ReBoot.

Technological Feasibility

Date: November 8th, 2019

Team Members:
Bianca Altman, Jennie Dallas,
Victoria Leafgren, Isai Martinez,
Cindy Valentin

Sponsor: Dr. Emery Eaves
Co-Sponsor and Mentor: Dr. Eck Doerry

Table of Contents

1.0 Introduction

Over 2 million Americans struggle with opioid addictions, costing the US $78.5 billion a year for healthcare, lost productivity, and addiction treatments. Every day, more than 130 people in the US alone die from opioid overdoses. Between 1999 and 2014, the rate for opioid use disorder for women at delivery quadrupled, with it currently at 6.5 per 1,000 deliveries. With these pregnancies comes the risk of neonatal abstinence syndrome in the baby, which can cause cognitive or behavioral impairments. In 2017, costs for educational services provisions were estimated at over $16 million in some states.

Pregnant women with opioid addictions who want to get clean and keep their babies are expected to go to Methadone clinics to receive treatment daily, in-person counseling appointments, and more every week. In some cases, women are expected to attend up to 17 appointments in a single week. For women that are in rural areas where the closest clinics and services are miles away, it makes this process significantly harder as they must have a way to travel to and from these appointments. In addition, many of these women fear the stigma surrounding being seen attending recovery-based appointments, especially while being pregnant. The goal of our project is to create a powerful tool for expecting mothers in recovery, so they have a safe space to learn and connect with others in the same situation and help lead them to a successful recovery.

We want to help solve this problem with a mobile application that can assist these women with their learning and receiving general support, without having to worry about traveling long distances or facing the stigma of being seen as only an addict, not a pregnant woman that is trying to change and actively learn. The expecting mothers can do this all while knowing they are using a safe and secure application. For our solution, we propose a completely anonymous application that will give women a way to learn about their pregnancy and what they should be expecting personalized to what stage of pregnancy they are in. This will also serve as a place for these women to express how they are feeling, receive, and give advice and support to other mothers knowing they are all dealing with similar issues.

To create this application, we are working with our client Dr. Emery Eaves. Dr. Eaves is a medical anthropologist and professor at Northern Arizona University, whose research focuses on reducing the opioid epidemic in Arizona and is currently working to improve the recovery system for pregnant women. After multiple meetings with Dr. Eaves, we have come up with a set of challenges we will be discussing solutions for in this document.

To begin our planning of this application's development, this technological feasibility document provides an analysis of available technologies for our design choices in creating our solution. The document is organized by first listing out our major technological challenges we are facing or are expecting to face with this application in section 2. Then, in section 3, we will be analyzing alternatives for the technologies we may use and based on a set of criteria we will choose the best solution. In section 4, we describe how our technologies integrate together.

2.0 Technological Challenges

The goal of this mobile application is to provide a place for pregnant women that are recovering from opioid addiction to learn and maintain their schedule around their recovery. This secure environment is to be a place where these women may share and receive help, keep track of their pregnancy journey, and be able to learn more about their pregnancy. Considering this, we have come up with the following challenges we will be facing while implementing this app:

- **Cross-platform support** – Ideally, our solution will produce both iOS and Android applications, so any technologies we choose must either directly support cross-platforms or be able to be used in a cross-platform supporting framework.
- **Tracking individual user's progress and tailoring their application** – This is an application that is specific to each user's own experience. So, this means we want the application to display the user's tailored information about the week of pregnancy they are in to be what they see when they open the application.
- **Surveys and question and answers** – A big part of this application is user input. There should be a survey that is consistently up on the homepage that asks the user how they are currently feeling. To keep users involved and actively learning, there should be a way that a user can be prompted to answer some questions based on a learning activity. Therefore, we need a way to provide a mechanism for distributing surveys, distributing learning modules, and gathering the results from these activities.
- **Video plugins** – An ideal way of learning for this application is videos. Videos are a lot easier to follow along with and can maintain the user's attention better than a long, black and white article. So, there must be a way to put videos into the application and store them.
- **Forums** – Another major component of this application is a forum. The users are going to be split into groups upon signing up. With this, there will be an option to see a forum. In the forum, users may ask questions, discuss, and give general support to one another, so the forum must allow users to post and comment. With this as well will be an information desk with a similar implementation, except the only person that may answer questions is a medical professional.
- **Calendar** – This application is going to be a way for the women using it to maintain their recovery, so we want a functioning calendar that allows them to input appointments, medication reminds, etc. into the calendar so they can receive notifications reminding them of these events.
- **Notifications** – As stated before, this application is to help maintain recovery. These notifications can remind users of appointments, new learning modules, other women posting, and more. Notifications will help keep the user interacting with the application.
- **Web application for administrative use** – This will be an application that has as little restrictions as possible in order to avoid users feeling distrusted by the application's administrators. However, with this requires monitoring. The web application will allow

administrators to manage content, monitor and manage activity in the forum, add new materials, and prevent negative social dynamics from developing.

- **Research analytics** – As this application is going to be used for a grant, understanding user activity can help with the future progress of this application. For example, being able to see the data and trends in video viewing times, responses to surveys, and traffic to different sections of the application are all relevant data that administrators would be interested in.

The core frameworks and technologies we choose must collectively support all of the above functionalities and integrate with each other. To accomplish properly implementing these requirements, we analyze multiple alternatives for technologies we may use in the next section. The following section will go in the analysis of the technology categories we will be looking into, which includes a mobile application framework, database technology, web application framework, research analytics software, and forum tool plugins.

3.0 Technology Analysis

As the technologies we choose are significant to part of the success of this project, we did some basic research to narrow down some options for each technology we are going to analyze. Each technology is being analyzed based on a set of criteria we decided is most important in deciding which technology is the best fit for this project. Using the criteria, we rate each technology on a scale of 1-5, 5 being the best.

As our mobile application framework will set up the decisions for our other technologies, the first item we will look into is the mobile application framework.

3.1 Mobile Application Framework

Mobile application frameworks are software frameworks that specifically support mobile application development. These frameworks provide tools to display all content in the application, access mobile device hardware, and handle communication with a server-side backend. Our project's entire mobile application features depend on the capabilities of the framework we choose, making this decision to be fundamentally shaping what our application will be and provide. With that, our first analysis is on which mobile application framework we will use. For our project, the mobile framework we choose will need to provide the following:

- **Ease of use and support** – The framework should have an active community with easy to find documentation. We will need a variety of different plugins and libraries.

  Overall, every team member should be able to grasp how the framework can be used, being able to find documentation and tutorials on the framework. We want this application to be supported even after this project is over and is easy to maintain and update as time goes on.

- **Security** – Any information that is stored should be completely secure. No user should be worried about any information they are putting on this application being leaked due to backend issues. Therefore, the framework we choose should provide some authentication and security plugins or built-in capabilities.
- **Ability to interact with phone notifications** – Notifications are very important to keep users interacting with this application and keeping up with any important dates and times. The framework must be able to send notifications on both iOS and Android. Therefore, the framework we choose should provide plugins compatible with iOS and Android to implement notifications.
- **Way to implement surveys / questions and answers** – There should be a way to implement different user input buttons for different variations of surveys and question / answer formats. Therefore, the framework we choose should provide the components necessary for implementing a typical survey format.
- **Video plugin** – Displaying videos onto this application is vital, as our client wants videos as the primary information medium. Therefore, the framework we choose should provide a plugin for embedding videos into a mobile application.

To decide which frameworks to test, we searched for popular, cross-platform frameworks. We wanted to find frameworks commonly used, as this would mean they are generally more trusted and can provide us with more documentation. Through our research, we decided on these four frameworks: Ionic, React Native, Apache Cordova, and Xamarin. The following subsections contain the analysis of four frameworks that all support cross-platform applications we found that would be the best options for this project based on the criteria listed above.

3.1.1 Ionic

Ionic is an open-source cross platform application development framework, meaning that it provides a way to create a single codebase that can be built into iOS, Android, and web applications. Minor adjustments in the code are needed to adapt functionality for the different operating systems. Ionic applications are not compiled to native code but are instead wrapped in a web view. Ionic was initially released in 2013 and was created by a few developers at a company called Drifty. In order to properly evaluate Ionic, we installed its command line interface and created a simple project. This allowed us to get a feel for how to build and preview projects. Additionally, we searched the internet for up-to-date examples of specific functionalities we're looking for.

- **Ease of use and support: 4/5** – Finding tutorials and documentation for Ionic was not too difficult. While their website features solid documentation for different plugins and built-in components, there are not a lot of Ionic tutorials other than the standard "your first application" tutorial. However, on YouTube there were a number of tutorials for various example implementations that were up to date with Ionic 4. When performing Google searches to research Ionic, there were more relevant results and helpful information and code examples. For complete Ionic beginners, this could make learning the framework easier. From this analysis, our team might still have to spend a good amount of time teaching ourselves how to use Ionic, but we won't be at a complete loss for direction.

Ionic has an abundance of plugins and ability to be integrated with tools we need for our application. For support, Ionic provides links to a community forum as well as a Slack channel to speak with other developers. Additionally, Ionic provides a way to submit a ticket for professional support, although it is unknown how quick or helpful this feature is.

Ionic also provides a free DevApp application that allows developers to preview their application directly from their own phone without physically connecting it to their computer. Applications developed in Ionic can also be previewed in a web browser, and it automatically updates whenever changes to the codebase are saved. This makes previewing and testing applications very easy.

As previously mentioned in the introduction, Ionic apps are wrapped in a web view instead of being compiled to native code for different operating systems. For us as developers, this can potentially make our lives easier by minimizing the amount of operating system specific code we need to customize for iOS versus Android.

- **Security: 5/5** – Ionic provides various plugins for secure user authentication. The Auth Connect plugin is a native plugin, meaning developers don't have to go through third parties, and as a result, this way of authentication is more secure. There are also secure plugins provided that authenticate using biometrics. Aside from these plugins, the security of an application mostly depends on our own design and implementation.
- **Ability to interact with phone notifications: 5/5** – Ionic provides a plugin for sending and receiving notifications, which can be easily integrated with a database such as Firebase to manage notifications to users.
- **Way to implement surveys / questions and answers: 3/5** – While in our research we did not find Ionic plugins or tools made for the purpose of creating surveys, there are examples of other developers' previous implementation of surveys using Ionic. The examples were created with an older version of Ionic; however, their main features were not centered around plugins or Ionic tools that could become outdated and unusable. Our team would be able to implement something similar using the latest version of Ionic.
- **Video plugin: 5/5** – For our purposes of supporting iOS and Android applications, there were two video plugins that would be compatible with our project. The first is a plugin specifically for YouTube videos, and the second is one for streaming media, whether this is video or audio. The Streaming Media plugin takes in a URL and streams the media from that URL. Also, possible to show different features based on if the user is an admin or regular user, so we could use this to distinguish between medical professionals and admins on the web application.

3.1.2 React Native

React Native is a fairly new, open source framework in development by Facebook for mobile application development, only being released in March of 2015. It is a JavaScript framework that can be used to implement cross-platform mobile applications. The JavaScript based application will interact with the phone OS asynchronously, so more resource-intensive operations can be done separately from the UI. To evaluate React Native, we downloaded the framework and tested with a simple button testing program, as well as research done on documentations, user reviews, and tutorials:

- **Ease of use and support**: **4/5** – Majority of the same code and components can be used for writing Native iOS applications, Native Android applications, and web applications. React Native renders the application using real UI components to build the JS code, so the rendering is not done with web views. Therefore, development can go a bit faster with quicker simulator updates. It is important to note that in order to build the iOS app, we will need access to a Mac. This can cause some problems as not all members have access to a Mac, but some do.

  To test this framework for basic use, we downloaded the framework. Running it is simple and just takes a few terminal commands to install React Native and to run it. A nice aspect is that all members can use their favorite text editor for writing the code. Using Expo tools, the application can be tested in real time on a phone. There is plenty of documentation on the React Native website for basic components, making the implementation go a lot smoother. We created a simple button tester for Android that would send alerts when buttons are pressed. The rendering takes little time, so testing goes fast. Overall, it is easy to use once playing with it for a while but gets more complex once the project grows.

  While the majority of the code can be written in one code base, there will have to be statements checking for the operating system anytime component is operating system specific or if we use functions that are interacting with the hardware, such as notifications. In most situations it is ideal to maintain separate folders for Android, iOS, and the overall app under the main app folder.

  This is a framework developed by Facebook, and while it appears to be growing, it could just be cancelled by Facebook at any time. Which does not seem likely but is something to consider. React Native is a fairly new framework, which can be a positive and a negative. Because of this, there are monthly updates being implemented to the framework. This can mean the application will need frequent updates, but this shows that the framework is growing and adding new features. As well, not all bugs and glitches have been addressed yet. This makes maintenance on the application something that will have to be done more frequently.

- **Security: 5/5** – React Native does have some libraries that may be installed for security implementation, such as Keychain which gives a wide variety of support in implementing passwords, biometrics, and more with the option to also work with a server.

- **Ability to interact with phone notifications: 4/5** – There is a package to download that will allow for enabling push notifications on iOS and Android, but the solution is implemented different for each operating system. From looking into examples, most people have two separate folders for iOS and Android to implement notifications. While trying to test this, we found using React Native to be quite complicated to learn and had trouble finding thorough documentation on errors we were getting. As of right now, we are unable to get this to work, even when using someone else's code. Therefore, we know push notifications can be implemented, but we are taking into consideration for this rating that we are having trouble to do it ourselves.
- **Way to implement surveys / questions and answers: 5/5** – As most of React Native uses premade components, there are components that have been created for user input that could work for surveys and question and answer formats. As well, there is a library for survey implementations in JS.
- **Video plugin: 5/5 –** Similar to the other components that are not already included in React Native, third-party plugins are supported. Specifically, for videos, there is a package that has ways to implement videos for iOS and Android.

3.1.3 Apache Cordova

Apache Cordova is an open source cross platform mobile application development framework. It aims to use one codebase to build applications for iOS and Android. Originally created and released by Nitobi in 2009, Cordova was acquired by Adobe Systems in 2011 and released as an open source project. In order to evaluate Apache Cordova, we installed it from the command line and created a simple project to build and test. Additionally, we searched online for relevant and up-to-date examples of functionality we need for our application.

- **Ease of use and support**: **2.5/5** – During our research, it was more difficult to find information and examples of different application implementations in Apache Cordova. The documentation on their website provides good insight into their plugins and built-in capabilities but provides no tutorials aside from the "your first application" tutorial. The number of tutorials and helpful examples on other websites like YouTube were minimal.

  Additionally, the process of previewing and testing applications takes a lot of time and sometimes does not work. We had difficulties using the Android emulator from the Cordova command line interface, and when researching these difficulties online, found that many other people had the same problem. It is also possible to preview the application from the browser, but any changes made to the code will need to be rebuilt before seeing them.

  Apache Cordova provides plugins for development, but there is no clear promotion or suggestion of which plugins are reliable. Instead, all third-party plugins are included in the search results. Cordova provides links to a couple forums for developers to get help from each other, but Cordova itself does not provide any professional support. The overall impression that Cordova gives is that developers will have to struggle a lot on their own to find solutions to their problems and development needs.

- **Security: 3.5/5** – Apache Cordova does provide authentication plugins, including one that connects to the Firebase database, and one that is used for biometric authentication, among others. However, since these are all open source contributions, some were last updated over a year ago. There is not a recommended set of plugins, so it is not clear how well these plugins are supported, and their security could be questionable.
- **Ability to interact with phone notifications: 5/5** – Searching for push notification plugins will show that there are multiple plugin options for this feature, including one that uses Firebase to manage notifications.
- **Way to implement surveys / questions and answers: 1/5** – In our research, we were unable to find a solid, working example of a survey in an application implemented with Cordova. We did find an open source project on GitHub that was described as being an application with survey features but were unable to get it to work on our own computer. This suggests that it is entirely possible, but we would have to find a way to implement it ourselves. We were also unable to find other templates or plugins with survey templates.
- **Video plugin: 5/5** – There are two video plugins that would be compatible with our project. One is a plugin specifically for YouTube videos, and the second is one for streaming audio and video media. The Streaming Media plugin takes in a URL and streams the media from that URL.

3.1.4 Xamarin

Xamarin is a cross-platform mobile application framework based on Microsoft technology released in 2013. Xamarin uses only C# and is natively compiled. Native libraries and C# are used wrapped in a .NET layer. Some applications that use Xamarin include Captio, Oro, and MRW. Our analysis on Xamarin was done primarily through user reviews, the Xamarin website, and tutorials:

- **Ease of use and support: 2/5** – It is expected that any development using Xamarin will be done using Windows, so running is also under this assumption. This is an issue as there is a member with a non-Windows computer. iOS and Android are compiled differently and are done with two different products, which is an addition to making using this framework more complicated.

  To get the native feel, writing the code is on separate frameworks for iOS and Android. There are different requirements for using VisualStudio (the IDE used for these frameworks) for Mac and Windows. After practicing with some code, the general implementations are not too complex if the programmers have some C# experience. The overall feel with the IDE is a little more complex than some previous frameworks tested.

- **Security: 2/5** – There are two different ways to implement security on the applications using Xamarin, one for iOS and one for Android. This makes security harder to deal with, as we will have to consider different references and tools for keeping everything secure. The authentication process is a bit complex compared to other frameworks.
- **Ability to interact with phone notifications: 5/5 –** Xamarin does provide access to platform-specific APIs and supports linking with native libraries, so there are implementations that allow for push notifications on iOS and Android.
- **Way to implement surveys / questions and answers: 1/5** – Through research online, we found little to no solutions for this. There are ways to implement user input and after some testing, we could probably find a solution. However, considering previous frameworks have premade components or clear solutions for this implementation already online, trying to deal with this is not good for time.
- **Video plugin: 3/5** – Similar to push notifications, there are plugins that support video playing on both platforms. However, if this does not work for us, there are compatibility issues with using third-party libraries and tools.
- **Support and Robustness: 2/5** – Xamarin is owned and supported by Microsoft, so we do not see this framework going anywhere any time soon. Maintenance and updates can be simple with deploying changes or updates to the source file, which will update both iOS and Android applications.

  However, there are different ways to approach creating applications, used with Xamarin.iOS, Xamarin.Android, and Xamarin.Forums. Xamarin.Forums allows for UI design to be shared across both platforms, but this can compromise the application feeling like a native built application. But it would be an added time and coding issue to have to write specific UI for Android and iOS.

3.1.5 Chosen Approach

The following table is a summary of the results of each framework based on the set criteria we chose:

Mobile Application Framework/Criteria	Ionic	React Native	Apache Cordova	Xamarin
Ease of Use	4	4	2.5	2
Security	5	5	3.5	2
Notifications	5	4	5	5
Surveys / Questions and Answers	3	5	1	1
Video Plugin	5	5	5	3
Total	27	26.5	17	18

Considering our evaluation resulted in a very close score, we will be performing further testing on Ionic and React Native to see which framework is capable of fulfilling our criteria, rather than just basing our final decision on primarily internet research and simple tests. We are currently leaning towards Ionic but have not made our final decision yet. With the general idea of “write once, run anywhere” for Ionic and “learn once, write anywhere” we think we will be able to save more time not worrying about general UI having to be written for each platform (iOS, Android, and the web app) with Ionic like it would be with React Native. Ionic is a better platform for creating one codebase for multiple operating systems, as opposed to React Native, which requires more operating system specific customization.

Also, with our research we saw that there is a steep learning curve for React Native, and we want to spend as much time on implementation as possible rather than learning about the framework on top of implementations. However, going with a React Native could provide an application that has a more native feel for each operating system.

3.1.6 Proving Feasibility

In order to prove that our framework will provide all the functionality we need for our application, we plan to create a sample application with the following capabilities:

- Plays video

- Sends notifications
- Provides survey and records answers
- User authentication

We envision the sample application will have a basic login screen that allows us to demonstrate that users will be able to enter the application with a valid username and password. Inside the application will be a sample video and survey that users should be able to interact with. Additionally, we will demonstrate that the application is able to send push notifications by using a button that triggers a notification being sent.

Next, we will be deciding which database will work best based off of the decision that Ionic or React Native will be our framework.

3.2 Database

Databases are needed to collect and store information in an organized and easily accessible and updatable manner. This application will need to not only store local user data, but also different file formats and users, so we need to have a secure database that can grow with this application. With our application's requirements in mind, we decided to analyze and rank potential databases with the following criteria:

- **Works with chosen framework** – The database technology we choose is required to work with the framework we have chosen to develop this application on, otherwise it is useless. Therefore, the database we choose should be able to integrate with either Ionic or React Native, our top two mobile application framework choices.
- **Ease of use** - As developers, the chosen database framework should be relatively easy for us to learn and use. Therefore, the database we choose should have adequate documentation and informational resources on how to implement different features.
- **Ease of communication with mobile application** - There will be many aspects of our application that will be populated with data from our database. We need to consider how difficult it will be to update the data as it's changed in the application or database. Such as whether manual calls to the database are needed, or if data fields are linked to the database and automatically updated as information is changed. Therefore, the database we choose should ideally be able to automatically update data as it is edited or changed in either the database or mobile application.
- **Scalability** – This application is going to start with smaller scale tests; however, we still want this application to be able to grow and the database needs to be able to handle a decent number of users and information it will hold. Therefore, the database we choose should be demonstrably able to handle large amounts of operations and activity.
- **Security** – While all users are under pseudonyms, we still do not want the risk of any information the user must give to not be stored in a secure place. Therefore, the database we choose should be able to handle secure authentication and data access restrictions.

To select databases to analyze, we researched which databases were compatible with the frameworks we evaluated. We examined which databases are most commonly used in software development and favored databases with more documentation and information available. We also specifically considered databases that would potentially work with our top two mobile application frameworks, Ionic and React Native. We did this by looking into the database websites, user reviews, and watching tutorials. Based on these factors, we chose to evaluate Firebase, MongoDB, and SQLite. These databases were analyzed and ranked to determine which would be best suited for our application.

3.2.1 Firebase

Firebase is a mobile and web application development platform built on Google infrastructure that provides backend services such as databases, user authentication, cloud storage, analytics, and more. Firebase was originally developed by Firebase, Inc. in 2011 and then acquired by Google in 2014. It currently integrates with various Google services and is used by companies such as Lyft, Venmo, and The New York Times for their applications.

In order to evaluate Firebase, we created a Firebase account and interacted with Cloud Firestore as well as Realtime Database to get a better idea of how good the user interface was, and to ensure that with a free account we would be able to use Firebase services. Additionally, we looked through official Firebase documentation and searched through other sources online for relevant examples of the functionality we need from Firebase.

- **Works with chosen framework**: **5/5** – Firebase is specifically able to be integrated with Ionic, and on the Ionic website it mentions its plugin for Firebase. Firebase also works with React Native. There is a website for the documentation on integration.
- **Ease of use: 5/5** – The Firebase website contains a helpful amount of documentation for getting started, including some code examples. Other online sources such as YouTube provide a helpful selection of tutorials. Overall, when searching for information about Firebase, the search results were helpful. Additionally, the online user interface for Firebase's Cloud Firestone database, which would likely be used for our application, is easy to navigate and use. From this analysis, our team's learning curve regarding Firebase might still be steep, but we will still have support from documentation and other online sources.
- **Ease of communication with mobile application: 5/5** – Firebase provides a method to sync data between the database and a mobile application. By binding database data with a certain scope in the mobile application, whenever changes are made to the database, they will be pushed to the scope.
- **Scalability: 5/5** – Firebase has the potential to be used on a larger scale. One of its repeating selling points on its website is that it scales automatically and can be used for large applications. As previously mentioned, companies like Lyft, Venmo, and The New York Times use Firebase as their database, so that is a proven factor.

- **Security: 5/5** – This database provides user authentication services and has a special security language to customize user authentication. For their Cloud Firestore database, it specifically mentions having strong user-based security and being able to restrict data access based on some conditions provided. Additionally, Firebase provides Firebase Authentication, a user authentication tool that is developed by the teams that developed Google Sign-in and Smart Lock. This authentication tool is able to be integrated with Cloud Firestore, which could possibly be a feature that makes our application more secure.

3.2.2 MongoDB

MongoDB is a cross-platform document database, initially released in 2009. MongoDB is a NoSQL database, meaning data is stored in JSON-like documents so fields can vary from each document. Major companies like Uber, Codecademy, and eBay use MongoDB as their database technology. The following metrics were decided on after researching the MongoDB website, user reviews, and tutorials with example code:

- **Works with chosen framework**: **5/5** – MongoDB is able to be integrated with Ionic, both for the web and mobile applications. MongoDB is also able to be integrated in React Native applications.
- **Ease of use: 1/5** – There is a decent amount of documentation and tutorials on integrating MongoDB with the Ionic framework. Installation is just a few simple terminal commands and implementing it in the code is not too complex either. MongoDB allows for indexing and real time aggregation. It allows for more flexibility as it can store any type of data in separate documents.

  As most of our members have little knowledge on building databases, learning this NoSQL/document-oriented database style should not be too complicated as we are not that familiar with SQL databases. However, MongoDB does not adhere to a relational schema, which can cause some issues with retrieving data that not all users have inputted, such as for our application we discussed having the option to input a general location.

  Currently, there is a feature release of MongoDB for mobile applications called MongoDB Realm. The first beta release; however, is estimated to be around January 2020. This will result in MongoDB Mobile and Mobile Sync being replaced with the Realm version. This may be an issue, as full functional release is estimated between January through July 2020, which could cause issues for our mobile application implementation during this time.
- **Ease of communication with mobile application: 2/5** – There is an option for mobile syncing, where whenever connectivity is restored, documents between local and remote databases are automatically synchronized. The functionality of this will be changed as the merge of MongoDB and Realm occurs. In the future release discussed in the ease of use section, there is real time data sync. But for now, the stability of being able to have synced connections is low.

- **Scalability: 5/5** – At the core of MongoDB, it is a distributed database. This means there is high availability, horizontal scaling, and geographic distribution. As our application will start on the smaller side with number of users and general data stored (such as videos or modules), we need to keep in mind the application will be getting bigger. So, having the ability to grow is important.
- **Security: 5/5** – The MongoDB configuration file can enable authentication and also has the option for role-based authentication, which can be helpful in that we only want one to two people to be able to access the user data stored, but all administrators should be able to access files in this iteration. If the application is hosted on a provider that does not support firewalls, MongoDB has documentation on configuring it using iptables (AWS can use 'Security groups' to restrict access). Adding a keyfile parameter to config files can set up a replica set helping enhance security. MongoDB has a way to restrict what interfaces it will listen to. Communication may be encrypted, as well as the data. There is further documentation on security on the MongoDB website.

3.2.3 SQLite

SQLite is an open source library that implements a small, self-contained, full-featured SQL database engine in C, initially released in 2000. It is one of the most used database engines, being used by Apple, Adobe, and Facebook. It is built into mobile phones and most computers. Files are commonly used as containers to transfer rich content between systems and good for long-term archival formats for data. Rather than being client-server, it is embedded into the end program. To analyze this database technology, we researched user reviews, the SQLite website, and tutorials with example code and came to the following conclusions:

- **Works with chosen framework**: **5/5** – SQLite can be implemented with the Ionic framework with a plugin and has a documentation page on the Ionic framework website. SQLite can also be integrated with React Native by installing a plugin.
- **Ease of use: 4/5** – The Ionic framework website has some basic documentation on their website, as well as a link to more extensive documentation on the GitHub repository. There are also some tutorials on how to get started and with setting up a database in the recent version of Ionic. It is supported on Android, iOS, macOS, and Windows, which is helpful since this application will be not only the mobile application, but also the web application. With basic knowledge on C programming and database implementations, it should not be hard for everyone in the team to understand.

  For React Native, there is a page documenting how to use the SQLite plugin and tutorials on how to integrate SQLite with React Native as well.
- **Ease of communication with mobile application: 0/5** – From our research, we could not find features that would allow aspects of our mobile application to be synced in a way where the data would automatically be uploaded in the database. It seems that in order to update information in the database, explicit method calls will have to be made.

- **Scalability: 1/5** – SQLite can only embed in a single application and single instance, so the opportunity for scaling up to support multiple users that are sending or receiving data is not possible. Since SQLite is not client/server but built into the application, it is not a good database for storing data elsewhere than the device being used. So, storing user information on a web application separate from the mobile device will not work. It is also not a good choice for an application that will have growing data to a size that cannot fit into a single disk file.
- **Security: 1/5** – SQLite has no built-in encryption for data at rest or in transit. Other than this finding, there is little documentation on implementing security with SQLite, which is a major downfall as we want this application to be as secure as possible.

3.2.4 Chosen Approach

The following table is a summary of the results of each database we researched based on the set criteria we chose:

Database/Criteria	**Firebase**	**MongoDB**	**SQLite**
Works with chosen framework	**5**	**5**	**5**
Ease of Use	**5**	**1**	**4**
Ease of communication with mobile application	**5**	**2**	**0**
Scalability	**5**	**5**	**1**
Security	**5**	**5**	**1**
Total	**25**	**18**	**11**

We chose Firebase as our database for this project. It integrates directly with our chosen mobile and web application framework, Ionic, and meets all the criteria we need. Additionally, it provides extra functionality that makes developing other aspects of our application easier, such as push notifications, user authentication, and application analytics.

3.2.5 Proving Feasibility

To test Firebase, we want to integrate it with a basic Ionic or React Native mobile application, depending on our final decision for our mobile application framework. We will populate the database with hardcoded values and make an attempt to retrieve the data and have it displayed on the application.

Now that we have analyzed and chosen a database technology for our product, we will look into web application frameworks that will be used to develop an integral part of our front-end component.

3.3 Web Application Framework

As there will be an administrative web application, we need a framework for this. While working with the framework directly related to our chosen mobile framework could be best, we still want to look into the different frameworks in case one could provide better features based on our criteria. We found our alternatives based on what web frameworks are most commonly used, as well as looking into Ionic and React Native's web application components, since those are our top choices for mobile application frameworks. Our decision for web framework will be made based on the following criteria:

- **Ease of use** – All team members should relatively easily be able to learn how to use this framework. Therefore, the framework we choose should have adequate documentation available on their website, as well as other online tutorials and examples to learn from.
- **Synchronization and compatibility with the mobile application framework** – This web application is to be used for administrative purposes, so communication from the web application to the mobile application is required. This also means the web application should have access to the data that comes from the mobile application, and vice versa. Therefore, the framework we choose should ease the process of communication for web application and mobile app.
- **Database interaction** – There will be a significant amount of data being stored with this that administrators need access to, so we need to have a way for this framework to work with our database. Therefore, the framework we choose should support the use of our database of choice.
- **Security** – As this web application is for administrative use, any user information and general information that is on the mobile application such as the forums are going to be accessible through our web application depending on administrative status. This information needs to stay secure, so we need a way to ensure those accessing this information have the authority to do so. Therefore, the framework we choose should have security measures available that could prevent access to this information without authorization.

- **Administrative use** – There should be a way to implement administrative specific functionalities, as those are the only people that will be using this web application. Therefore, the framework we choose should ease administrative functionalities such as adding/removing surveys, games, informational videos, etc.

The web frameworks we are looking into are Ionic, Django, Angular, and React Native. Ionic is being looked at as a possible option because it was recommended to us and because it is a great candidate as our web application framework. Django, Angular, and Reach Native were discovered while searching for popular and useful web application frameworks that could work well with our project. The following contains the information we gathered on these four web frameworks based on the criteria above.

3.3.1 Ionic

Ionic is a cross-platform framework released in 2013, being cross-platform means that it can be used to create both mobile and web applications. This is particularly useful for us as we need to create both. We came across Ionic while researching frameworks that would fit specific criteria, in this case cross-platform functionality. Ionic is used for many successful applications, some of which include: MarketWatch, Pacifica, Sworkit, and more. Research on the Ionic web framework was done by running a quick program to get a sense of how it would function, we also looked at documentation to learn about some of the Ionic features it provides.

- **Ease of use: 4/5** – With Ionic being a free and easily accessible framework anyone can start programming with, it is also built on top of AngularJS and Apache Cordova making it easy to extend their functionality with Ionic. Another fact worth mentioning is that Ionic is written in JavaScript, a very common and easy to learn language for web applications. Ionic also uses Typescript, which is a superset of JavaScript.
- **Synchronization and compatibility with the mobile application framework 5/5** – As mentioned before, Ionic is cross-platform, meaning that it is compatible with mobile and web applications. This is because Ionic uses technologies that are usually used for web development like HTML, CSS, and JavaScript for the development of mobile applications. This makes it incredibly easy to make a web and mobile app with Ionic, as well as making it simple to communicate between the two apps as they would both be built with one framework.
- **Database interaction: 5/5** – Ionic supports databases such as Adobe AEM, AWS Amplify, Couchbase Lite, Firebase, Offline Storage, SQLite, SQLite Db Copy, SQLite Porter. Firebase is the database we are going to use, making Ionic a great choice for these criteria.
- **Security: 4/5** - There are many pre-existing security functionalities on this framework we could use, one that is particularly useful for us is *Auth Connect* which adds login/logout and token refresh to an Ionic app and keeps these secure. Auth Connect would be used for our Admin verification so that only the right people have access to the information in the web application. Other security functions that Ionic has include:

Identity Vault, Offline Storage and Keychain. However, there is still a lot of implementation from our part that we would need to look into, like encryption of data on the database side.

- **Administrative use: 4/5** – Ionic does not have an administrative interface like Django. The administrative interface is a feature in Django that creates an admin page automatically with the developer having to use little to no code. The admin interface is explained further in 3.3.2. Although Ionic does not have this administrative interface, it allows for the implementation of an admin page using tables, forms, and login functions. This makes it easy to implement admin use manually.

3.3.2 Django

Django is a popular, Python-based framework that was released in 2005. Django's goal is to make the creation of complex web features very simple. It does this through a large selection of libraries and built-in functionalities. We came across Django while researching web frameworks that would be easy to use and provide administrative uses. Django provides that, and more. Django is so popular and capable that it is used for some of the biggest websites out there, including Instagram, Spotify, YouTube and more. Research on Django was done by writing Python code to test its various features, and by looking at documentation and quick tutorials.

- **Ease of use: 5/5** – This framework has plenty of documentation on not only the framework in general, but the packages and libraries that can be used as well. Django is useful for small-scale projects which are simple web applications that do not require complicated technologies, a web application for a local restaurant that allows ordering food online may be considered small scale. However, Django is also great for high-scale applications that may require more complex functions like support for a large number of traffic and handling a large amount of data. An example of a high-scale web application might be Facebook because they have millions of users adding large amounts of data every second.

  Due to the fact that Django has been around for a decent amount of time, there are a good number of answers that can be found to questions we may have while working on it. Django is also written in Python, a dynamically typed language that is also easy to learn.

- **Synchronization and compatibility with the mobile application framework: 1/5** – Django is a full stack web framework, meaning that for mobile integration and compatibility, Django would need to be used alongside Angular, React, or a Mobile SDK. This is due to the fact that Django's architecture is a lot different from a mobile application architecture because it is built to run on a single service. For this reason, Django would not be a good option for compatibility with a mobile application framework.

- **Database interaction: 3/5** – This framework works with many different databases, such as MySQL, SQLite, and Oracle. However, it does not officially support the Firebase database, which is currently ranked number one in our selection for this project.
- **Security: 4/5** – Django has many useful security functionalities built-in such as XSS protection which protects against users injecting malware, CSRF protection which protects against users using credentials of a different user, SQL injection protection, and other features that aren't going to be too useful for this project. There would still be a significant amount of implementation in this area from our part due to the fact that security is one of our main focuses.
- **Administrative use: 5/5** – Django has an automatic administrative interface; this means that whenever the developer is ready to implement an administrative function to their web application, they can do this automatically with a few commands through the terminal. This admin interface is created using metadata from the developer's models and has many features already built-in such as deleting, adding, and updating information. This would be useful for us as it would make the implementation of the admin features incredibly simple.

3.3.3 Angular

Angular is a Typescript and JavaScript based framework released in 2016. Angular is a rewrite of AngularJS which was released in 2012. Angular is made by Google and can be used for high scale web applications, meaning that they require more computational power and more complicated functionalities. There are many web applications pages that use Angular as their framework some of which include: Samsung, Delta, GitHub, and Google. Research on Angular was done by downloading and testing some programs to try out a few things, such as making a basic task list program that can be checked and unchecked depending on what tasks have been completed. This program was used to get a basic understanding on how Angular might be used. We also looked into Angular documentation and tutorials.

- **Ease of use: 2/5** - Learning this framework may be more difficult to learn than other web frameworks. This is because Angular requires a more in-depth knowledge in JavaScript than the other frameworks require.
- **Synchronization and compatibility with the mobile application framework: 1/5** – Angular is not specifically made for mobile implementation, rather it is geared more towards web applications. Making it interact in a mobile environment may be a difficult task. This is because in order for Angular to be able to interact with a mobile platform, it would need to be paired up with another mobile framework like NativeScript to make it function well.
- **Database interaction: 3/5** – Angular supports the use of many databases, it may require more work the implement them than with other web frameworks. This additional effort is not too complicated though, as the only difference is that database code would need to be implemented in JSON format.

- **Security: 3/5** – Angular does not have built-in security functionalities, so we would have to implement it all if this framework were to be used. However, there are resources available with Angular that would help us with this implementation. There are also general tips on the Angular Documentation website about things that could be done to make the web application more secure.
- **Administrative use: 2/5** – Angular does not have a built-in administrator interface like Django, so all admin features would have to be implemented manually, which would require a lot of time and resources. We are basing all of the web framework's administrative capabilities with Django as it is the framework with the best administrative implementation.

3.3.4 React Native

React Native was released in 2015 and created by Facebook. React is a mobile application framework that can be used for web application development, meaning that it is cross-platform. React Native is a popular framework used by applications such as: Instagram, Facebook, Pinterest, Skype and more. React Native was mostly evaluated by reading documentation, as well as creating a simple "Hello World" program to test some functionality.

- **Ease of use: 5/5** – Looking into how React Native works, we can say that it is very simple to turn an application that works on mobile to one that works as a web application with a few simple commands. This makes building a web application version of a mobile application simpler.
- **Synchronization and compatibility with the mobile application framework: 3.5/5** – React Native can be used to make a web application from a mobile one, making it very much a cross-platform framework. If we decide to choose Ionic as our mobile application framework, we would have to rewrite all of our code for this web application. However, if we decided to go with React Native we would require a few extra dependencies and scripts, and with a simple command you could have a hybrid application that works as a mobile as well as a web application.
- **Database interaction: 5/5** – React works with many databases including Firebase, SQLite PouchDB, RealM. Firebase is currently looking like our database of choice; this would make it easier for us to work with React and integrate our data.
- **Security: 4/5** – React Native has a few security features, one of them includes keychain. Keychain can help protect passwords, account information, credit cards and more. This feature can make it easier for us to implement functionality that will protect user information. There are other implementations in security that we would need to make but React Native provides a great help.
- **Administrative use: 4/5** – Although React does not have an admin interface built-in like Django, it does provide the resources to easily implement one. While looking for examples that developers have made with administrator pages, we have found multiple that function well.

3.3.5 Chosen Approach

The following table is a summary of the results of each web framework based on the set criteria we chose:

Web Framework/Criteria	Ionic	Django	Angular	React Native
Ease of Use	4	5	2	5
Synchronization and Compatibility with the Mobile Application Framework	5	1	1	3.5
Database Interaction	5	3	3	5
Administrative Use	4	5	2	4
Security	4	4	3	4
Total	22	18	11	21.5

Similar to our results with the mobile application framework, Ionic and React Native are very close in how they scored based on our criteria. In order to come to a final decision on which framework to choose, for now we will test both of them in parallel to get a better sense of the workflow with each. Through more testing, we will be able to decide which framework better fulfills our criteria and come to a final decision.

3.3.5 Proving Feasibility

Since our web application needs to be able to work with our mobile application, we will show with a prototype that data between our mobile and web applications synchronize with each other as well as demonstrate the administrator's role in keeping all the data organized. Whether this means adding or removing users, generating a code unique to each user, or managing activities for the application, we plan to be able to demonstrate this in a demo.

In the next section we will decide what technology will be best for us to use to retrieve research analytics on our application, now knowing that we will either be using Ionic or React Native with a Firebase database.

3.4 Research Analytics Technology

A Research Analytics Technology is a tool that uses basic and advanced analysis, data visualization and reporting processes that provide valuable information and new insights based on the collected information. Research Analytics will allow us to view usage of our mobile application and be able to help generate reports based on our data. This includes the ability to generate a graph-based visualization, track how often our mobile application is used and how individuals are using our mobile application. Our solution will require a full-bodied Analytic Research Software. As such we have outlined what we believe we would need as important components of the Analytic Research Software we would use. The research analytics software are analyzed based on the following criteria:

- **Implementation in Android and iOS:** Our client has asked us to build a multiplatform application and having software that analyzes both iOS and Android applications is vital as we would like to reach more people. The research analytics software we choose must be able to have the ability to be used to analyze Android and iOS applications.
- **User usage tracking:** We have been asked to be able to analyze user usage of the mobile application to monitor how the application is being used and what is being used. Therefore, the research analytics that we choose must be able to keep track of user usage and be able to break down the information into a graph-based visualization.
- **Individual tracking for user:** Our application should be able to have or generate a report for women who use our mobile application to be able to see their own usage of the application. In order to have this requirement fulfilled we must have a research analytics tool to be able to keep track of individual users by allowing us to pick a user and see their history. The research analytics we choose should be able to allow individual tracking easily.
- **Provide visualization of information collected:** The main components of a research analytics tool are to be able to analyze data. However, we are focused on the analyzation to be easy to see by visualization of inputs. Therefore, the research analytics we choose should be able to graph components and be easy to see.

Without these components, an Analytic Research Software is not suitable for this mobile application. We are examining four possible candidates which are Firebase, Mixpanel, Countly, and Flurry. Analytics Research Software were narrowed down to four candidates by the availability of the software being advertised as free and were in recommendation articles by other mobile developers.

3.4.1 Firebase

Firebase is a mobile and web application development platform that is used as a backend service that provides features to building mobile applications. Firebase has three services: a real time analysis, database, user authentication and hosting. Firebase is a server that can use GoogleCloud for advance applications. In order to evaluate Firebase, we analyzed the documentation that was provided by Firebase, searched for tutorials that tried implement Firebase, and looked at documentation from other mobile developers.

- **Implementation in Android and iOS: 4/5** – Firebase separates coding for mobile application for Android and iOS or use Firebase with a framework. The recommendation with Firebase is to download an application called Flutter, which is a framework to develop mobile applications. The recommended frameworks on the Firebase website is Android Studio, IntelliJ, Xcode, or VS Code. Firebase gives steps on how to connect the framework you choose to Firebase, but the steps do not seem to be simple. This takes a toll on choosing Firebase as we are not using any of the recommended frameworks, it may cause a problem in development.

  Firebase has React Native as an official Firebase SDK for performing admin actions in privileged environments, but none of the other frameworks we are researching. There is mention that Firebase supports Ionic as a database, but the only video that was related to Firebase Analytics in Ionic was not in English. However, there was a little more information on Ionic being used with Firebase after a little more digging.

- **User usage tracking: 4/5** – Automatically tracking screens is a feature that Firebase provides, but there is no information that describes the capability of tracking screens. The document stated that there an ability to set events and be able to tag these parameters and capture that occurrence. Firebase offers Realtime updates usage. Firebase allows you to take action on almost anything that you are able to measure in your mobile application.
- **Individual tracking for user: 3/5 –** Firebase allows you to construct a valid user ID. This tool will help analyze individual usage and track history. For example, calculate values such as a user's total time spent across an application. The downside to this is that Firebase forces you to access this data in BigQuery, since Firebase automatically tags a user's value with events and sends it to BigQuery. We were unable to find out how the individual tracing looked like or find documentation how it worked with BIgQuery and if we actually needed it.
- **Provide visualization of information collected 4/5** – Firebase supports data visualization through graphs. In a video by Firebase, they introduce different graphs that Firebase supports in order to analyze data. They provided a variety of graphs using data input, which is one of the criteria that we would like our research analytics to fulfill. In each of the individual tracking for user and the user tracking usage we would be able to analyze that data with a graph.

3.4.2 Mixpanel

Mixpanel is an analytics platform that focuses on measurement via firing events. Mixpanel offers solutions for desktop, mobile sites, and desktop applications. They offer behavioral analytics, data science, sending targeted messages and experiments, and data governance which keeps the data clean, organized, and can let Mixpanel validate your data automatically. One service we will focus on for Mixpanel is behavioral analytics, which is an analyzation of behavior with reports. In order to evaluate Mixpanel, we analyzed the documentation that was provided by Mixpanel, searched for tutorials that tried to implement Mixpanel, and looked at documentation from other mobile developers.

- **Implementation in Android and iOS: 2/5** – Mixpanel separates coding for mobile application for Android and iOS, and you must implement it using JavaScript, iOS Objective C, iOS Swift, Android Studio, Python, PHP, Ruby,Unity, or Node.js. This could potentially cause a problem depending on the framework that we decide to use for our mobile application. The steps in order to integrate Mixpanel into one of those software types appear to be simple, but there is still confusing language. We were not able to find information about implementation of Mixpanel with other frameworks, and we will not be able to integrate with some of the frameworks we chose to analyze.
- **User usage tracking: 2/5** – Mixpanel has an analytics tool that allows behavioral data analysis for groups by groups, which allows analyzation of behaviors. However, it turns out that although Mixpanel has what we need for behavior analysis, it does not have this plugin for free, which makes it unusable to us.
- **Individual tracking for user: 2/5** –Mixpanel has the ability to allow researchers see single user's profile and activity feed to understand who they are and where they're hitting roadblocks in the application. However, like the user usage section above, Mixpanel does not offer this service for free.
- **Provide visualization of information collected 4/5** – Mixpanel provides various ways for the mobile developers to be able to see the data that they collect. For example, Mixpanel allows you to input various key metrics. However, a certain constraint stood out when analyzing this feature. Mixpanel only allows certain key metrics for free and in order to customize they charge.

3.4.3 Countly

Countly is an analytics platform that can be used in web, desktop and mobile applications. Countly can be installed in a secure private cloud or on-ground. Countly focuses on mobile analytics retention, funnels, and revenue analytics. Countly has automated push, formulas, A/B testing, cohorts, and user profiles as plugins. In order to evaluate Countly, we analyzed the documentation that was provided by Countly and looked at documentation from other mobile developers.

- **Implementation in Android and iOS: 3/5** – Countly separates coding for mobile application for Android and iOS. You can only integrate Countly into Android specifically by using Android Studio, Xamarin (Android) or Eclipse. You can add to both Android and iOS by using, React Native (Bridge), React Native (unmaintained), Cordova, and Unity. For iOS you can use the software already mentioned or can use Xamarin (iOS), watchOS, and MacOS. Countly makes it easy to use on one of these softwares as they give steps on how to integrate it into those programs. However, we were not able to find documentation to integrate Countly with other frameworks that we are analyzing.

- **User usage tracking: 2/5** – Countly has custom events which allow you to send any type of data to a Countly server. This feature allows us to create actions that would trigger an event which we would be able to put into our analysis. Countly lets you have cohorts, which allows for grouping users based on their in-application behavior and segment all the data using these groups. This tool is extremely helpful, especially since this is what we are looking for in an analytic research software. However, Countly offers limited features to be free, and unfortunately, the previously mentioned feature is not free.
- **Individual tracking for user: 2/5** – Countly lets you track individual level customer information and actions. You can attach values to the customer and explore analytic data at an individual customer level of detail. You are able to see historical data of the user for sections and events and have custom properties. We can in some way figure out how to be able to get that analysis to the individual user when they request their usage report. However, Countly offers limited features to be free and unfortunately, this feature is not free.
- **Provide visualization of information collected 2/5** – Countly allows data visualization by showing the data in different kinds of graphs. However, a certain constraint stood out when analyzing this feature. Countly only allows certain key metrics for free and in order to customize they charge. Thus, we knocked their rating to 2 out of 5 because they have what we need, but we are not able to access it.

3.4.4 Flurry

Flurry is a mobile analytics, monetization and advertising company. The platform analyzes consumer interactions with mobile applications for Android and iOS. Flurry has real-time analytics, in-application revenue analytics, crash analytics and reporting, and application usage metrics. In order to evaluate Flurry, we analyzed the documentation that was provided by Firebase, looked at documentation from other mobile developers, searched for tutorials that tried implement Firebase, and played with the demo that is provided by Flurry.

- **Implementation in Android and iOS: 3/5** – Flurry has easy-to-follow steps for implementing or adding Flurry into an Android application. They have separate steps to follow for iOS implementation that are more difficult to implement than Android. Flurry gives separate steps on how to have custom events which can be used to track specific actions that users make within the application. They also give steps on how to set advanced features that can be used to track user IDs and demographics. In order to implement these specific events, we must implement it using Objective-C or Swift for iOS. We must also use it with Android, iOS, watchOS, Unity, tvOS, and React Native.

There is no mention and no documentation for any other framework that can be used with Flurry. After several searches we were able to find developers who use Ionic and Flurry. There reviews stated that Flurry is a good analytics tool, but it is not as mature as you would expect it to be.

- **Single user usage tracking: 4/5** – Flurry lets you track new users, active users, and sessions. You can run instant analysis and get insights on specific users' groups and user actions. We would be able to see overall user usage but have a difficult time in tracking single user usage. Flurry states that they can report user paths based on the custom events that a user triggers (these events would be created by ReBoot). Flurry's demo showed that we would be able to separate user filters by gender and other characteristics that are in the mobile application demonstrated.
- **Individual tracking for user: 4/5** – Flurry has a new feature and documentation called User Journey. It is stated to be able to investigate the actions of users that have shown specific behavior in the application based on the criteria specified. You are able to review actions of specific types of specific users. Fortunately, Flurry has a demo that you are able to explore, and we went through some of the possibilities that we are able to find out about a specific user. The downside is that we were not able to customize this setting to see its full capability, but it was helpful in demonstrating a possibility in Flurry.
- **Provide visualization of information collected 4/5** – Flurry has several data visualization-based graphs. Using the demo that Flurry provided, we were able to see the different graphs and how adding certain pieces of data changes the graph. However, when it came to adding different aspects to the graph, it was a bit difficult to comprehend what data set could be added to the graph. We are not sure if it was difficult to add a component to the graph because we didn't know the data information or if it was because Flurry is difficult.

3.4.5 Chosen Approach

The following table is a summary of the results of each research analytics based on the set criteria we chose:

Analytic Research Software/Criteria	**Firebase**	**Mixpanel**	**Countly**	**Flurry**
Implementation in Android and iOS	**4**	**2**	**3**	**3**
Single user usage tracking	**4**	**2**	**2**	**4**
Individual tracking for user	**3**	**2**	**2**	**4**
Provide visualization of information collected	**5**	**2**	**2**	**5**
Total	**16**	**8**	**9**	**16**

The research analytics software we decided to go with is Firebase. Firebase and Flurry were neck and neck in fitting the criteria that we set, but since we chose Firebase as our database, it would be more convenient for us to also use it as our analytics tool. Although both research analytics met our criteria something that we continue to analyze is how would be able to fully incorporate our analytics to our mobile application. We took into consideration many developer articles that have stated that Firebase is easy to implement in a mobile application, which is something that separates it from other analytics tools. We were unable to find experiences that other developers had when implementing Flurry into their mobile application. For the reasons stated, we decided to use Firebase, as it seems to have better reviews when it comes to integrating with a mobile application.

3.4.6 Proving Feasibility

In order to prove that the research analytics will provide all the functionality we need for our application, we plan to create a simple program with a few buttons. We plan to incorporate track user usage, individual usage, Android and iOS implementation and be able to generate a graph based on data collected. In a short demo we will develop a small mobile application that a user will be able to press the button and implement an analyzation of that mobile application as to how many times people have been pressing certain buttons. We will be able to track individual use by tagging users to see how often they press a button. All these analyzations must be able to be shown on a graph-based visualization using Firebase.

Now that we have analyzed and chosen our research analytics technology, we will be investigating potential forum plugins to provide the capability of users to join and participate in group chats within our mobile application.

3.5 Forum Plugin

An important part of this application is that there will be a forum where the users will be able to talk to other future mothers. For the forum functionality, we are going to need a forum tool plugin that meets the following criteria:

- **Free** - This plugin needs to be free for us to use.
- **Support grouping of users -** As the users that sign up for this app will be grouped together for their forums, we need a functionality that will be able to support this. Therefore, the forum plugin we choose should provide some feature or capability that supports grouping users in the forum.
- **Users can create forums/threads** - Users should be able to create threads where they can ask for advice or give advice. Therefore, the forum plugin we choose should include a feature that allows users to create a discussion thread.
- **Users can comment on threads** - Users should be able to answer questions on other users' threads or be able to comment with general advice. Therefore, the forum plugin we choose should allow users to access and comment on discussion threads.

In this forum users should be able to create threads, reply to posts, and be able to be placed into designated groups based on their birth month. We will be using WordPress, which is a content management system that can support specific forum plugins. WordPress works with Ionic or React Native and can be free to use when combined with AWS. There is also a plugin available where one can use both Firebase and WordPress. While researching, we could not find any forum plugin tools made specifically for Firebase. The forum plugin tools chosen were found by going to wordpress.org and looking at their most popular forum tools, this includes bbPress, Asgaros, and wpForo.

3.5.1 bbPress

bbPress is one of the most downloaded forum tool plugins that can work with WordPress. The first version was created in 2004, and it is one of the most used forum tool plugins. Some companies that use bbPress are Sleep Number, Score Sense, and Videomaker Inc. This plugin was researched by going on Wordpress and bbPress's website and tested by using their example forum. This plugin has many features that can be added on, which allows for better customization.

- **Free: 5/5 -** This plugin is 100% free.
- **Supports grouping of users: 1/5 -** After doing research, we have found that there isn't a way to group users. This is a big issue as users need to be organized by groups.
- **Users can create forums: 5/5 -** Users are able to create their own threads. This was tested by going on their website and trying to create a forum topic in their support section. It was a little difficult to navigate.
- **Users can comment on threads 5/5 -** Users are able to comment on other users' posts. This was also tested on bbPress's support section.

3.5.2 Asgaros

Asgaros is a forum tool plugin that has the highest ratings on WordPress. According to the Asgaros GitHub, the plugin was created in 2015. This plugin was researched by visiting WordPress's and bbPress's websites and going through the listed features. We also visited a website that compared Asgaros and WpForo. This plugin was tested by using their support forum. It comes with many free features while also being lightweight. The only downside of this forum tool plugin is that it is less customizable in terms of stylization.

- **Free: 5/5 -** This plugin is 100% free.
- **Supports grouping of users: 5/5 -** There is a way with this plugin to filter groups. This plugin has the ability to add permissions for different parts of the forums and can prevent certain groups from viewing specific content from other groups. This is exactly what we need as we need to make sure that mothers are only seeing the forums of their birth month.

- **Users can create forums: 5/5 -** Users are able to create their own threads. This was tested by going onto the Asgaros website and going to their support section where they use their own forum plugin to create a help forum. There we were able to create a new topic.
- **Users can comment on threads 5/5 -** Users are able to comment on other users' posts. Testing for this was done by going to their support section and replying to someone's post.

3.5.3 wpForo

wpForo is also a highly rated forum tool plugin on WordPress's website. This first version of this plugin was created in 2016 and has over 340,000 installations. Research was done by going searching for articles comparing popular forum tools, as well as going on WordPress's website. Testing was done by trying out the example forum on wpForo's website. It comes with many free features and has a decent variety of styling options.

- **Free: 3/5 -** This plugin is generally free, but for extra features/additions there is a paywall. Each add-on generally costs between $11 - $35. Some add-ons include polls, emoticons, and attachments. While these add-ons may not currently be necessary for the project, not having these features could become limiting.
- **Supports grouping of users: 5/5 -** There is a way to be able to group users together. On their website, it says they have "forum access sets per user group per forum".
- **Users can create forums: 5/5 -** Users are able to create their own threads. On the wpForo website, there is a forum for users to discuss their plugin. On this page there is a button where a new topic can be added.
- **Users can comment on threads 5/5 -** Users are able to comment on other users' posts. On their website, if the thread is clicked, a reply can be made in the text box at the bottom of the page, or a reply can be to a specific person and a text box will appear.

3.5.4 Chosen Approach

The chart below shows how we ranked each forum tool plugin in terms of the criteria:

Forum Plugin/Criteria	bbPress	Asgaros	wpForo
Free	5	5	3
Groups	1	5	5
Create forum thread	5	5	5
Comment	5	5	5
Total	16	20	18

For our project, Asgaros is the best option, because it fits all of our criteria and is also completely free. bbPress wouldn't have been useful as grouping users is not possible. wpForo does everything we would want it to, but not everything on it is free. Asgaros has none of these problems. The fact that it is completely free, allows grouping and preventing certain groups from seeing certain parts of the forum, and that users can create forum threads and reply to others makes it a great choice for our mobile application.

3.5.5 Proving Feasibility

In the tech demo, we can create a basic program that for each tool will allow us to test out how a user would create a new topic, how they could reply to a post, and then make sure that the user can only see posts for their designated group. We will also want to test to make sure that it is able to work inside of the mobile application shell.

4.0 Technology Integration

Figure 4.1 is a system diagram that shows how we are planning to have all the chosen technologies interact with one another.

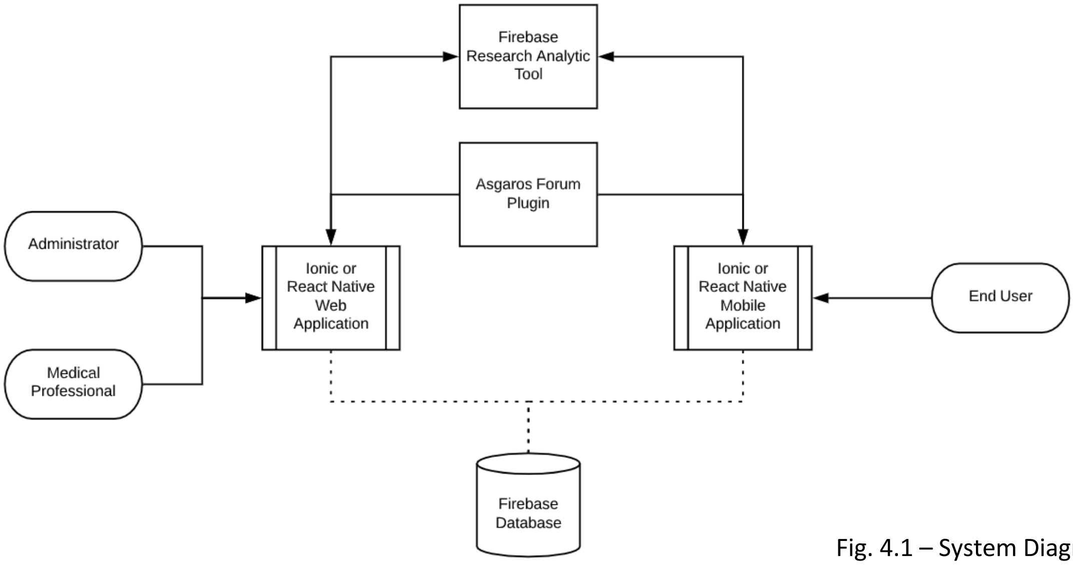
Firebase Research Analytic Tool
Asgaros Forum Plugin
Administrator
Medical Professional
Ionic or React Native Web Application
Ionic or React Native Mobile Application
End User
Firebase Database

Fig. 4.1 – System Diagram

As indicated in Figure 4.1, our front-end web and mobile application will be developed using either the Ionic or React Native framework. Back-end development, such as storing user information in the database, will be done using Firebase. Firebase also provides application analytics capabilities, which will be used for analyzing how users interact with our mobile application. This information will be displayed on the web application for administrators to see. Firebase specifically integrates with Ionic and React Native, so there will be no issues connecting these two technologies together for these front-end and back-end capabilities.

We plan on using Asgaros as our forum plugin tool, which will allow users to be in group chats and communicate with each other. In our research, we discovered that in order to integrate Asgaros with Ionic or React Native, we would first have to integrate Ionic or React Native with WordPress, and then use Asgaros as a plugin for WordPress. While this is an extra step, we will be able to use Asgaros as a forum plugin with Ionic or React Native.

During our research process, we purposely chose technologies and criteria that would support system wide integration. Our database and analytics tool integrate with our front-end framework, as does our chosen forum plugin. We do not expect to have issues when it comes to getting our chosen technologies to work together.

5.0 Conclusion

The path to recovery from opioid addiction demands a lot of time, effort, and determination. Recovering pregnant women with opioid addictions may need to attend up to 17 appointments in a single week to receive the medical and counseling treatment required in order to get clean and prove they are capable of raising their child. The stigma surrounding these women adds another obstacle to the many responsibilities they bear during the recovery process. Many women are not seen as pregnant mothers, but just another addict. Although they may be trying their best to recover and keep their child, case managers are seldom convinced by these women's efforts. To help empower and alleviate some of the struggles these women face, we are creating an anonymous application that will allow pregnant women in recovery track their pregnancy progress, participate in forums with other women in recovery, and learn more about what they should be expecting and doing with modules. There will also be a web application for administrative users to manage data and monitor usage.

To create this application, we need a reliable and secure mobile application framework, database, web application framework, research analytic software, and forum plugin. After analyzing multiple alternatives for each of these technologies, we decided on Ionic or React Native as our mobile application framework, Firebase as our database technology, Ionic or React Native as our web application framework, Firebase for our research analytics software, and Asgaros for our forum plugin. As mentioned in the mobile application and web application framework sections, we will be conducting further research to come to a final decision between Ionic and React Native. Through our research we know all of these components should be able to be implemented together to solve our primary technical challenges.

To summarize our findings relative to our technical challenges:

Technical Challenge	Proposed Solution	Confidence Level (1 – 10)
Cross-platform support	Ionic and React Native works on iOS, Android, and for web.	**10**
Tracking individual user's progress and tailoring their app	We have found implementations that tailor based on user using Ionic and React Native.	**9**
Surveys / Questions and Answers	Plugins can be used for this with Ionic and React Native, as well as user input components.	**10**
Video plugins	There is a plugin for videos that can be used with Ionic and React Native.	**9**
Forums	Using WordPress, we can implement Asgaros forum plugin.	**7**
Calendar	We have found multiple implementations of creating a calendar using Ionic and React Native.	**8**
Notifications	We have implementations of this, will only have to tailor it to our reasons for sending notifications.	**10**
Web application for administrative use	Ionic and React Native have ways to get authentication and can change permissions based on this.	**10**
Research analytics	Firebase has a way to store data and create analytics.	**8**

Storing videos in the database needs more research as well based on the plugin for videos. We know Ionic works with WordPress, so we have the ability to use Asgaros for our forum plugin. We just need to do some more research on implementing it fully. For our calendar, we want to look more into adding important times and dates. We still need to look more into solutions of tracking a user's individual mobile application data.

Overall, after this analysis, we feel a lot more confident in being able to implement our technical challenges and create an end product that will be helpful to pregnant women in recovery from opioid addiction.