



College of Engineering

CS CAPSTONE FINAL REPORT

JUNE 10, 2018

"I HEART CORVALLIS" MOBILE APPLICATION

PREPARED FOR

CORVALLIS COMMUNITY RELATIONS OFFICE

LYNDI-RAE PETTY

PREPARED BY

GROUP 14

OMEED HABIBELAHIAN

BRADLEY IMAI

DYLAN TOMLINSON

Abstract

This document acts as the complete notebook for the I Heart Corvallis mobile application project. It details the project requirements and design, technology reviews for various aspects of the project, and our team's weekly blog posts detailing our progress. It also includes documentation on how the project works and how to install and run the application, and it provides some functionality flowcharts. The team also provides conclusions and reflections on the past year, and the appendices provide notable code listings and screenshots from the project.

CONTENTS

1	Introduction	3
2	Requirements Document	4
2.1	Requirement Modifications	12
2.2	Updated Gantt Chart	12
3	Design Document	13
3.1	Design Modifications	25
4	Tech Review	26
5	Weekly Blog Posts	50
5.1	Omeed	50
5.1.1	Fall 2017	50
5.1.2	Winter 2018	53
5.1.3	Spring 2018	56
5.2	Bradley	59
5.2.1	Fall 2017	59
5.2.2	Winter 2018	60
5.2.3	Spring 2018	61
5.3	Dylan	62
5.3.1	Fall 2017	62
5.3.2	Winter 2018	64
5.3.3	Spring 2018	65
6	Final Poster	67
7	Project Documentation	68
8	Recommended Technical Resources to Learn More	69
9	Conclusions and Reflections	70
9.1	Omeed	70
9.1.1	What technical information did you learn?	70
9.1.2	What non-technical information did you learn?	70
9.1.3	What have you learned about project work?	70
9.1.4	What have you learned about project management?	70
9.1.5	What have you learned about working in teams?	70
9.1.6	If you could do it all over, what would you do differently?	71
9.2	Bradley	71
9.2.1	What technical information did you learn?	71
9.2.2	What non-technical information did you learn?	71

9.2.3	What have you learned about project work?	71
9.2.4	What have you learned about project management?	71
9.2.5	What have you learned about working in teams?	72
9.2.6	If you could do it all over, what would you do differently?	72
9.3	Dylan	72
9.3.1	What technical information did you learn?	72
9.3.2	What non-technical information did you learn?	72
9.3.3	What have you learned about project work?	72
9.3.4	What have you learned about project management?	73
9.3.5	What have you learned about working in teams?	73
9.3.6	If you could do it all over, what would you do differently?	73
10	Appendix I: Essential Code Listings	74
10.1	EventListAdapter Class	74
10.2	PassportLoader Class	77
10.3	Recording a Completed Event	79
10.4	Updating a Prize	81
11	Appendix II: Screenshots	83
11.1	Application	83

1 INTRODUCTION

The "I Heart Corvallis" mobile application was conceptualized by the Corvallis Community Relations office to complement their larger "I Heart Corvallis" initiative. The CCR office noticed that many OSU students do not engage much with the greater community and prefer to interact with the OSU community itself, so they started this initiative to build and promote a strong and cooperative relationship between the OSU community and the greater Corvallis community. The goal of the mobile app is to inform students about the various community events, activities, and service opportunities available to them around the community, as well as to give them an incentive to engage in these activities. Our client was Lyndi-Rae Petty, the Graduate Teaching Assistant at the Corvallis Community Relations office, a subset of the Office of Student Life. The development team consisted of Omeed Habibelahian, Bradley Imai, and Dylan Tomlinson. All three of us were heavily involved in nearly all aspects of the application, and we often helped each other out when one of us was stuck on a particular section of the development. It was not so much that one of us took a particular role; we were all involved in the various aspects of development and implementation, as well as in communicating updates and changes with our client. There were, however, a few aspects of the development that we each took more control of. Omeed handled more of the implementation of the administrative website, Dylan was in charge of user security on both the mobile application and the administrative website, and Bradley put in quite a bit of work on the design of the user interface of the mobile application. Our client primarily took the role of supervisor of the project. Every week we would meet with her to discuss the changes we made and new features we implemented, as well as what we had to do next, and she gave input on how she felt about our changes and anything she wanted done differently. When we needed to reach out to another organization or source for guidance or help, we would describe our situation to her, and she would take charge of reaching out to these organizations.

I Heart Corvallis - Mobile Application Requirements Document

Capstone I Fall 2017

Omeed Habibelahian

Bradley Imai

Dylan Tomlinson



Abstract

This document defines the perspective and scope of the I Heart Corvallis project. It defines key terms that will be used throughout the document, the final product's perspective and functions, the main user characteristics, and constraints that we will face throughout the project. It also defines the specific requirements necessary for completion of the project, such as the external interface requirements, functional requirements for each class of users, performance requirements, software system attributes, and our stretch goal for this project. It also includes a Gantt chart graphing the time frames for completing various aspects of the project.

1 INTRODUCTION

1.1 Purpose

This document will define the scope and requirements for completing the development of the I Heart Corvallis mobile application. It defines key terms central to the development of the application and its background, constraints and software complications that could influence completion of this project, and what key features will need to be implemented in the application before it can be considered completed. This document's intended audience is the Corvallis Community Relations office, who we are building this application for, and our instructors that will be grading and evaluating the project at the end of the year.

1.2 Scope

In this project, we will be producing the "I Heart Corvallis" mobile application. The app will showcase events happening around the Corvallis community, such as city council meetings, service and volunteer projects, and other community activities. It will also act as a passport for users to show that they have attended these activities. The app will give the user stamps upon completion or verification of attendance for each activity and will offer rewards to the user for accumulating enough stamps. On top of this, the application will showcase other resources available to community members.

The application will be available for Android devices and aims to inform members of the Corvallis community, both students and others, about various initiatives and resources around the community, as well as get community members more involved with community projects, events, and meetings by giving them an incentive to do so.

Another goal of the app is to help students be more aware of community events. To accomplish this, the application will utilize the Google Maps API to show where events and various community resources can be found. The app will also include a separate page that will provide additional information about the city of Corvallis, such as links to website in the community and information about the Corvallis Community Relations (CCR) office and the initiative.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 *I Heart Corvallis*

The application being developed in this project.

1.3.2 *Corvallis Community Relations*

A subset of the Office of Student Life. The Corvallis Community Relations office is the leader of the I Heart Corvallis initiative and the client for this project.

1.3.3 *Stamp*

In-app verification that the user has attended a particular community activity. Some activities will be worth more stamps than others.

1.3.4 *IDE*

Integrated Development Environment; a software application that provides comprehensive facilities for software development. An IDE typically consists of a source code editor, build automation tools, and a debugger.

1.3.5 Xcode

The official IDE for software development on Apple's mobile and desktop operating systems.

1.3.6 Android Studio

The official IDE for software development on Google's Android operating system, built on JetBrains's IntelliJ IDEA software and designed specifically for Android development.

1.3.7 Database

A structured set of data held in a computer, especially one that is accessible in various ways.

1.3.8 Node.js

A runtime environment used for executing server-side JavaScript code.

1.3.9 Meteor

A web framework written in Node.js that allows for the production of cross-platform code between Android, iOS, and webpages.

1.3.10 MongoDB

A document-oriented database program.

1.4 Overview

The client of this application is the Corvallis Community Relations office. They have asked us to create a mobile application to aid them to list the various community events, projects, and meetings being put on by different organizations, authenticates the user, successfully tracks the events they engage in, and provides rewards for completing enough community activities. The I Heart Corvallis mobile application will be dedicated to OSU students and community members.

2 OVERALL DESCRIPTION

2.1 Product Perspective

This product will be visually similar to that of other Oregon State University mobile applications and desktop pages, but functionally it is completely independent of any other systems. The Corvallis Community Relations office wants to use the I Heart Corvallis application in conjunction with their OSU Corvallis Community Relations website, but the two will not be directly reliant on each other.

2.2 Product Functions

When the user first opens the application, it will show the login page. This page asks if the user is an OSU student or a permanent resident. Students will be redirected to the ONID login screen for authentication, while permanent residents will be asked to log in with their username and password. They will be given the option to create a username and password if they have not previously done so.

If the user is logging in for the first time, the app will display a short survey with a few questions for users to answer before their home page loads. Once the user has logged in, the app will display the home page, which provides quick links to the user's passport, the list of upcoming events, the stamp leaderboard, prize page, community resources page, and a page about the creators of the application and information about the I Heart Corvallis initiative.

Both the event list page and the passport page list community events in chronological order. Non-timed, location-based events will be listed first, followed by the events with a specific date and time being listed afterwards. Each event's information box will show the event title, location, and the date and time if applicable. The event list page will also show the number of stamps that each event is worth, whereas the passport page will show an indicator of whether they attended that event or activity, and how many stamps they received for doing so.

The user can press on an event to pull up the detailed information page for that event, which will include a picture for the event, as well as the event title, location, date and time if applicable, description, and any relevant links provided by the event host. The passport page will also provide a button that indicates the user's current level, which will be determined by the number of stamps they have accumulated. There will be bronze, silver, and gold levels, each of which will correspond to different prizes. When the user presses the button, they will be taken to a page that lists the available prizes, separated by level. This prize page will also be accessible from the home page.

The community resources page will show other resources available to community members, as well as a link to the Corvallis Living Guide website. This page will also include a Google Map that will display various community establishments such as entertainment locations, grocery stores, restaurants, shopping locations, and city offices. The "about" page will provide a short description of the app and its purpose, as well as information about the Corvallis Community Relations office.

The Corvallis Community Relations office will also have access to a back-end page that they will use to add, edit, and remove events, as all events will have to be approved by the office before being added to the application's event list, and the office will have the sole authority to post, edit, and remove these events. They will also be able to edit survey questions and other pages throughout the app, as well as add or remove stamps for users. They will be access this page through a special administrative login.

2.3 User Characteristics

The intended users of the I Heart Corvallis app are OSU students and permanent residents of Corvallis. The app does not require any special technical or educational expertise, as it will be designed for anyone to be able to navigate without needing to learn any new technical skills. As long as the user knows how to use a mobile application, the app will be designed to be simple enough for even non-tech savvy users to navigate.

2.4 Constraints

There are a couple of constraints that can potentially limit our options as far as project completion is concerned. The Corvallis Community Relations office wants the application to be developed for both Android and iOS, but Android and iOS apps are usually coded in different languages and tested in different IDEs. Therefore, efficiently coding for Android and iOS simultaneously in a cross-platform fashion will be a challenge. Because of this, our primary focus will be on completing the Android version of the app, and if time allows after completing the Android version, we will build the iOS version.

Another constraint is the availability of testing environments across multiple desktop platforms. While Android Studio is available for both Mac and Windows, Xcode is a Mac-exclusive IDE, and not everyone on the development team has a Mac computer. Therefore, we will have to find the best way for all three of us to build the iOS version of the application.

2.5 Assumptions and Dependencies

The main factor that will influence the requirements for this project is whether or not we can successfully build the application in a cross-platform fashion. The Corvallis Community Relations office has mentioned Meteor, or MeteorJS, as an option for cross-platform implementation of the app. Meteor would allow us to build the application using JavaScript and would allow for synchronous cross-platform implementation across both Android and iOS without the use of either Android Studio or Xcode. However, Meteor uses MongoDB as its database, which we have historically found to be more difficult than other databases such as SQL. Meteor is also a platform that none of us on the development team have prior experience with, whereas we do have experience with IDEs like Android Studio.

3 SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

When the application is opened for the first time, it will display a log-in screen. There will be two log-in systems implemented into the app, as the login for students will be separate than for permanent residents.

The app will display the home page whenever the user reopens the app. This page will display quick links to other pages, such as the user's passport, stamp leaderboard, prizes available, and a page about the creators of the application and the I Heart Corvallis initiative.

The passport system will allow the user to accumulate points, or "stamps," after completing or attending various community activities and events, and the user will be able to win prizes for accumulating enough stamps.

The application will also be connected to two databases, one for events and one for user account information. The event database information will hold all of the information for each event being advertised by the app. The user account database will hold the user's login credentials, which will be kept secret to minimize the risk of the wrong user getting access to someone's account, as well as their passport information.

The application will provide a map that displays the locations of important and notable community resources, events, and activities, as well as sources of entertainment, restaurants, shopping centers, and city offices. We will utilize the Google Maps API to implement this map into the application. The locations of the events advertised in the app will be shown on the map as well, and the user will be able to click on a pin on the map for more information about the event, activity, or establishment that pin represents.

3.1.2 Hardware Interfaces

The only hardware restriction on the application is that it will initially be exclusively available on the Google Play Store for Android devices. Our stretch goal is to build an iOS version of the application and publish it to the App Store as well. Other than this, the application itself doesn't have any designated hardware interfaces.

3.1.3 Software Interfaces

The application will use the user's smartphone's GPS to track their current location to check if they are at the location of a community event or activity. It will also require an internet connection to retrieve event information from the event database and retrieve the user's information from the account database.

3.1.4 Communications Interfaces

Communication between different sections of the app is vital for the app's success, so links and buttons will have to take the user to the correct page. Links will redirect to the smartphone's Internet browser, pressing on an event box will bring up more information about that event, and quick links will redirect to their respective full pages. The underlying operating system will also be vital to successful communication between sections of the app.

3.2 Functional Requirements

3.2.1 Student and Permanent Resident Users

Student users and permanent resident users shall be able to:

- View the list of events happening around Corvallis (both general information and detailed descriptions)
- View their passports
- Check in using their current location and enter a code to receive a stamp for time-based events
- Check in using their current location to receive a stamp for location-based events
- View a map of community establishments
- (Permanent residents) create an account with a username and password and log in
- (Students) Log in using their ONID account (no need to create an account; ONID is the account)
- View available prizes
- See and answer survey questions upon their first login (student survey will be different than the resident survey)
- View a leaderboard of who has accumulated the most stamps
- View an informative page about the I Heart Corvallis initiative, the creators of the app
- View a page informing the user about various resources available in the community

3.2.2 *Administrative Users*

Administrative users shall be able to:

- Add, edit, and remove events
- Edit survey questions
- Edit the community resources page
- Edit the "about us" page
- Log in to edit content via their own special administrative login page
- Add and remove stamps for users

3.3 **Performance Requirements**

The application will support up to 20 simultaneous users. The user will not have to input very much information, as the app will generally show them upcoming community events and activities, how much of their passport they have completed, what prizes they can win and how they can redeem it, and various community resources available to them.

3.4 **Software System Attributes**

3.4.1 *Reliability*

The application will have a user interface that 8 out of 10 users will be able to navigate without difficulty. The application will also function without crashing 95% of the time.

3.4.2 *Availability*

The application will initially be available for Android, and if we can meet our stretch goal, it will be available for iOS upon launch as well. The application will require an Internet connection to function properly. If the Internet connection is lost or the app crashes, the app will recover any account and event information that had already been successfully stored in the database.

3.4.3 *Security*

For OSU students, the authentication system will be the same as for other services that require an ONID login for authentication. The student will have to log in with their ONID username and password to be able to access the app. To protect student accounts, we will not store their ONID password in the account database, as the ONID interface itself authenticates the user. We will only store their ONID username. For permanent residents, we will encrypt their password to prevent potential hackers from obtaining users' passwords, and the encrypted password will be stored in the database.

3.4.4 *Maintainability*

We will create separate functional and formatting code files for each page in the app. This way, the app will be modularized in such a way that editing the makeup of one page will not affect the behavior of the other pages of the app. This will result in many code files being created but it will allow us to edit one page independent of the other pages.

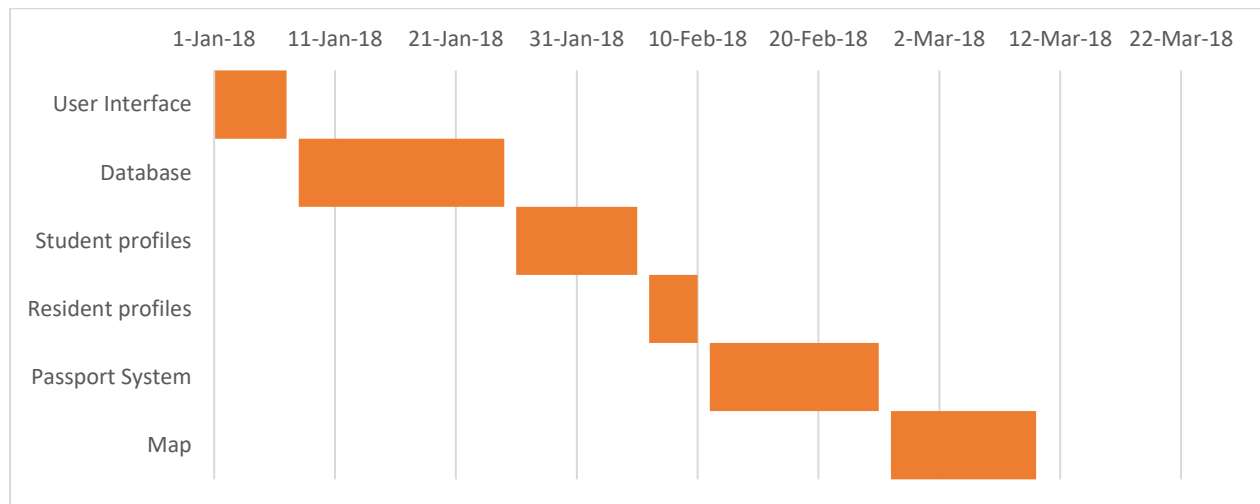
3.4.5 Portability

Since the primary programming languages for Android are different than for iOS, a majority of our components and code will be host-dependent. About 85% of the application's components will include host-dependent code, and about 80% of the total code for this project will be host-dependent. This is due to the fact that Android apps are generally coded in Java and XML, while iOS apps are generally coded in Objective-C and Swift. For the databases, we will code in SQL and PHP, both of which are portable programming languages.

3.5 Stretch Goals

Our main stretch goal is to build and publish an iOS version of the application, complete with all of the features available on the Android version of the application.

3.6 Timeline



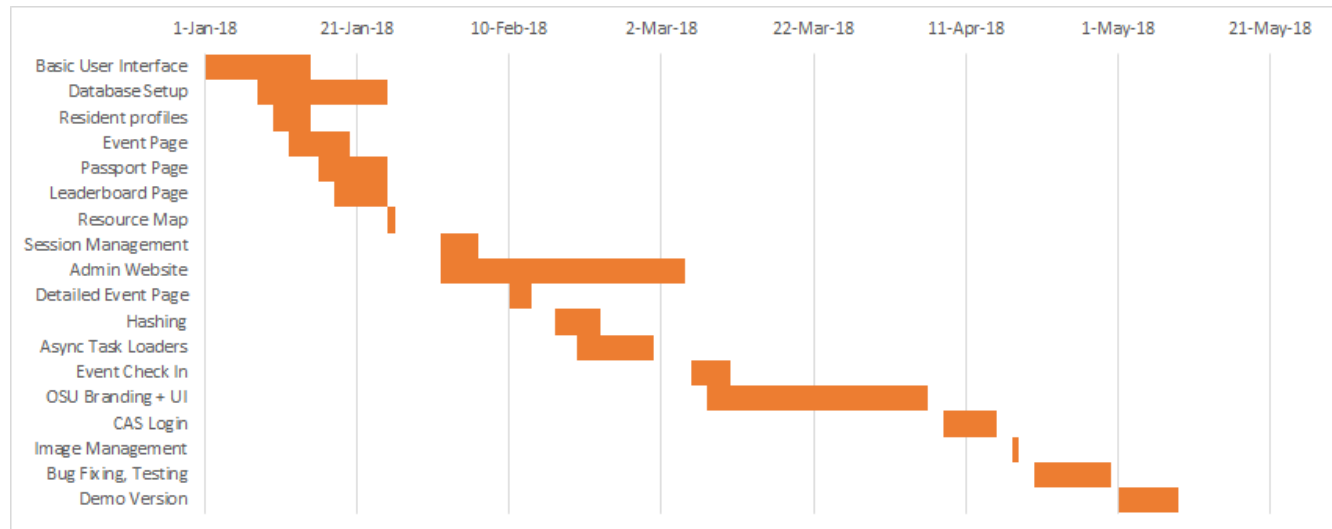
2.1 Requirement Modifications

The two requirements that were changed were the implementation of the ONID login and the database usage.

We did not implement the ONID login system, or CAS, into the application, as we were unable to figure out how to implement CAS on a mobile device. The documentation for integrating CAS into a website is fairly straightforward, but this documentation does not translate to mobile applications the same way. We did reach out to some other resources within OSU, like Derek Whiteside (Director of Web and Mobile Services) and Andrew Morgan (Systems Administrator for Identity and Access Management), and we received some valuable information from the latter source, but we were nonetheless unable to complete the requirement. We made sure our client was aware of this, and she gave us the green light to proceed without implementing CAS. Instead, we implemented a universal login system that asks the user if they are a student/faculty member during the signup process.

We also did not use two databases for the project, as mentioned in the document. We are not exactly sure what our reasoning was behind that idea. It is possible that we meant two tables rather than two databases, but we ended up only needing one table for all of the tables for the application and administrative website. Our MySQL database consisted of 11 tables.

2.2 Updated Gantt Chart



I Heart Corvallis - Mobile Application

Software Design Document

Capstone I

Fall 2017

Omeed Habibelahian

Bradley Imai

Dylan Tomlinson



Abstract

This document discusses the various aspects of the I Heart Corvallis mobile application, what technologies we will use to implement those pieces of the application, and our rationale for making our design decisions. It consolidates our research and the conclusions we came to in our respective Technology Review documents, explaining what our client requires for each aspect of the app, what implementations and technologies we will use to meet their requirements, and why the decisions we have made make sense in the context of the application.

CONTENTS

1	Overview	4
1.1	Scope	4
1.2	Purpose	4
1.3	Intended Audience	4
1.4	Conformance	4
2	Definitions	5
3	Conceptual Model for Software Design Descriptions	5
3.1	Software Design in Context	5
3.2	Influences on the Preparation of this Document	5
4	Design Choices	5
4.1	Data Storage and Handling	5
4.1.1	Overview	5
4.1.2	Our Client's Concerns	6
4.1.3	MySQL	6
4.1.4	Design Rationale	6
4.1.5	Connections to Other Parts of the Application	6
4.2	UI Design Architecture	6
4.2.1	Overview	6
4.2.2	Our Client's Concerns	6
4.2.3	Semantic UI	7
4.2.4	Design Rationale	7
4.3	Administrative Access	7
4.3.1	Overview	7
4.3.2	Our Client's Concerns	7
4.3.3	In-App Administrative Interface	7
4.3.4	Design Rationale	7
4.4	User Security	8
4.4.1	Overview	8
4.4.2	Our Client's Concerns	8
4.4.3	PBKDF2	8
4.4.4	Design Rationale	8
4.5	Interface Scaling	8
4.5.1	Overview	8
4.5.2	Our Client's Concerns	8
4.5.3	Semantic UI	8
4.5.4	Design Rationale	9

		3
4.6	Application Testing	9
4.6.1	Overview	9
4.6.2	Our Client’s Concerns	9
4.6.3	Android SDK Emulator	9
4.6.4	Design Rationale	9
4.7	Social Media Integration	9
4.7.1	Overview	9
4.7.2	Our Client’s Concerns	9
4.7.3	Instagram API	10
4.7.4	Design Rationale	10
4.8	Geolocation Services	10
4.8.1	Overview	10
4.8.2	Our Client’s Concerns	10
4.8.3	Google Maps Geolocation API	10
4.8.4	Design Rationale	10
4.8.5	Connections to Other Parts of the Application	10
4.9	Event Display	11
4.9.1	Overview	11
4.9.2	Our Client’s Concerns	11
4.9.3	Google Maps API	11
4.9.4	Design Rationale	11
4.9.5	Connections to Other Parts of the Application	11

References		11
-------------------	--	----

1 OVERVIEW

1.1 Scope

In this project, we will be producing the "I Heart Corvallis" mobile application. The app will showcase events happening around the Corvallis community, such as city council meetings, service and volunteer projects, and other community activities. It will also act as a passport for users to show that they have attended these activities. The app will give the user stamps upon completion or verification of attendance for each activity and will offer rewards to the user for accumulating enough stamps. On top of this, the application will showcase other resources available to community members.

The application will be available for Android devices and aims to inform members of the Corvallis community, both students and others, about various initiatives and resources around the community, as well as get community members more involved with community projects, events, and meetings by giving them an incentive to do so.

Another goal of the app is to help students be more aware of community events. To accomplish this, the application will utilize the Google Maps API to show where events and various community resources can be found. The app will also include a separate page that will provide additional information about the city of Corvallis, such as links to websites in the community and information about the Corvallis Community Relations (CCR) office and the initiative.

1.2 Purpose

This document will define the design specifications of the "I Heart Corvallis" mobile application. It defines key design decisions that we will be utilizing to implement the requirements for the application throughout the scope of this project. It explains the concerns and desires that the CCR office has in regards to the app, attributes related to various elements of the app and various design constraints we face in the development of the app.

1.3 Intended Audience

This document's intended audience is the Corvallis Community Relations office, who we are building this application for, and our instructors that will be grading and evaluating the project at the end of the year.

1.4 Conformance

The user interface design of the I Heart Corvallis application will conform to that of other official Oregon State University mobile applications.

2 DEFINITIONS

- **I Heart Corvallis:** The application being developed in this project.
- **CCR:** Corvallis Community Relations, a subset of the Office of Student Life. The Corvallis Community Relations office is the leader of the I Heart Corvallis initiative and the client for this project.
- **XML:** A software- and hardware-independent tool for storing and transporting data.
- **Database:** A structured set of data held in a computer, especially one that is accessible in various ways.
- **JavaScript:** A programming language that is commonly used in web development.
- **jQuery:** A fast, small, and feature-rich JavaScript library that makes HTML document traversal and manipulation, event handling, and animation much simpler.
- **Android Studio:** The official integrated development environment (IDE) for software development on Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.
- **Stamp:** In-app verification that the user has attended a particular community activity. Some activities will be worth more stamps than others.
- **MySQL:** An open-source relational database management system based on Structured Query Language (SQL).

[1]

3 CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTIONS

3.1 Software Design in Context

The I Heart Corvallis application is a subset of the I Heart Corvallis initiative, spearheaded by the CCR. The initiative's goal is to enhance OSU students' sense of belonging to the Corvallis community and cultivate students' commitment to the common good. The application is intended to be a mobile extension of the initiative, and below we will showcase how we plan to design this application to best suit the interests of the CCR office and the I Heart Corvallis initiative.

3.2 Influences on the Preparation of this Document

The preparation of the software design document is driven by the Requirements Document, which states the specific requirements for the project to be completed. Also, the time constraint placed on the project by the Capstone courses has an effect on how exactly we will design and implement the application, as we may have to find simpler solutions for some aspects of the app to complete it within our allotted timeframe.

4 DESIGN CHOICES

4.1 Data Storage and Handling

4.1.1 Overview

Data storage and communication between databases will be crucial in our application. The application will utilize at least three separate databases, one for event information, one for prize information, and at least one for user account information (depending on whether or not we make separate databases for students and permanent residents), so information will frequently be shared and synchronized across the databases.

4.1.2 *Our Client's Concerns*

The CCR office requires that the database system used in the app can efficiently store, modify, and remove entries from the database. The database needs to be secure enough that only they can modify its contents, and it needs to be fast enough that changes made to the database are nearly immediately reflected in the app.

4.1.3 *MySQL*

MySQL is an open-source relational database management system written in C and C++. It works on many different system platforms, such as Linux, macOS, Windows, FreeBSD, and OpenBSD. Developers can also use PHP to enable access to MySQL databases in Android applications. MySQL is used in many well-known websites, like Facebook, Twitter, Flickr, and YouTube. To access our MySQL databases in our application, we will code parts of the app in PHP. [2]

4.1.4 *Design Rationale*

When researching database systems, we found MySQL, MongoDB, and PostgreSQL to all be viable options. However, we decided to use MySQL for our database implementation. MySQL provides great security features and is fast, which is vital to our application due to the constant communication between the user and the database. PostgreSQL is a good option but may be more powerful than necessary for this application, and it is not quite as fast as MySQL. MongoDB has some great features, but we have tried implementing MongoDB as a database management system for a website in the past and spent weeks trying to set it up. We were never able to successfully implement MongoDB into our website and ended up switching to MySQL anyway.

4.1.5 *Connections to Other Parts of the Application*

How application data is stored and handled is central to the functionality of the app, since all of the events' information will be stored in a database and referenced in different sections of the app. The app's geolocation functionality will use the user's current location to show and direct them to events near them, and that implementation will need access to the information stored in the event database. On top of this, the in-app map that shows the locations of all the events also needs access to the information in the event database. [2] [3]

4.2 **UI Design Architecture**

4.2.1 *Overview*

The user interface is going to be a major part of the application, as people will not use the app if it does not look visually appealing. The app also has to be easy to use so users continue using the app. Because of this, it's critical to choose a UI architecture that makes navigating the app as simple and visually clean as possible.

4.2.2 *Our Client's Concerns*

The CCR office requires that the user interface architecture we use is modifiable enough that the interface can be made to resemble other Oregon State University applications where necessary.

4.2.3 *Semantic UI*

Semantic UI is a user interface component library implemented using a set of specifications designed around natural language. It uses HTML and CSS and provides tons of templates for cards, boxes, buttons, menus, and icons. Because Semantic UI is implemented using HTML and CSS, we would code the functional parts of the app in JavaScript instead of Java because JavaScript, HTML, and CSS work together very well, and you can reference JavaScript functions and CSS style guidelines within HTML code. [4]

4.2.4 *Design Rationale*

While researching the UI interface options available to us, we considered Android SDK, Apache Cordova, and Semantic UI all to be strong potential choices. Although all three options definitely could come in handy in this program, we decided to use Semantic UI along with the Android Studio IDE from Android SDK. Semantic UI will allow us to implement a clean interface without resorting to the stock Android style, and although Android's default languages are Java and XML, you can edit the XML file corresponding to each code file to recognize JavaScript as the language for that file. Semantic UI will also give us flexibility and customizability in creating our desired interface, as well as interfaces that align with that of other OSU applications. [5]

4.3 **Administrative Access**

4.3.1 *Overview*

The CCR office is in charge of what content gets added to the app. They approve all the events in the app, and they have sole authority to modify or remove any content in the app. Because they are the only ones that can do this, they need to have an exclusive platform on which they can do this.

4.3.2 *Our Client's Concerns*

The CCR office requires that they are able to add, edit, and remove events, as well as edit user information, such as their stamp count. They prefer that they can make these modifications within the app itself via an administrative login.

4.3.3 *In-App Administrative Interface*

By creating an in-app administrative interface for our clients, they would be able to make changes to in-app content by simply logging into the app through a special administrative login. The interface for this administrative account would structurally look similar to the rest of the app, but we would implement pages, buttons, and quick links to allow the CCR office to make changes to the app on their mobile device. Changes made would directly affect the databases and would allow for quick content modification and quick resolution of any problems users could be having with their content or account.

4.3.4 *Design Rationale*

We considered a few different options when looking into the best way to make it so that the CCR office can edit content within the app. We considered making a separate website for our clients that they can use to edit database content, giving them direct access to the literal database, and creating a special administrative interface within the app that they can access via an administrative login. Though all three options would be viable, we are going to implement the special in-app administrative login for our clients to edit in-app content. It will allow us to focus on a single end product instead

of building both a mobile and web interface for the app, and it would be the simplest option for our clients, as they do not have a lot of technical experience. Therefore, they need a simple way to edit content within the application, and an in-app administrative login would be the simplest solution.

4.4 User Security

4.4.1 Overview

Users of the app who are not students of OSU will need to make accounts before accessing the full functionality of the application. These accounts will require a unique email address and a password. We are obligated to protect the user's passwords as it has been shown that most users will repeat their passwords for different - and possibly more important - accounts.

4.4.2 Our Client's Concerns

The CCR office requires that the users of the application have a guaranteed level of security for their account credentials.

4.4.3 PBKDF2

Password-Based Key Derivation Function 2, or PBKDF2, is a hash function aimed at reducing an encrypted password's vulnerability to brute force attacks. It was originally written in 2000, but is still recommended for password hashing by RFC 8018 in 2017. PBKDF2 applies a pseudorandom function to derive keys, and the length of the derived key is essentially unbounded. [6] [7]

4.4.4 Design Rationale

We decided on PBKDF2 over a few other hashing algorithms because it hits the middle ground of what is needed for this project. It guarantees a certain level of security for our users without requiring a large and complicated implementation. [6]

4.5 Interface Scaling

4.5.1 Overview

The I Heart Corvallis application is going to be an Android application available on the Google Play Store. This means that we have to deal with several different types of mobile devices with different screen sizes and resolutions. We will need an easy way to ensure that our application retains its visual quality regardless of which Android phone it is running on.

4.5.2 Our Client's Concerns

The CCR office is requiring that this application be available on a variety of Android devices. They are also requiring that the visual quality of the user interface remains consistent across these devices.

4.5.3 Semantic UI

Semantic UI is a HTML5-based framework that is powered by jQuery. It is used for developing Android applications, as well as websites. It focuses on using semantics-based commands, which makes its syntax very easy to understand. Semantic UI uses the HTML5 and CSS3 programming languages, and we would utilize JavaScript as well since it integrates well with HTML and CSS. [4] [8]

4.5.4 *Design Rationale*

There were a few different choices that we could implement for this application using the Android SDK with varying levels of success. But using the built in commands given in Semantic UI will help us stay consistent in our program without having to rely too much on external libraries. There is already enough control in the base language for us to easily scale this user interface for different screens.

4.6 **Application Testing**

4.6.1 *Overview*

Throughout the app implementation process, we will have a list of requirements that we need to complete to constitute a finished product. In order to determine that a project requirement has been completed, we have to be able to test if the functionality of that requirement meets our expectations and desires. This testing includes ensuring that our in-app functions perform as expected and testing that our application's interface design is as we planned it to be.

4.6.2 *Our Client's Concerns*

The CCR office wants an application that functions as intended 90% of the time. They are concerned that an application riddled with bugs will not be accepted by its target audience.

4.6.3 *Android SDK Emulator*

The Android Software Development Kit (SDK) contains an emulator that allows you to emulate many different android devices including their screen sizes, hardware characteristics, and android version. This emulator can be used to run our application and perform necessary tests on our features for many different devices. [9]

4.6.4 *Design Rationale*

Out of several different choices we decided to utilize the emulator provided in Android Studio: the Android SDK Emulator. It can emulate a vast number of Android devices and it has a vast amount of documentation in case we run into errors or other problems while testing the functionality of the app. [9]

4.7 **Social Media Integration**

4.7.1 *Overview*

Integrating a social media aspect such as Twitter and Instagram photos from an event into our application will allow the user to view what others have done at those particular events and also entice them into exploring them. Having tweets and photos posted on that specific event will be crucial to keeping that event's page organized.

4.7.2 *Our Client's Concerns*

The CCR office desires that users can post about their experiences at an event and view other users' experiences from that event. They would like the app to filter photos in such a way that photos are only presented for the event they correspond to.

4.7.3 *Instagram API*

Instagram's API will allow us to post Instagram photos or a photo stream on our mobile application from any personal account. This API will also create a gallery of images that will automatically update as new pictures are added. The process of implementing these features into our application will first start off by setting up the library, followed by obtaining the web API keys, configuring the login page, building the API output, and lastly dumping the data out. To implement Instagram's API into our application, we will code in JavaScript and jQuery. [10]

4.7.4 *Design Rationale*

The potential choices we chose for this feature were Instagram API, Twitter API, and Facebook API. While all three APIs provide great features to our application, we will first be implementing Instagram's API. By providing a feed of photos in each event filtered by location and relevant hashtags, other users could be encouraged to attend that event.

4.8 **Geolocation Services**

4.8.1 *Overview*

Retrieving the geolocation (current location) for our application is a critical tool to our mobile application. By obtaining a user's current location, we will allow the user to see various events that are happening around them, and the app will be able to prove that they have actually attended an event, providing an added layer of security.

4.8.2 *Our Client's Concerns*

The CCR office desires that the app can retrieve a user's geolocation to show them and direct them to events near them, as well as authenticate a user's location to prove that they are at a particular event before giving them a stamp.

4.8.3 *Google Maps Geolocation API*

The Google Maps Geolocation API is a useful tool for retrieving the current location of an individual. Provided on the Google Maps Geolocation API website is a tutorial on how to display the geographic location of a user or device on Google Maps using the browser's HTML5 Geolocation feature along with the Google Maps JavaScript API. This website also provides example code and comments on how it works, making this a very useful tool. Both Android and iOS Google Maps APIs require the app to prompt the user for consent to use their location services. [11]

4.8.4 *Design Rationale*

The potential choices we chose for this feature were Google Maps Geolocation API, Cordova - Plugin - Geolocation API, and Android Location API. Even though all three location service APIs work for our application, we will be using the Google Maps Geolocation API. It provides a more powerful, high-level framework that automatically handles location providers, user movement, and location accuracy. It also handles location update schedule based on power consumption parameters we provide. In most cases, you will get better battery performance, as well as more appropriate accuracy, by using the Google Maps Geolocation API. [11]

4.8.5 *Connections to Other Parts of the Application*

The application's geolocation implementation will need access to the event database to properly display the events near the user's current location, so it's important that these two parts of the app can efficiently communicate with each other.

4.9 Event Display

4.9.1 Overview

This application will also serve as a platform to highlight useful resources around Corvallis. By implementing two maps into our application, we will be able to fulfill this requirement. The first map will showcase notable establishments around Corvallis, including activities and entertainment, grocery stores, restaurants, shopping centers and city offices. The second map will display events approved by the Corvallis Community Relations office (CCR), at which students will be able to accumulate stamps. Having separate maps will allow the user to easily navigate to local events and notable establishments.

4.9.2 Our Client's Concerns

The CCR office would like to include a previously created Google Map that pins various establishments around Corvallis, as well as implement another map that displays the locations of the events that are listed in the application.

4.9.3 Google Maps API

Google Maps API provides a feature called Markers. A Marker identifies a location on a map and can display a custom image of that location. According to the Google Maps Marker developers page, "Markers are designed to be interactive. By default they receive 'click' events, so you can add an event listener to bring up an information window displaying custom information." Another feature of this API allows the user to remove a Marker from the map. Google's website explains the process on how to implement Markers or events on the map and provides sample code for us to implement into our application. We will utilize JavaScript to implement this API. [12]

4.9.4 Design Rationale

The potential choices we chose for this feature were Google Maps API, MapBox, and Microsoft Bing Map. Although all three APIs provide great features, we will be using Google Maps API as our source to display events to our application. The Google Developer Manual for this API provides detailed information on how to implement their code into our application. [12]

4.9.5 Connections to Other Parts of the Application

Like the geolocation implementation, the event map will need access to the event database to properly display the locations and details of events around town. Therefore, it's crucial that these two pieces of the application can easily communicate with each other.

REFERENCES

- [1] "jQuery," 2017, [Online; accessed 30-November-2017]. [Online]. Available: <https://jquery.com/>
- [2] "Mysql," 2017, [Online; accessed 29-November-2017]. [Online]. Available: <https://www.openhub.net/p/mysql>
- [3] DigitalOcean, "Sqlite vs mysql vs postgresql: A comparison of relational database management systems," 2014, [Online; accessed 12-November-2017]. [Online]. Available: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
- [4] S. UI, "Semantic ui," [Online; accessed 13-November-2017]. [Online]. Available: <https://semantic-ui.com>
- [5] J. Hutber, "Add js and css support to android studio project," 2013, [Online; accessed 13-November-2017]. [Online]. Available: <https://stackoverflow.com/questions/17137577/add-js-and-css-support-to-android-studio-project>

- [6] J. Jarmoc, "Enough with the salts: Updates on secure password schemes," 2015, [Online; accessed 29-November-2017]. [Online]. Available: <https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2015/march/enough-with-the-salts-updates-on-secure-password-schemes/>
- [7] B. Kaliski, "Pkcs #5: Password-based cryptography specification, version 2.0," 2000, [Online; accessed 29-November-2017]. [Online]. Available: <https://tools.ietf.org/html/rfc2898#section-5.2>
- [8] I. Gerchev, "Introducing: Semantic ui component library," 2014, [Online; accessed 29-November-2017]. [Online]. Available: <https://www.sitepoint.com/introducing-semantic-ui-component-library/>
- [9] "Run apps on the android emulator," [Online; accessed 29-November-2017]. [Online]. Available: <https://developer.android.com/studio/run/emulator.html>
- [10] Connecting with api data from instagr.am using php. [Online]. Available: <http://spyrestudios.com/connecting-with-api-data-from-instagram-using-php/>
- [11] The google maps geolocation api. [Online]. Available: <https://developers.google.com/maps/documentation/geolocation/intro>
- [12] Markers. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/markers>

3.1 Design Modifications

There were a few design aspects that we changed or removed altogether.

The plan to integrate a social media API, such as Instagram, was scrapped, as it did not fit into the user interface of our application and was deemed unnecessary. The plan for an in-app administrative interface was also changed to a web-based interface. This change was requested by our client, who told us that it would be easier for her to maintain the content of the application on her computer rather than on her phone in the application.

We changed the application from an HTML-based application to a stock Android, Java-based application. As a result of this, we also strayed away from using Semantic UI in the application. We made this change because in order to enable PHP on the application, we had to install PHP onto the device, which we were having trouble implementing. Android also has significantly more documentation for these languages, which helped us out quite a bit in the long run.

We did not focus on scaling the UI of the application. This design decision fell out of our scope as we focused on the main UI and functionality of the app. Our client was satisfied with the condition of the application, prompting us to consider our development of our application finished in early May.

I Heart Corvallis - Mobile Application

Technology Review

Capstone I

Fall 2017

Omeed Habibelahian



Abstract

This document takes a deeper look at a few of the technologies that will be used in the implementation of the I Heart Corvallis application, specifically data storage and handling, the UI design architecture, and administrative access. Each technology will be broken down into three different possibilities, and these three will be explained, compared, and contrasted. Then, for each technology, we will make a decision on which implementation or option we will use for our application.

CONTENTS

1	Data Storage and Handling	3
1.1	Overview	3
1.2	Criteria	3
1.3	Choices	3
1.3.1	MySQL	3
1.3.2	MongoDB	3
1.3.3	PostgreSQL	3
1.4	Discussing the Choices	3
1.5	Conclusion	4
2	UI Design Architectures	4
2.1	Overview	4
2.2	Criteria	4
2.3	Choices	4
2.3.1	Android SDK	4
2.3.2	Apache Cordova	4
2.3.3	Semantic UI	5
2.4	Discussing the Choices	5
2.5	Conclusion	5
3	Administrative Access	5
3.1	Overview	5
3.2	Criteria	5
3.3	Choices	5
3.3.1	Web Interface to Edit Content	5
3.3.2	Direct Access to Databases	6
3.3.3	Special Administrative Login Within App	6
3.4	Discussing the Choices	6
3.5	Conclusion	6
	References	6

1 DATA STORAGE AND HANDLING

1.1 Overview

Data storage and communication between databases will be crucial in our application. The application will utilize at least three separate databases, one for event information, one for prize information, and at least one for user account information (depending on whether or not we make separate databases for students and permanent residents), so information will frequently be shared and synchronized across the databases.

1.2 Criteria

We will need the ability to modify database information as users interact with the app. On top of users needing to be able to view the event information, show interest in an event, view prizes, and be rewarded for attending events, the Corvallis Community Relations office needs to be able to edit the information in these databases. Any changes made to events by the office need to be synchronized across any other databases that also hold that information.

1.3 Choices

1.3.1 *MySQL*

MySQL is an open-source relational database management system written in C and C++. It works on many different system platforms, such as Linux, macOS, Windows, FreeBSD, and OpenBSD. Developers can also use PHP to enable access to MySQL databases in Android applications. MySQL is used in many well-known websites, like Facebook, Twitter, Flickr, and YouTube. [1] [2]

1.3.2 *MongoDB*

MongoDB, classified as a NoSQL database program, is a free and open-source cross-platform document-oriented database program that uses JSON-like documents with schemas. MongoDB supports field and range queries, as well as regular expression searches. The queries can return specific fields of documents and also include user-defined JavaScript functions. MongoDB can also be integrated into Android applications via the MongoDB Stitch Android SDK. [3] [4]

1.3.3 *PostgreSQL*

PostgreSQL, or simply Postgres, is an object-relational database management system with an emphasis on extensibility and standards compliance. It can handle workloads ranging from small single-machine applications to large Internet-facing application, and it is the default database on macOS. An Android application can communicate directly with a Postgres database by using the PostgreSQL JDBC driver. [5] [6]

1.4 Discussing the Choices

All three database management systems can be used in Android applications. MongoDB supports the most programming languages at 27, followed by MySQL at 19 and Postgres at 9 languages. However, all three support both Java and PHP. MongoDB only provides their JSON-based proprietary protocol, however, whereas MySQL and Postgres both provide JDBC drivers as a potential access method. MySQL is also feature-rich, fast, and offers some rather advanced security features. One downside of MySQL is that it suffers from relatively poor performance scaling. It also isn't fully SQL-compliant, and its functionality tends to be heavily dependent on add-ons, but it is extremely popular, and as a

result there are a lot of third-party applications, tools and integrated libraries which help greatly with many aspects of working with it. Postgres is not as popular as MySQL, but there are still many great third-party tools and libraries that are designed to make working with Postgres simple. Postgres is very powerful and as a result does a great job of handling many tasks very efficiently. However, it is so powerful that it can be overkill in some cases and may appear less performant than MySQL. It's not quite as recommended for speedy operations and simple setups. [7] [8] [9]

1.5 Conclusion

Although all three of these database management systems have their upsides, we will be using MySQL for our database implementation. MySQL provides great security features, and is fast, which is vital to our application due to the constant communication between the user and the database. Postgres is a good option but may be more powerful than we need for this application, and it's not quite as fast as MySQL. MongoDB has some great features, but we've tried implementing MongoDB as a database management system for a website in the past and spent weeks trying to set it up. We were never able to successfully implement MongoDB into our website and ended up switching to MySQL anyway.

2 UI DESIGN ARCHITECTURES

2.1 Overview

The user interface is going to be a major part of the application, as people won't use the app if it doesn't look visually appealing. The app also has to be easy to use so users continue using the app. Because of this, it's critical to choose a UI architecture that makes navigating the app as simple and visually clean as possible.

2.2 Criteria

The user interface needs to include features like access to other pages, quick link buttons, and cards/boxes for events. Clutter needs to be kept to a minimum, and event listings need to be aligned with each other. The app interface needs to stand out and not look like a generic stock Android app. It also needs to incorporate OSU color schemes, fonts, and logos to remain consistent with the schemes of other OSU applications.

2.3 Choices

2.3.1 *Android SDK*

Android SDK is the official software developer kit for creating Android applications, provided by Google. It provides you with the API libraries and developer tools necessary to build, test, and debug apps for Android. It provides you with Android Studio, which is the official integrated developing environment (IDE) for Android and also allows you to emulate Android devices on your computer to see how your app would function on an actual Android device. Android's default programming language is Java, and its default formatting language is XML. [10]

2.3.2 *Apache Cordova*

Apache Cordova is an open-source mobile development framework that allows you to use HTML5, CSS3, and JavaScript for cross-platform development. Applications execute within wrappers targeted to each platform. According to its overview guide, Cordova is a good option if you are a mobile developer interested in either "mixing native application components with a special browser window that can access device-level APIs" or extending an application across platforms "without having to re-implement it with each platform's language and tool set." [11]

2.3.3 *Semantic UI*

Semantic UI is a UI component library implemented using a set of specifications designed around natural language. It uses HTML and CSS and provides tons of templates for cards, boxes, buttons, menus, and icons. Because Semantic UI is implemented using HTML and CSS, we would code the functional parts of the app in JavaScript instead of Java because JavaScript, HTML, and CSS work together very well, and you can reference JavaScript functions and CSS style guidelines within HTML code. [12]

2.4 **Discussing the Choices**

Android SDK is the official SDK provided by Google and provides Android Studio, which is really helpful because it's a one-stop shop for building, testing, debugging, and emulating Android apps. It's also pretty easy to use. When using Android SDK, apps are coded in XML and Java, which we have prior experience with. However, unless you do some thorough design formatting, the end product will utilize the stock Android design style, which is fine if not basic. We want to build an app that is more consistent with the design of other OSU applications, and we do not want our app to look like every other stock Android app. Apache Cordova uses HTML5, CSS3, and JavaScript, which allows for quite a bit of control over how the interface looks. XML allows for this too, though. Semantic UI also utilizes HTML and CSS, but it also has the big upside of providing tons of templates for cards, boxes, menus, and buttons. It also provides a large library of icons, which will come in great handy throughout our application.

2.5 **Conclusion**

Although all three options definitely can come in handy in this program, we will be using Semantic UI along with Android Studio. Semantic UI will allow us to implement a clean interface without resorting to the stock Android style, and although Android's default languages are Java and XML, you can edit the XML file corresponding to each code file to recognize JavaScript as the language for that file. [13]

3 **ADMINISTRATIVE ACCESS**

3.1 **Overview**

The Corvallis Community Relations (CCR) office is in charge of what content gets added to the app. They approve all the events in the app, and they have sole authority to modify or remove any content in the app. Because they are the only ones that can do this, they need to have an exclusive platform on which they can do this.

3.2 **Criteria**

Our clients (the CCR office) needs to be able to add, modify, and remove events and all related information about the event from the database of events. They also need to be able to do this with the prizes available in the app. On top of this, they need to be able to edit users' information if necessary.

3.3 **Choices**

3.3.1 *Web Interface to Edit Content*

One way to handle this is to create a desktop/web interface that allows them to edit in-app content. They would be able to go to a webpage that gives them access to the user information database(s), the event database, and the prize database, and the site would provide them with fields for adding and modifying information, as well as the option to delete information from a particular database.

3.3.2 *Direct Access to Databases*

Another way to give our clients administrative access to the app is to give them direct access to the databases. They would be able to directly make changes to fields in the databases, and instead of us creating a platform on which they could make changes, the database interface itself would provide all of these features to our clients.

3.3.3 *Special Administrative Login Within App*

One more route would be to create a special login within the I Heart Corvallis app itself. Instead of logging in as a regular student or permanent resident user, our clients would have a separate login form that would allow them to essentially enter the back end of the app. We would create an interface for this admin account type as well, and it would structurally look similar to the rest of the app, but we would implement pages, buttons, and quick links to allow the CCR office to make changes to the app on their mobile device. Changes made would directly affect the databases and would allow for quick content modification and quick resolution of any problems users could be having with their content or account.

3.4 **Discussing the Choices**

Both the web interface and the in-app administrative login would require a special log-in form exclusively for our clients. Both options are feasible, but the web interface would require making a separate website on top of the app, whereas the special administrative login within the app would just require a bit more work on the app itself. Giving our clients direct access to the databases could work, but our clients do not have very much technical experience, so it could be a more complicated option for them. The special in-app interface would allow everything regarding management of the app to stay within the app instead of having to create an external service.

3.5 **Conclusion**

Though all three options would be viable, we are going to implement the special in-app administrative login for our clients to edit in-app content. It would allow us to focus on a single end product instead of building both a mobile and web interface for the app, and it would be the simplest option for our clients, as they do not have a lot of technical experience. Therefore, they need a simple way to edit content within the application, and an in-app administrative login would be the simplest solution.

REFERENCES

- [1] Wikipedia, "Mysql — wikipedia, the free encyclopedia," 2017, [Online; accessed 12-November-2017]. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=MySQL&oldid=809470205>
- [2] TutorialsPoint, "Android - php/mysql," 2017, [Online; accessed 12-November-2017]. [Online]. Available: https://www.tutorialspoint.com/android/android_php_mysql.htm
- [3] Wikipedia, "Mongodb — wikipedia, the free encyclopedia," 2017, [Online; accessed 12-November-2017]. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=MongoDB&oldid=809465469>
- [4] MongoDB, "Integrate the mongodb service with your android application," 2008-2017, [Online; accessed 12-November-2017]. [Online]. Available: <https://docs.mongodb.com/stitch/examples/mongodb-android/>
- [5] Wikipedia, "Postgresql — wikipedia, the free encyclopedia," 2017, [Online; accessed 12-November-2017]. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=PostgreSQL&oldid=809546300>
- [6] M. Wong, "Android & postgresql," 2011, [Online; accessed 12-November-2017]. [Online]. Available: <https://www.pgcon.org/2011/schedule/track/Applications/278.en.html>

- [7] D. Engines, "System properties comparison mongodb vs. mysql vs. postgresql," 2017, [Online; accessed 12-November-2017]. [Online]. Available: <https://db-engines.com/en/system/MongoDB%3BMySQL%3BPostgreSQL>
- [8] DigitalOcean, "Sqlite vs mysql vs postgresql: A comparison of relational database management systems," 2014, [Online; accessed 12-November-2017]. [Online]. Available: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
- [9] J. Mack, "Five advantages & disadvantages of mysql," 2014, [Online; accessed 19-November-2017]. [Online]. Available: <https://www.datarealm.com/blog/five-advantages-disadvantages-of-mysql/>
- [10] Stackshare.io, "Android sdk vs. semantic ui," 2017, [Online; accessed 13-November-2017]. [Online]. Available: <https://stackshare.io/stackups/android-vs-semantic-ui>
- [11] T. A. S. Foundation, "Overview," 2012, 2013, 2015, [Online; accessed 13-November-2017]. [Online]. Available: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>
- [12] S. UI, "Semantic ui," [Online; accessed 13-November-2017]. [Online]. Available: <https://semantic-ui.com>
- [13] J. Hutber, "Add js and css support to android studio project," 2013, [Online; accessed 13-November-2017]. [Online]. Available: <https://stackoverflow.com/questions/17137577/add-js-and-css-support-to-android-studio-project>



College of Engineering

CS CAPSTONE TECHNOLOGY REVIEW

JUNE 4, 2018

I HEART CORVALLIS

PREPARED FOR

LYNDI-RAE ELISABETH PETTY

Signature

Date

PREPARED BY

GROUP 14

I HEART CORVALLIS

BRADLEY IMAI

Signature

Date

Abstract

This document takes a closer look at the technologies that will be carried out into the I Heart Corvallis Mobile application. Social media integration, user action verification (geolocation) and event display will be the three topics discussed. Each topic will be broken down into three potential options which are all compared and contrasted. Lastly, we will determine which implementation will be used in our application.

CONTENTS

1	Social Media Integration	2
1.1	Overview	2
1.2	Criteria	2
1.3	Potential Choices	2
1.3.1	Instagram API	2
1.3.2	Twitter API	2
1.3.3	Facebook API	2
1.4	Discussion	3
1.5	Conclusion	3
2	User action - verification (Geolocation)	3
2.1	Overview	3
2.2	Criteria	3
2.3	Potential Choices	3
2.3.1	Google Map API	3
2.3.2	Cordova - Plugin - Geolocation API	4
2.3.3	Android Location API	4
2.4	Discussion	4
2.5	Conclusion	4
3	Event Data Display	4
3.1	Overview	4
3.2	Criteria	4
3.3	Potential Choices	5
3.3.1	Google Map API	5
3.3.2	MapBox	5
3.3.3	Microsoft Bing Map	5
3.4	Discussion	5
3.5	Conclusion	5
	References	6

1 SOCIAL MEDIA INTEGRATION

1.1 Overview

Integrating a social media aspect such as twitter and Instagram photos from an event into our application will allow the user to view what others have done at those particular events and also entice them into exploring them. Having tweets and photos posted on that specific event will be crucial to keeping that events page organized.

1.2 Criteria

One of the main features to this application is to allow the users to post and also view others experiences from that event. Each events information page will list photos and tweets from the most recent posted. By giving each event a specific hashtag, users will be able to upload their photos and tweets to the correct events. On the back end, we will implement a filter that searches for specific locations and event hashtags so that the user can easily navigate and view others posts.

1.3 Potential Choices

1.3.1 Instagram API

Instagram's API will allow us to post Instagram photos or a photostream on our mobile application from any personal account. This feature will also create a gallery of images that will automatically update as new pictures are added. The process of implementing these features into our application will first start off by setting up the library, followed by obtaining the web API keys, configuring the login page, building the API output, and lastly dumping the data out. [1]

1.3.2 Twitter API

Twitter Kit SDK will allow us to display Twitter content onto our mobile application. Below will be a list of kits we can used to implement into our application.

- TwitterCore - Let users log in with Twitter via Single Sign-On. Make authenticated requests to the Twitter API to load Tweets, users, search results, and other Twitter content. [2]
- TweetUI - Embed media-forward or compact Tweet views into your views and list views to show users information that is relevant to your app. Let users share Tweets discovered via your app. [2]
- TweetComposer - Let users compose new Tweets as part of your app or share content discovered via your app. [2]

For our feature, we will most likely be incorporating the TweetUI kit in order to display content into our application. Within TweetUI we will be using the tweetView and CompactTweetView to render our tweets. The Tweets will be requested through the Tweet API or TweetUtils which will cache recent requests. From there we will be able to display the correct content to our application. [3]

1.3.3 Facebook API

Facebook provides its own SDK to integrate with our Android application. There are three different APIs that perform different functions. We will need to first get an Access token to access those APIs. Reviewing the developer's page on access tokens, we will be using the user access token and page access token. These functionalities will be provided below.

- User access token - this kind of access token is needed any time the app calls an API to read, modify or write a specific person's Facebook data on their behalf. [4]
- Page Access Token - This token is very similar to the User access token however, it provides permission to APIs that read, write or modify the data belonging to a Facebook Page. [4]

To obtain a page access token, we will first obtain the user access token and get the manage pages permissions. Once that is achieved, we will get the page access token or the Graph API which will allow us to make API calls on those pages.

1.4 Discussion

All three of these APIs can be implemented into our application. The features that these APIs provide are all very similar, however the information that they display is a little different. Facebook's API seems to be the middleman of Instagram and Twitter. It will be able to display photos and status of users, where Instagram only posts photos, and Twitter only postes statuses.

1.5 Conclusion

While all three APIs provide great features to our application, we will first be implementing Instagram's API. By providing a feed of photos in each event, other users could be encouraged to attend that event.

2 USER ACTION - VERIFICATION (GEOLOCATION)

2.1 Overview

Retrieving the geolocation (current location) for our application is a critical tool to our mobile application. By obtaining a user's current location, we will allow the user to see various events that are happening around them. After reviewing a few potential choices, I came up with the Google Maps API, Cordova-Plugin-Geolocation API, and Android Location API.

2.2 Criteria

The user will be able to turn on their location services which will allow them to view the events that are happening around them. By clicking on an event, the user will be prompted if they would like directions to that event. If so, this feature will direct the user to their event in the fastest way.

2.3 Potential Choices

2.3.1 Google Map API

Google Maps API is a useful tool for retrieving the current location of an individual. Provided on the Google Maps API website is a tutorial on how to display the geographic location of a user or device on Google Maps using the browser's HTML5 Geolocation feature along with the Google Maps JavaScript API. This website also provides example code and comments on how it works, making this a very useful tool. Both Android and iOS Google Maps APIs require the app to prompt the user for consent to use their location services. [5]

2.3.2 Cordova - Plugin - Geolocation API

This API provides the location information of any device in the form of its latitude and longitude. According to Cordova's website, Common sources of location information include Global Position System (GPS) and location inferred from network signals such as IP address, WiFi, and Bluetooth MAC addresses. A quick note is that the API is based on the W3C geolocation API specification and only executes on devices that don't already provide an implementation. [6]

2.3.3 Android Location API

Within the Android API, you can call `android.location.LocationListener`. There are three main components of the API. The `Location` class gives geographic location, including the latitude and longitude. `LocationManager` provides access to the system location services. And lastly, `LocationListener` is used for receiving notifications from the `LocationManager` when the location has changed. [7]

2.4 Discussion

All three APIs can be incorporated into our Android application. They all can retrieve the location of a mobile device. Cordova, on the other hand, can retrieve a few more features such as the IP address, WiFi, and Bluetooth MAC addresses. These features are interesting but not what we are looking for. Android Locations API has a built function within Android Studio making it very simple to implement a function to retrieve the location. However, according to the Android developer's page, it recommends to use the Google Location Service API.

2.5 Conclusion

Even though all three location service APIs work for our application, we will be using the Google Location Services API. The Google Play Services provides a more powerful, high-level framework that automatically handles location providers, user movement, and location accuracy. It also handles location update schedule based on power consumption parameters we provide. In most cases, you'll get better battery performance, as well as more appropriate accuracy, by using the Location Services API.

3 EVENT DATA DISPLAY

3.1 Overview

This application will also serve as a platform to highlight useful resources around Corvallis. By implementing two maps into our application we will be able to fulfill this requirement. The first map will showcase notable establishments which consist of activities and entertainment, grocery stores, restaurants, shopping centers and city offices. The second map will display events approved by the Corvallis Community Relations office (CCR) which students will be able to accumulate stamps from. Having separate maps will allow the user to easily navigate to local events and useful establishments.

3.2 Criteria

Students will be able to access two maps which display useful resources around Corvallis and events that are approved by the CCR. We will incorporate a sidebar on the first map (useful resources) that filters out establishments by type which will help students narrow down their searches.

3.3 Potential Choices

3.3.1 Google Map API

Google Maps API provides a feature called markers. A marker identifies a location on a map and can display a custom image of that location. According to the Google Maps marker developers page, markers are designed to be interactive. By default they receive click events, so you can add an event listener to bring up an information window displaying custom information. Another feature of this API allows the user to remove a marker from the map. Under the title, adding a Map with a marker on Google's website explains the process on how to implement markers or events on the map. The website provides sample code for us to implement into our application. [8]

3.3.2 MapBox

MapBox is very similar to Google Maps on displaying various icons on a map. However, you can customize every aspect of the map from the colors, hiding or showing specific layers, to choosing which information to present on the map, all while the users are interacting with the map. The only downside to MapBox is that there is a fee after so many views and requests to the API. [9]

3.3.3 Microsoft Bing Map

The Microsoft Bing Maps platform provides many controls and service APIs for our application. A list of them will be provided below from their website.

- Bing Maps V8 Web Control - the latest Bing Maps JavaScript API. Combine the AJAX map control with the Bing Maps REST Services and the Bing Spatial Data Services to create powerful Web sites and mobile applications with the latest imagery and location functionality. [10]
- Bing Maps REST Services - perform tasks such as creating a map with pushpins, geocoding an address, retrieving imagery metadata or calculating a route. [10]
- Bing Maps WPF Control - The Bing Maps WPF Control SDK lets developers integrate Bing Maps into rich Windows Presentation Foundation (WPF) applications. [10]

In a nutshell, Bing Maps provides many API features and platforms that can be incorporated into our application. All of the API keys are also provided on their website for easy access.

3.4 Discussion

Reviewing the three APIs, they all have their ups and downs. All three of the APIs provide a map in which we can modify to our own specifications. However, MapBox has a fee after so many requests to its API of which is a huge downfall. On the other hand, Google Maps and Bing's API are very promising for the features they provide. Implementing Bing's API may be a challenge, on the other hand, Google Maps API seems very reasonable as it provides example code and good explanations on how to apply it to our application.

3.5 Conclusion

Although all three APIs provide great features, we will be using Google Maps API as our source to display events to our application. The Google Developers manual for this API provides detailed information on how to implement their code into our application.

REFERENCES

- [1] Connecting with api data from instagr.am using php. [Online]. Available: <http://spyrestudios.com/connecting-with-api-data-from-instagr-am-using-php/>
- [2] Twitter kit for android. [Online]. Available: <https://dev.twitter.com/twitterkit/android/overview>
- [3] Show tweets. [Online]. Available: <https://dev.twitter.com/twitterkit/android/show-tweets>
- [4] Facebook access tokens. [Online]. Available: <https://developers.facebook.com/docs/facebook-login/access-tokens/#pagetokens>
- [5] The google maps geolocation api. [Online]. Available: <https://developers.facebook.com/docs/facebook-login/access-tokens/#pagetokens>
- [6] cordova-plugin-geolocation. [Online]. Available: <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-geolocation/>
- [7] Android location and maps. [Online]. Available: <https://developer.android.com/guide/topics/location/index.html>
- [8] Displaying marker and evnts google map. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/markers>
- [9] First steps with the mapbox android sdk. [Online]. Available: <https://www.mapbox.com/help/first-steps-android-sdk/>
- [10] Mocrosoft bing maps. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff428643.aspx>

I Heart Corvallis Technology Review

Prepared by Dylan Tomlinson of Group 14

CS Senior Capstone, Fall 2017

June 4, 2018

Abstract

This document covers three pieces of the I Heart Corvallis project and the technologies needed to create these pieces. The pros and cons of each technology will be discussed for each piece, and one technology will be chosen for each piece.

Contents

1	Purpose	3
2	User Security	4
2.1	Overview	4
2.2	Criteria	4
2.2.1	Technology 1: SHA-3	4
2.2.2	Technology 2: PBKDF2	4
2.2.3	Technology 3: Argon2	5
2.3	Discussion	5
2.4	Selected Technology	5
3	User Interface Scaling	6
3.1	Overview	6
3.2	Criteria	6
3.2.1	Technology 1: Scalable Layout	6
3.2.2	Technology 2: Android SDK	6
3.2.3	Technology 3: Semantics UI	7
3.3	Discussion	7
3.4	Selected Technology	7
4	Application Testing	8
4.1	Overview	8
4.2	Criteria	8
4.3	Potential Choices	8
4.3.1	Technology 1: GenyMotion	8
4.3.2	Technology 2: Android SDK Emulator	8
4.3.3	Technology 3: Visual Studio Android Emulator	9
4.4	Discussion	9
4.5	Conclusion	9
5	Bibliography	10

1 Purpose

In this document we will be going over in detail three pieces of the I Heart Corvallis project, User Security, Application Scaling, and Application Testing. We will then discuss several technologies we can use to implement them. For each piece, we will list three different technologies and we will analyze the pros and cons of each technology. We will then pick one of the three technologies, indicating why that decision was made.

2 User Security

In the I Heart Corvallis application, users who are not students of OSU will need to make accounts before accessing the full functionality of the application. These accounts will require a unique email address and a password. We are obligated to protect the user's passwords as it has been shown that most users will repeat their passwords for different and possibly more important accounts.

2.1 Overview

Our goal is to require a third party expend a certain amount of resources in order to obtain our user's passwords. In order to do this we will be implementing a hashing algorithm. This will keep the user passwords more secure, as with hashing algorithms the passwords themselves will never be stored in the database. Instead they will be replaced with the resulting, irreversible hash [1].

2.2 Criteria

We will be deciding which technology to use based upon the programming difficulty, it's effectiveness, and the memory required to store the hashes.

2.2.1 Technology 1: SHA-3

Secure Hash Algorithm 3 is a hashing algorithm that utilizes sponge construction. Which is a type of algorithm with finite internal states that take an input bit stream of a given length and give an output bit stream of a that length. It is easy to implement and can be implemented in any of the languages we are using for this application. SHA-3 is a fast hashing algorithm, so it is somewhat susceptible to brute force attacks [3].

2.2.2 Technology 2: PBKDF2

Password-Based Key Derivation Function 2 is a hash function aimed to reduce the encrypted keys vulnerability to brute force attacks. It was originally written in 2000, but is still recommended for password hashing by RFC 8018 in 2017. PBKDF2 has many different implementations, including Javascript, PHP, Java, and C. So Regardless of our design decisions for the rest of the application, we will still be able to use PBKDF2. It is rather easy to implement as well. One of it's weaknesses is that brute force attacks using a GPU can be done somewhat cheaply [3].

2.2.3 Technology 3: Argon2

Argon2 is a key derivation function designed in 2015. It has three different versions, argon2d, argon2i, and argon2id. Each of these has a different strength and are specified by three common parameters. These are execution time, memory required, and degree of parallelism [2].

2.3 Discussion

The Argon2 hash is the most secure of the three hashes. In contrast, it also requires the most work to implement and the most memory to store the hashes. The SHA-3 hash requires little implementation and a small amount of space to store the hashes. Compared to Argon2 and PBKDF2, SHA-3 not nearly as effective at creating hashes as it's hashes are vulnerable to brute force attacks [3].

2.4 Selected Technology

We chose Choice 2, PBKDF2, because it hits the exact middle ground of what we need for this project. Our audience for this application consists of users in the Corvallis area, so we do not expect the application to be used or even known outside of this area. The argon2 hashing algorithm is a little overkill for our needs, and in contrast SHA-3 has been proven to be crackable with too little effort. We currently do not know our resources regarding database storage, but from what we have gathered so far we do not have much to work with.

3 User Interface Scaling

Our application is going to be available on android platforms. This means that we have to deal with several different types of mobile devices with different screen sizes and resolutions. We will need an easy way to ensure that our application retains its quality regardless of the phone being used. There are several technologies that aid in this, and we are going to choose one to implement.

3.1 Overview

Our goal is to have the I Heart Corvallis application be easily view-able on most android phones built in the last few years. We want this goal to be achieved rather fluently without having to create ten different versions of our user interface.

3.2 Criteria

Our criteria for this piece is its ease of implementation, it's effectiveness, and how well it meshes with our other user interface technologies.

3.2.1 Technology 1: Scalable Layout

An Independent android library released on Github that allows for easy scaling of the user interface. It includes measurements for objects that use relational units rather than pixels or dpi [4]. This ensures that objects that are created will take up the same percentage of the space on larger or smaller screens. It is implemented in Java and XML. There are no tutorials on how to use it so learning it will have to be trial and error.

3.2.2 Technology 2: Android SDK

The Android Software Development Kit consists of development tools, an emulator, required libraries to build android applications, and lots of sample projects with source code. It is one of the more popular development kits for android platforms. The kit itself already has libraries with several ways to build an interface with scaling capabilities [5]. It requires a little more knowledge of the android language and will require a lot of tinkering. But there is a vast amount of walk-throughs, guides, and examples of other projects using these libraries.

3.2.3 Technology 3: Semantics UI

Semantic UI is a html5 based framework that is powered by jQuery. It is used for developing Android apps as well as websites. It focuses on using semantics based commands, which makes its syntax very easy to understand [6]. In order to utilize Semantic UI you must install nodeJS and gulp, which increases the overhead of the framework. It is used in several applications such as Snap Chat. Its downsides include a strange build system and large CSS files with few ways to reduce their size [6].

3.3 Discussion

Scalable layout has a few great examples of how easy scaling can be done in similar applications. In contrast it has very little documentation so we would have little outside help with any problems we may run into. Semantic UI has the benefit of being used commercially in many different applications and websites. There is a great deal of documentation as well. A downside to it though is the huge CSS file, it could prove to be difficult to navigate through it to make a single change. The Android SDK has the benefit of being supported by Google with more documentation than we will need. The downside to it is it becomes difficult to separate your application from other applications using the base kit without extensive knowledge of the system.

3.4 Selected Technology

We chose the Semantic UI largely because we like the way the project can look utilizing this technology rather than using the Scalable layout or the Android SDK. We will also be using Semantic UI for the rest of our user interface design, so that will keep compatibility problems to a minimum.

4 Application Testing

In order to determine that a requirement has been completed, we have to be able to test that requirement's functionality. There are many different technologies that will aid us by allowing us to emulate running the application on an android phone.

4.1 Overview

Our goal here is to find a technology that can allow us to easily simulate and test our application on a variety of devices. The Android market is filled with many different types of mobile devices, and we have to ensure that our application works on the majority of these devices.

4.2 Criteria

We will be determining our choice in technology upon three ideals. The technology's effectiveness, the technology's ease of use, and the price of the technology.

4.3 Potential Choices

4.3.1 Technology 1: GenyMotion

GenyMotion is a multi-platform Android emulator. It can emulate most Android devices and supports older versions of Android as well. GenyMotion allows you to simply drag and drop an application (.apk file) to start running it on your emulator [8]. It utilizes the OpenGL capable GPU in your desktop, so even if the run times are not the same it takes a substantially less amount of time to emulate your app [7]. It also has a very simple install process.

4.3.2 Technology 2: Android SDK Emulator

The Android Software Development kit mentioned earlier, contains an android emulator. This emulator allows you to simulate many different android devices including their screen sizes, hardware characteristics, and android version. Utilizing this technology would allow us to easily test our program on any android device we need. [9]

4.3.3 Technology 3: Visual Studio Android Emulator

The Visual Studio Android Emulator can be used either through the Visual Studio program or as a stand alone emulator. It shares many similarities with the other SDKs. It can simulate many different Android devices including their screen sizes and hardware characteristics [10].

4.4 Discussion

Each of these technologies have their advantages and disadvantages. GenyMotion is very easy to use with it's drag and drop concept, and it runs much faster than the other two choices. In contrast to the other two technologies which are free, Genymotion requires a subscription cost of 119 dollars per year. In contrast, the Visual Studio Android Emulator is only slightly faster than the Android SDK emulator. It also has the benefit of having a usable stand alone version, or it can be implemented inside of Visual Studios itself. It is also compatible with the Xamarin IDE, which we may decide to use.

4.5 Conclusion

In conclusion, we chose the Android SDK emulator due to it's vast documentation and it's ability to emulate most devices with most versions of android. We also chose it because unlike GenyMotion, the Android SDK emulator is free.

5 Bibliography

[1] Coates, Michael, "Safely Storing User Passwords: Hashing vs. Encrypting", Dark Reading, 2014, <https://www.darkreading.com/safely-storing-user-passwords-hashing-vs-encrypting/a/d-id/1269374>

[2] Dinu, Khovratovich, "phc-winner-argon2", GitHub, 2017, <https://github.com/P-H-C/phc-winner-argon2>

[3] Rietta, Frank, "Use Bcrypt or Scrypt Instead of SHA* for Your Passwords, Please!", Rietta, 2016, <https://rietta.com/blog/2016/02/05/bcrypt-not-sha-for-passwords/>

[4] Kim, Youngmann. "Scalable Layout for Androi", Github, 2017, <https://github.com/ssomai/ScalableLayout>

[5] "Supporting Multiple Screens", Developer.Android, 2017,

[6] Gerchev, Ivaylo, "Introducing: Semantic UI Component Library", Site Point, 2014

[7] Hindy, Joe, "15 best Android emulators for PC and Mac of 2017" Android Authority, 2017, <https://www.androidauthority.com/best-android-emulators-for-pc-655308/>

[8] Favaro, Filippo, "Genymotion: the best Android emulator for app development" Android Pub, 2015, <https://android.jlelse.eu/genymotion-the-best-android-emulator-for-app-development-24f5c8933768>

[9] <https://developer.android.com/studio/run/emulator.html>

[10] Gorski, Edmund, "Using the Visual Studio Emulator with Android Studio", Clearly Agile Inc, 2017, <https://www.clearlyagileinc.com/blog/using-the-visual-studio-android-emulator-with-android-studio>

5 WEEKLY BLOG POSTS

5.1 Omeed

5.1.1 *Fall 2017*

5.1.1.1 Week 1: This week I thought about which projects I wanted to work on, and I submitted my project preferences. I also spoke with Bradley about working on the I Heart Corvallis mobile app project, contacted Lyndi-Rae Elisabeth Petty (the project's client), and got approved to work on that project.

5.1.1.2 Week 2: This week we were officially assigned to our project groups. Brad emailed our client our contact information and asked for good meeting times for next week. When she responded with potential meeting times for next week, I initiated discussion with my groupmates to figure out when would be the best time to meet with her, and when we figured out the best time, Brad let our client know, and she confirmed that time with us. I also met with the full group for the first time after class on Thursday.

Next week we will meet with our client planned for next Wednesday (October 11) at 2pm to have a deeper discussion about what the project entails.

5.1.1.3 Week 3: This week I wrote the rough draft of the problem statement for my project. At that point, we hadn't met with our clients yet so I wasn't really able to base much of the rough draft on the client's explanation of the project. Now that we've met with our clients, we'll be able to write a more complete and thorough final draft of the problem statement.

On Wednesday we met with our clients for the first time. They told us their vision for the app and their timeline for completing various aspects of the app, and we had a conversation about the app's design and features, as well as some of what we'll need from them. Hopefully soon they'll nail down exactly what they want in the app so we can put these specifications in our design document.

We met with our TA for the first time this week, and we're going to meet with him at the same time (Wednesdays at 3:20pm) every week this term.

I also started making some drawings of what the app's pages could look like. Next week I'm going to share my drawings with the rest of the team, and they'll share any drawings they make as well.

I also found some good software that could help make it easier to implement the app for both Android and iOS. Since not all languages work well on both Android and iOS, it's important that we're able to find a way to implement the app in such a way that we don't have to implement it completely differently (such as in a completely different language) for Android than for iOS. I found Xamarin (which is in Visual Studio) and it will allow for cross-platform app implementation in C#. I'll continue to look into the best options for easy cross-platform development, since coding

in different languages for Android and iOS would significantly increase the amount of time it takes to complete this project.

5.1.1.4 Week 4: This week we focused on completing and turning in our Problem Statement. On Monday the three of us got together at the library to write the final draft of the Problem Statement. We got most of it finished on Monday, and on Tuesday we went to Kirsten's office hours and the Writing and Research Center to receive feedback on our writing. I posted our final draft to OneNote and emailed Lyndi our completed final draft on Thursday. Kirsten gave us more feedback on OneNote, and Brad and I got together to figure out how to take her of her recommended fixes. I completed the fixes and reposted the final draft on OneNote and re-emailed it to Lyndi early Friday morning.

Next week we plan to write the rough draft of the requirements document, and we will meet with Lyndi and Jon (our clients) again soon to continue talks on the application.

5.1.1.5 Week 5: This week we scheduled another meeting with Lyndi to discuss the next steps for the planning of the project. We now have a tentative meeting time with her each week (Wednesdays at 2pm). I showed her the Photoshop app mockups I made of the initial login screen and the homepage, and we told her we can come up with more designs for the app so she can get an idea of what the final version of the app could look like. We also informed her that depending on whether we take the route of implementing the app in a cross-platform fashion or we take the route of building the Android app in Android Studio and the iOS app in Xcode, we could potentially run into a time constraint, as the latter option would essentially require us to make two separate apps, since we would finish the Android app first and then the iOS app. She emailed us some resources to help us figure out what would be the best option, but she was receptive to and accepting of the fact that we may potentially not have enough time to code both versions of the app.

We also wrote the rough draft of our Requirements Document this week. We began compiling information for the document on Thursday, and we met in the library Friday afternoon to continue working on the rough draft. I finished the rough draft Friday night. We plan to use the feedback we receive to make a more complete Requirements Document for next week.

5.1.1.6 Week 6: This week I met with Lyndi along with the rest of our group. We have pretty much nailed down a weekly meeting time of Wednesdays at 2pm. We discussed the focus group that was happening later that day, as well as more app design concepts, the requirements document, and cross-platform implementation and keeping the iOS version of the app a stretch goal. We also met with our TA. We recorded a lot of useful information based on the discussions that occurred in the focus group that day.

I also worked with the other team members to complete the final draft of the requirements document and send it to Lyndi. She emailed back with a couple of small corrections, and I made those corrections and resent it to her.

She emailed Kevin and Kirsten on Friday letting them know that she received and approved the final draft of the requirements document.

Next week I will work on completing the rough draft of the technology review document, and next Wednesday (11/8) our group will discuss the focus group discussion that occurred this past Wednesday.

5.1.1.7 Week 7: This week we met with Lyndi on Wednesday to reflect on last week's focus group and the feedback we were given by the focus group attendees. There's another focus group scheduled for next Tuesday, and we'll use the information we've compiled from both focus group sessions to make more concrete decisions on what we're going to include in the app. We've been able to nail down much of what the app will entail already, though. We also met with our TA on Wednesday. That same day I created a Google Doc with a table that we will fill with topics for the Tech Review document and decide who will cover each topic. We were originally planning to meet this week to fill out the table and figure out who will cover what topics, but that has been pushed back to this weekend.

This weekend and next week we will write our Tech Review rough draft and take a look at what feedback we receive at the second focus group session that's scheduled for Tuesday.

5.1.1.8 Week 8: This week was a pretty light week for our group. I completed the rough draft of my Tech Review document on Tuesday. Our client held another focus group for the I Heart Corvallis initiative on Tuesday. We didn't meet with her this week, as there wasn't a lot of new things to discuss this week, but during our next meeting with her, we will go over the feedback given in this week's focus group. We also met with our TA this week. This weekend and next week I will complete my final draft of the Tech Review document and work on the rough draft of the Design Document so I can receive feedback on it. Our next meeting time with both our client and our TA are not completely determined yet, either, as Thanksgiving is next week. We will communicate with both our client and our TA to figure out what the plan is for next week.

5.1.1.9 Week 9: This week was a short and light one. I completed the final draft of my Tech Review document on Tuesday. On Thursday, I made a Google Doc for the team containing the template for the Design Document. We will work on the Design Document next week. We also plan to meet with Lyndi next week to get back up to speed (since we haven't met since November 8) and discuss the feedback she received at the last focus group.

5.1.1.10 Week 10: This week we completed the final draft of the Design Document. I went to Kirsten's office hours twice with Bradley to get some more clarification on the composition of the Design Document. Once we completed the final draft, I emailed it to Lyndi so she could give us any feedback she had and approve it if it looked good. She approved the final draft and emailed Kevin and Kirsten letting them know she had received and approved it. I've

already started the Fall Term Progress Report, and next week will complete it and make the corresponding presentation. Over Winter Break, we plan to begin implementation of the application.

5.1.2 *Winter 2018*

5.1.2.1 Week 1: Over Winter Break we got a lot of the basic app layout and page creation completed, and we started implementing various layouts and attributes to each page. We met with Lyndi this week to discuss our progress on the app, what both our and their timelines look like for this term and next term, and what we're currently working on with the app. We also set up weekly group work times so we can have set times dedicated to working on the app as a group. Next week we will continue working on creating our MySQL databases and connecting the app to the databases, as well as implementing more layouts and attributes to the various pages of the app.

5.1.2.2 Week 2: Over Winter Break we got a lot of the basic app layout and page creation completed, and we started implementing various layouts and attributes to each page. We met with Lyndi this week to discuss our progress on the app, what both our and their timelines look like for this term and next term, and what we're currently working on with the app. We also set up weekly group work times so we can have set times dedicated to working on the app as a group. Next week we will continue working on creating our MySQL databases and connecting the app to the databases, as well as implementing more layouts and attributes to the various pages of the app.

5.1.2.3 Week 3: This week was a very productive week for us. After working for quite a bit of time to connect the app to our database, on Wednesday I finally got the app connected to my PHP scripts that communicate with the database. Once I got that working for our non-student login page, I connected it to our signup page, as well as other pages that will be behind the authentication wall, such as the events page, leaderboard, and resource map. We met with Lyndi on Thursday to show the progress we've made in the past week, and during this meeting we all went to meet with Derek Whiteside of Web and Mobile Services to discuss storage location options for our database, as well as other important mobile application topics, such as security, app branding, CAS (the ONID login), and app lifecycle/maintenance. Next week we will continue integrating our app with our database, and we will begin work on our administrative interface, which Lyndi told us yesterday she would prefer to be in the form of a website rather than an in-app interface.

5.1.2.4 Week 4: This week was a bit lighter of a work week, as we had a lot of work in other classes to do this week. We mostly worked on implementing the administrative interface, which our client told us last week she wanted to be in the form of a website instead of an in-app interface. Bradley initialized the website, while I worked on styling it and furthering its functionality. I integrated Semantic UI to the site, which made it more visually pleasing. I also added some more specific columns to our event database, such as address, latitude, and longitude. We also went through the details of our project to determine what we've completed so far and what we have left to do, and we divided up the

remaining pieces among the three of us. Next week we will start to work more heavily on our respective pieces of the projects so we can have a Beta version of the app ready by the end of the term.

5.1.2.5 Week 5: This week was a light work week for me as I had midterms in other classes on Wednesday and Friday this week. Nonetheless, I did put in some work on the administrative website. I got most of the Manage Events page working, and I created a home page (index.html) that has quick links to a bunch of other admin pages. We also had our weekly client meeting with Lyndi to discuss our progress the past week and next steps for the project. Dylan got the user session implemented so that the app remembers who's logged in, even when the app is closed. This weekend and next week we plan to create a draft of our Expo poster, as well as create and submit our midterm progress report and presentation. I will also be able to pick up the pace on my share of the work next week as I've now gotten both midterms out of the way.

5.1.2.6 Week 6: This week we got a lot of work done on our app and website, and we also put a decent amount of time aside to work on the midterm progress report and presentation, as well as the draft of our Expo poster. We recorded our presentation and worked on the Expo poster draft on Sunday, and I put together the pieces and finished the video the same day. This week I also set up the event maps, set up the event management webpage, and set up the prize addition and management webpages for our administrative website. We finished the Expo poster draft on Wednesday, and Dylan submitted it the same day. We completed and submitted the midterm progress report Friday afternoon. Now that all of our midterms are out of the way (Brad and Dylan both had midterms this week), next week we're going to focus on working towards finishing the website, as well as implementing geolocation services and the event PIN verification page. We're also going to have to start looking at options for third-party database services to store our data tables long-term.

5.1.2.7 Week 7: This week we got some important parts of the application implemented. Dylan set up password hashing so that account passwords aren't stored as plaintext in the database, but are rather hashed using the PBKDF2 hashing algorithm. He also started implementing PHP prepared statements into our PHP scripts to protect them from MySQL injections. Brad got geolocation services set up for the event map. I replaced AsyncTasks in several activities to AsyncTaskLoaders (and put these loaders in their own pages instead of leaving them as anonymous classes in the activity). This will allow for both less resources being used when performing activity tasks and caching of retrieved data, both of which increase app efficiency. Our main problem this week was that our client recently found out that her position will not be retained after this school year, so we'll need to figure out if that will have any implications for our project. We will look into that situation more next week. Next week we plan to implement features such as geofencing, more AsyncTaskLoaders for activities, updating the completed events database upon entering the correct PIN for an event, and implementing more pages for the administrative website.

5.1.2.8 Week 8: This week we focused on fixing our PHP prepared statements, implementing more AsyncTaskLoaders (to replace AsyncTasks) for our various app pages, and adding the ability for our client to add and manage resource page information and resource map markers. On Wednesday, I implemented the AsyncTaskLoaders, and on Thursday I implemented the resource management pages on the administrative website. We've been fairly consistently looking at our PHP prepared statements, since they'll be important in preventing hackers from messing with our database. However, most of them are not yet working properly, so we're still looking into what's wrong with them and how to fix those statements. We'll continue examining those errors next week. This week we also found out that our project is not in jeopardy despite the uncertain status of our client's position for next year, which is definitely a good sign. Next week we'll continue working on our administrative website and add some of the final remaining pieces into our application.

5.1.2.9 Week 9: This week was a critical week in the progress of our project. We now have a near-complete website, including the ability to log in and out of the site and putting the webpages behind a login wall so users have to log in before accessing any of the webpages. This allows our client to control who can edit in-app content, as they'll be able to authenticate new administrative users after logging into the website themselves. I also enhanced the resource map this week, as the legend is now collapsible and the user can now restrict the map to only show a specific type of resource. I also initialized my final progress report for this term.

One important bug that I ran into on Thursday is that the user's information is not always passed between pages, and the app crashes when trying to log the user's ID as a result. It's not a consistent issue, and it only happens sometimes after officially completing an event and returning to the Dashboard. We'll be looking into that error and figuring out what's causing it in the near future, as it will likely become an annoying issue for the user if it's not resolved.

5.1.2.10 Week 10: This week was largely spent on refining various aspects of the app and patching bugs. On Sunday we set up geofencing for event attendance verification. Now, when a user wants to check into an event, they can only do so if they are within range of the event (within 0.1 miles). On Tuesday, I patched a bug where user information wouldn't always get passed around the app and the app would crash when trying to access user information. On Wednesday I added an end date/time for each event and reflected this change on both the website and in the application. And on Thursday I corrected the hex color code for the official Beaver Orange color in our app. We also recorded our final progress report presentation for the term on Thursday. Over the weekend and into next week I'll write my final progress report for the term and prepare the presentation video. We're also meeting (along with our client) with the systems administrator of Identity & Access Management at OSU to discuss integrated CAS and Single Sign On into our application for the student log-in.

5.1.3 Spring 2018

5.1.3.1 Week 1: This week we got a lot of smaller miscellaneous things done. We set up weekly meeting times with Lyndi (our client) and emailed Andrew (our TA) to schedule weekly TA meetings. I created the survey over Spring Break, and got it fully connected to our app and database throughout the week. On Wednesday I changed the "age" column of the user database to "birthdate" and made it a datetime column to better handle user ages, and I added a date picker for the birthdate to the Settings page. This week I also started editing the app's style to fit OSU branding guidelines. I added branding-approved colors, fonts, icons, and a background image. On Wednesday Dylan introduced the idea of changing the format of the survey response table to allow for the ability to change the number of survey questions, and I recreated that table and restructured the survey response loader and corresponding PHP script to reflect this change. Dylan found a signup bug that caused the signup PHP script to not work and a FileNotFoundException to be thrown, meaning new accounts could not be created. On Friday, Dylan and I successfully fixed this bug, and users can once again sign up for the app. Next week we plan to begin implementation of the CAS login to allow students to sign up and log in with their ONID accounts. We will also continue fixing bugs that we find.

5.1.3.2 Week 2: Due to grad school visits and planning, this week was a lighter week for me. However, Brad and Dylan did a good job of helping out with implementation this week. I've been working on enhancing the student survey to get more information from the user upon signup, such as grade, birth date, and user type (domestic student, international student, resident, visitor, etc.). I also re-added the birthdate preference to the Settings fragment for all users (I had previously made it only shown for non-student users, such as residents or visitors).

We also began implementing the CAS login system for OSU students. We ran into some issues where we would get a Stale Request when trying to use the ONID login, and we're stuck on how to get the user's data back down to the application after validating their ticket. We had our client email Derek Whiteside at Web Services for help directing us to someone who could help with that. Next week we'll continue implementing the CAS login, and I'll continue updating the survey. We'll begin working on adding the ability to add custom images for events soon as well.

5.1.3.3 Week 3: This week was a big week for us. We finally got image upload working for events and resources, so now newly added events and resources can have custom images associated with them. I also implemented the ability for the user to choose, take, and change a profile picture for their account. The profile picture is unique to the device the app is installed on, as it's stored in a local database on the device. I also added indicators for faculty users of the app. This addition can be seen in the page where we ask the user for more basic information, as well as in the Settings page, where you can change your grade and user type preference among other account attributes. I also implemented prepared statements for nearly every PHP script for the app and website. This will really help secure our tables against PHP injection attacks.

We reached out to Derek Whiteside at Web Services, since we've ran into a wall integrating CAS login for OSU students and faculty. He has not responded since our client emailed him, and we're beginning to run out of time to implement

the feature, so we may ditch CAS integration in favor of one universal login system. Our client is okay with us doing this if we cannot get CAS integrated.

Next week we will patch a couple of remaining bugs and move on to building the demo version of our app, since we will not have reliable Internet access at the Engineering Expo.

5.1.3.4 Week 4: This week we focused on patching some final bugs and implementing the remaining features of the app and website. I fixed the form validation bug on the website and implemented more prepared statements in the PHP scripts. On Wednesday I got the app permissions working, so the app now asks the user for permission to use location services, the camera, and the photo gallery on the device. The former permission is for the event map and event check-in, while the latter two permissions are for the profile picture.

On Thursday we finalized with our client that we're not implementing the CAS login system for OSU students and faculty. We had run into a wall implementing it, and when our client reached out to Derek Whiteside of Web Services for guidance, we did not hear back from him. We are instead implementing a universal login system that will ask the user for their student ID number and ONID username if they are an OSU student or faculty member. Our client has confirmed that she is okay with us keeping track of students this way.

On Friday I fixed the bitmap memory bug by replacing the bitmaps with Picasso load statements that load the file directly into the ImageView. I also got the event images to load into the ImageView on the detailed event page on Friday, and I did some more work on giving our client the ability to change event and resource images via the edit pages on the website that day. I also started working on giving our client the ability to choose a custom image for the About page and the ability to change that image via the Edit About Page page on the website. Next week we will focus on finishing up the implementation of changing the event and resource images, and then we'll begin working on the demo version of the app for the Expo.

5.1.3.5 Week 5: This week was an important week for us. We managed to patch up some important bugs and fill some loose ends of the application, and we now have a near-final product. We also nearly completed implementing the demo version of the application. We started the midterm report and the script for the midterm presentation on Monday, filmed and recorded the presentation on Tuesday, and completed and submitted both the report and presentation on Wednesday. We also submitted our Expo poster for printing on Monday.

Next week we will focus on recording the walkthrough video for the website, since we won't have a reliable Internet connection at the Expo. Other than that and patching/implementing any remaining loose ends in the application and website, we may be able to take it a bit easier leading up to Expo, since we're pretty close to being ready.

5.1.3.6 Week 6: This week was an extremely light week for us, as our project was near-ready for Expo coming into the week. On Wednesday we recorded the walkthrough video of the website for Expo and fixed a couple of final bugs, one on the event page of the app and one on the home page of the website. Next week we'll be able to take it easy and get ready for Expo!

5.1.3.7 Week 7: This week was a very light week for the project, as we implemented the demo version of the app last week. I made a quick fix on Friday to make the detailed event page scrollable, and we presented at Expo on Friday. Next week we'll begin looking at the remaining assignments for the course and handing the final product over to our client.

5.1.3.8 Week 8: This week was a pretty light week for me as far as the senior project is concerned, since the implementation is already done. We started working on the Final Report on Thursday, and we're going to continue working on the Final Report and Presentation next week and the week after. We're also going to prepare the app, website, and code for handoff to our client in the coming weeks, as well as the group evaluations.

5.1.3.9 Week 9: This week a lot of my time was taken up by my Honors thesis and homework for other courses, so I did not do any Capstone work this week. Next week we will complete both the final presentation and final report, and I will submit the group evaluation next week.

5.1.3.10 Week 10: This week we focused on finishing up the final tasks for the project. On Sunday I submitted my group evaluation, and on Monday and Wednesday we put in more work on the Final Report. On Wednesday we recorded our Final Presentation, and I edited and submitted it that same day. On Wednesday and Thursday I added documentation comments to the code for the administrative website and scripts, and on Friday I added comments to the PHP scripts for the mobile application. On Friday we also officially handed off our project code to our client by emailing her the link to our code on GitHub and giving instructions on how to download the code. This weekend and early next week we will submit our Final Report and complete all other final requirements for Capstone.

5.2 Bradley

5.2.1 Fall 2017

5.2.1.1 Week 1: This was the first week of Capstone. I was told by a friend of my that took Capstone last year to review all of the Capstone projects and if you have one that you have interest in, contact the client. In so, I contacted a couple of groups and they all got back to me. However, one client said that the team was already full and the I Heart Corvallis group got back to me and put me on the team.

5.2.1.2 Week 2: This week my group members, Omeed and Dylan set up meeting times with our clients every Wednesday from 2- 3 at Snell 150. We also had a Problem Statement assignment due at the end of this week. This assignment was quite difficult since we had not met with our client and get full details about the project. However, I spoke to our professor and they said to write as much as you know.

5.2.1.3 Week 3: This week we met up with our clients for the first time and took great notes on a shared Google Doc. Got a better understanding of what the project was and what type of needs the application would need. I also finished the final draft on Monday and started thinking of what needed to into our final draft for our Problem Statement which is our first group document.

5.2.1.4 Week 4: This week we continued to write up the final draft. Once all of the group members were satisfied with what we came up with, we took it to the Writing Center at the library to get it pre-reviewed. We then sent it off to our client who revised a few things in the document and emailed the final draft to our professor and client. Got more feedback from them both and revised it again and uploaded it to our GitHub.

5.2.1.5 Week 5: This week we meet up with our client and discussed our next steps in documentation which will be the Requirements Document. After our meeting with our clients, we had our TA meeting in which he discussed what the Requirements Document was. Our next steps were to begin the writing of the Requirements Document. We created a Google Doc for it where we can share all of our information.

5.2.1.6 Week 6: This week we continued to work on the Requirements Document and and also met up with client to discuss the document with her. We later then submitted the document to our TA and GitHub for grading.

5.2.1.7 Week 7: This week was an interesting week! Our client held a focus group that consisted of other students (about 10) who had interest in helping with the idea. They also helped give us feedback on what would be interesting and cool to be in the application. A lot of them loved the idea and gave us great feedback to take along with us.

5.2.1.8 Week 8: This week we continued to do research about individual Tech Reviews. We had also met with our client and discussed this document to her. In which it was a little difficult to discuss since it was more of a technical document but we were able to teach her a few new things. We had created a Google Doc for this in which we all through our collaborations in it.

5.2.1.9 Week 9: This week was a short week due to Thanksgiving! However, we had to complete the final Tech Review which was due on Tuesday. We had completed it successfully. After that we came together as a group and begin discussing the next writing assignment that is due at the end of the term. The last document is the Design doc.

5.2.1.10 Week 10: This was a busy week due to finals coming up but we were able to complete the design document. We did a little bit of writing over break through Google Docs and google hangouts. We met up again later this week to discuss the Design Document and sent it to our client for feedback. We took in consideration of the focus groups ideas in the document.

5.2.2 Winter 2018

5.2.2.1 Week 1: Over break we had met up a few times as a group to start working on the implementation of the application. I was told to get started early on the application instead of waiting. In so, we got the group together and began the implementation. This week we brought our prototype to our client and received feedback from her.

- Showcased our application to our client
- Got feedback from our client
- Set up meeting times for our group
- Tuesday and Thursday at 4PM
- Client meeting at 2PM Wednesday
- Still waiting to hear from TA

5.2.2.2 Week 2:

- Met up with client and showcased our application
- I created a splash screen when application is opened
- Created a Google Map with markers that client wanted

5.2.2.3 Week 3:

- Got great information on what needs to into our application
- We got our database to retrieve data
- Received great information about hosting our database as well as other features to our application

5.2.2.4 Week 4:

- Make a simple website for the client to access
 - Able to update events
 - Still need to add more functionality to the website

5.2.2.5 Week 5:

- Met up with group later in the week to start the midterm progress reports
- Showcased our client our updates to the application
 - Got helpful feedback
- Figured out what else we need to finalize our project

5.2.2.6 Week 6:

- Completed all of our midterm progress reports, video and slides
- Continued to work on application and week 7 will begin the implementation of the geolocation

5.2.2.7 Week 7:

- Was able to get the hash passwords working with the database and application
- Will need to hash the passwords for the admin website as well

5.2.2.8 Week 8:

- Fixed the prepared statements
- Added a few more functionality to the website

5.2.2.9 Week 9:

- Completed the admin website
- Continuing to work on the geofencing feature

5.2.2.10 Week 10:

- Finished geofencing feature for application
- Meet up with another group on wed to do extra credit review session @2
- Begin planning for the final presentation/progress reports

5.2.3 *Spring 2018*

5.2.3.1 Week 1:

- Figure out times to meet with client
- Created a toolbar
- Began the implementation of the CAS login system

5.2.3.2 Week 2:

- Spent a lot of time figuring out the implementation of the CAS login system
- We had figured to use a webview in order to get access to the ONID login screen
- But ran into problems

5.2.3.3 Week 3:

- Completed the alternative route to take for our client to receive the information she needs
- Will be discussing it to her next week and will plan on finalizing the project with her so that we can begin the implementation for our Expo

5.2.3.4 Week 4:

- Received the okay to start working on the demo version
 - Client is very satisfied on all of our accomplishments over the year and is happy if we stop working on the project and focus our attention onto the demo for Expo

5.2.3.5 Week 5:

- Successfully created a demo version of the mobile application to showcase to people
 - We had to create a demo version of the application due to no internet connection
 - We had to take out all of the code that was referencing the database and hard code in values.
- Next steps will be to create a video of our web administration website
 - Planning on running through a short video of how to add, modify and delete information on the application

5.2.3.6 Week 6: This week we continued to work on the administration website video and completed it fully. Everything is running up to our standards for Expo.

5.2.3.7 Week 7:

- Had a successful Expo presentation
- All group members showed up to Expo on time
- Plan on working on our Final Report and video in the upcoming weeks

5.2.3.8 Week 8:

- Started working on the Final Report with group members
- Created a Google Doc for it
- Discussed what will be going into it and what we will be needing to be doing in the upcoming weeks to complete the assignments

5.2.3.9 Week 9:

- Had group member's thesis defense this week
 - Went extremely well
- Continue working on Final Report
- Planning on meeting up right after capstone class next week 10 to finish up our report and video
- Wednesday will be a busy day for us

5.2.3.10 Week 10:

- Met with group multiple times this week to finish up the final document and video

5.3 Dylan

5.3.1 Fall 2017

5.3.1.1 Week 1:

- Picked my 5 ideal projects and gave reasons for them.

5.3.1.2 Week 2:

- Began work on problem statement, contacted client and group

5.3.1.3 Week 3:

- Set up weekly meetings with TA for 3:15 Wednesdays
- Refreshed my git flow
- Met with client to figure out what exactly our project is going to be. We now have a much better picture of what the end goal will be.

5.3.1.4 Week 4:

- Added a lot of extra details of what the project actually is. Such as specific events that our app is involved with.
- We made changes on our performance metrics section, most of the content in it needed to be moved to the solution section. We also went to the writing center at the library for grammar checking and additional help. We then emailed Lyndi (our client) our final draft.
- Met with TA, discussed managing client expectation, and made plans to make more revisions to our problem statement.
- Finished the final draft and turned it into the group one note, later in the evening Kirsten gave us feedback on our submission and Badley and Omeed made final changes to the problem statement and turned it in.

5.3.1.5 Week 5:

- Met with our client on Wednesday, discussed requirements in further detail and determined that we are basically in charge of what the specific requirements are. We changed iOS development to a stretch goal, looking in to it we will be able to create a much more complete and bug free app if we focus on a single platform. We do not have experience with cross platform work, but with the research we have done into it. It seems that keeping our development to one platform for now is a better way of going about it.
- Next week we will do more thinking about our requirements to possibly break them up further or go into more detail about them for our final draft of the requirements document.

5.3.1.6 Week 6:

- Met with Lyndi and attended the focus group, we got some good information out of the focus group and will attempt to implement a social media aspect into our application. It just depends on the requirements we need to meet in order to use an instagram API, will require some research.
- Also did some error checking on the Requirements draft after getting feedback from Kirsten.

5.3.1.7 Week 7:

- We decided to see if we can implement the social media aspect to our project. The rest of the ideas were either not doable or not very good ideas.
- Turns out our project is a little difficult to split up. I have a feeling that we will be reaching to get 9 different pieces out of this project for the technical review.

5.3.1.8 Week 8:

- I finished most of the first two pieces of the technical review. I still need to come up with a third piece to write about for the final draft.
- The paper is now 2/3s done, which is good enough for this rough draft. I will be coming up with a third piece throughout the week and will hopefully have something for the final draft.
- My third piece for the project is "Application Testing", which is basically just what technology are we going to use to test and emulate our application.

5.3.1.9 Week 9:

- Finished the document, just waiting on feedback for it now. We will probably fill in a little bit for the design document but I think our current plan is to just go into office hours and get feedback next week.

5.3.1.10 Week 10:

- Finished the Design Document and turned out final draft in, Planning to create the final progress report and video this Saturday.

5.3.2 *Winter 2018*

5.3.2.1 Week 1:

- We shared schedules and found times that we can meet throughout the term to work on the project together.
- Scheduled a meeting time with Lyndi, Wednesdays at 2pm
- Worked on UI for About Us page

5.3.2.2 Week 2:

- Met with Lyndi
- Added UI for passport, and leaderboard pages
- Implemented recyclerview for leaderboard page
- Emailed TA about scheduling meetings

5.3.2.3 Week 3:

- Met with Lyndi
- Met with Derek the director of web and mobile services at OSU
- Updated Design Document to reflect design changes (no longer using semantic UI for application)

5.3.2.4 Week 4:

- I began the implementation for logging out and session management and we met with Lyndi on Wednesday.
- Created google document to keep track of necessary features needed for the app to be considered complete, a more goal oriented version of the requirements + design document

5.3.2.5 Week 5:

- Created midterm progress report and presentation
- Created first draft of our poster board for Expo
- Implemented session management + logout functionality
- Met with Lyndi and TA

5.3.2.6 Week 6:

- Implemented hashing of user passwords
- Finished midterm progress paper
- Met with Lyndi and TA

5.3.2.7 Week 7:

- Began implementation of prepared statements, to prevent SQL injections
- Met with TA (Brad and Omeed also met with Lyndi, I couldn't attend due to having to move the meeting to another time)

5.3.2.8 Week 8:

- More Prepared statement work
- Began implementation of new dashboard
- Changed weekly meetings with Lyndi to Thursdays
- Met with TA and Lyndi

5.3.2.9 Week 9:

- Implemented PHP hashing for admin website
- Met with Lyndi

5.3.2.10 Week 10: We worked on the final progress report and presentation, planning on finishing them this weekend/next week. We also have a meeting set up with Andrew Morgan of Identity and Access at OSU. He works with CAS, so hopefully he will be able to give us some insights on how to implement CAS in our app.

5.3.3 *Spring 2018*

5.3.3.1 Week 1: We figured out each others schedules and picked a meeting time with us and Lyndi. There were also some app breaking bugs that we had to fix towards the end of the week. We got those taken care of and now the app works again.

5.3.3.2 Week 2: We spent most of the week working on different CAS implementations, such as directing the user via webview to the CAS login screen, and redirecting the user through a PHP script. But none of our ideas were able to work, due to the layout of our application. We emailed Derek Whiteside to see if he could aid us or point us in the right direction for help.

5.3.3.3 Week 3: We figured out how we can host pictures on the database, we simply store the name of the picture in the database, and we store the actual image on the server. We know the URL so we can append the images name to the URL to retrieve it.

Met with Lyndi's contact Sara, who sadly does not have the resources to host an application such as this.

5.3.3.4 Week 4: Completed the poster and submitted it. Met with the TA and it looks like we are all set for Expo once we complete the demo version of our app. We also took care of a few bugs in the app over the weekend, such as not being able to change a photo for an already created event/resource.

We also met with our client and decided we will not be implementing the CAS login, as we have not heard back from Derek and our client is satisfied with simply prompting the user for their ONID.

5.3.3.5 Week 5: We completed the midterm report and presentation. And we also managed to complete the demo version of the app in one day.

5.3.3.6 Week 6: Made a few small fixes, app looks ready to go for Expo!

The meeting went well, unfortunately Visit Corvallis does not have the resources to host an application, they have about 3 employees, and none of them with CS experience. However they are interested in contributing to the app by populating it with events/resources and maybe even prizes.

5.3.3.7 Week 7: Attended Expo

5.3.3.8 Week 8: Did nothing for Capstone this week, which felt weird.

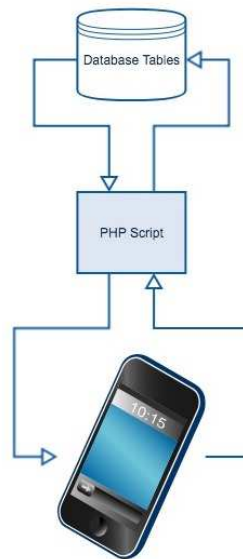
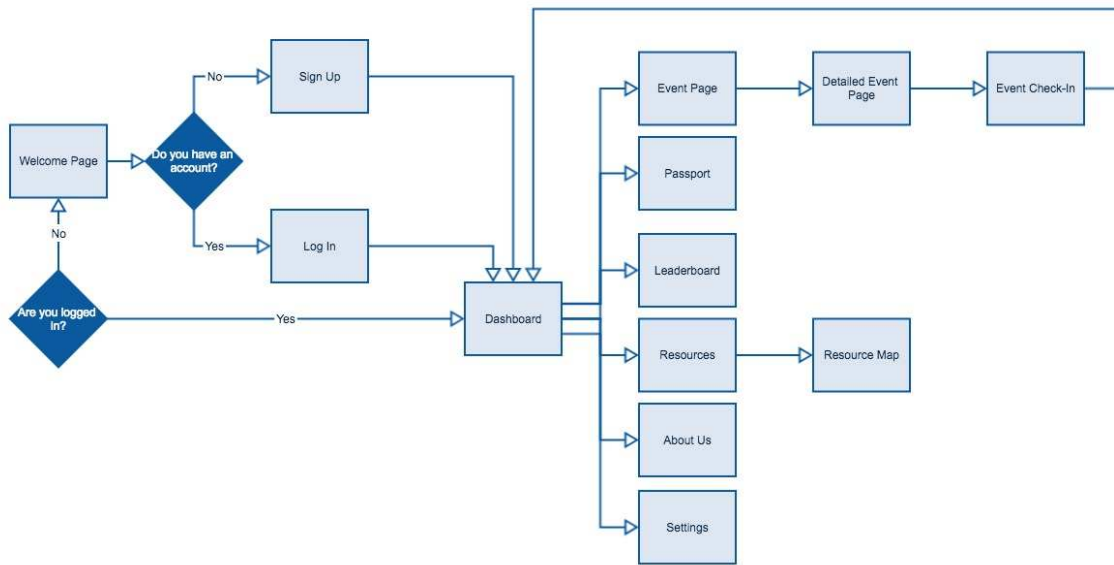
5.3.3.9 Week 9: Met with Martha from the OSU marketing department on Friday, who seemed very interested in the application. She has a few contacts that she wants to talk with, but hopefully she will be able to point us to someone who can carry on the application.

5.3.3.10 Week 10: Completed the video presentation as well as the paper. We will be handing over the code to Lyndi some time next week. Also, we have been going through the code and adding comments to describe what each class is doing, and some details about how it does it.

6 FINAL POSTER

COLLEGE OF ENGINEERING	Electrical Engineering and Computer Science		14
<p>About the Project</p> <p>The I Heart Corvallis project is a mobile application we created for the Corvallis Community Relations (CCR) office and is part of a larger initiative ran by the office to help students become more involved in the community.</p> <p>The application focuses on increasing student awareness of events and service opportunities happening around Corvallis, as well as creating an incentive for students to attend these events.</p> <p>Students will be able to view upcoming community events and service opportunities, receive stamps for attending these events, and win rewards for accumulating enough stamps. They can also view community resources and more information about the initiative.</p>  	<h3 data-bbox="505 254 1068 306">“I Heart Corvallis” Mobile App</h3> <p data-bbox="505 323 1052 396">Incentivize student involvement with the greater Corvallis community</p> <div data-bbox="565 405 704 428"> <p>Passport System</p>  <ul style="list-style-type: none"> • Users can see the events they completed in their passport • When a user has completed enough events, they can turn in their stamps to redeem prizes from the CCR office </div> <div data-bbox="873 405 1036 428"> <p>Detailed Event View</p>  <ul style="list-style-type: none"> • Users can see the events they completed in their passport • When a user has completed enough events, they can turn in their stamps to redeem prizes from the CCR office </div> <div data-bbox="488 867 805 1056"> <p>Client Administrative Interface</p>  <ul style="list-style-type: none"> • The CCR office has the ability to add, modify, and remove events, resources, and prizes from the application via the administrative website • The office can also view summaries for events and organizations, such as how many students attended an event and how students are rating events by a particular organization </div>	<p>Implementation</p> <p>App Development Java, XML Android Studio Emulator Android devices (smartphones, tablet)</p> <p>Website Development HTML, CSS, Javascript</p> <p>Database Integration PHP, MySQL</p> <p>Corvallis Community Relations Office Lyndi-Rae Petty, PI and project organizer, Outreach Coordinator for Center for Civic Engagement Jonathan Stoll, Director of Corvallis Community Outreach</p> <p>CS Capstone Team (Pictured from left to right below): Dylan Tomlinson, tomlinsd@oregonstate.edu Omeed Habibelahian, habibelo@oregonstate.edu Bradley Imai, imab@oregonstate.edu</p> 	

7 PROJECT DOCUMENTATION



The application provides users with a passport that shows them which events they have completed. Users receive stamps upon completing events, and when a user accumulates a sufficient number of stamps, they can redeem prizes. On the administrative website, the CCR office will be able to add, modify, and delete any content from the application, such as events, prizes, resources, resource map locations, survey questions, and the information displayed on the About Page. The office will also be able to authorize other users to access the administrative website.

The app has not yet been published to the Google Play Store, as the iOS version of the application has not yet been created. Once both versions of the application have been fully developed, the application will be published to both the Google Play Store and the Apple App Store. Once it is available on the Play Store, devices with Android 4.4 "KitKat" or

later installed will be able to download the application from the Google Play Store.

In the meantime, in order to run the app, the device you want to run the app on must be connected to a computer that has both Android Studio installed and the full code directory for the app stored on it. To run the program in Android Studio, hit the Run button (the green Play button) at the top of the IDE and you will be able to choose whether you want the app to run on an emulator or on your own connected device.

8 RECOMMENDED TECHNICAL RESOURCES TO LEARN MORE

- Android Developers Website: <https://developer.android.com/>
- PBKDF2: <https://medium.com/@kasunpdh/how-to-store-passwords-securely-with-pbkdf2-204487f14e84>
- Derek Whiteside, Director of Web and Mobile Services at OSU
- Andrew Morgan, Systems Administrator, Identity and Access Management at OSU

9 CONCLUSIONS AND REFLECTIONS

9.1 Omeed

9.1.1 *What technical information did you learn?*

I learned a lot about the visual design of applications, as well as how interactions between applications and databases are set up. It was really cool learning how each piece of the user interface visually comes together in the code. It took me a while to figure out exactly how applications connect to databases, and even though I would look through code tutorials, I continued to run into walls implementing it myself, but once I figured out an efficient implementation for one page, it was a lot easier to mold the code to fit for other pages of the app as well.

9.1.2 *What non-technical information did you learn?*

I learned a lot about how to communicate my creations to other people and how to modify my use of technical jargon to make my explanation as clear as possible to anyone who inquired about the project. This also applies to meeting with our client, as I had to effectively and clearly communicate our progress and next steps to our client, who did not have as much technical experience as us. I also learned how to schedule the completion of each task of a long-term endeavor, as the application and website had a slew of bits and pieces that we needed to make sure got completed in a timely manner.

9.1.3 *What have you learned about project work?*

I have learned that its important to communicate your progress and planned tasks to your teammates so there is no confusion as far as who is in charge of what tasks. I have also learned that dedicated teammates are crucial to completing everything that needs to get done for the project. This project has also taught me that plans change, and something that you thought would be a huge factor in the development of the project may end up being completely abandoned in the end. As you implement the project, your client may bring up new desires and change their mind on previous design decisions, and it is important to be aware of this when embarking on a time-intensive project.

9.1.4 *What have you learned about project management?*

As far as project management, this project taught me that it is incredibly important to plan out each task of the project and to set up a timeline for when to complete each task by. You also need to be clear when communicating your timeline to your teammates and client, as you need to ensure that not only your teammates are on board with your plan in relation to their plans, but also that your client agrees with your timeline. The final product is ultimately for the client, so they need to confirm that your schedule for completing each piece of the project aligns with their plans to deploy or market the final product. It is also very important to keep your progress saved in multiple places so that not only can you continue your work on different machines, but you have a backup of your data if you somehow lose your local files.

9.1.5 *What have you learned about working in teams?*

I have learned that each member of your team brings a different perspective to the table, whether it be for a particular piece of the project or for the project design as a whole. It is important to maintain your concept for the final product while simultaneously considering the concepts your teammates have created. They will undoubtedly have some idea for the project that you did not even think of, and the final product will reflect the decisions your team made on which

features to implement and which ones to ignore or push to a later time. It is also very important to keep consistent contact with your team so you are not left in the dark about what the rest of your team is implementing.

9.1.6 *If you could do it all over, what would you do differently?*

If I were to do this project over again, I would definitely consider implementing the application with a cross-platform development environment. Our client's original goal was to have both an Android and iOS version of the application developed this year, and since we only had prior experience developing for iOS, we decided to focus on implementing the Android version of the application and making the iOS version a stretch goal. If I were to go back to Fall Term, I would do significantly more research on cross-platform development environments, such as React Native.

9.2 Bradley

9.2.1 *What technical information did you learn?*

From the very beginning of the implementation stage of Winter term, every aspect of the programming and research was a new experience for me. My past experience in programming an Android application was the CS496 course at OSU. We created a simple weather application that displayed weather data from an API. The I Heart Corvallis mobile application was an eye-opener for me in the Android development. Everything from the creation of an animated splash screen, utilizing Google Maps and displaying notable locations, geolocation, and implementing the ability for an Administration website to add, update and delete content that was displayed on the application was all a new experience for me.

9.2.2 *What non-technical information did you learn?*

Having the ability to talk nontechnical and technical in a working environment is essential to a successful project. In my past interviews for jobs, I have been asked every time to give an example of a nontechnical and technical experience. Over the course of this year, I had the ability to learn this skill and become proficient in it. Talking to our client whose background is nontechnical versus talking technical with group members was a slight barrier. However, I was able to pick up this skill easily and was able to successfully explain features and progress of our application to our client as well as discussing technical features of the application with group members.

9.2.3 *What have you learned about project work?*

This has been bar far the biggest project I have been a part of. Being able to organize and manage time in an efficient manner was a must in developing a successful application. Not only did I have to worry about completing my tasks for this application but I also had to worry about other coursework. The main factor that I took out of this coursework was that features took a lot longer than expected. Some of the smaller features of the application were able to develop in one setting but the majority of the features took at least a few days to program.

9.2.4 *What have you learned about project management?*

Reflecting this year, I noticed we took on a waterfall method for project management. This consisted of group members doing research, writing a requirement, design and technical documents in the fall, followed by implementation in the winter and bug fixing the spring. During the implementation phase, I noticed that we had to take a step back and modify our design and requirement decisions. This took us back a few days in which we could have been developing for our application. I have done some research on project management and many companies these days use the agile method. This consists of running through sprints and being able to develop a product in a more timely manner.

9.2.5 What have you learned about working in teams?

By working with an amazing group of guys, I was able to develop a specific set of experiences and skills that normal coursework could not. An example of this would be that groups make better decisions than individuals. Having multiple opinions of various features of the application gives ideas that may not have come up for an individual. Being able to work in a group environment is a must in the working field. No job, besides freelance will give you the option to work alone.

9.2.6 If you could do it all over, what would you do differently?

If I could do this all over again I would take a look at cross-platforming. After doing a few job search within mobile development I have seen a lot of React and React Native developers in need. From my understand React Native is a cross-platform framework that can develop for IOS and Android mobile application with use of javascript. Facebook developers had created this framework and have been widely used in Airbnb, Instagram, Discord and many more.

9.3 Dylan

9.3.1 What technical information did you learn?

Basically every piece of code I wrote for this project was something new for me. I have only made one program using Java before, and it was extremely simple compared to this project. I also have never had experience working on a mobile application before. In terms of technical information I learned about the components that make up an android application. Such as having a Java class and XML layout for every page. I learned about how an android application uses resources and how to pass variables between classes. I also learned how to work with a hashing algorithm that is not part of a library. I also learned about how an android application communicates with a database. I always assumed that the PHP scripts would simply live with the application. I did not know that most applications simply use SQLite, storing information on the users phone instead of in a seperate database.

9.3.2 What non-technical information did you learn?

For non-technical info I learned how to set up and meet with actual computer scientists. We had a few situations throughout the year that required us to meet with employees of Oregon State. Through these meetings I feel that I learned a bit about the protocol of meeting with people in the field. I also learned about focus groups and getting user data. I had only been in one focus group and attended a few hosted by our client throughout the year. I also learned a bit about selling your work and having to describe it to technical and non technical people. Over the past month we have been trying to find a way for our project to live on after our graduation. We have met with the Visit Corvallis office and the marketing department of OSU in order to build up interest in our app. Nothing has come to fruition yet but it was still a good experience communicating our work and vision.

9.3.3 What have you learned about project work?

It helps to have others who are interested in working on the project. For my previous senior design project each of us split up the aspects of the project and none of us made much headway in the project. Having my group to work with through these requirements this year helped tremendously in keeping steady progress throughout the year. We have been able to keep each other accountable and motivated to work on the application. I have also learned that it takes a lot of time, between getting approval from our client and the actual time it takes to code in a feature, a lot of time passes.

Most of these requirements took several days if not weeks to complete. When creating something new you generally cant expect to sit down and finish it in one sitting.

9.3.4 What have you learned about project management?

I have learned the importance of planning for the worst. Leaving extra time later in the development cycle is the main reason why we were able to complete this project. We began development during winter break as we knew this project was going to take a lot of planning. That planning paid out by allowing us to work when we had time instead of having to cram our schedules half way through the term. There were very few times that I was worried about the project as we were generally ahead of where we needed to be in the development cycle.

9.3.5 What have you learned about working in teams?

Communication has been the most important factor in working as a team. We have several different outlets for communicating that we have been using frequently throughout the year. Finding times to work together was always a challenge but we managed, spending many late nights at the Valley Library over winter term. It is also important to communicate your work to your teammates. If you are having a specific problem, they could usually offer some sort of insight to help out.

9.3.6 If you could do it all over, what would you do differently?

I would have started the CAS implementation earlier in the development cycle. We did not realize that it would be much more difficult to implement CAS in a mobile application than in a website. If we had started it in the winter, than we might have been able to finish it. I also would have looked further into cross platform development. We did a lot of research on the subject, but we had our worries that it wouldnt work out well. It would have been great to have an iOS and an Android version of our app.

10 APPENDIX I: ESSENTIAL CODE LISTINGS

10.1 EventListAdapter Class

```
package edu.oregonstate.studentlife.ihcv2.adapters;

import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.Date;

import edu.oregonstate.studentlife.ihcv2.R;
import edu.oregonstate.studentlife.ihcv2.data.Event;

/**
 * Created by Omeed on 1/18/18.
 */

public class EventListAdapter extends
    RecyclerView.Adapter<EventListAdapter.EventListViewHolder> {

    private ArrayList<Event> mEventList;
    private OnEventClickListener mOnEventClickListener;
    private final static String TAG = EventListAdapter.class.getSimpleName();

    public EventListAdapter(OnEventClickListener onEventClickListener) {
        mEventList = new ArrayList<Event>();
        mOnEventClickListener = onEventClickListener;
    }

    public void addEvent(Event event) {
        mEventList.add(event);
        notifyDataSetChanged();
    }

    @Override
    public int getItemCount() {
        return mEventList.size();
    }
}
```

```

public interface OnEventClickListener {
    void onEventClick(Event event);
}

@Override
public EventListViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
    LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
    View view = inflater.inflate(R.layout.event_item_listview, viewGroup, false);
    EventListViewHolder viewHolder = new EventListViewHolder(view);
    return viewHolder;
}

@Override
public void onBindViewHolder(EventListViewHolder holder, int position) {
    Event event = mEventList.get(position);
    holder.bind(event);
}

class EventListViewHolder extends RecyclerView.ViewHolder {

    private LinearLayout mEventListingLL;
    private TextView mEventMonthTV;
    private TextView mEventDayTV;
    private TextView mEventNameTV;
    private TextView mEventLocationTV;
    private TextView mEventTimeTV;

    private String[] monthShortNames = {"JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL",
        "AUG", "SEP", "OCT", "NOV", "DEC"};

    public EventListViewHolder(View itemView) {
        super(itemView);
        mEventListingLL = (LinearLayout) itemView.findViewById(R.id.ll_event_listing);
        mEventMonthTV = (TextView) itemView.findViewById(R.id.tv_event_month);
        mEventDayTV = (TextView) itemView.findViewById(R.id.tv_event_day);
        mEventNameTV = (TextView) itemView.findViewById(R.id.tv_event_name);
        mEventLocationTV = (TextView) itemView.findViewById(R.id.tv_event_location);
        mEventTimeTV = (TextView) itemView.findViewById(R.id.tv_event_time);

        mEventListingLL.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mOnEventClickListener.onEventClick(mEventList.get(getAdapterPosition()));
            }
        });
    }
}

```

```

}

void bind(Event event) {
    mEventNameTV.setText(event.getName());
    mEventLocationTV.setText(event.getLocation());
    Log.d(TAG, "start dt: " + event.getStartDay() + "-" + event.getStartMonth() + "-" +
        event.getStartYear() + " " + event.getStartTime());
    Log.d(TAG, "end dt: " + event.getEndDay() + "-" + event.getEndMonth() + "-" +
        event.getEndYear() + " " + event.getEndTime());
    if (event.getStartDay().equals("1") && event.getStartMonth().equals("1") &&
        event.getStartYear().equals("1900") && event.getStartTime().equals("12:00 AM")
        && event.getEndDay().equals("31") && event.getEndMonth().equals("12") &&
        event.getEndYear().equals("2099") && event.getEndTime().equals("11:59 PM")) {
        mEventMonthTV.setText("ANY");
        mEventDayTV.setText("TIME");
        mEventTimeTV.setText("Anytime");
    }
    else {
        int monthInt = Integer.parseInt(event.getStartMonth()) - 1;
        mEventMonthTV.setText(monthShortNames[monthInt]);
        mEventDayTV.setText(event.getStartDay());
        String dtRange = event.getStartTime() + " - " + event.getEndTime();
        mEventTimeTV.setText(dtRange);
    }
}

}

}

```

10.2 PassportLoader Class

```
package edu.oregonstate.studentlife.ihcv2.loaders;

import android.content.Context;
import android.support.v4.content.AsyncTaskLoader;
import android.util.Log;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;

/**
 * Created by Omeed on 2/22/18.
 */

public class PassportLoader extends AsyncTaskLoader<String> {
    private final static String TAG = PassportLoader.class.getSimpleName();

    private String userid;
    private String passportJSON;
    private final static String IHC_GET_COMPLETED_EVENTS_URL =
        "http://web.engr.oregonstate.edu/~habibelo/ihc_server/appscripts/get_completed_events.php";

    public PassportLoader(Context context, String userid) {
        super(context);
        this.userid = userid;
    }

    @Override
    protected void onStartLoading() {
        forceLoad();
    }

    @Override
    public String loadInBackground() {
        try {
            URL url = new URL(IHC_GET_COMPLETED_EVENTS_URL);
            String data = URLEncoder.encode("userid", "UTF-8") + "=" +
                URLEncoder.encode(userid, "UTF-8");

            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
```

```

conn.setRequestMethod("POST");
conn.setDoOutput(true);
OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
wr.write( data );
wr.flush();

BufferedReader reader = new BufferedReader(new
    InputStreamReader(conn.getInputStream()));

StringBuffer sb = new StringBuffer("");
String line = null;

while ((line = reader.readLine()) != null) {
    sb.append(line);
}

return sb.toString();
} catch (Exception e) { e.printStackTrace(); return new String("Exception: " +
    e.getMessage()); }
}

@Override
public void deliverResult(String data) {
    passportJSON = data;
    Log.d(TAG, "passport data: " + data);
    super.deliverResult(data);
}
}

```

10.3 Recording a Completed Event

```
<?php

ini_set('display_errors', 1);
error_reporting(E_ERROR);
ini_set('memory_limit', '1G');

$dbhost="oniddb.cws.oregonstate.edu";
$dbname="habibelo-db";
$dbuser="habibelo-db";
$dbpass="RcAbWdWDkpj7XNTL";

$alreadyExists = False;

$mysqli = new mysqli($dbhost,$dbuser,$dbpass,$dbname);
//Output any connection error
if ($mysqli->connect_error) {
    die('Error : (' . $mysqli->connect_errno .') ' . $mysqli->connect_error);
}

$userid = $eventid = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $userid = $_POST["userid"];
    $eventid = $_POST["eventid"];
    $rating = $_POST["rating"];
    $comment = $_POST["comment"];
    $dateandtime = date("Y-m-d H:i:s");

    $stmt = $mysqli->prepare("INSERT INTO ihc_completed_events (userid, eventid, dateandtime,
        rating, comment) VALUES (?, ?, ?, ?, ?)");
    $stmt->bind_param('iisis', $userid, $eventid, $dateandtime, $rating, $comment);
    $stmt->execute();
    if ($stmt->get_error == "") {
        $stmt2 = $mysqli->prepare("SELECT * FROM ihc_completed_events WHERE userid=?");
        $stmt2->bind_param('i', $userid);
        $stmt2->execute();
        if ($stmt2->error == "") {
            $result = $stmt2->get_result();
            $stampcount = $result->num_rows;

            $stmt3 = $mysqli->prepare("UPDATE ihc_users SET stampcount=? WHERE id=?");
            $stmt3->bind_param('ii', $stampcount, $userid);
            $stmt3->execute();
            if ($stmt3->error == "") {
```



```
        echo "COMPLETED EVENT ADDED";
    }
    $stmt3->close();
}
$stmt2->close();
}
else {
    echo "ADDERROR";
}
$stmt->close();
}

$mysqli->close();
?>
```

10.4 Updating a Prize

```
<?php

ini_set('display_errors', 1);
error_reporting(E_ERROR);
ini_set('memory_limit', '1G');

$dbhost="oniddb.cws.oregonstate.edu";
$dbname="habibelo-db";
$dbuser="habibelo-db";
$dbpass="RcAbWdWDkpj7XNTL";

$alreadyExists = False;

mysqli = new mysqli($dbhost,$dbuser,$dbpass,$dbname);
//Output any connection error
if ($mysqli->connect_error) {
    die('Error : ('. $mysqli->connect_errno .') '. $mysqli->connect_error);
    echo "Connection failed!<br>";
}

$prizeid = $name = $level = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $prizeid = $_POST["prizeid"];
    $name = $_POST["name"];
    $levelVal = $_POST["level"];

    if ($levelVal == "1") {
        $level = "gold";
    }
    else if ($levelVal == "2") {
        $level = "silver";
    }
    else if ($levelVal == "3") {
        $level = "bronze";
    }

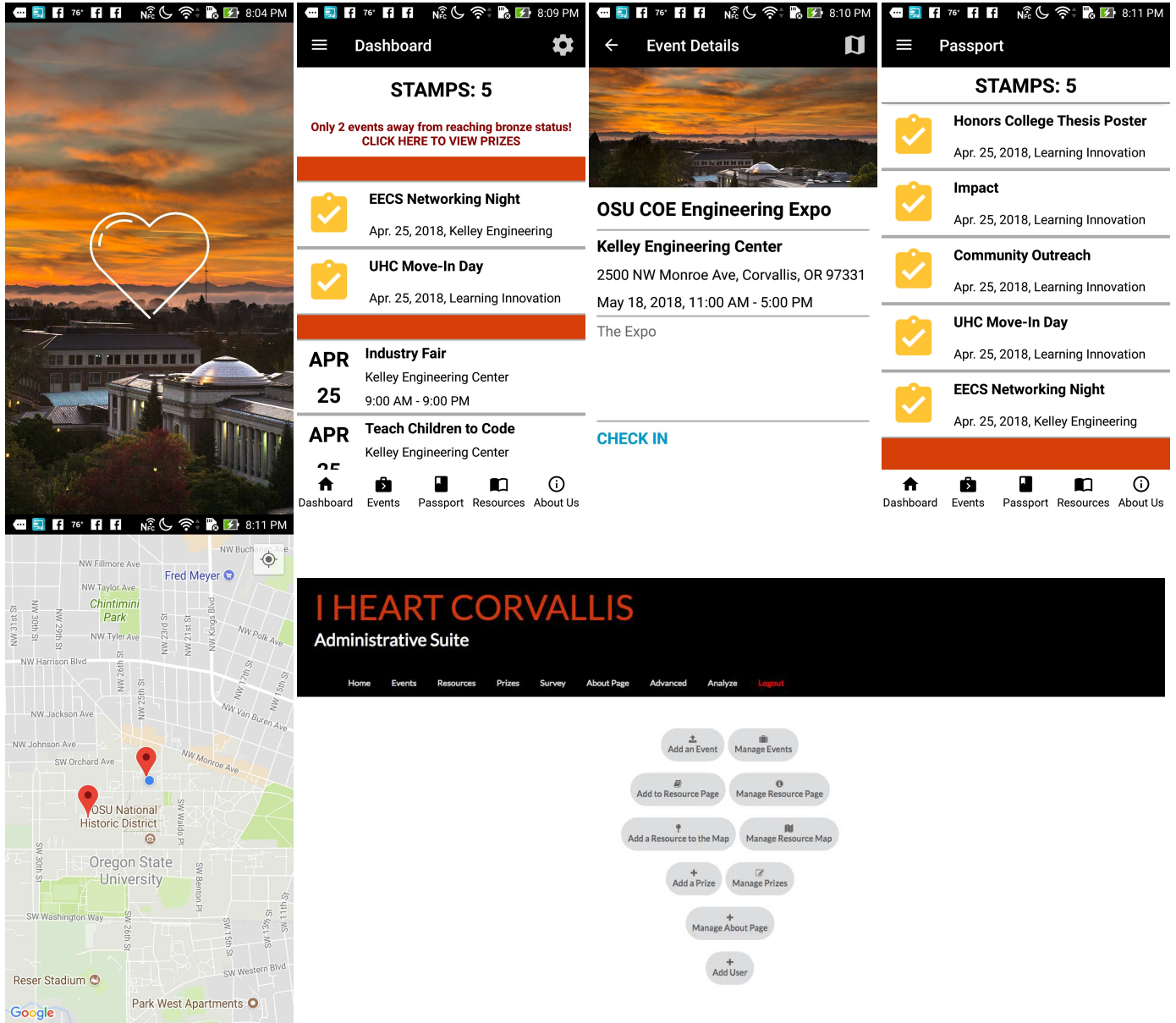
    $stmt = $mysqli->prepare("UPDATE ihc_prizes SET name=?, level=? WHERE prizeid=?");
    $stmt->bind_param('ssi', $name, $level, $prizeid);
    $stmt->execute();

    if ($stmt->error == "") {
        $message = "Prize has been updated!";
    }
}
```

```
else {  
    $message = "Error updating prize!"; # error updating prize in database  
}  
$url = "../manage_prizes.php";  
  
$stmt->close();  
$mysqli->close();  
echo "<script type='text/javascript'>alert('$message');</script>";  
echo "<script type='text/javascript'>document.location.href = '$url';</script>";  
}  
$mysqli->close();  
  
?>
```

11 APPENDIX II: SCREENSHOTS

11.1 Application



Top left: Splash screen

Top second from left: Dashboard

Top second from right: Event Detail Page

Top right: Passport

Bottom left: Event Map

Bottom right: Administrative Website Homepage