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

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 JetBrain's IntelliJ IDEA software and designed specifically for Android development.
- **Stamp:** In-app verification that the user has attended a particular community activity. Some activies 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 Android SDK

The Android Studio libraries that are built into the SDK allows us to create the user interface of our application. It has all of the required functionalities that are necessary to create our interface in a way that satisfies our desires and that of our clients. The Android SDK uses XML files called layouts to handle the structure and parts of the content of the interface.

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 originally decided to use Semantic UI along with the Android Studio IDE from Android SDK. Semantic UI allowed us to implement a clean interface without resorting to the stock Android style, and although Androids 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. Problems began to arise with Semantic UI while implementing the database. We originally were going to attach PHP files to our project in Android Studio but were quickly overwhelmed with the number of obstacles needed for PHP to run on Android Studio. There was also the concern of how much slower the application would run with these work arounds that had to run. We decided to stick to the Android SDK because it allows for a flexible interface and a faster communication between the application and its database.

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 psuedorandom function to derive keys, and the length of the derived key is essentially unbounded. [4] [5]

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. [4]

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 Android SDK

Android Studio contains a multitude of layouts that can be used in conjunction to create a user interface that supports several different screen sizes and densities. The layouts can be configured in ways that will allow for the contents of a page to appear similarly across many devices.

4.5.4 Design Rationale

The original design decision we made for our application was to utilize Semantic UI to control our interface scaling. But as was mentioned in section 4.2.3 we ran into complications with using HTML and Javascript in our program. In the end we decided to change our decision to the Android SDK. It contains the necessary functionality to keep our interface consistent across many devices and has plenty of documentation to aid us in our development.

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. [6]

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. [6]

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. [7]

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 Java. 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. [8]

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. [8]

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. [9]

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. [9]

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] 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/

[5] 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

[6] “Run apps on the android emulator,” [Online; accessed 29-November-2017]. [Online]. Available: https://developer.android.com/studio/run/emulator.html

[7] Connecting with api data from instagr.am using php. [Online]. Available: http://spyrestudios.com/connecting-with-api-data-from-instagr-am-using-php/

[8] The google maps geolocation api. [Online]. Available: https://developers.google.com/maps/documentation/geolocation/intro

[9] Markers. [Online]. Available: https://developers.google.com/maps/documentation/javascript/markers