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

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 it's 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