Delaware State University
Dover, Delaware

Helping the Muslim Community Navigate Through a Western World

Tuba S. Abbasi

Advised by Dr. Tomasz Smolinski

A Project Submitted in Partial Fulfillment
of the Requirements of the
Bachelor of Science in Computer Science – Professional Development
May 2018

ABSTRACT

Follow. Subscribe. Comment. These are all words that most of use on a daily basis. In our current era, we have all the information we want and need at our fingertips. We can "subscribe" to or "follow" various organizations or places, on our favorite social media platform so that we can be notified about their latest updates. Even though this concept of "subscribing" has been around for a while, most non-profit organizations and locations are still playing catch up due to lack of funding. Most smaller scale non-profit organizations do not have the funds to hire a developer to build and maintain an app, which caters to the organization's specific needs.

One community where this is most evident is the Western Muslim[1] community; as it is a minority within the Western world. Numerous Mosques[2] around the United States either have outdated apps or no apps at all. This introduces problems for Muslim individuals such as difficulty finding Mosques, keeping up-to date on important holiday announcements, etc.

Thus, I am proposing and developing a mobile application as-a-service called "Ummah[3]", which will target the Muslim community. Though each Mosque is different from one another, they all seek to share the same basic information, such as the Sect they belong to, prayer times, events, etc. Hence, the Ummah app will; (1) Allow numerous Mosques around the U.S. (initially) to claim themselves and be listed on the app. After doing so, they can share the information mentioned above; (2) Allow the Ummah app users to subscribe to any Mosque in the area in which they live, or frequently visit. Once they do so, they will be able to get updates on the prayer times, events, or announcements posted by that Mosque.

[1] Muslim: a person who follows the religion, Islam. Islam is a monotheistic religion teaching there is one God and that prophet Muhammad is the messenger of God [5].
[2] Mosque: a place of worship for Muslims [6].
[3] Ummah: a common Arabic word which in literal translation means nation, people group, or community [7].

BACKGROUND

According to a study conducted in 2015, the Muslim population makes up about 24% of the world's overall population [1]. In addition, 74% of the countries within the world are Non-Muslim majority countries [1]. These statistics imply that we have a decent Muslim population in Non-Muslim majority countries – for the purposes of this project I will be focusing on the Western community (specifically, the United States of America). As an American Muslim myself, I have first handedly experienced some of the problems the Muslim community faces while living in a Western country. The most prominent issues for Muslim individuals are, 1) finding Mosques and, 2) keeping up-to date with a Mosque of interest. On the flipside, Mosque owners are facing some challenges as well. The Mosque owners spend a considerable amount of money to get curated apps built for their Mosque. A lot of this money is actually spent on managing the app, and overtime funding these apps becomes too expensive. Due to lack of funding, these apps often become too outdated for the community to use. For my capstone project, I am developing a mobile application called "Ummah[1]" to help mitigate some of these issues for the Muslim community.

The "Ummah" application will allow users to easily find Mosques, keep up-to date with Mosques, and view details of various Mosques in a uniform format. This uniformity is established with the "app as a service" model that I will be utilizing. We have seen this model in apps like Yelp, where many businesses claim themselves on one platform and their information is displayed to the users in almost an identical format. From my previous work and insight from the community, I realize that uniformity in the design aspect of the app is vital, as this is what makes some of the best apps intuitive to use. Similar to the Yelp app, my app will allow users to search for places of interest based on location. Specifically, my application will list the points of interest that are relevant to the Muslim community. Each Mosque listing will have some information for the user such as, Mosque Sect, prayer times, latest announcements, upcoming events, etc. In addition, the application will have a subscription system allowing users to "subscribe" to their favorite Mosque. Once they do so, they will be able to get updates on that Mosque's prayer times, upcoming events, announcements, *etc*. directly on their phone.

In short, the functionalities of this app are: (1) Mosques all over the U.S. (initially), will be able to claim their business/organization and be listed on the app; (2) The users will have the ability to subscribe to any Mosque in the area in which they live, or frequently visit.

PREVIOUS WORK

Most Mosques and Churches are considered nonprofit organizations. Due to this, funding can be limited, and decisions have to be made on where to invest that money. A lot of Mosque owners have invested in hiring a developer to build an app curated especially for their Mosque. The Mosque owners try to ensure the app is low cost; however, in the long run it ends up costing them a lot more than they accounted for. The reason being, an app such as this is encapsulated with a notification, location, and a blogging system. These features can make the app not only complex, but as a result, also costly in terms of managing it. Such apps require constant updates in order to keep them up and running. Eventually, the allocated funding falls short and the app starts getting outdated.

Most of these apps have faults in even the basic functionalities such as: regular crashes while using the app, missing data, lack of utilization of useful tools built into smart phones today, *etc*. For example, most of the existing apps for such organizations do not have push notifications, instead they are either sending out emails to their users for any updates or announcements or posting them on their Facebook page, both of which require effort on the user's part to either check their email or Facebook. In addition, Muslim users have to rely on either emailing or calling the Mosque to find information regarding important dates. For example, some Islamic Holidays are based on the lunar calendar, for which individuals have to call the Mosque to confirm the dates of the holiday. This is a great example of an instance where a simple push notification would have saved them time. Although push notifications might seem like a minute example, it is a useful functionality that most smartphone users rely on and use on a daily basis.

The issues listed above were derived from first hand experiences. The figures below show screenshots of a few Mosque apps and their flaws.

Figure 1.0 – Apps Downloaded for Research

The “ISD” app is Delaware based, whereas “Muslim Pro”, and “MoqueFinder” apps are general apps.

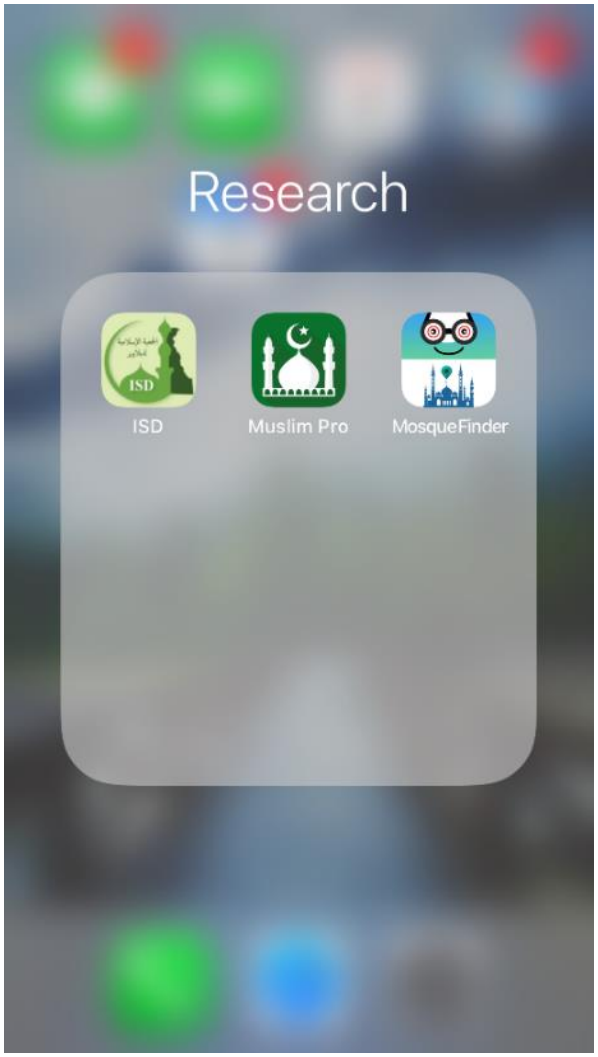
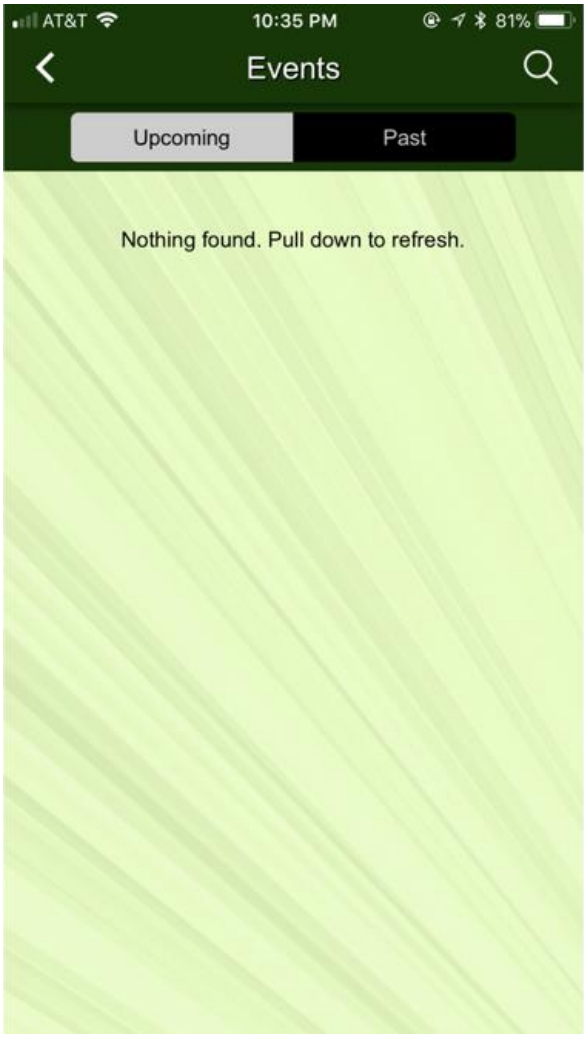
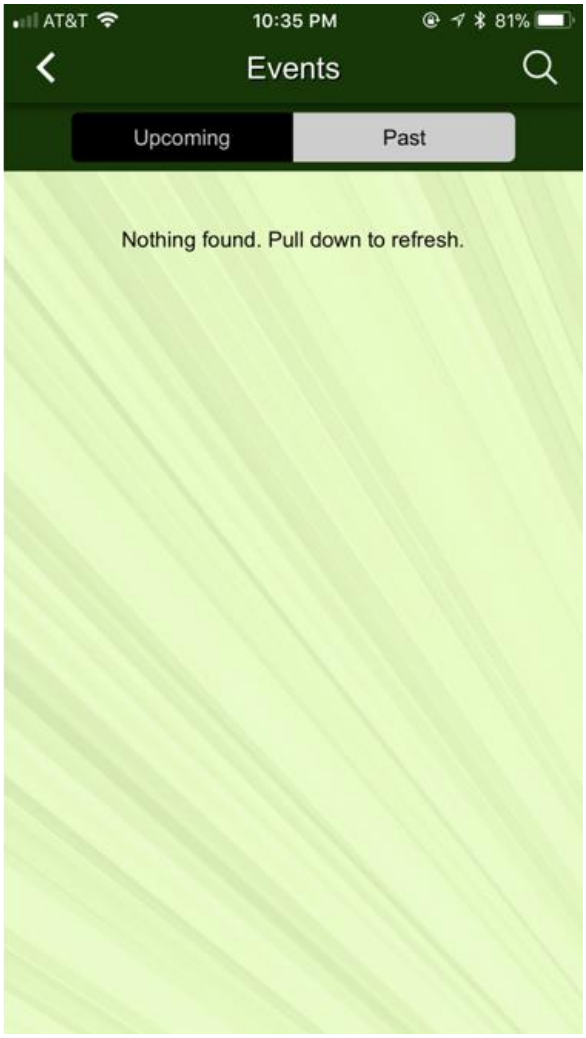


Figure 1.1 – ISD Mosque Events

This is an example of one of the apps for a Mosque in Delaware. Here I show both, “Upcoming”, and “Past” events and you can see that there is no data displayed. However, this same Mosque is quite active with posting events on Facebook. I talked to the Mosque owner and was confirmed that the app has not been updated since 2016.





Consequently, in the same app, the Mosque announcements section has not been updated either.

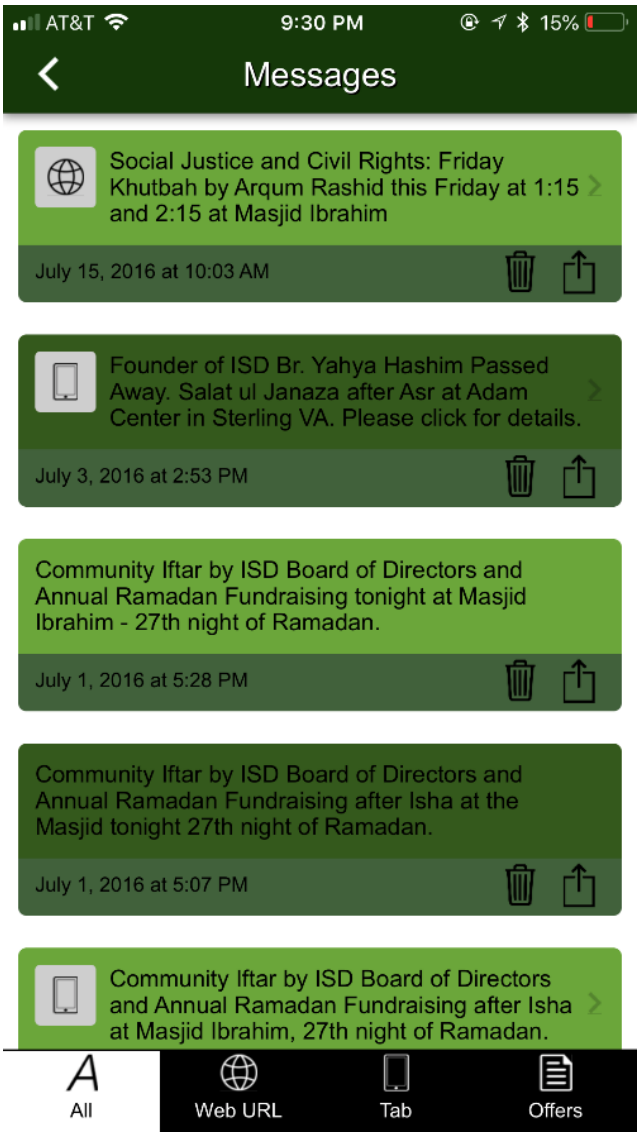


Figure 1.2 – Mosque Finder

The idea behind this app was to display a list of Mosques based on the user's current location. However, as you can see when I first open up this app there are no Mosques shown in the list view or the map view.

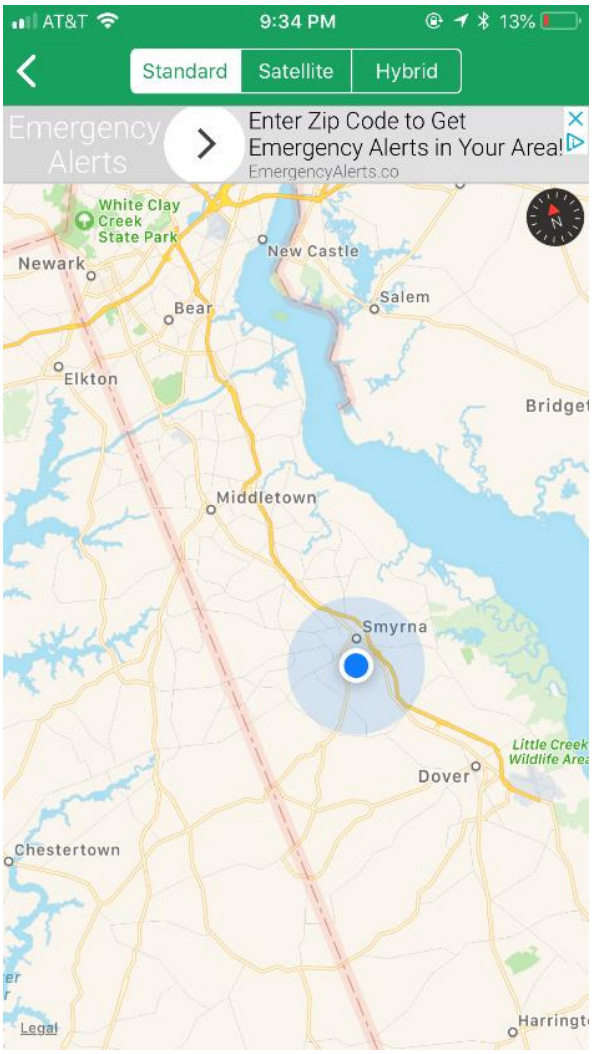
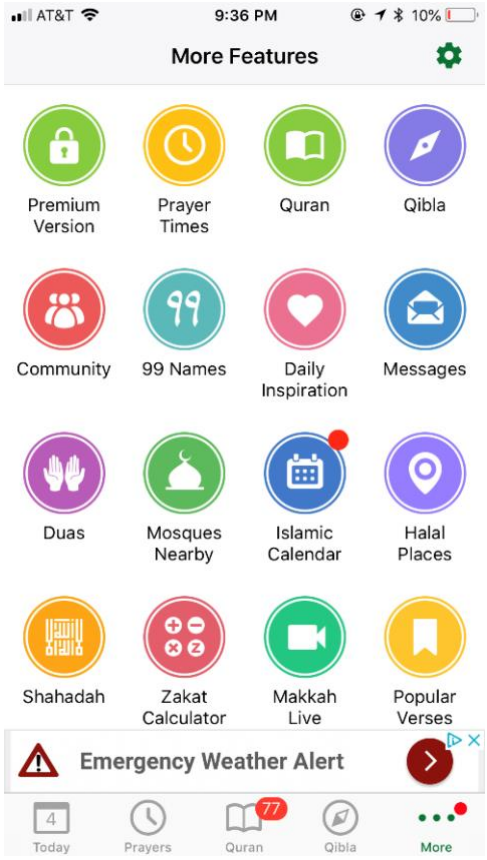


Figure 1.3 – Muslim Pro

This app had a lot more features; I however, focused on the Mosque nearby feature. When I tapped on Mosques nearby, the app reports that there are no Mosques near my area. This is far from reality — there is a Mosque just 10 miles from my current location.

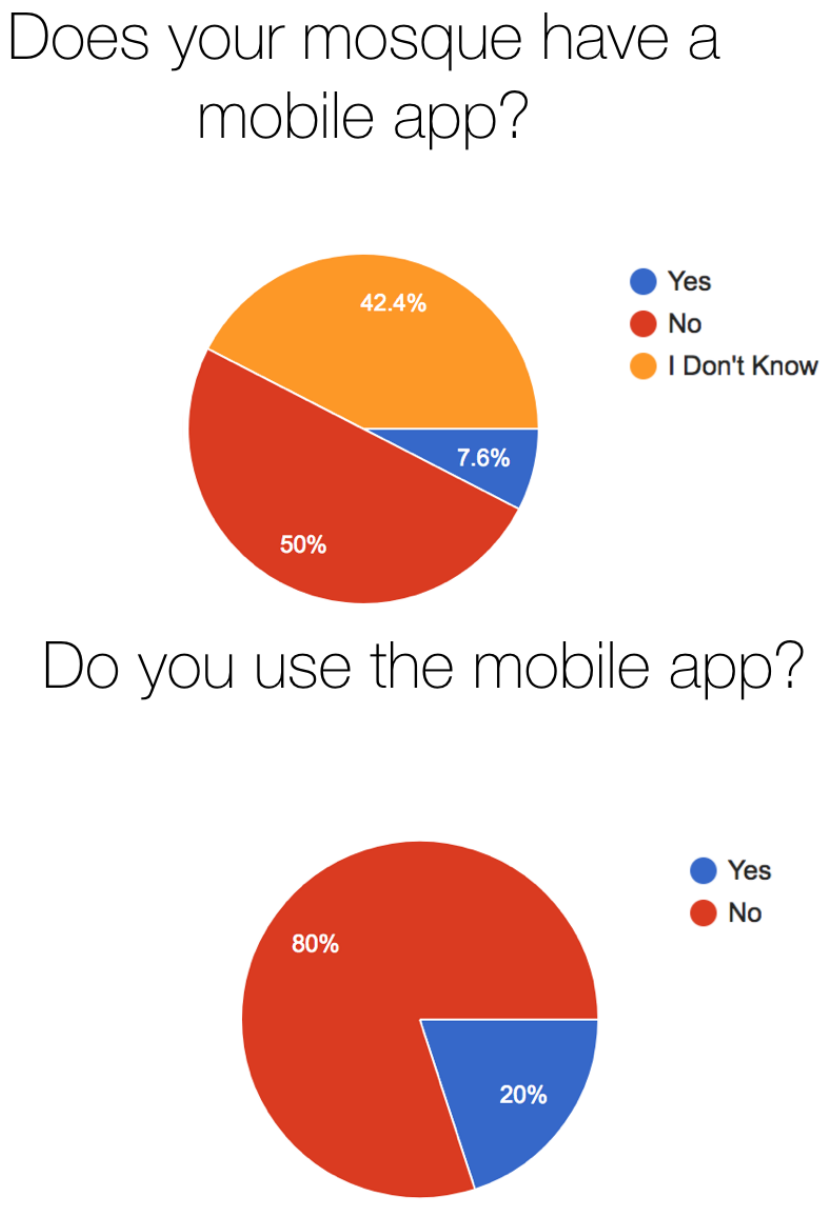


For the reasons defined above, developing an app as a service will be quite useful not only for the consumer, who will be able to subscribe to various Mosques, but to the Mosque owners as well. Though each Mosque is different from one another, they all seek to share the same basic information, such as the Sect they belong to, prayer times, events, *etc.* The Ummah app model will save Mosque owners' money and the users' time and energy, making it a better solution for the Muslim Ummah.

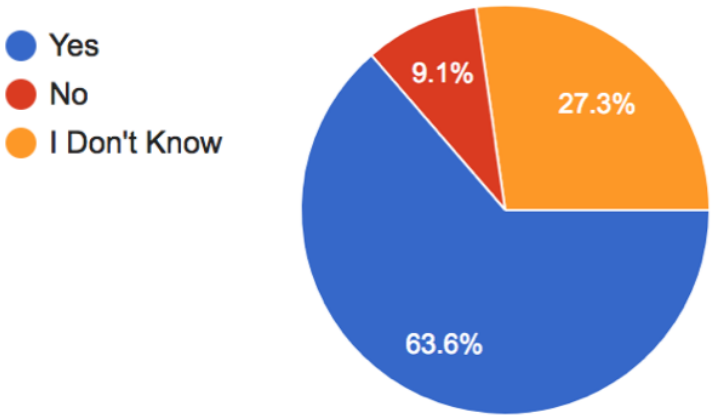
SIGNIFICANCE

The Ummah app is given significance purely through its' user base. In order to determine whether this app will be successful a survey was sent out to part of the Muslim community living within the U.S.A. or Canada. This survey received 65 responses in over 2 days. The results shown in the figures below confirmed that this majority of the community would be open to using the Ummah app.

Figure 1.4 – Survey Results



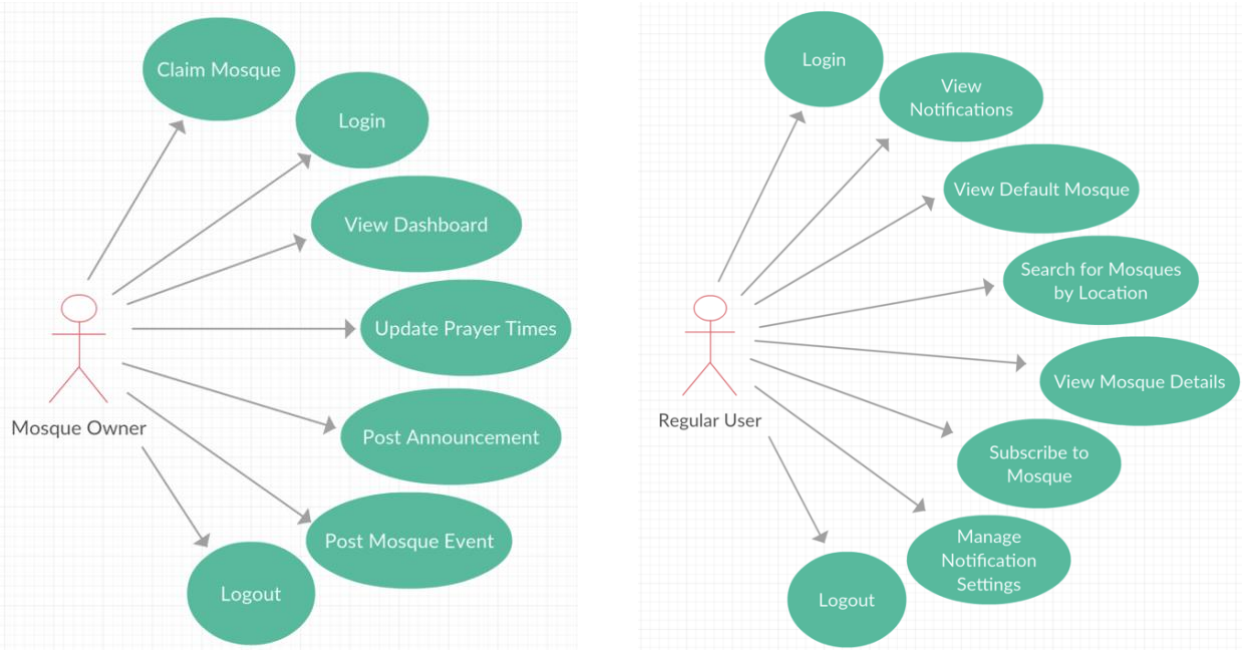
Would you use an app where you can **subscribe** to any mosque, and **get updates** about that mosque's events, announcements, & prayer times?



IMPLEMENTATION AND TOOLS

The Ummah mobile application has two main types of users; the *Mosque owners* and the *Regular users* of the app. The Mosque owners claim their Mosque on the app through a web-based dashboard. The dashboard will require each Mosque owner to enter detailed information about their Mosque such as the Mosque name, the Sect they belong to, their address, etc. Additionally, they can utilize the dashboard to update the Mosque prayer times, post Mosque events, and even make announcements. This content will be displayed on the front-end side on the mobile app for the Regular users. The Regular users create an account and login to the app. Depending on their location privacy setting, they will either be shown a list view of the Mosques in their area, or they will be able to enter a specific location (zip code, city, state, etc.) resulting in a list view of Mosques in the specified area. The Regular users can view any Mosque in detail and if they wish, they can "Subscribe" to one of the Mosques they wish to follow. This allows the Regular user to get updates posted by the Mosque owner for that Mosque, directly on their phone.

Figure 1.5 – Use Cases for Mosque Owner and Regular User



The Ummah system is has the following components; a mobile app (iOS), database, web-based dashboard, and a server to enable communication between all the components. Keeping the server-side code separate was intentional as it will save time when developing the Android version of the application. In other words, majority of the backend code will remain the same for all platforms.

Figure 1.6 – System Overview

The overview below shows the flow between the Ummah system components as the user is utilizing the app. First, the Mosque owner dashboard is shown, this is where the Mosque owners claim their mosque – allowing them to be visible on the Ummah app. Any data the Mosque owner posts (such as events, announcements, etc.) is stored in the Firebase database. This data is retrieved in the Node.js server. Then, the server sends this data to application while the user is using it. This flow ensures that the user always gets the most up to date data. These components are explained further in detail within the "Implementation Breakdown" section below.

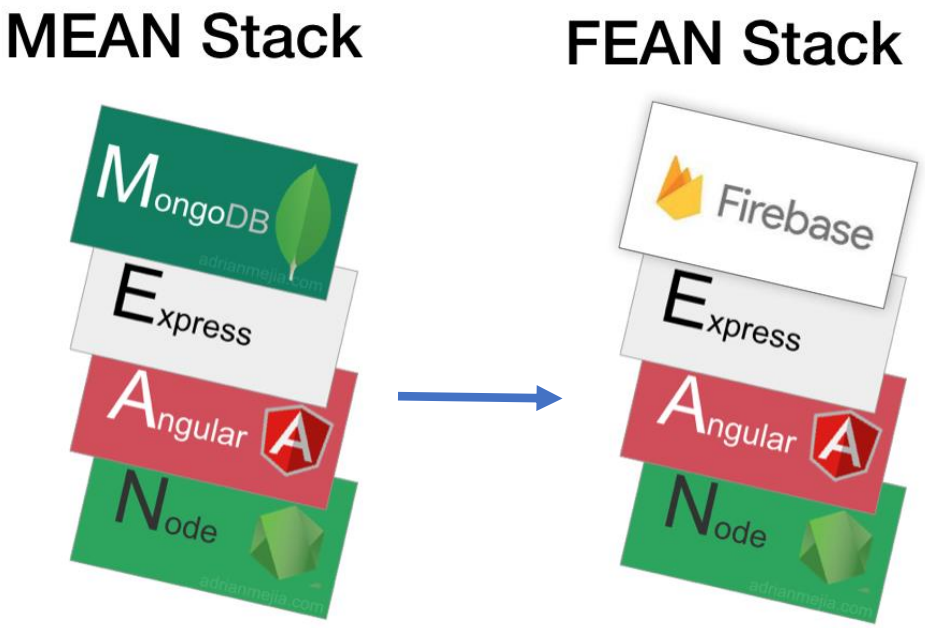
IMPLEMENTATION BREAKDOWN

Stack Implementation:

Deciding on the right stack was a vital step in delivering a successful prototype within 2 and a half months. The decision had to be made between the two leading stacks, LAMP and MEAN stack. After some research the MEAN stack was selected due to it requiring only JavaScript for both, the frontend and the backend [9-10]. This saves a lot of time in learning multiple languages, which can be allocated towards building the prototype.

The MEAN stack stands for its components MongoDB, Express, Angular.js, and Node.js [11]. The MongoDB is a no SQL database and was swapped out for Firebase database instead. Firebase database, is a cloud hosted real-time document store [12-13]. Since the data is stored in the cloud it was readily available everywhere, which saved a lot of code work. In addition, Firebase provides authentication systems for multiple platforms (including Web, iOS, and Android), and can be easily integrated with the platform without writing the code for it from scratch.

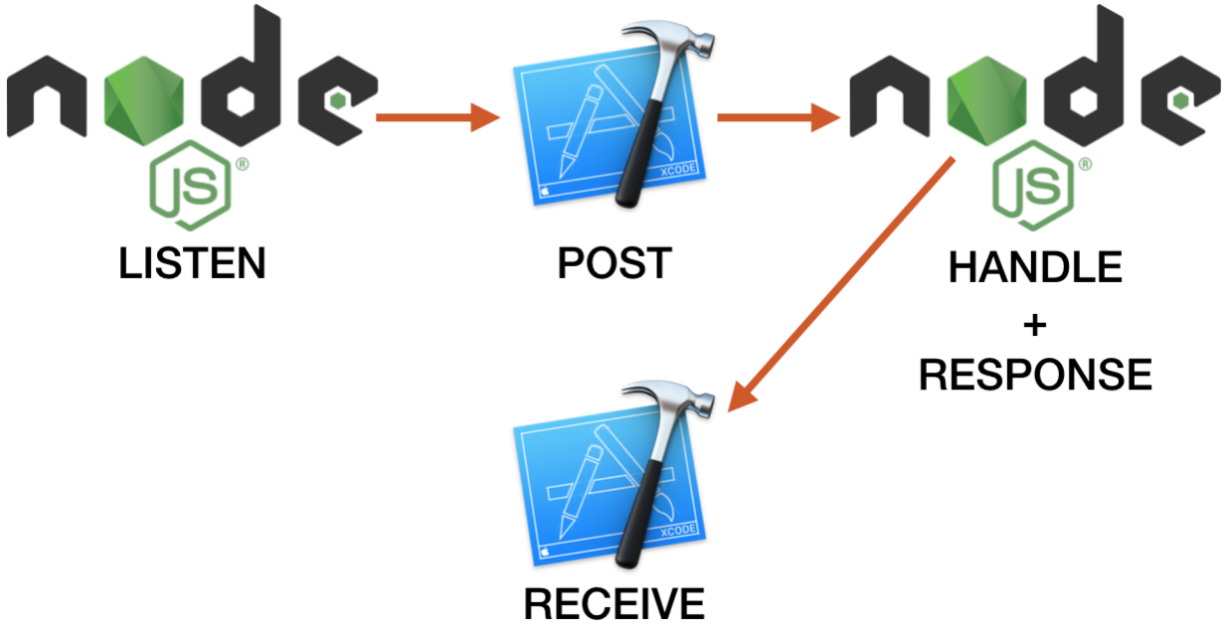
Figure 1.7 – Stack Implementation



Setup Server:

The Node.js server was setup locally to make POST and GET requests between App.js and the Xcode application. The server was tested by first, creating a basic html form and then rendering it through the server. Then Xcode was setup to POST to the server by utilizing the Xcode's Alamofire library [14]. After this, the server was setup successfully. The server communication process is displayed below.

Figure 1.8 – Server Implementation

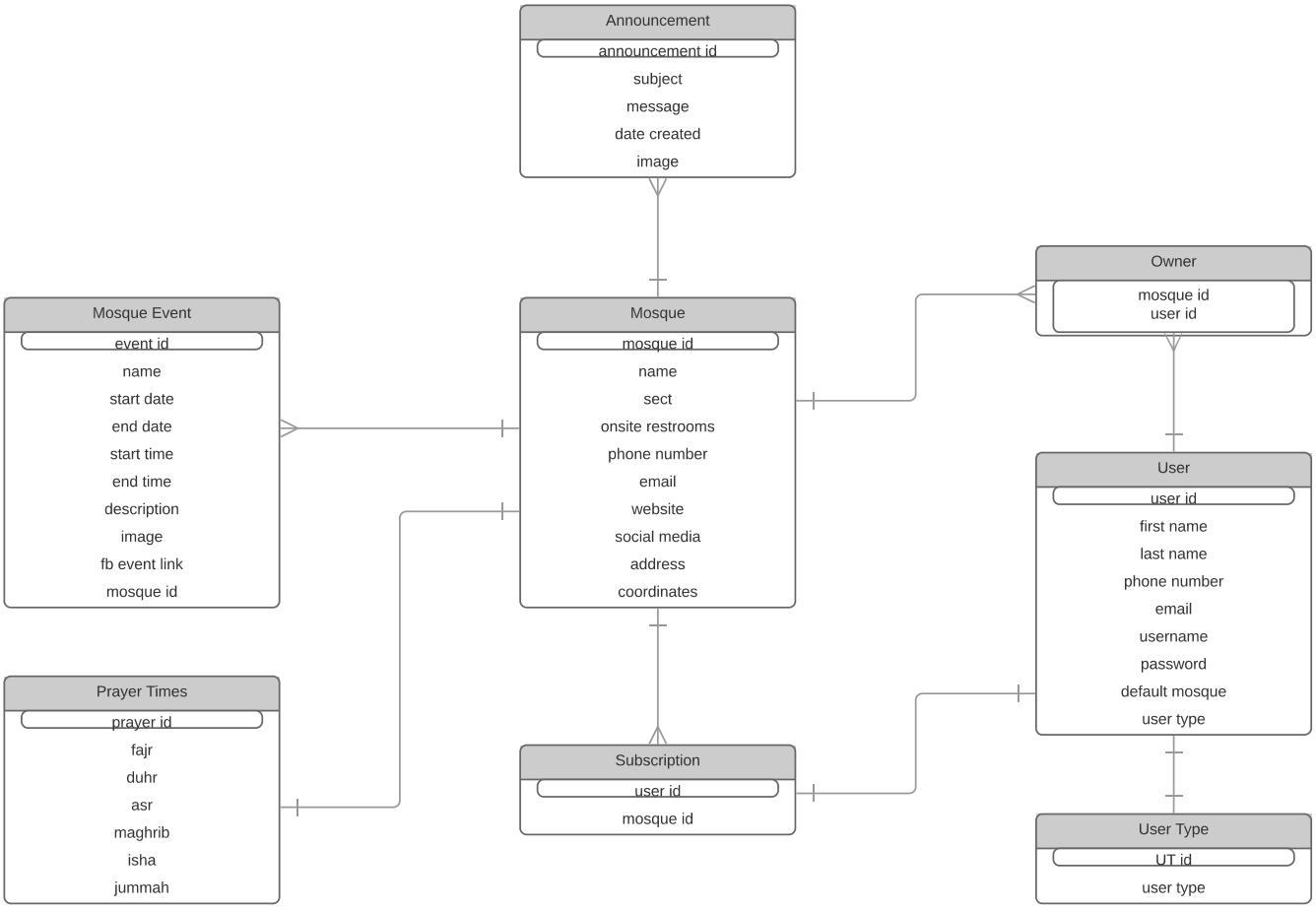


App Structure:

The Ummah database stores all the Mosque data, including the Ummah mobile app user's basic information (i.e. username, email, etc.). The database component is one of the most complex components within the Ummah system and required multiple revisions. The final model of the database is shown below.

Figure 1.9 — Ummah Entity Relation Diagram

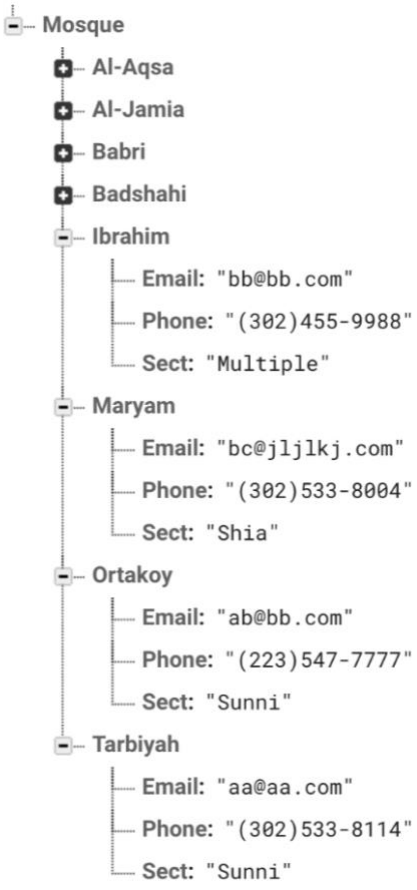
The database has the following entities; Mosque, Owner, Announcement, Mosque Event, Prayer Times, Subscription, User, and User Type. Note that the Mosque entity is in the center and it the main component connecting all the other entities together.



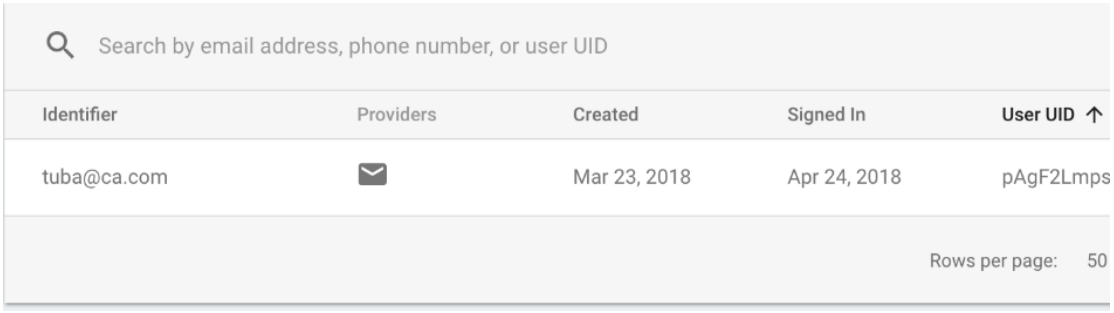
A basic implementation for the prototype was developed in Firebase database.

Figure 2.0 — Firebase Database Console

First, we can see the nodes created for the Mosques within the Firebase database console.

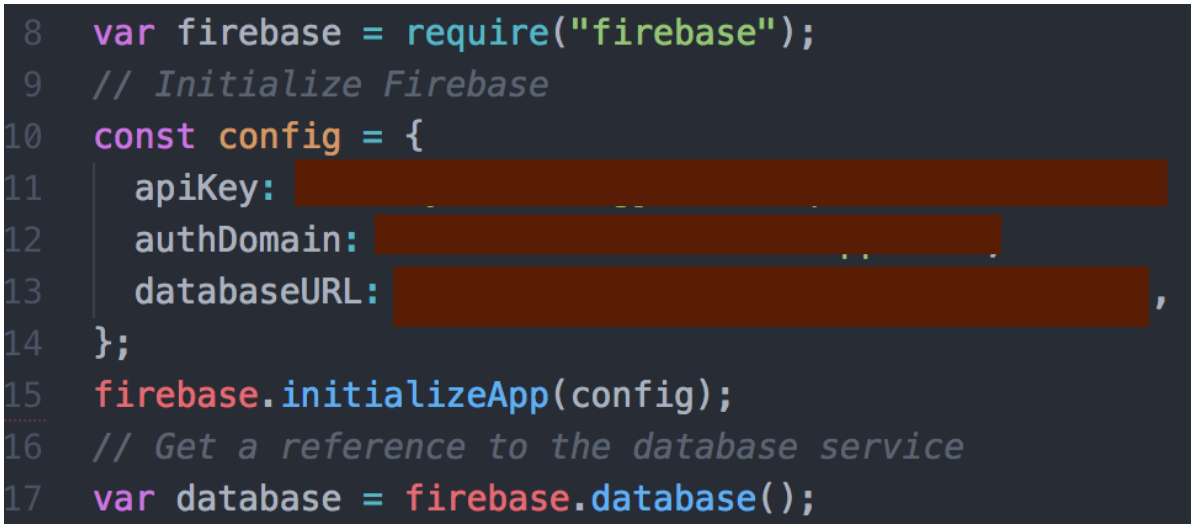


Next, we see a Firebase authenticated user, which was created for testing purposes.



Connecting the database to the server took some time and understanding. However, the actual setup required only 7 lines of code.

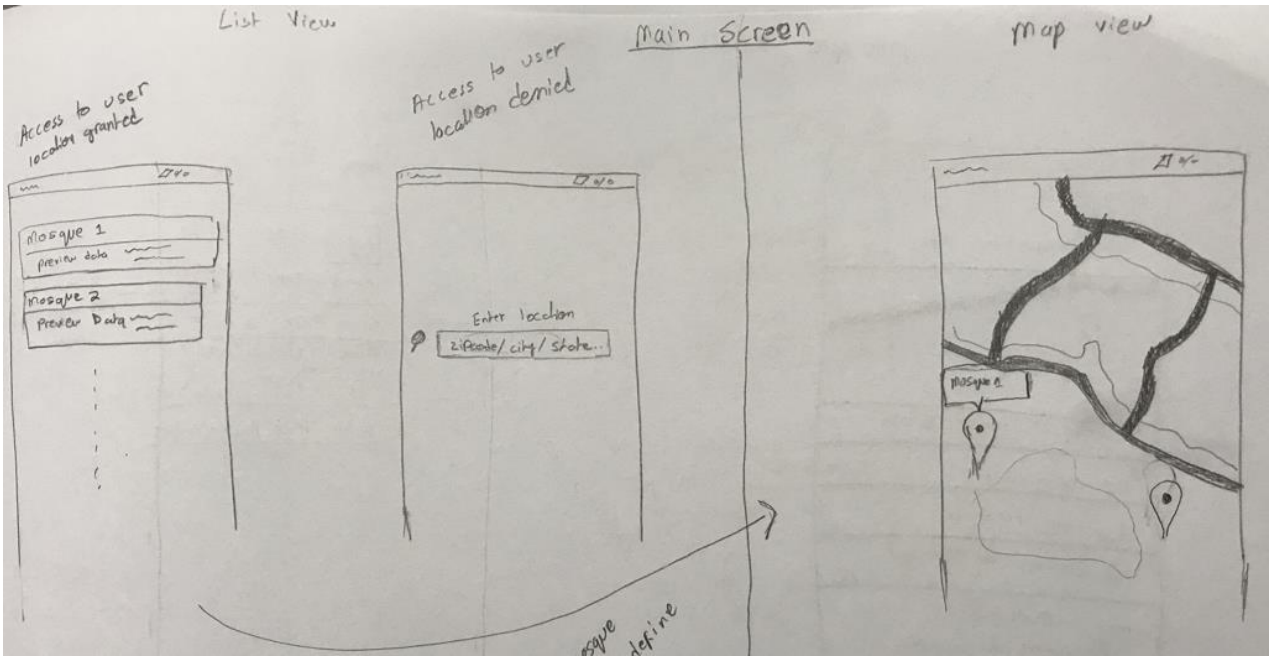
Figure 2.1 — Firebase Database to Server Connection



Once the database was initialized and a reference was created, it was quite easy to interact with the nodes within the database.

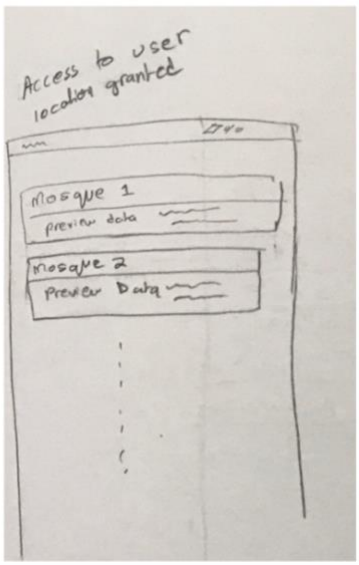
Another component of the app structure was creating the app mockups. Drawing the mockups also helped with finalizing the database design.

Figure 2.2 — Elementary App Mockup Designs

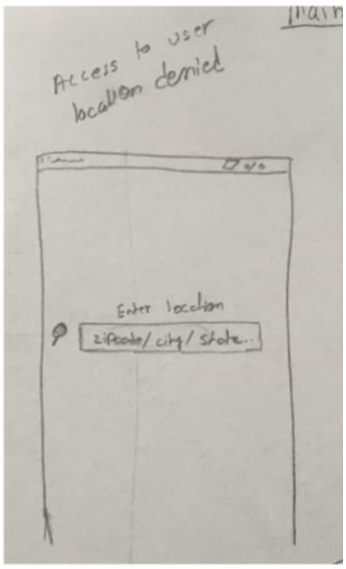


Landing Screen | Location Access Dependent

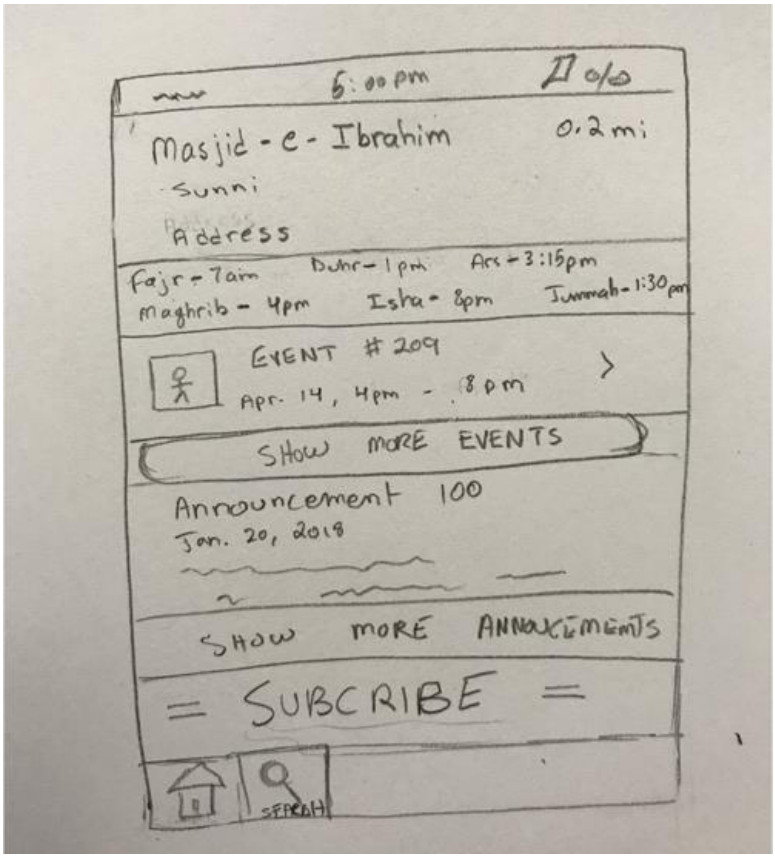
Location Request Accepted



Location Request Denied

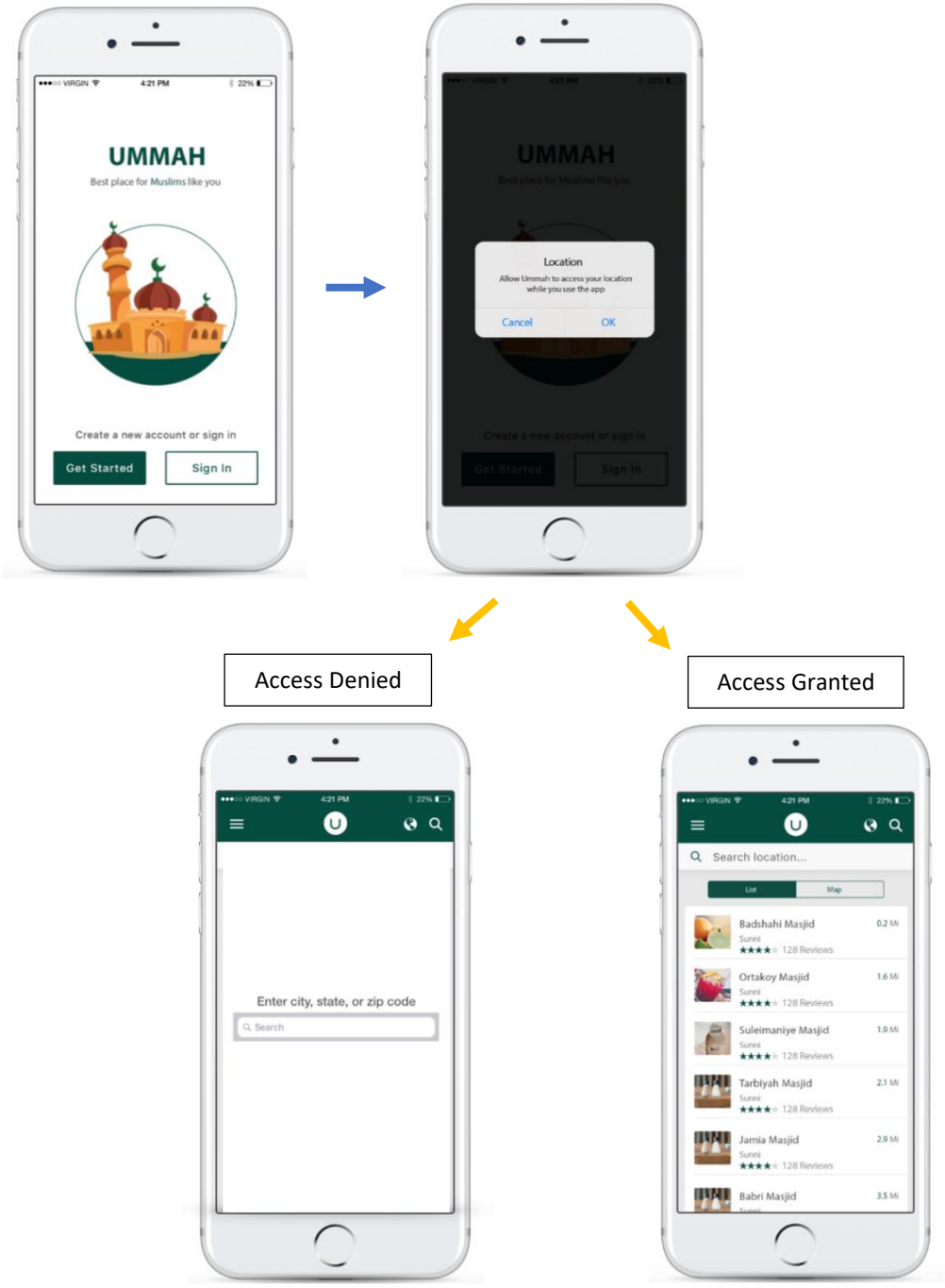


Mosque Detail View Screen

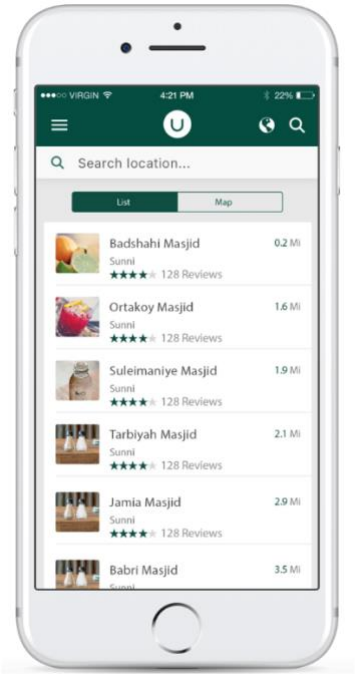


These designs were later modified and then brought to life using Adobe Photoshop and marvelapp.com [2], a platform for bringing prototype designs to life. The design is shown in the figures below.

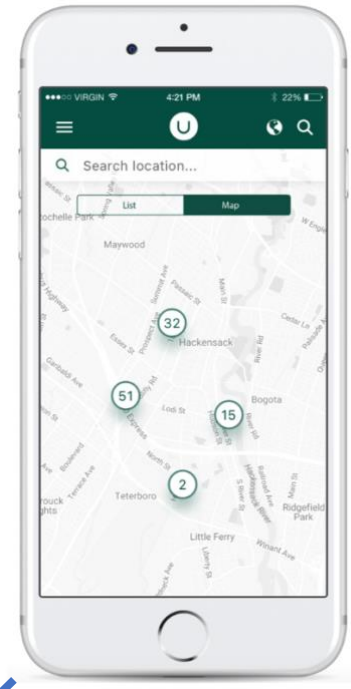
Figure 2.3 — Ummah Application Design and Flow



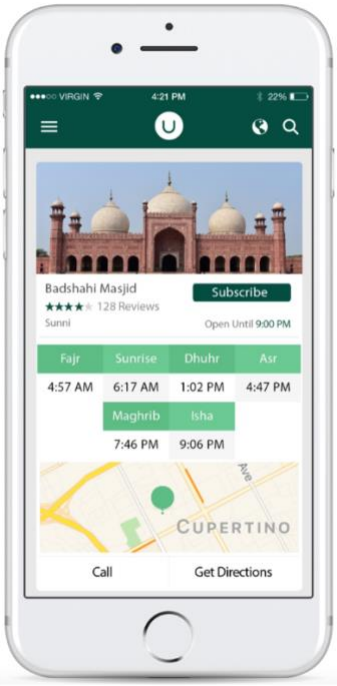
List View



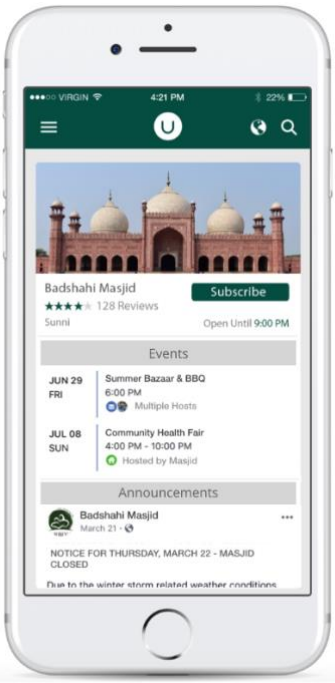
Map View



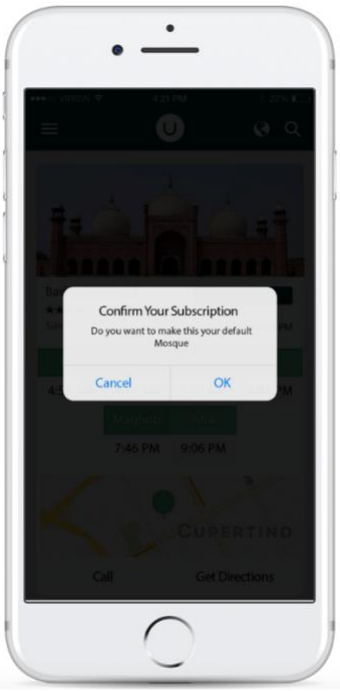
Detail View



Detail View



Subscribe

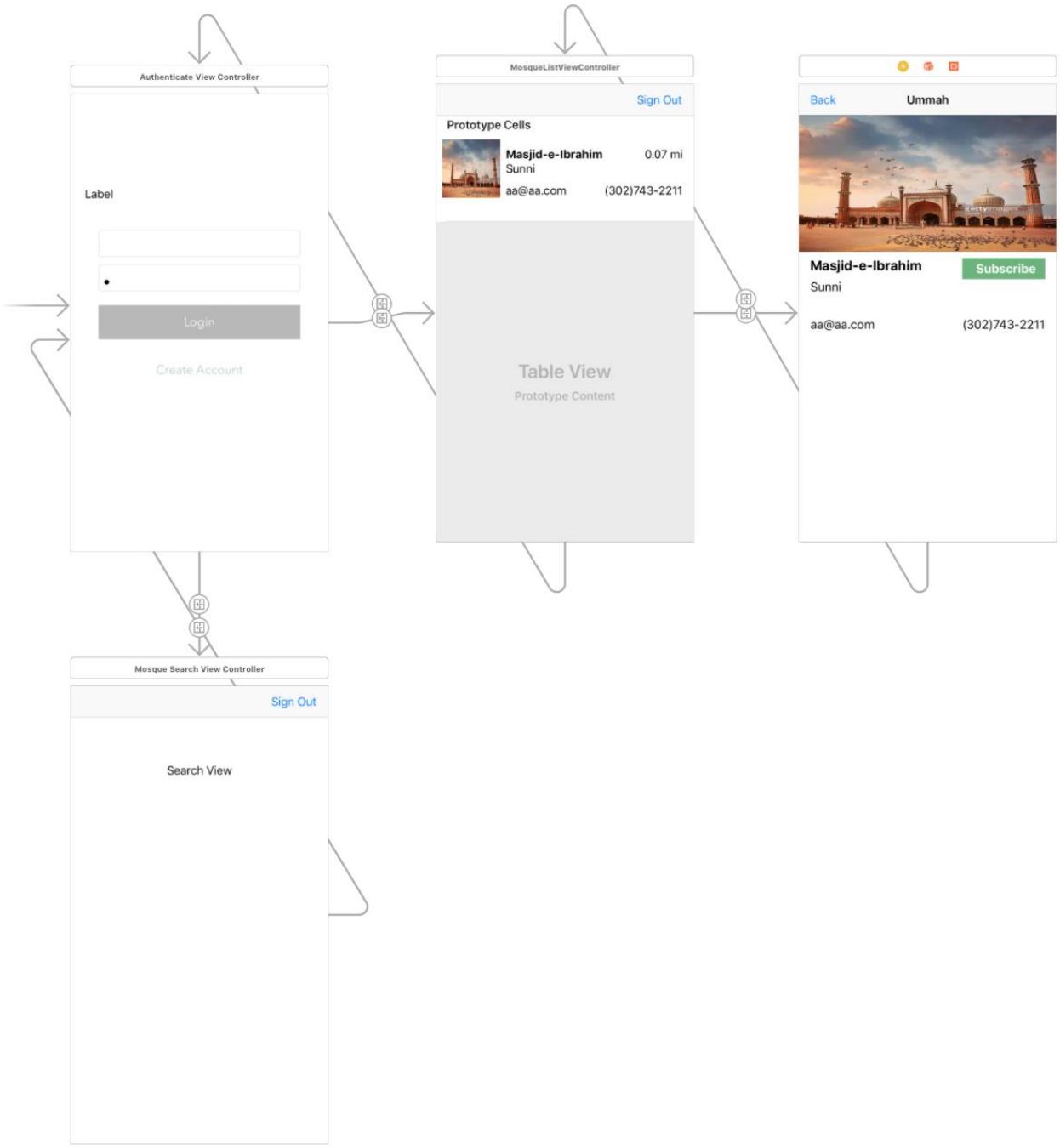


Mobile App:

After setting up the main components of the Ummah system, it was time to start developing the mobile application. Each feature within the app started with the development of the UI UX component, and then adding functionality to it. The development steps closely followed the Application Flow shown in Figure 2.3. Thus, first the user login was implemented, which required sending a post to the server to allow authentication from Firebase. Then the landing screen logic based on location access was implemented. The logic requests permission to access user's location information, and lands user on the appropriate screen depending on whether permission was granted or denied. These screens are namely, the Mosque list view, and Mosque search view.

The list view was the final component with completed functionality as a part of the porotype.

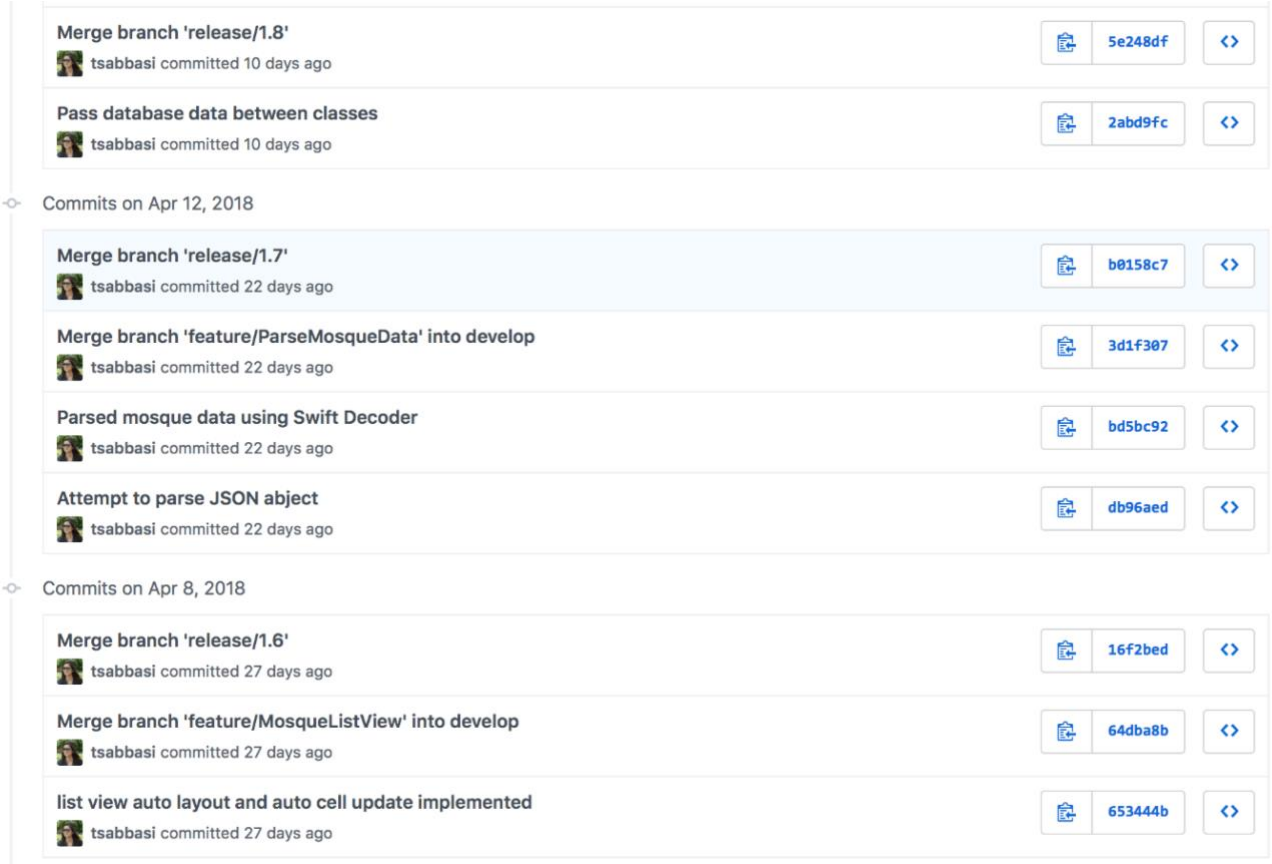
Figure 2.4 — Storyboard on Xcode



Version Control:

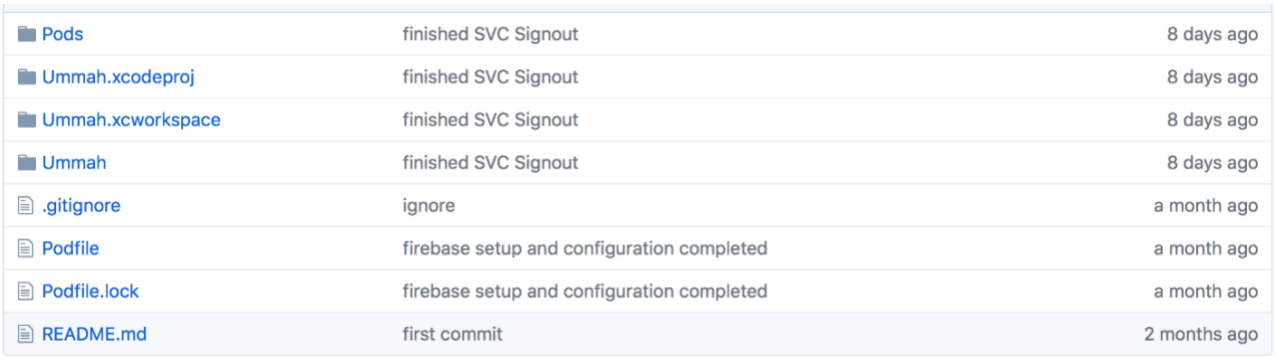
Git was used to track and store all the versions for the Ummah system. Specifically, GitHub's feature release process was incorporated throughout the project [3-4]. The feature release process made the implementation for this entire project a lot easier. This process forces the developer to break down the various milestones to be broken into smaller pieces. For example, the Ummah mobile app had the following implementation milestones; User login, Mosque list view, and search view. The mosque list view implementation required numerous steps including, retrieving Mosque data from the server, parsing the data, putting the data in the appropriate struct, etc. These steps were defined as releases in the GitHub feature release process.

Figure 2.5 — Example of Feature Release for Ummah



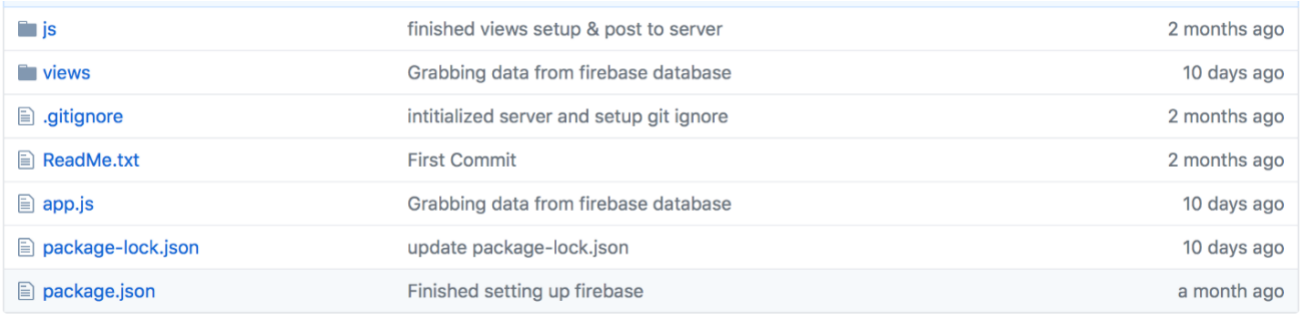
Important Note: The code for the Ummah system was pushed to a private repository on GitHub, due to which the link for the GitHub repository cannot be shared at the moment. In the future, there will be a public version made available for any contributors or interested party. However, the repository views are shown in the figures below.

Figure 2.6 — Ummah Mobile App Repo



Pods	finished SVC Signout	8 days ago
Ummah.xcodeproj	finished SVC Signout	8 days ago
Ummah.xcworkspace	finished SVC Signout	8 days ago
Ummah	finished SVC Signout	8 days ago
.gitignore	ignore	a month ago
Podfile	firebase setup and configuration completed	a month ago
Podfile.lock	firebase setup and configuration completed	a month ago
README.md	first commit	2 months ago

Figure 2.7 — Ummah Backend Repo



js	finished views setup & post to server	2 months ago
views	Grabbing data from firebase database	10 days ago
.gitignore	intitialized server and setup git ignore	2 months ago
ReadMe.txt	First Commit	2 months ago
app.js	Grabbing data from firebase database	10 days ago
package-lock.json	update package-lock.json	10 days ago
package.json	Finished setting up firebase	a month ago

FUTURE WORK

As of April 25, 2018, the prototype was completed. This prototype is a minimal viable product (MVP), which serves to get the concept of the app across to the target audience. The MVP consisted of the Ummah system database structure, stack implementation, and finally, the mobile app screens/views that were shown in Figure 2.4. Given that, there is quite a bit of work still left to do. The figure below shows the timeline that was followed for delivering the prototype.

Figure 2.8 — Ummah Capstone Project Prototype Timeline

The text highlighted in green represents the milestones that were completed. In yellow are those milestones that have a few more components that need attention/modification. The text that is not highlighted, represent the milestones that still need to be completed.

Milestone/Deliverable	Due Date
Prospectus Submission	Feb. 4, 2018
App Mockup & Database Structure	Feb. 15, 2018
Database Structure	Feb. 8, 2018
First Update Presentation	Feb. 16, 2018
Server Setup	Mar. 1, 2018
Stack Implementation	Mar. 20, 2018
Ummah App API's	Mar. 22, 2018
User Login Implementation	Mar. 22, 2018
Second Update Presentations	Mar. 23, 2018
Abstract	Mar. 28, 2018
UI UX Development	Mar. 29, 2018
Mosque Owner Dashboard API's	Apr. 3, 2018
Mosque Owner Dashboard	Apr. 8, 2018
List of Mosques	Apr. 10, 2018
Default Mosques Implementation	Apr. 15, 2018
Map View	Apr. 18, 2018
Final Presentation	Apr. 26, 2018
Final Artifact	May. 5, 2018

Aside from completing the application, there are some ideas to expand the Ummah application as well. These include, 1) allowing users to find nearby halal restaurants or grocery stores, and 2) developing this "app as a service" model for more businesses such as churches, parks, schools, etc.

Find Nearby Halal Restaurants/ Grocery Stores:

When Muslim individuals search for Halal[4] restaurants, rarely ever do they get the desired results. For example, while I was touring the Bay Area in California, I noticed that there were many South Asian Halal restaurants. The surprising part however was that even in such a diverse area, searching for these restaurants online (i.e. Google search) was still a challenge. The keyword "Halal" usually had little to no effect on the search results, as it returned a list of Middle Eastern or South Asian restaurants regardless of whether they served Halal food or not. This forces the individuals to call and ask the restaurants in advance prior to selecting them. The Ummah app can implement "Find a Halal Restaurant/Store" feature similar to how "Find a Mosque" is being implemented.

"App as a Service" model for more Business/ Organizations:

Through this model people will be able to perform location-based search for parks. Then, they can "Subscribe" to their favorite park (or parks) and get updates about any events or announcements posted by that park, directly on their phone. Similarly, this model can be extended to churches, schools, and many other areas.

[4] Halal: Meat prepared as prescribed by the Muslim law [8].

CONCLUSION

Ummah is a one stop shop for the Muslim community. It allows Muslim individuals to not only search for Mosques, but also subscribe to any one of the Mosques. This allows the users to keep up to date about events, holidays, and announcements from that Mosque. Additionally, users will be able to easily find Halal restaurant in a specified location — a task which currently presents a plethora of difficulties within a non-Muslim majority country.

I look forward to delivering the Ummah app to the Muslim community in the near future.

ACKNOWLEDGEMENTS

There are many people that contributed to my success on this project. First, I would like to acknowledge Dr. Tomasz Smolinksi for his wonderful guidance throughout the project development. I would also like to thank my brother, Taha Abbasi for his help and insight on the technology within the project. I would also like to thank him for inspiring me to push the boundaries and go after opportunities that I otherwise would not have – he, along with those opportunities served a great role in the success of this project.

I also want to thank my parents for being there for me and for all their prayers. Specifically, I am grateful for my mother, Nuzhat Abbasi waking up at 3 in the morning on numerous occasions to make sure that I was studying and/or working on the project; and my dad, Salahuddin Abbasi for making breakfast for me as I studied and for always being concerned about my well-being. The sacrifices made by both my parents have been the biggest motivation for me to put forward my very best work.

I would also like to thank my fiancé, Haris Usmani for proofreading my material countless times, for the motivation, and for the love and compassion that kept me sane throughout the whole journey.

Lastly, I would like to thank all the professors within the CIS department for challenging me and motivating me to do my very best. Thank you Dr. Rasamny, Dr. Holness, Dr. Smolinski, Dr. Kong, and the rest of the CIS faculty for all your help. I am blessed to have encountered some of the most brilliant professors. My journey in completing my degree as well as the completion of this project has been one I feel truly proud and accomplished about. Thank you all.

REFERENCES

[1] Conrad Hackett. 2015. Religious Projections 2010-2050. Retrieved from http://www.pewforum.org/2015/04/02/religious-projections-2010-2050/.

[2] Marvel App. Retrieved from https://marvelapp.com.

[3] GitHub. Retrieved from https://github.com.

[4] Daniel Kummer. Git-flow cheatsheet. Retrieved from https://danielkummer.github.io/git-flow-cheatsheet/.

[5] Wikipedia. Muslim. Retrieved from https://en.wikipedia.org/wiki/Muslim.

[6] Wikipedia. Mosque. Retrieved from https://en.wikipedia.org/wiki/Mosque.

[7] CBN. What is the Muslim Understanding of "Ummah"? Retrieved from http://www1.cbn.com/onlinediscipleship/what-is-the-muslim-understanding-of-%22ummah%22%3F.

[8] Oxford Dictionaries. Halal. Retrieved from https://en.oxforddictionaries.com/definition/halal.

[9] Manish Shewaramani. 2017. How to Choose Your Mobile App Technology Stack. Retrieved from https://www.credencys.com/blog/how-to-choose-your-mobile-app-technology-stack/.

[10] Peter Wayner. 2017. MEAN vs. LAMP for your next programming project. Retrieved from https://www.infoworld.com/article/2937159/javascript/mean-vs-lamp-for-your-next-programming-project.html.

[11] Mean.IO. Retrieved from http://mean.io/.

[12] DB-Engines. System Properties Comparison Firebase Realtime Database vs. MongoDB. https://db-engines.com/en/system/Firebase+Realtime+Database%3BMongoDB

[13] Firebase. Retrieved from https://firebase.google.com/.

[14] Alamofire Library. Retrieved from https://github.com/Alamofire/Alamofire/blob/master/Documentation/Usage.md.

ADDITIONAL REFERENCES

Adnan Rahić. 2017. “Hello World!” app with Node.js and Express. Retrieved from https://medium.com/@adnanrahic/hello-world-app-with-node-js-and-express-c1eb7cfa8a30.

Shahid (UnixRoot) Shaikh. 2015. Render HTML file in ExpressJS. Retrieved from https://codeforgeek.com/2015/01/render-html-file-expressjs/.

Maxim Shoustin. 2014. Stack Overflow: How to validate an e-mail address in swift? Retrieved from https://stackoverflow.com/questions/25471114/how-to-validate-an-e-mail-address-in-swift.

Chris Esplin. 2016. Firebase 3.0: Node Client. YouTube Video. (8 July 2016). Retrieved from https://www.youtube.com/watch?v=G_FlX41qADE&t=288s.

The Bearded Programmer. 2016. Basic iOS Tutorial 2 - Creating a button and making it click through to another view. YouTube Video. (18 April 2016). Retrieved from https://www.youtube.com/watch?v=y_5YFTvpQsM.

Swift documentation. Retrieved from https://developer.apple.com/documentation/swift.

Lets Build That App. 2017. Parsing JSON Just Became Super Easy in Swift 4 with Decodable. YouTube Video. (1 July 2017). Retrieved from https://www.youtube.com/watch?v=YY3bTxgxWss.

cartoonsmart. 2017. How to Parse JSON data with Swift 4. YouTube Video. (17 November 2017). Retrieved from https://www.youtube.com/watch?v=XqNvHHAi08s.

David Siegel. 2017. Stack Overflow: Pulling data from TicketMasters API (Swift 4.0/Decodable). Retrieved from https://stackoverflow.com/questions/46738857/pulling-data-from-ticketmasters-api-swift-4-0-decodable?rq=1.

Ben Scheirman. 2017. Ultimate Guide to JSON Parsing with Swift 4. Retrieved from https://benscheirman.com/2017/06/swift-json/.

ACM. Association for Computing Machinery: Citation Style and Reference Formats. Retrieved from https://www.acm.org/publications/authors/reference-formatting.