

# Mobile Student Lookup

## Milestone 3 Hand-In Document

10/18/2011

Brandon Knight, Mark Vitale, Ann Say, Katie Greenwald

## Contents

Executive Summary .....	3
Introduction .....	3
Feature Listing .....	4
Use Cases .....	8
Global Pre Conditions .....	8
Global Alternate Flow .....	8
Mobile Student Lookup Use Case: Call a User .....	9
Project Background and Functionality Not In Use Cases .....	10
Usability Requirements .....	10
Performance Requirements .....	10
Reliability Requirements .....	10
Supportability Requirements .....	11
Hardware and Software Interfaces .....	11
Documentation, Installation, Legal and Licensing Requirements .....	11
Design Constraints .....	11
User Interfaces .....	11
Index .....	12
Glossary .....	13
References .....	14

## Executive Summary

In the progress of the Mobile Student Lookup Application, we have designed and created a User Interface mock-up than a potential user or client can play around with. The User Interface we created covers all of the Use Cases previously defined in Milestone 2 and the Features described in Milestone 1. We also included multiple requirements for the application, including Usability Requirements, Performance Requirements, Supportability Requirements, and Reliability Requirements. These requirements are to inform the team and the client of the requirements we must abide by for the rest of the development of the application. We cross checked these with the client, and received valuable feedback with our other documentation sections. These sections include Hardware and Software interfaces, Design Constraints, Installation, and Legal Requirements.

## Introduction

From the Use Case and Feature Mapping in Milestone 2, we have created a user interface prototype in a Vector PDF which will be used to demonstrate to our clients the product that we have envisioned. Also, the requirements mentioned in the Executive Summary are a very important part of this document as they will help us with the rest of the documents and the development of code later in the project. In addition, we added a feature and a use case which involve a user clicking on a phone number; it will call the number. We thought it would be best to add it rather than have it slip through without formally acknowledging it.

## 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. The only change in the feature listing from the previous milestone document is feature 14.

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

## Use Cases

To reflect the potential uses of the new feature added in the previous section, a single use case has been added.

### 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.



## 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.

## Project Background and Functionality Not In Use Cases

The mobile student lookup application will be a native application for iOS devices (Apple Inc, 2011). 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 (Schedule Lookup, 2011). 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.

Jimmy Theis 10/15/11 12:28 AM

**Comment [1]:** I like that this one is conservative. Way to be realistic.

## 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.

Jimmy Theis 10/15/11 12:29 AM

**Comment [2]:** You might double check that this is something you can control. If the system can be in a state where you get memory warnings no matter what you do, don't promise it.

Jimmy Theis 10/17/11 3:25 PM

**Comment [3]:** You might want to be careful with your definitions of bugs here. A misspelling of a word can be a bug that manifests in a critical workflow, but doesn't actually affect anything. Is that a critical bug?

Jimmy Theis 10/17/11 3:25 PM

**Comment [4]:** Not to nit-pick, but it seems like you can always satisfy this requirement by filing a bunch of bogus minor bugs. Maybe rethink or add some stipulations to make this a little more well-defined.

Jimmy Theis 10/17/11 3:25 PM

**Comment [5]:** I guess this satisfies my above comment, but I still think it could use a little clarity. Maybe I'm just being picky. Probably.

## 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.

Jimmy Theis 10/15/11 12:41 AM

**Comment [6]:** You should probably at least mention what the long-term support situation looks like, even if you're not going to be the ones who have to do the support work.

## 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. (Apple Inc, 2011)

## 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 via e-mail.

## Index

Git, 11

Human Interface Guidelines, 10

iOS, 11

Rose-Hulman, 8, 13

server, 8

## Glossary

Git	A free and open source distributed version control system
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
Use Case	A description of the interactions between a user and the system that leads to a useful outcome for the user

## References

- 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>
- (2011). iPhone Mobile Lookup Survey. (K. Greenwald, Interviewer)
- Schedule Lookup*. (2011, 4). Retrieved 10 18, 2011, from [www.rose-hulman.edu](http://www.rose-hulman.edu): <https://prodweb.rose-hulman.edu/regweb-cgi/reg-sched.pl>
- Apple Inc. (2011). *iOS 5 for Developers*. Retrieved October 11, 2011, from [www.apple.com](http://www.apple.com):  
<http://developer.apple.com/technologies/ios5/>
- Ekl, T., & Stokes, E. (2011, 9 21). Client Meeting. (M. Vitale, B. Knight, K. Greenwald, & A. Say, Interviewers)
- Yourdon, D. L. (2003). *Managing Software Requirements, 2nd edition, A Use Case Approach*. Addison Wesley: Pearson Education.