Meals-on-Wheels Android App

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Kevin Luk

Spring, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Daniel Graham, Department of Computer Science

Meals-on-Wheels Android App

CS4991 Capstone Report, 2021

Kevin Luk
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
khl7wh@virginia.edu

ABSTRACT

The Meals-on-Wheels program in Charlottesville, Virginia, needed an Android app that would enable vendors with Android phones to keep track of volunteers and clients and ensure that meals were being delivered to the right locations. As an intern for Meals-on-Wheels, I used Kotlin as the programming language to write the app and SQLite as the relational database management system to store information such as client and volunteer names, customer addresses and contact information. I also used various software development methods to design and implement this app, including model-view-view-model (MVVM) and software architecture object-oriented programming (OOP). While the app was not finalized during the internship, the company anticipates the new Android app will provide Android users with an easier method of tracking their meal distribution process.

1 Introduction

Meals-on-Wheels America is an organization that supports thousands of community-based programs across the country that help senior citizens dealing with isolation and hunger (Meals on Wheels America, 2021). These programs consisting of millions of staff and volunteers ensure that seniors are able to live independently and with dignity. These volunteers deliver meals and perform regular safety checks for seniors in need.

Historically, the Meals-on-Wheels program in Charlottesville, has been a paper-based company. Staff were given papers with information about their volunteers, and volunteers were given papers with information about the seniors for whom they were providing meals and safety checks. This resulted in an excessive amount of paper being wasted. In recent years, the increased attention towards environmental protection and sustainability caused the Charlottesville Meals-on-Wheels program to reevaluate its paper use. ZippyZen, LLC, a small software company based in Charlottesville, created Zippymeals, a cloud-based software solution for Meals-on-Wheels (Zippy Meals 2017). A web application and iOS app were created, but an Android version did not exist.

2 Related Work

ZippyZen created a web application that enabled staff members to manage their volunteers and clients (Figure 1). All data was centralized into a single location where staff can control client and volunteer scheduling, create route sheets with maps and directions, and receive updates about clients and volunteers. All the data and information needed by staff is automatically backed up on the cloud, where it is protected and can be accessed from anywhere. While this application presented a way for staff to better organize all of information related to their clients and volunteers, a great amount of paper was still being used to create ride sheets.

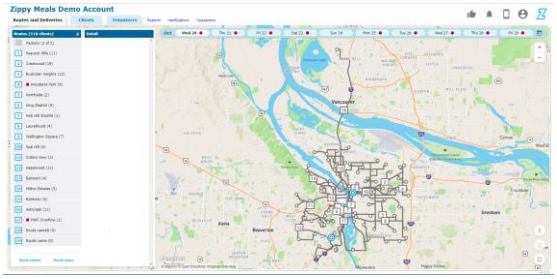

Figure 1: Zippy Meals Web App at https://zippymeals.com/login/

An iOS app was also created in order to help the program to transition to using less paper, as well as provide an easier method of keeping track of clients and volunteers (Zippy Meals 2021). With this new app, staff members are able to use interactive maps to view client locations and volunteer routes. Another important feature of this app is the ability to view all data in offline mode, since the data is stored locally on the smartphone (Figure 2).

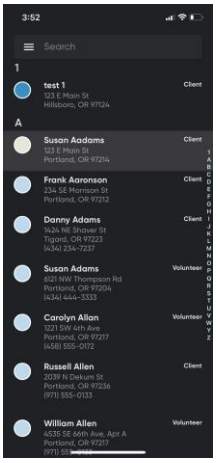

Figure 2: iOS App with dummy information

Unfortunately, while staff were able to view program information via the app, volunteers still had no way of seeing client information. This problem would be addressed in the app.

3 Design Process

The design of the Android app was based on the existing iOS version of the app. As a result, the code base for the iOS app was used as a reference for the new code base for the Android app. I used Kotlin as the programming language to write the app and SQLite as the relational database management system. The app was developed in Android Studio, an Android development environment.

3.1 Development

I used object-oriented programming (OOP) in order to organize the client and volunteer data into objects. Each of these objects contain specific information for that object. Many of the objects were related to one another, so I created a Unified Model Language (UML) diagram in order to better visualize the relationships between all of the objects, as well as the specific attributes of each object (Figure 3). This made it easier for other team members that were working on other aspects of the app to quickly understand the relationship between all of the data that would be used in the app as well as the attributes that were specific to a data object.

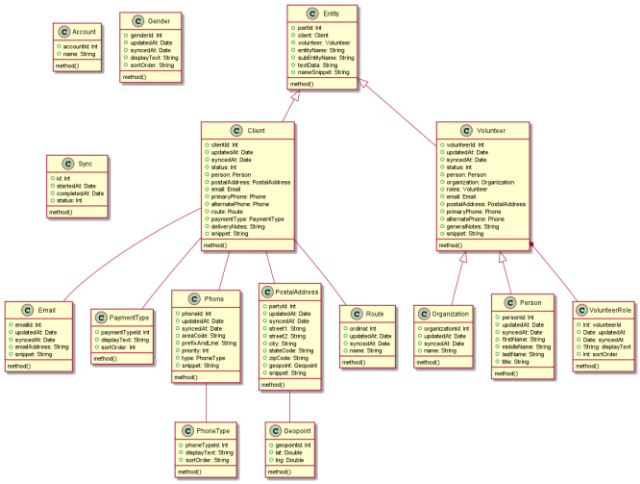

Figure 3: UML Diagram of Class Objects

The design pattern that was used to create this app was model-view-viewmodel (MVVM) (Figure 4). The models are the data objects, the view is the presentation of the data that the users interact with and the viewmodel controls models are displayed in different views (Britch, 2021).

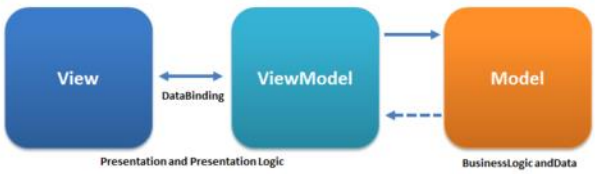

Figure 4: MVVM Design Pattern

The work that I focused on within my team was to design and create the screens for the app. Using the iOS app as a reference, I created the screens for the Android app to match those of the iOS app. I created the basic layout of the login screen (Figure 5), main menu (Figure 6), and the individual volunteer (Figure 7) and client screens (Figure 8).

Since the main focus for those screens was to display the volunteer and client information, the color schemes for the screens did not match those of the iOS app. The plan was to change the color schemes to match the iOS app once the app was completed. Since these were static screens, dummy data was used to simulate fetching data from a server and displaying it on the screen. In the final product, there should a sorting and searching functionality for the client and volunteer information screen to allow for easier access to information.

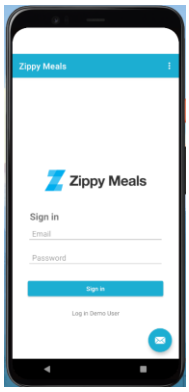

Figure 5: Login Screen

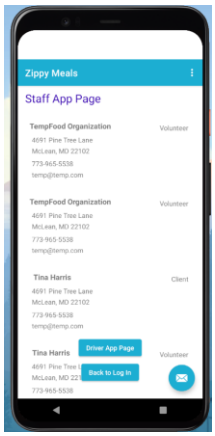

Figure 6: Client and Volunteer Information Screen

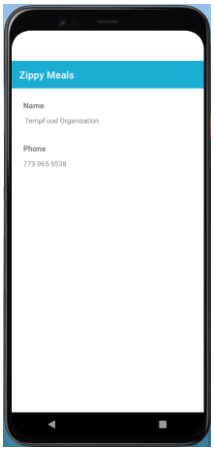

Figure 7: Volunteer Information Screen

Figure 8: Client Information Screen

Once the basic information could be displayed on the screens, I switched my work to implementing database functionality with the app. This app has offline functionality, meaning that if a user were to sign into their account and lose connection to the internet, all the information fetched from the server would remain on the phone since it would be stored locally.

I used SQLite as the database since it is a serverless database that is used for local data storage. Room, a persistence library that is part of the Android Jetpack library collection, was used to integrate SQLite into the app. This library adds an abstraction layer over SQLite which simplifies database access and validates SQL queries when the app is compiled. I used a pre-existing database file of volunteer and client information to see if the information could be queried from the app. In the final product, this database file would be retrieved from a server and then stored on the mobile device to allow for offline functionality.

I used the Junit5 testing framework for testing the app. Extensive testing of the app could not be done during the time of the internship. Basic testing was still performed in order to test small aspects of the app, and to ensure that data objects were being properly created and data was being correctly inputted and retrieved from the database.

3.2 Challenges

The main problem that my team and I faced was figuring out how to integrate SQLite with the app. Before using Room, it was very tedious to convert between the data retrieved from the SQL queries and data objects since the data is formatted differently. It was also very difficult to ensure that the SQL queries being written were correct. These problems were solved once we learned about the Room persistence library.

Another difficulty that my team and I faced was the learning how to develop Android apps. No one on my team, including myself, has developed an Android app before or used Kotlin. Much time was spent learning the language and learning how Android apps are developed.

4 Results

As the Meals-on-Wheels program in Charlottesville continues to grow, there must be a solution for participants of the program to efficiently access information. This solution will increase the efficiency of the program as everyone will have easier access to needed information such as volunteer and client contact information. Also, this solution will slowly eliminate the program's issue of using large amounts of paper since information will now be available on mobile devices.

5 Conclusion

This early prototype of the Android app set a good starting point for future developers to continue the work. Eventually, the Android app will have the same features as the iOS app, allowing for members with Android devices to quickly access information just like their iOS counterparts.

6 Future Work

Many aspects of the app still need to be implemented since only a very basic prototype was completed during the summer. One feature that we were working on but were not able to finish by the end of the internship was the user authentication feature. During the internship, we used dummy data instead of real staff data so that it would be easier to test how the information would be displayed on the UI. Once the data was properly formatted on the UI, we began to work on the user authentication feature so that we could simulate a staff member logging in to their account to see information about their volunteers and clients.

Another feature that was being implemented was the data access layer objects (DAO) for all the data objects in the app. This is an interface that allows for create, read, update and delete operations for each of the data objects and their respective tables in the database.

As development continues, more time and greater focus should be devoted to testing the app. During the internship, testing was rarely done, which can cause major issues that might be more difficult to solve as the code base continues to grow.

7 UVA Program Evaluation

Numerous classes prepared me for the work that I would be doing during this internship. Data Structures and Algorithms (DSA 1 and 2) were extremely helpful. Object-oriented programming was heavily used in order to group the data together, and data structures

was used to organize that data. Database Systems (CS 4750) was also very helpful because it introduced me to storing and retrieving data from databases, which I needed to know how to do in order to retrieve the client and volunteer information from the database. Software Development Essentials (SDE) was helpful because it taught me how to utilize version control when working on a software development project, how to organize code so that it is scalable, and basic software testing skills.

While the computer science at UVA is extremely helpful in the way that they teach their students, some changes could still be made to further improve the student's learning experience. The introductory programming class should devote more time to basic data structures so that students are not overwhelmed when moving on to higher level CS courses such as CS 2110, which is known to be an extremely difficult course. Introduction to Programming (CS 1112) is taught in Python, and spending more time learning about lists and dictionaries would help students as they move onto CS 2110 or DSA.

Another change that should be made all CS courses should be project-based. Projects will help students better understand how the information they are learning will be implemented in real world situations. Creating more projects will also help students build their resumes and provide more topics to discuss during interviews.

REFERENCES

[1] Meals on Wheels. 2021.(2021). Retrieved October 27, 2021 from https://www.mealsonwheelsamerica.org/

[2] ZippyZen LLC. 2017. Zippy Meals. (2017). Retrieved October 27, 2021 from https://zippymeals.com/

[3] ZippyZen LLC. 2021.(March 2021). Retrieved October 27, 2021 from https://apps.apple.com/us/app/zippy-meals/id1557599310

[4] David Britch. 2021. The Model-View-ViewModel Pattern. (August 2021). Retrieved October 27, 2021 from https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm