

Always UC Mobile

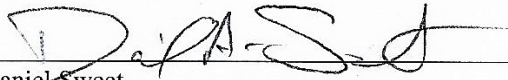

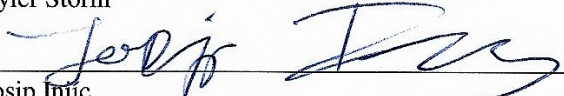


By

Dan Sweet, Josip Injić, Tyler Storm, Kenneth Keith

Submitted to CECH Faculty
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright Always Education

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

 Daniel Sweet	<u>4/17/17</u> Date
 Tyler Storm	<u>4/17/17</u> Date
 Josip Injić	<u>4/17/17</u> Date
 Kenneth Keith	<u>4/17/17</u> Date
 Russell McMahon	<u>4/17/2017</u> Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2017

Always UC Mobile



Prepared for:

Always Education

University of Cincinnati

Prepared by:

Dan Sweet, Josip Injić, Tyler Storm, and Kenneth Keith

Students of University of Cincinnati

College of Education, Criminal Justice & Human Resources

School of Information Technology

<http://cech.uc.edu/it.html>

19 April 2017

Table of Contents

Abstract	i
1. Introduction.....	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Project Description	2
1.4 User Profile.....	3
1.4.1 Use Case Diagram	6
2. Technical Components.....	7
2.1 Technical Elements	7
2.1.1 App Layout & Design	7
2.1.2 iOS.....	8
2.1.3 Android	9
2.1.4 Middleware.....	9
2.2 Testing.....	10
2.2.1 Testing Overview	10
2.2.2 Scope	11
2.2.3 Logging & Reporting	12
2.2.4 API Documentation and Test Bench	13
3. Project Management	14
3.1 Objective	14
3.2 Project Budget	14
3.3 Schedule.....	15
3.4 Deliverables.....	16
4. Conclusion	17
Bibliography	18
Appendix.....	24

Table of Figures

Illustrations

Figure 1: User Profile.....	3
Figure 2: Use Case Diagram.....	6

Figure 3: Mobile Application Design	8
Figure 4: Gantt Chart	16
Figure 5. Test Case Documentation	24
Tables	
Table 1: Project Budget	14

Abstract

Always UC Mobile is a native iOS and Android application for the Always UC platform. Always UC is the system that replaces UC's current integration of Blackboard as an all-encompassing digital education platform. In addition to serving as a platform for delivering, receiving, grading, and tracking course assignments, the system allows students and faculty to search and connect with students and associates based on interest, skills, and associations to schools, colleges, classes, and clubs within the university. Always UC creates a 'news feed' that displays information about things happening on campus, upcoming assignments, announcements, and more based on the user's role within the university. The mobile application is developed for the latest versions of iOS and Android, sporting modern material design UI components that are friendly and intuitive, engaging users and encouraging more involvement in the college experience by connecting them in a digital social education experience unrivaled by competitors.

1. Introduction

1.1 Introduction

Always Education as a company needs a mobile implementation of their Always UC project, which exists currently as a web application. Always UC is a University of Cincinnati specific implementation of the Always Education software, which joins a series of disconnected data silos, organizing and aggregating the data relevant to each university associate into a single logical layer and interface. From this web application, users have immediate access to all information relevant to their role within the organization. Students and faculty alike have the ability to create a network based on the courses, clubs, or ties to the university, as well as the capability of searching the entire university for any user connected to those accrued interests and experiences.

1.2 Problem Statement

Always UC is a digital education delivery platform for universities that replaces the University's current implementation of the Blackboard software, enhancing user capability to connect with peers and colleagues based on interests, skills,

experience, and associations to the university. The university's current implementation of Blackboard presents a problem in the inherent lack of direct student interaction due to the lackluster functionality of the Blackboard mobile client. Additionally, the Always UC project lacks a native mobile application for iOS and Android, creating a void in which students and faculty lack access their course information and relevant context for the courses currently being taught using the Always UC platform.

1.3 Project Description

Always UC Mobile is a mobile application that implements the Always UC platform into a native mobile app for iOS and Android. The mobile application has an enhanced, user friendly interface designed with the latest Swift (iOS) and Android material design standards to create a simple and intuitive interface for users. The application offers a multitude of features, such as push notifications to alert students about campus events or overdue homework assignments, as well as the ability to navigate to course information, users within their network, a dashboard that displays a news feed relevant to that user, and role-specific information.

1.4 User Profile

The figure (1) below provides a detailed illustration of the team's assessment of the user profile for the Always UC Mobile applications.

PROJECT: Always UC Mobile
POTENTIAL USERS: <ul style="list-style-type: none">- UC Students- Adjunct/FTE Professors- University Staff/Officers
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This application will be developed in Swift 3.0 using XCode as the compiler and emulator to develop in and delivering an app that is to be used on any up to date version of the iPhone or iPad. The current version of iOS Always UC Mobile will run on is iOS 10.0.2. It will be compatible with the last version of iOS 9. The interface will have an

“easy to navigate” feel in order to engage the user to access their assignments, their individual profile, and news feed.

EXPERIENCE WITH SIMILAR APPLICATIONS:

Outside of the developers the clients (student and faculty) will have not seen or used Always UC or its mobile application. With that in mind, the application is being tailored for ease of use and capability to communicate with other users very similar to Blackboard mobile.

TASK EXPERIENCE:

Not all business owners will have mobile application development experience

FREQUENCY OF USE:

The goal is to stand the application up for daily use of all student and faculty members. This is to allow for constant communication between all University affiliates. This will keep all users up to date with what is happening on campus as well as any time sensitive appointments or assignments.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Adaptation for business owners and business continuity personnel
- Technical information and demonstration of disaster recovery solution.
- Overview of requirement and risk assessment
- Engaging presentation
- Visually and aesthetically pleasing designs
- Images that inform and enhance the presentation

Figure 1: User Profile

1.4.1 Use Case Diagram

As suggested by the user profile, the use case diagram below provides a visual representation of the typical use case for the program.

Use Case Diagram: AlwaysUC Mobile iOS Application

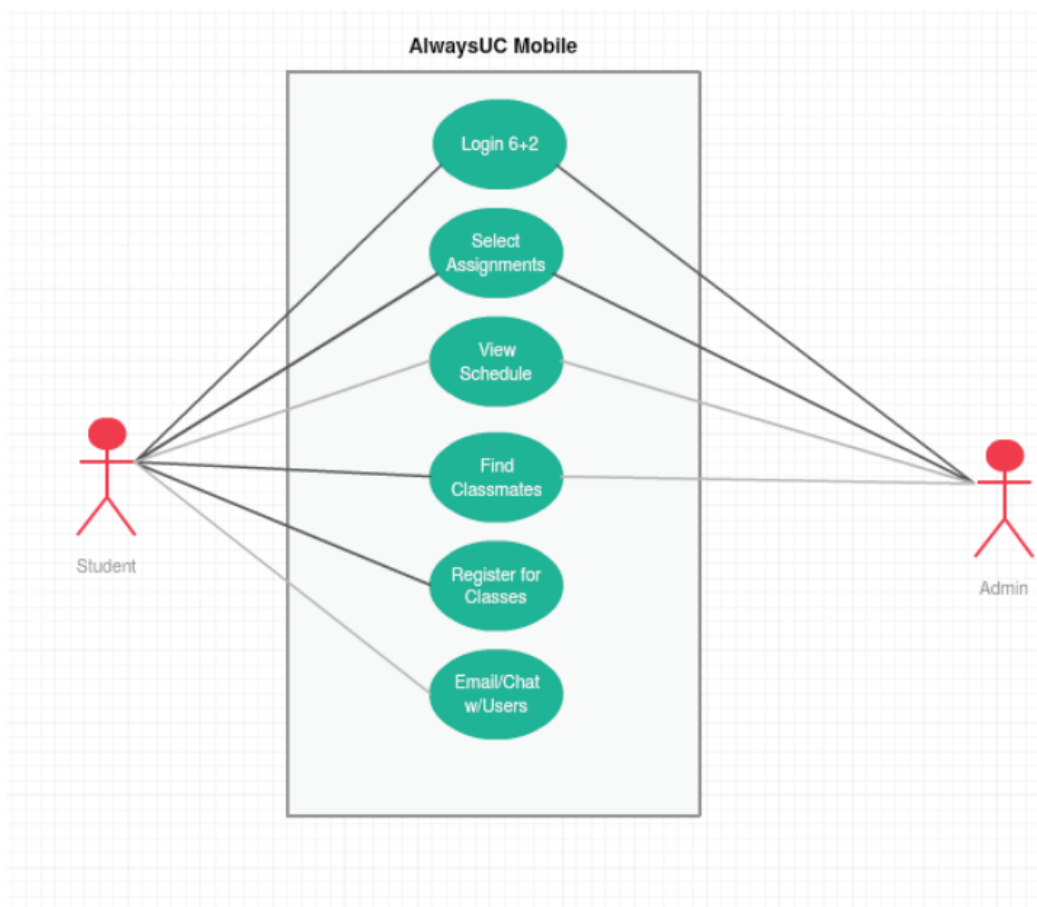


Figure 2: Use Case Diagram

2. Technical Components

2.1 Technical Elements

The technical elements of the software are comprised of three separate categories of custom software, the iOS application, the Android application, and the middleware server. Each of these applications consist primarily of the proprietary closed-source custom code written by the Always UC Mobile team, as well as implementing several open-source technologies.

2.1.1 App Layout & Design

The figure (3) below contains an illustrated design provided by the Always Education team. This design details the expectations of the Always Education team for a mobile application that conforms to their design and branding standards.

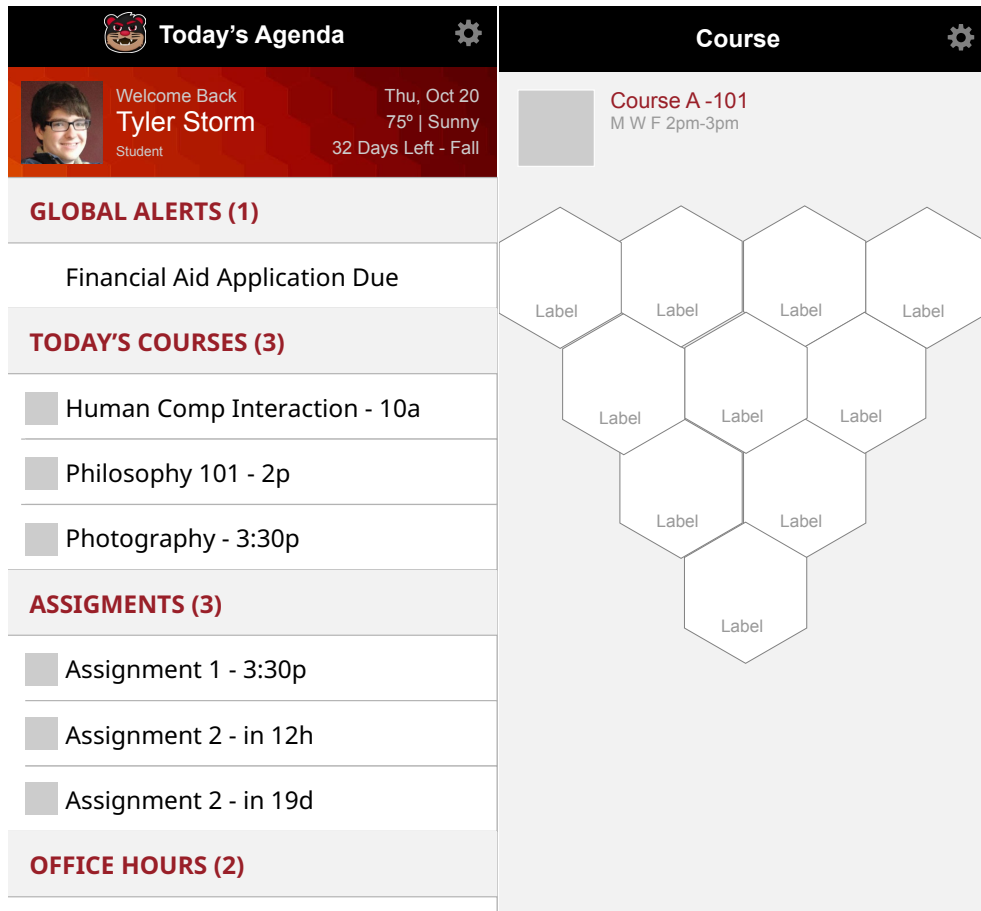


Figure 3: Mobile Application Design.

2.1.2 iOS

The iOS application is developed in Swift 3, Apple's latest custom coding technology used for Apple's iOS, macOS, tvOS, and watchOS operating systems. Additionally, the iOS application implements the Appboy SDK for push notifications, as well as the following open-source technologies:

-
- **Kingfisher**
 - **SSHoneyCombView**
 - **PINCache**
 - **TSMardown Parser**
 - **SDWebImage**
 - **Whisper**

2.1.3 Android

The Android application is developed in IntelliJ's Android Studio, targeting a minimum Android OS version of API 24 targeting a total Android market share of over 81.4% for maximum compatibility. In addition to standard native Android controls written in Java and XML, the Android application implements the Appboy SDK for push notifications, as well as the following open-source technologies:

- **MaterialDrawer**
- **Picasso**
- **New Relic Agent**
- **ExpandableTextView**
- **ButterKnife View Binding**
- **MarkdownJ Parser**
- **Joda-Time**

2.1.4 Middleware

The middleware server exists to provide standardized database calls through a RESTful webservice API developed in Node.JS. This middleware server provides both iOS and Android with a standard method of retrieving and storing information in the AlwaysUC databases, as well as handling client authentication

through UC's Shibboleth login system and Redis session storage. This node.js

backend also implements the following open-source technologies:

- **Asyncawait**
- **Body-parser**
- **Compression**
- **Connect-redis**
- **Marked**
- **Katex**
- **Highlight.js**
- **Cookie-parser**
- **Crypto**
- **Ejs**
- **Expres**
- **Gulp-api doc**
- **Gulp**
- **Express -partials**
- **Sanitize-html**
- **Express-session**
- **Moment**
- **Request**

2.2 Testing

2.2.1 Testing Overview

The strategy employed during the development of Always UC Mobile has been the test driven development (TDD) methodology. This development methodology ensures that pivotal functions have significant if not complete test coverage throughout every stage of development. At the end of the first semester, a beta release of Always UC Mobile was published and made available for all student

users with the IT Senior Design class for ADHOC testing as the project progresses.

Additionally, our partners at Always Education have teamed us up with an “iOS Mentor” and “Android Mentor” who regularly meet with the development team for code reviews. Additionally, Always Education has provided the team with a dedicated beta tester, an individual with no knowledge of programming who performs user acceptance testing, providing us with feedback on ease of use. This ensures that the application is user friendly, which is key to the project’s success as the target audience for the product consists primarily of non-technical users.

2.2.2 Scope

The intention of the extensive testing covered in the overview is to ensure seamless deployment to the following mobile devices:

- iPhones using the latest release of iOS (iOS 10.0 and later covering 80% of current iOS users)
- iPad
- Android phones running Android KitKat 4.4 or higher (an estimated 81.4% of current Android users).
- Android Tablets running Android KitKat 4.4 or higher.

During each iterative update to the current production version of the software, the software is segmented into the smallest testable parts, and unit tests are run against each of these units. A passing grade on every unit test is an essential step in ensuring that the first public production release of Always UC Mobile functions as intended from beginning to finish. During the final weeks of the semester, during the team's weekly sprint review meeting, a testing report will be analyzed and discussed to help determine key areas of focus for execution of the following week's development sprint. Effective employment of these strategies promotes significant coverage and works to mitigate the probability of encountering serious issues and bugs during the final weeks leading up to our initial production release.

2.2.3 Logging & Reporting

For iOS, logging is handled in each unit test using NSLogger, a software library which is built into Swift. For Android, logging is handled in each unit test using Junit, a library built into Java. This ensures that with the addition of enhancements core functionality stays intact as well as providing insight on where updates are needed with future releases. Crash reporting has been implemented in order to investigate points of failure during our beta trial phase. These logs are

kept within Test Flight, a web application provided by Apple which provides the team with detailed reports on every critical error our beta users experience.

2.2.4 API Documentation and Test Bench

Always Education has a test bench web application to aid our developers in testing the web service end points that the mobile applications hit to read and write data to the AlwaysUC system. Not only does this allow our developers to ping the endpoints to validate that the endpoint server is up and running, but additionally shows the web service responses in plaintext JSON format. This allows our developers to see a full-screen JSON response from a web service call, which is a valuable feature when tracking down the root cause of any exceptions our developers encounter. The URLs for our API documentation and test bench are as follows:

API Documentation: https://aucdev.cech.uc.edu/api/docs/index.html#api-Board-News_Feed

API Test Bench: <https://alwaysuc.cech.uc.edu/api/v1/test>

3. Project Management

3.1 Objective

Our purpose is to create mobile applications for the iOS and Android platforms that engage the user, specific to their role, in order to empower and encourage participation in the university. We have delivered an application that spans the entire use of the already free standing Always UC web application.

3.2 Project Budget

The table below contains estimated costs for completion of the project as of 01/23/17. All Budget is rounded to the nearest dollar.

Item	Rate	Units	Total
Labor			
		840	
iOS Developer	30.00	Hours	\$25,200.00
		840	
Android Developer	30.00	Hours	\$25,200.00
		840	
Backend Developer	30.00	Hours	\$25,200.00
		840	
Project Manager	50.00	Hours	\$42,000.00
Total			\$117,600.00
Licensing			
Adobe Photoshop	700.00	1 Year	\$700.00
Apple Development	100.00	1 Year	\$100.00
Adobe Illustrator	220.00	1 Year	\$220.00
Sketch	60.00	1 Year	\$60.00

Total			\$1,080.00
Devices			
		1	
iPhone 5	286.00	Device	\$286.00
		1	
iPhone 6	320.00	Device	\$320.00
		1	
iPhone 7	740.00	Device	\$740.00
		1	
HTC One A9	299.00	Device	\$300.00
		1	
Samsung Galaxy S6	399.00	Device	\$400.00
		1	
iPad Pro	599.00	Device	\$600.00
Total			\$2,646.00
Project Total			\$121,326.00

Table 1: Project Budget

3.3 Schedule

The Gantt chart below provides a detailed illustration of the proposed Gantt chart for the duration of the project’s initial development period, spanning a single academic year.

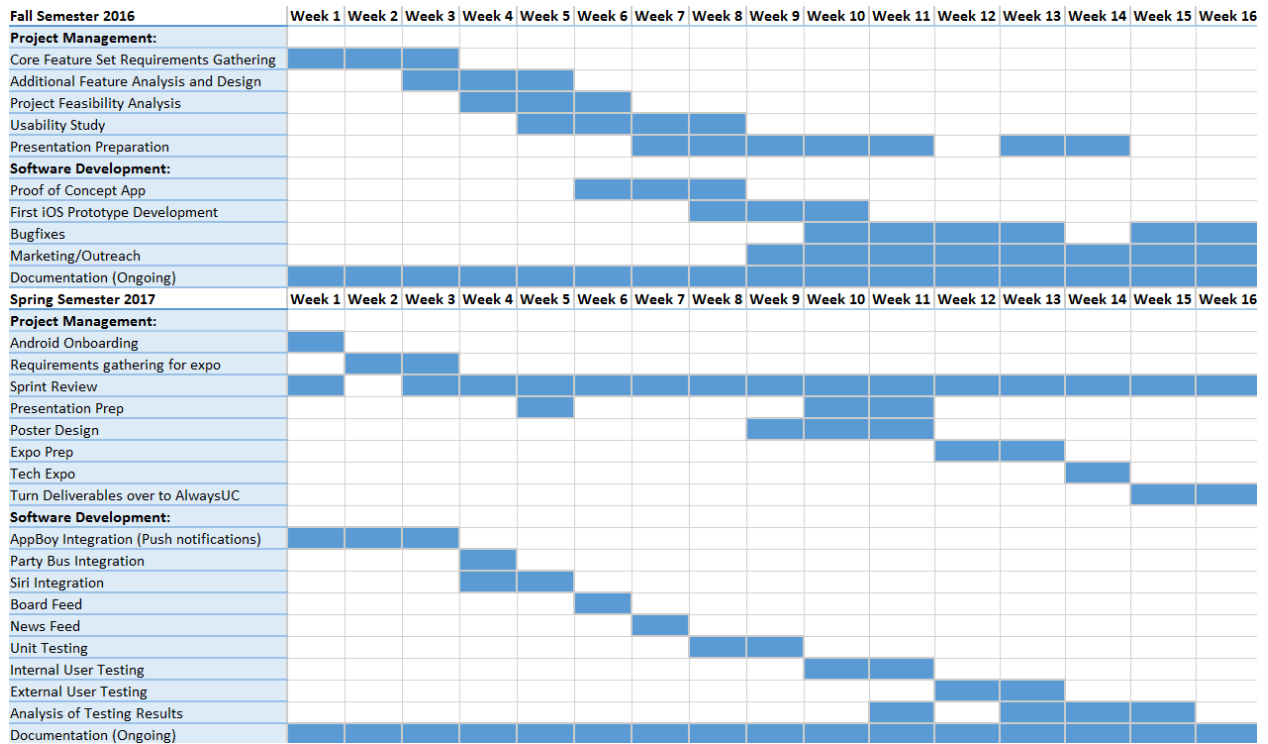


Figure 4: Gantt Chart

3.4 Deliverables

Build iOS and Android applications that mirror each other in functionality, to be delivered to the Always Education team. The application has the following features:

- Create an Agenda
- View Course Enrollment
- View Assignments both submitted and upcoming

-
- Connect University users via instant messaging
 - Integration with Siri for iOS
 - Connect via bulletin boards
 - Links to other University websites
 - Customizable interface based on user settings

4. Conclusion

Through role-based instruction, Always UC Mobile presents a piece of learning-management-software with the capability of making major quality-of-life improvements for each and every University student, affiliate, teacher, or other. The ability to easily collaborate with peers and share the passion of learning exists at the fingertips and within the pocket of every mobile user. Always UC Mobile strives to deliver a user friendly and reliable product that will encapsulate the development team's passion for learning and bring it to life.

Bibliography

Tadayon, Vali, interview by Josip Injic, Tyler Storm, & Daniel Sweet. *Always Education Needs a Mobile Client*. (August 17, 2016).

Tadayon, Vali, and Fariborz Grayli, interview by Josip Injic, Tyler Storm, & Daniel Sweet. *Technical Requirements for AlwaysUC Mobile*. (August 29, 2016).

Grayli, Fariborz, interview by Josip Injic, Daniel Sweet, & Tyler Storm. *AlwaysUC iOS Client* (October 4, 2016)

Hudephol, Trent, interview by Kenneth Keith *AlwaysUC Android Client* (December 12, 2016).

NPM, *asyncawait v1.0*, An open-source JavaScript library

for reduction of callback spaghetti code,
<https://www.npmjs.com/package/asyncawait>

expressjs, *body-parser*, an open-source node.js library for parsing middleware,

<https://github.com/expressjs/body-parser>

Express, *compression*, an open-source node.js library for middleware

compression, <https://github.com/expressjs/compression>

tj, *connect-redis*, an open-source Redis session store for Connect,

<https://github.com/tj/connect-redis>

chjj, *marked*, an open-source markdown parser and compiler,

<https://github.com/chjj/marked>

Khan, *KaTeX*, an open-source fast math typesetting for web development,

<https://github.com/Khan/KaTeX>

highlightjs, *highlight.js*, an open-source syntax highlighter for web development,

<https://highlightjs.org/>

NPM, *cookie-parser*, an open-source cookie header and req.cookies populator

With an object-keyed by the cookie names for node.js middleware,

<https://www.npmjs.com/package/cookie-parser>

NPM, *crypto-js*, an open-source JavaScript library of cryptography standards,

<https://www.npmjs.com/package/crypto-js>

Express, *express.js*, an open-source JavaScript library minimalist web framework

for node.js, <https://expressjs.com/>

gulpjs, *gulp*, The streaming build system for JavaScript,

<https://github.com/gulpjs/gulp>

Public Class, *express-partials*, an open-source JavaScript library for Layout and

Partial support, <https://github.com/publicclass/express-partials>

NPM, *sanitize-html*, an open-source simple HTML input sanitizer,

<https://www.npmjs.com/package/sanitize-html>

Express, *session*, an open-source simple session middleware for Express,

<https://github.com/expressjs/session>

Moment, *moment*, an open-source parser, validator, manipulator, and formatter

For dates in JavaScript. <https://github.com/moment/moment>

Request, *request*, an open-source library for constructing a simplified HTTP request client, <https://github.com/request/request>

onevcats, *Kingfisher*, an open-source lightweight purely Swift library for and caching images from the web. <https://github.com/onevcats/Kingfisher>

Pinterest, *PINCache*, an open-source fast, non-deadlocking parallel object cache for iOS, tvOS, and OS X. <https://github.com/pinterest/PINCache>

rs, *SDWebImage*, an open-source library for asynchronous image downloading with cache support such as the UIImageView category. <https://github.com/rs/SDWebImage>

Cocoa Controls, *SSHoneyCombView*, a honey comb View that looks like a collection view with hexagon shaped items for Swift. <https://www.cocoacontrols.com/controls/sshoneycombview>

laptobbe, *TSMarkdownParser*, an open-source library for parsing Markdown Text and Images, <https://github.com/laptobbe/TSMarkdownParser>

hyperoslo, *Whisper*, an open-source component that makes the task of displaying messages and in-app notifications simple.

<https://github.com/hyperoslo/Whisper>

Mike Penz, *MaterialDrawer*, A flexible, easy to use, all in one open-source drawer library for Android. <https://github.com/mikepenz/MaterialDrawer>

New Relic, *New Relic Monitoring*, Performance monitoring for Android applications, <https://newrelic.com/>

Jake Wharton, *ButterKnife*, an open-source library for asynchronous UI injections for Android, <http://jakewharton.github.io/butterknife/>

dlew, *joda-time-android*, an open-source library for parsing and formatting DateTime objects in Android, <https://github.com/dlew/joda-time-android>

Square, *Picasso*, a powerful image downloading and caching library for Android, <http://square.github.io/picasso/>

Manabu-GT, *ExpandableTextView*, an open-source android TextView that can expand/collapse like the Google Play Store's app descriptions, <https://github.com/Manabu-GT/ExpandableTextView>

MyABC, *markdownj*, a purely Java port of John Gruber's Markdown,

<https://github.com/myabc/markdownj>

Appendix

Testing	Status	Result
Unit Testing		
Objects	Complete	Success
Network Features	Complete	Success
User Testing		
User Acceptance Testing	In Progress	
Bug Bash Event	Complete	Feedback Received
Crash Reporting		
Firebase Implementation for Android	Complete	Success
Appboy Implementation for iOS	Complete	Success

Figure 5. Test Case Documentation