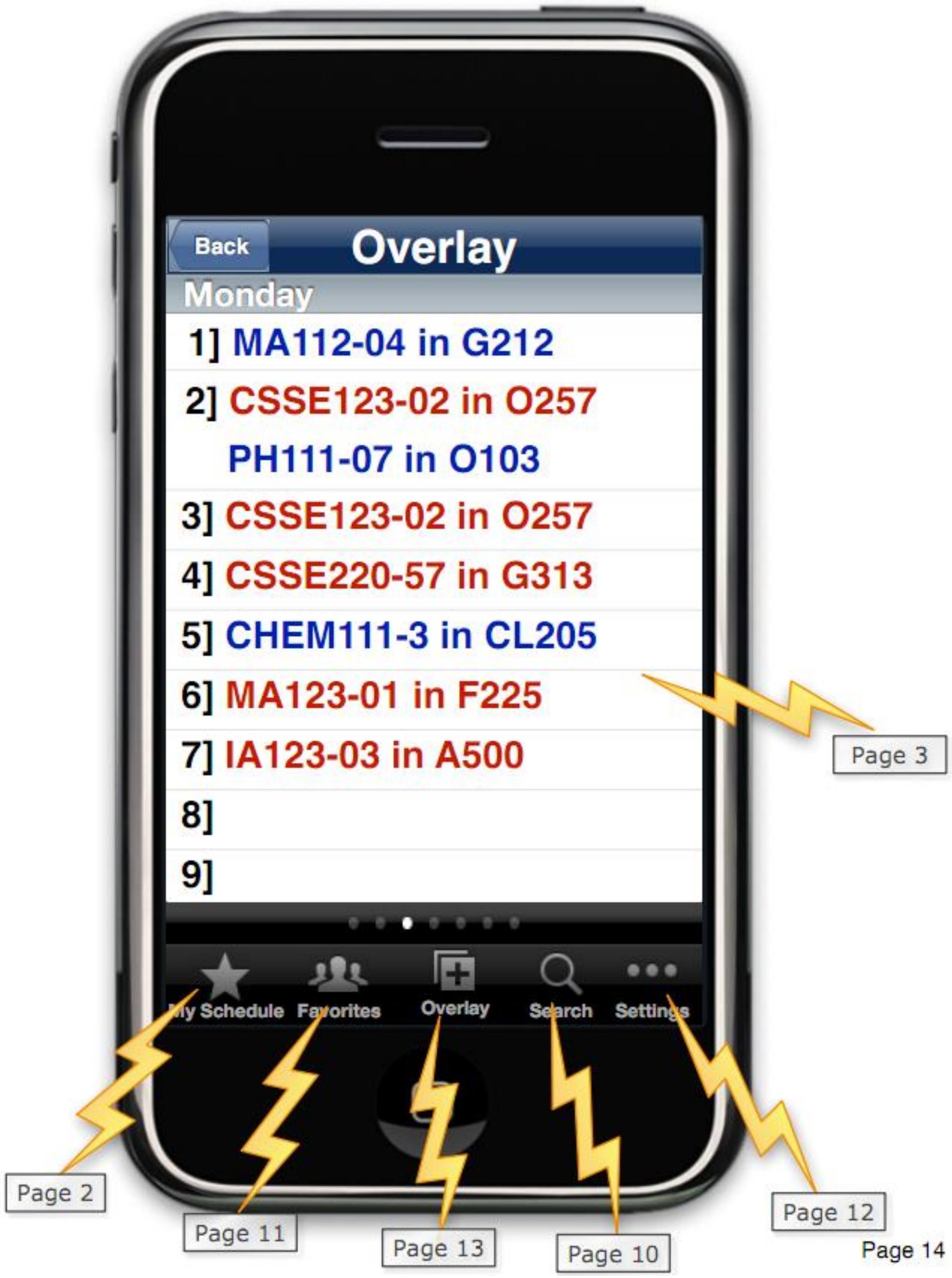


Settings

Username	egglesty
Password	****************
Sync Schedule to Calendar	

Page 12


Overlay
Edit
Timmy Eggleston
Samuel Bo
View Overlay
Page 14
My Schedule
Favorites
Overlay
Search
Settings
Page 2
Page 11
Page 13
Page 10
Page 12
Page 13


Back

Overlay

Monday

1] MA112-04 in G212

2] CSSE123-02 in O257

PH111-07 in O103

3] CSSE123-02 in O257

4] CSSE220-57 in G313

5] CHEM111-3 in CL205

6] MA123-01 in F225

7] IA123-03 in A500

8]

9]

Initial Interface Design

The initial interface design was submitted via e-mail as it is an extremely large vector PDF and the interactive functionality is not present in paper form.

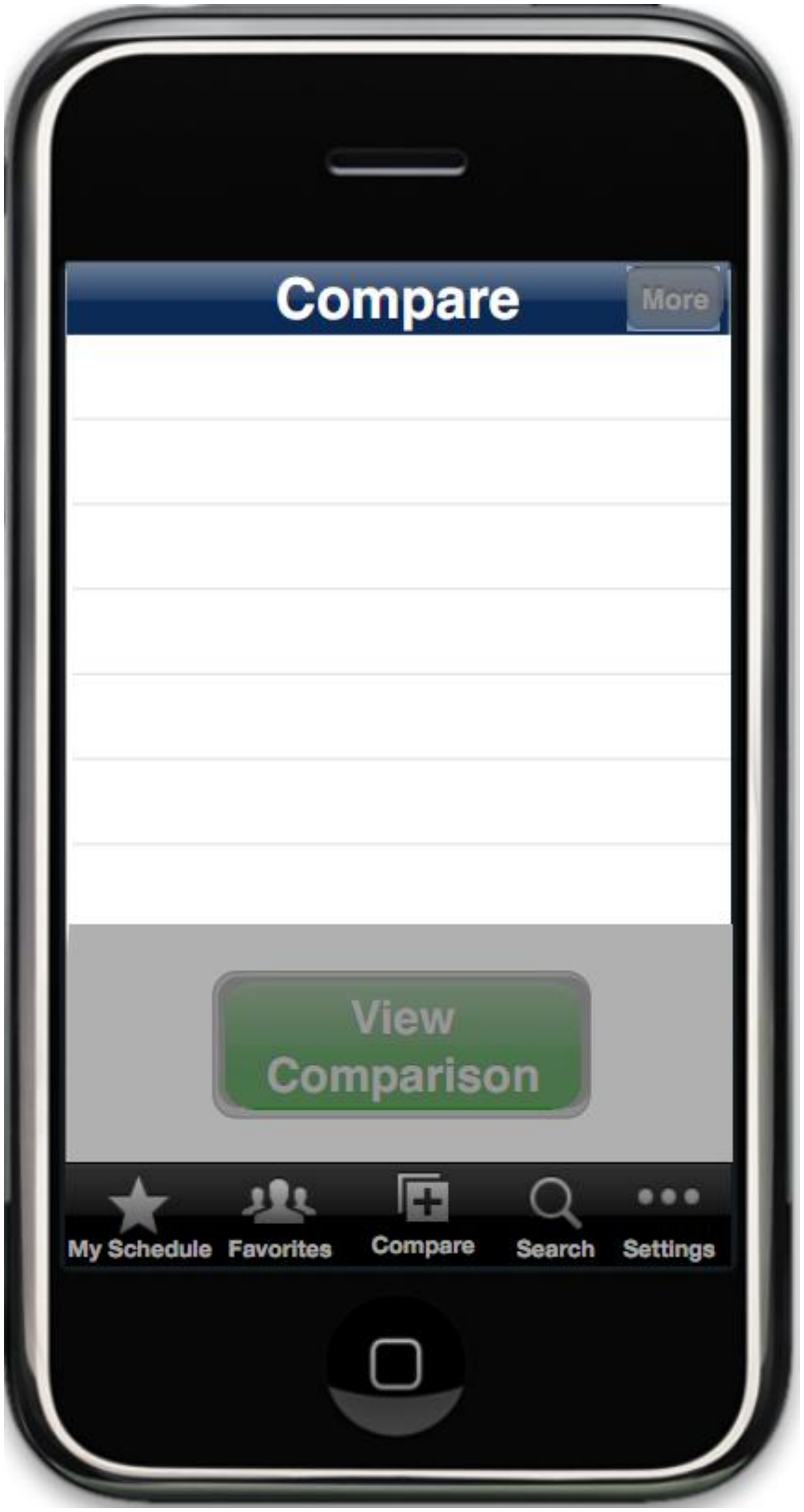
Revised Interface Design

As a result of the usability study, a few changes were made to accommodate for common issues faced by users. First, the term “Overlay” was confusing for many users who didn’t know what functionality to expect from that feature. Based on the responses of several usability study participants, the term was changed to “Compare.”

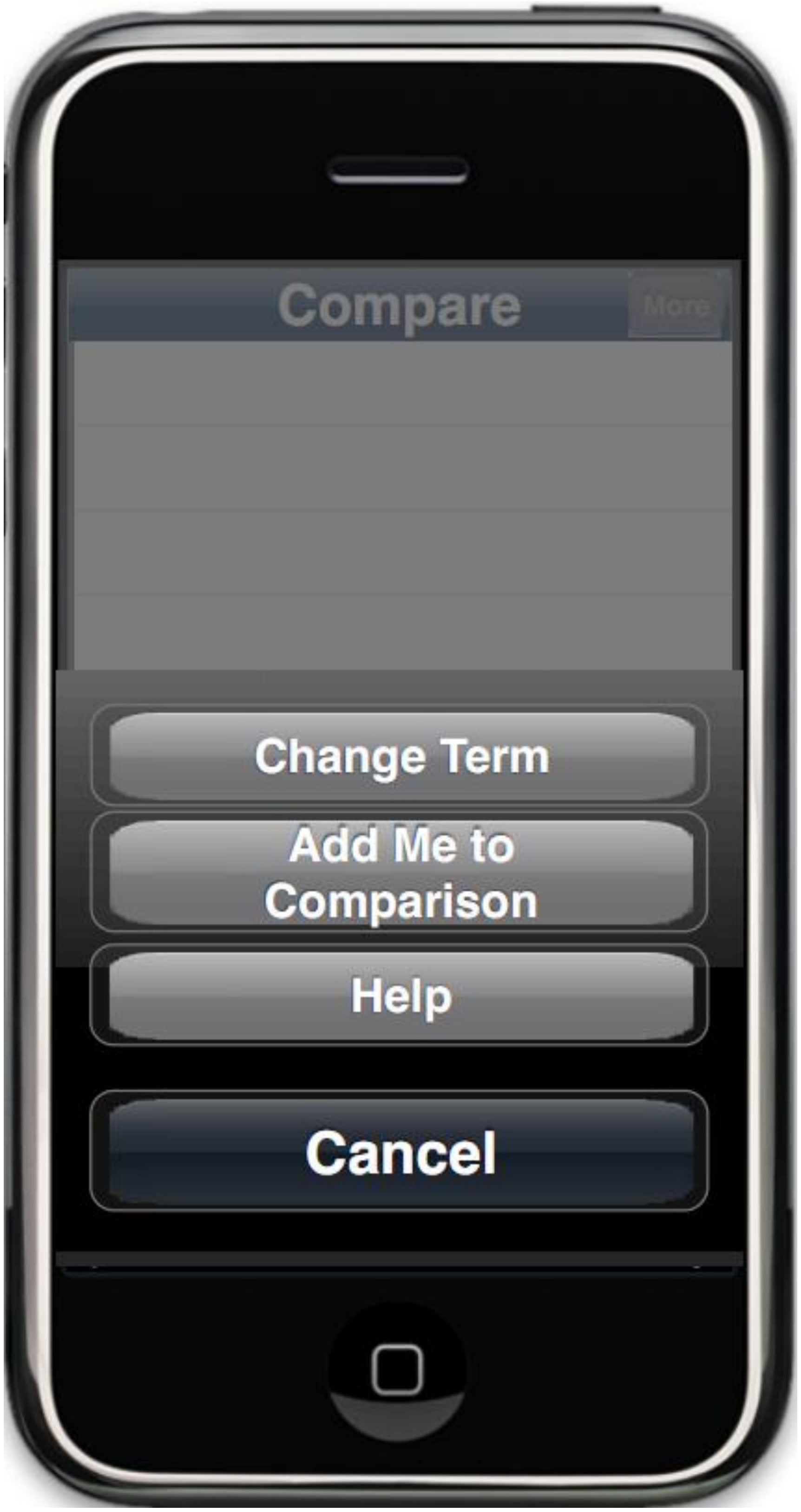
Secondly, in iOS apps, the expected functionality when a user selects the ‘+’ button is for something new to be created, not for an options menu to appear. To fit in more with the operating system, the ‘+’ button has been replaced with a “More” button to indicate that pressing that button may reveal more functionality.

Due to the issues with the previous overlay page, as few users actually discovered what it could do, the “Edit” button was replaced with a “More” button. Deleting items from the list is now performed with the “swipe to delete” gesture. On the “More” menu, the new options are to add yourself to the comparison, view the comparison for a term other than the current term, and view a help page that better describes the functionality of the comparison feature.

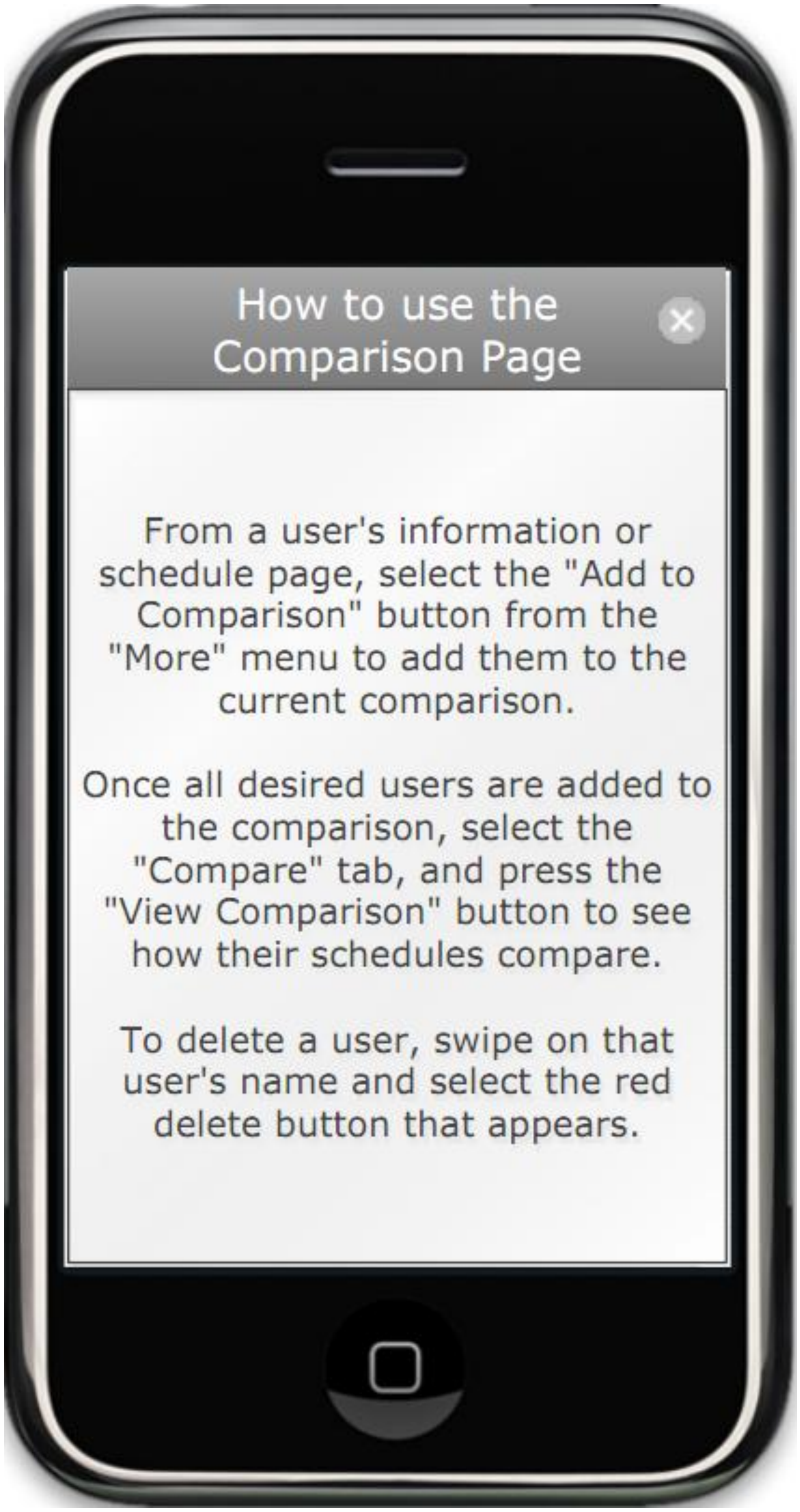
Finally, a few buttons were left active even when there was no effect due to the current state of the application. To help prevent users from being confused, all buttons are now grayed out when not active.



This screen indicates that the overlay page has been changed to a Compare page. The tab has been updated accordingly. As neither button will do anything in an empty comparison, both buttons are grayed out.



The new "More" page for the comparison page. This includes adding yourself to the comparison, viewing a help page, and changing the term for the current comparison.



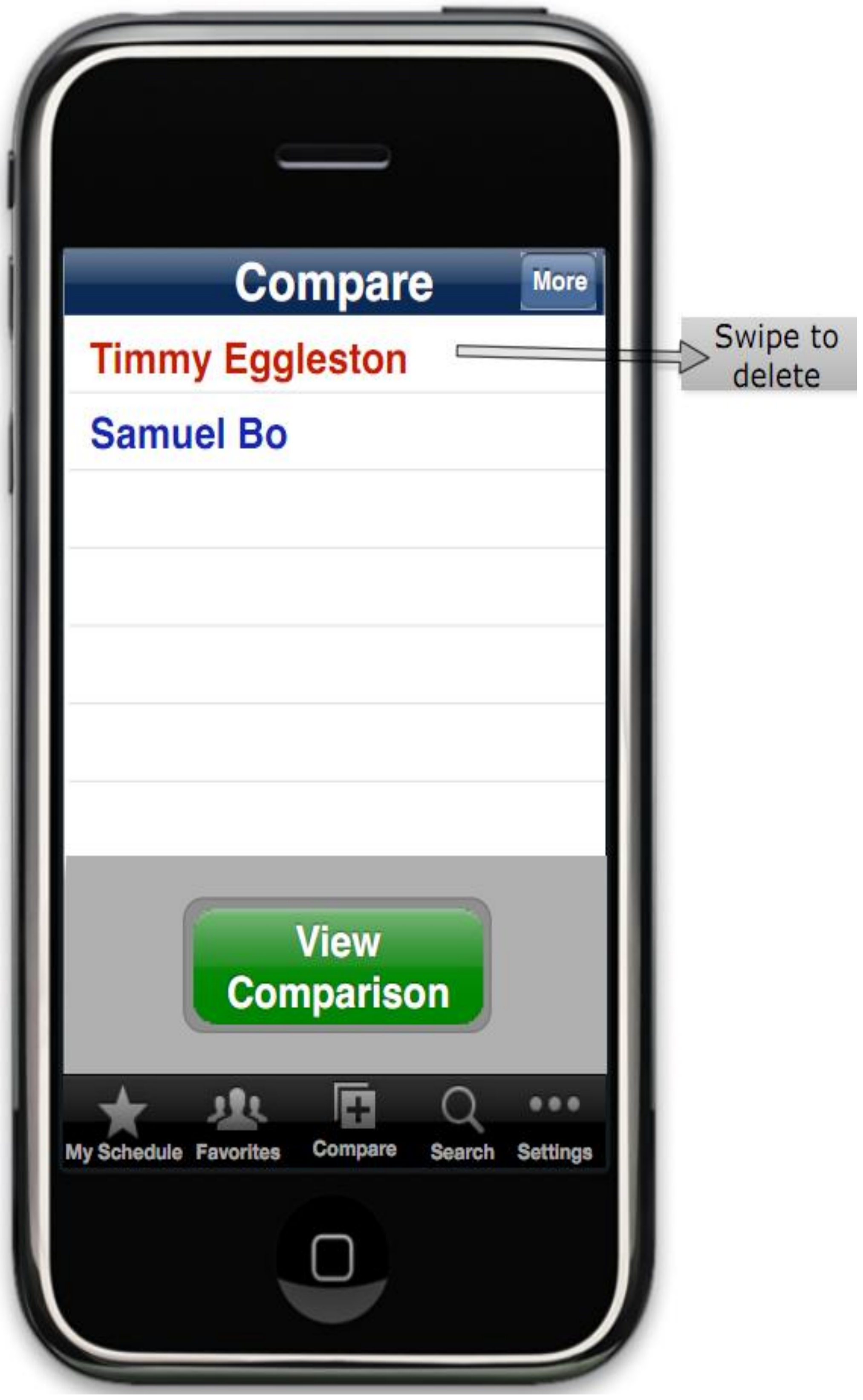
How to use the Comparison Page

From a user's information or schedule page, select the "Add to Comparison" button from the "More" menu to add them to the current comparison.

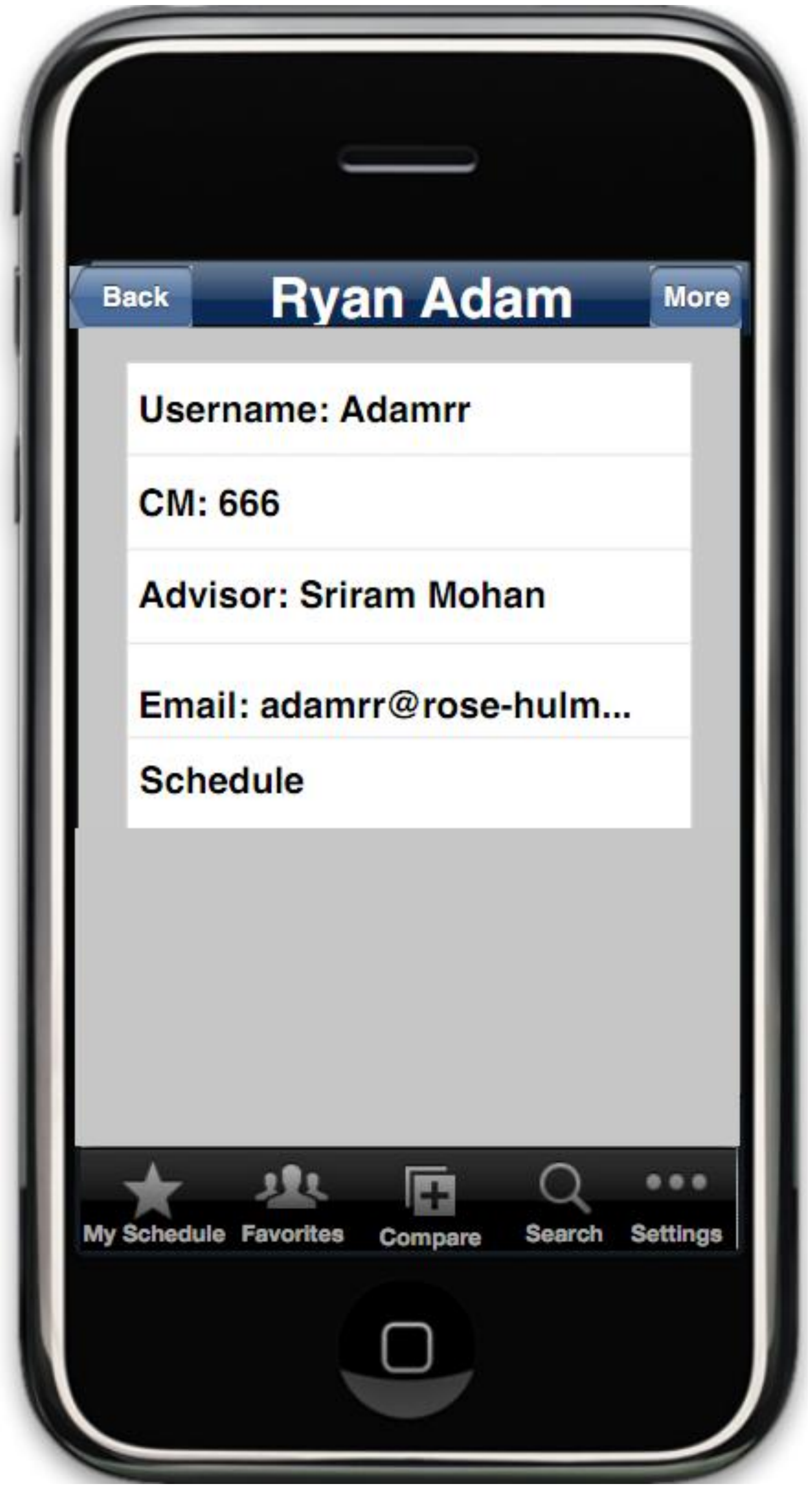
Once all desired users are added to the comparison, select the "Compare" tab, and press the "View Comparison" button to see how their schedules compare.

To delete a user, swipe on that user's name and select the red delete button that appears.

The help page for the comparison page.



Users are now removed using “swipe to delete.” Notice that the buttons are no longer grayed out due to the active users in the comparison.



Username: Adamrr

CM: 666

Advisor: Sriram Mohan

Email: adamrr@rose-hulm...

Schedule

The new user info page has "More" instead of "+" in the top right corner. Compare has replaced Overlay in the tab bar.

Index

Index

Glossary

iOS	The mobile operating system developed by Apple that runs on iPhone, iPod Touch, and iPad devices
Rose-Hulman	The number one undergraduate engineering school in the nation, and the school at which the schedule lookup app will be used
Usability Study	A study to determine ease of use of a software application.
Git	A free and open source distributed version control system

Bibliography

Ekl, T., & Stokes, E. (2011, 9 21). Client Meeting. (M. Vitale, B. Knight, K. Greenwald, & A. Say, Interviewers)

iOS Developer Library Human Interface Guidelines. (2011). Retrieved from Apple: Apple Inc. (2011) Human Interface Guidelines https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html

iOS Developer Library iOS Programming Guide. (2011). Retrieved from Apple Inc.: https://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007072

(2011). iPhone Mobile Lookup Survey. (K. Greenwald, Interviewer)

Yourdon, D. L. (2003). *Managing Software Requirements, 2nd edition, A Use Case Approach.* Addison Wesley: Pearson Education.

Who Done It: Team Member Names	Section/Part Completed	Task/Comments	# of hours effort
Mark Vitale	Milestone 1	Combined all documents into one, removed duplicate/ unnecessary items. Led team meeting about iOS development.	3 hours
Brandon Knight	Milestone 1, Engineering Journal	Looked over combined document, added Index, References, updated Table of Contents. Wrote up sections in Engineering Journal. Met with team.	2 hours
Ann Say	1.2	Created Engineering Journal Met with team.	1 hours
Chris Gropp	Document	Met with team.	1 hours
…	…	…	…