iLETU

Software Requirements Specification

Version 1

September 2, 2013

Prepared for

COSC 4853 - Software Engineering II

LeTourneau University

2100 S Mobberly Ave.

Longview, TX 75607

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description of Changes |
| 9/2/2013 | 1.0 | Brady Steed | First Version |
|  |  |  |  |
|  |  |  |  |

# Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| Signature | Name | Title | Date |
|  | Eric Papke | Librarian and Scribe |  |
|  | Terry Penner | Lead Programmer |  |
|  | Ethan Sikes | Project Leader |  |
|  | Brady Steed | Lead Designer |  |
|  | Remington Wells | Quality Assurance Manager |  |

# Table of Contents

# 1 Scope

## 1.1 Identification

This is the Software Requirements Specification document for iLETU. The software is already published by another author, but it is no longer being developed. The existing software is being used and modified under license and with the consent of the original author. This document contains the necessary information and requirements for the planning, development and testing of the iPhone app, iLETU.

## 1.2 System Overview

The new version of the app keeps the same basic functions that the original app has, but it is designed with more current industry standards. The software is designed to make accessing campus information and resources simple and efficient; it will do this by automatically logging the user in to secure areas of the LeTourneau website and providing public information in convenient formats.

This is an event oriented application, so at any time, the user will be able to change between views. The views are listed below in 3.1 Required States and Modes

## 1.3 Document Overview

This document contains the necessary information and requirements for the planning, development and testing of the iLETU iPhone app. Section 2 contains any directly or indirectly referenced document. Section 3 contains requirements specifications. Section 4 lists testing methods. Section 5 contains the testing methods for the requirements.

# 2 Referenced Documents

MIL-STD 498 SRS-DID. Dr. Tevis. December 05, 1994

iOS Developer Library. https://developer.apple.com/library/ios/navigation/. September 3, 2013.

# 3 Requirements

## 3.1 Required States and Modes

The software shall have several views:

* Auto-login view
* Announcement view
* Calendar view
* Yellowjacket view
* Campus Bullet view
* Intramural view
* Menu view
* Campus Hours view
* Schedule view

These views shall be event oriented and change in real time when the user selects a new view.

## 3.2 CSCI Capability Requirements

### 3.2.1 Auto-login and user specific data.

The software shall securely store the user’s Letnet credentials. The software shall then allow the user to automatically login to and view the following online resources (also see 3.3 CSCI external interface requirements):

* School email
* Blackboard account
* Web Services
* The student directory
* Chapel attendance
* Daily class schedule

### 3.2.2 Common LeTourneau resources.

The software shall download and display the following information from LeTourneau’s server (or store the data if it never changes):

* Recent Announcements
* Calendars including the academic, YAC, and athletic calendars
* The menu at the Corner Café
* Hours of the Airport bus and the pool.
* The Campus Bullet
* The Yellowjacket (school newspaper)
* Intramurals website

See 3.3 CSCI external interface requirements for more detailed information

The software shall store the Corner Café, The Hive, Common Grounds, the library, the computer labs, Solheim, and the Mail Center.

### 3.2.3 Calendars.

The software shall display all calendars in a monthly calendar format. The weekly class schedule shall be displayed as a 7 day calendar, but implementation shall be left to design.

## 3.3 CSCI external interface requirements.

The software shall interface with the following websites using TCP with no authentication:

* Bus schedule - http://www.letu.edu/opencms/opencms/homepage-links/student-resources/abbottbus.lnk
* Pool hours - http://www.letu.edu/\_Other-Resources/solheim-hours.html
* YAC calendar - http://www.google.com/calendar/embed?showTitle=0&showCalendars=0&showTabs=0;&showPrint=0&height=600&wkst=1&bgcolor=%23EEEBDB&src=yac%40letu.edu&color=%231B887A&ctz=America%2FChicago
* Athletics page - http://www.letuathletics.com/index.aspx

The software shall also load the following RSS feeds:

* Announcements - http://letustartpage.blogspot.com/feeds/posts/default?alt=rss
* Campus Bullet - http://campusbullet.net/home/category/all/?feed
* Yellowjacket - http://letuyellowjacket.org/feed/
* Saga menu - http://legacy.cafebonappetit.com/rss/menu/147
* Intramurals schedule - http://www.imleagues.com/School/Portal.aspx?SchID=39d7b8cb08db465ab4f27a061607a7c4&Portal=SchOfDay

The software shall securely login to and display the following pages:

* Mail - https://mail.letu.edu/owa/auth/logon.aspx
* Web services - https://my.letu.edu/ICS/
* Blackboard - https://courses.letu.edu
* Student directory - http://www.letu.edu/search/student\_directory.html
* Chapel attendance - http://www.letu.edu/student\_life/chapel/attendance.html

The implementation of the Class schedule interface shall be left to design.

## 3.4 CSCI internal interface requirements

Internal interfaces shall follow the standards set by Apple.

## 3.5 CSCI internal data requirements

The software shall store the user’s Letnet username and password securely.

## 3.6 Adaptation requirements

The software shall have different versions made to fit the standard iPhone, iPhone 5, and iPad screens.

## 3.7 Safety requirements

N/A

## 3.8 Security and privacy requirements

The software shall secure confidential data at all times. The software must encrypt the data in storage and use a secure connection when transmitting it.

## 3.9 CSCI environment requirements

The software shall run on iOS. The only requirement for full service is an internet connection, but the app will successfully run with limited capabilities without the internet.

## 3.10 Computer resource requirements

## 3.10.1 Computer hardware requirements

The software must be able to run on the hardware of an Apple mobile device.

## 3.10.2 Computer hardware resource utilization requirements

This software shall be implemented and written using the Objective-C language. All code shall be written following K&A-Stroustrup style conventions. The software shall be developed in the Xcode integrated development environment.

## 3.10.3 Computer Software requirements

The software shall require an iOS environment to run.

## 3.10.4 Computer communications requirements

The software shall use the industry standards of data communication and will cache information that does not change often.

## 3.11 Software quality factors

The software will be able to perform all functions without fail or crashing, in a timely manner. The source code of the software will be properly commented such that all functions are explained clearly. New versions of the source code will have clear explanations to what has been improved on.

## 3.12 Design and implementation constraints

This software shall be implemented and written using the Objective-C language. All code shall be written following K&A-Stroustrup style conventions. The software shall be developed in the Xcode integrated development environment.

## 3.13 Personnel-related requirements

The software shall be designed for one user at a time.

## 3.14 Training-related requirements

N/A

## 3.15 Logistics-related requirements

N/A

## 3.16 Other requirements

All other requirements shall be left to design.

## 3.17 Packaging requirements

All packaging requirements shall be left to design.

## 3.18 Precedence and criticality of requirements

3.8 Security and privacy is the most important requirement. After that, all of the features (3.2 CSCI Capability Requirements) are the most important, followed by implementation constraints.

# Qualification provision

1. Demonstration: The operation of the CSCI, or a part of the CSCI, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
2. Test: The software or specific part of the software shall be tested according to a prewritten test specific to the portion of software being tested. The prewritten test shall consist of expected results and exemplary results. The expected results are those that are vital to the software’s functionality and most critical to the current stage of development. Exemplary results are those that are not vital or critical to the current stage of development and that can be improved later.
3. Analysis: After testing, the results of testing shall be compared to expected results. If test results do not match expected results, the tested portion of software is deemed to be in failure. If test results match exemplary results but not expected results, the tested portion of software is deemed to be in failure. If test results match expected results but not exemplary results, the tested portion of software is deemed to be satisfactory. If test results match both expected and exemplary results, the tested portion of software is deemed to be in full operation.
4. Inspection: The source code of the tested portion of software will be observed for proper functionality.
5. Special qualification methods: Any special qualification methods for the software, such as special tools, techniques, procedures, facilities, and acceptance limits.

|  |  |  |
| --- | --- | --- |
| Requirement | Qualification Methods | Comments |
| 3.1 | Demonstration |  |
| 3.2.1 | Demonstration |  |
| 3.2.2 | Demonstration |  |
| 3.2.3 | Demonstration |  |
| 3.3 | Demonstration, Test |  |
| 3.4 | Inspection |  |
| 3.5 | Test |  |
| 3.6 | Demonstration |  |
| 3.8 | Test |  |
| 3.9 | Demonstration |  |
| 3.10.1 | Demonstration |  |
| 3.10.2 | Inspection |  |
| 3.10.3 | Demonstration |  |
| 3.10.4 | Test, Inspection |  |
| 3.11 | Test |  |
| 3.12 | Inspection |  |
| 3.13 | Demonstration |  |

# 5 Requirements traceability

TBD

# 6 Notes

TBD