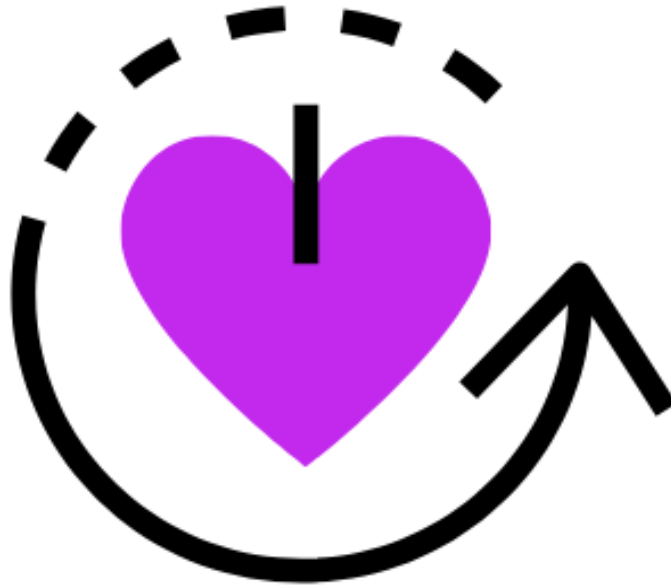


# Team ReBoot.



## Software Design Document

Version 1.3

**Date:** February 18, 2019

**Team Members:**

Bianca Altman, Jennie Dallas,  
Victoria Leafgren, Isai Martinez,  
Cindy Valentin

**Sponsor:** Dr. Emery Eaves

**Mentor:** Dr. Eck Doerry

# **Table of Contents**

1.0 Introduction.....	1
2.0 Implementation Overview.....	4
3.0 Architectural Overview.....	7
4.0 Module and Interface Descriptions.....	8
5.0 Implementation Overview.....	24
6.0 Conclusion.....	29

## 1.0 Introduction

Over two million Americans struggle with opioid addictions, resulting in more than 130 people in the US alone dying every day from opioid overdoses and costing the US \$78.5 billion a year for healthcare, lost productivity, and addiction treatments<sup>1</sup>. Not only are overall rates of opioid addiction increasing, but the rates of pregnant women who suffer from opioid use disorder are growing as well. Between 1999 and 2014, the number of women with opioid use disorder at the time of delivery quadrupled, with the rate currently at 6.5 per 1,000 deliveries<sup>2</sup>. Use of opioids during pregnancy is a particularly serious problem, as it can result in newborns being born with a withdrawal syndrome known as neonatal abstinence syndrome (NAS). Babies born with NAS may experience poor feeding, breathing problems, body shakes, excessive crying, and more<sup>3</sup>. Longer term, NAS can cause cognitive or behavioral impairments<sup>4</sup>. From 2004 to 2014, there was a five-fold increase in the number of babies born with NAS in the United States, which has consequently resulted in a steady increase of the amount of money spent on treating these babies. In 2014, an estimated \$563 million in hospital costs was spent on treating infants with NAS, compared to the \$90.9 million spent in 2004<sup>5</sup>.

Pregnant women with opioid use disorder who want to start on the path to recovery are typically placed in a comprehensive outpatient treatment program. The treatment program involves medication, addiction counseling, and attentive prenatal care and education. Traditionally, the recovery process involves making daily trips to a Methadone clinic to receive that day's dose of Methadone, which is a commonly used medication for treating pregnant women with opioid use disorder and is administered daily. Being treated with this medication not only helps these women in their recovery, but also reduces the rates and severity of NAS in their newborns<sup>6</sup>. Counseling appointments help these women cope with feelings or situations that might lead to relapse and is integral to an effective treatment program<sup>7</sup>. They may be in a one-on-one or group setting depending on the needs of the individual and must also be attended on a regular basis.

The issue with existing conventional recovery programs is the number of appointments and obligations that accumulate throughout each week. A pregnant woman in recovery needs to

---

<sup>1</sup> National Institute on Drug Abuse. (2019, January 22). Opioid Overdose Crisis.

<sup>2</sup> National Institute on Drug Abuse. (2019, January 22). Dramatic Increases in Maternal Opioid Use and Neonatal Abstinence Syndrome.

<sup>3</sup> The Recovery Village. (2019, October 14). Long-Term Effects of Neonatal Abstinence Syndrome.

<sup>4</sup> Seattle Children's. (2018). *Neonatal Nursing Education Brief: The Long-Term Outcomes of Infants with Neonatal Abstinence Syndrome*.

<sup>5</sup> National Institute on Drug Abuse. (2019, January 22). Dramatic Increases in Maternal Opioid Use and Neonatal Abstinence Syndrome.

<sup>6</sup> National Institute on Drug Abuse. (2017, July). Treating Opioid Use Disorder During Pregnancy.

<sup>7</sup> Center for Substance Abuse Treatment. Substance Abuse: Clinical Issues in Intensive Outpatient Treatment. Rockville (MD): Substance Abuse and Mental Health Services Administration (US); 2006. (Treatment Improvement Protocol (TIP) Series, No. 47.) Chapter 4. Services in Intensive Outpatient Treatment Programs.

make daily trips to a Methadone clinic, up to five counseling sessions weekly<sup>8</sup>, and extra trips to a prenatal care clinic if she has any pregnancy questions. Depending on the stage of pregnancy a woman is in and how many additional questions or concerns she has each week, she may have to attend up to two or three prenatal care appointments weekly<sup>9</sup>. All of these responsibilities are added to a woman's regular day, which might include working full time and taking care of other errands and responsibilities. Consequently, attending the many meetings and responsibilities required in a recovery program can become a major obstacle for pregnant women in recovery. This is especially challenging for women living in rural areas or who are transportation-limited, as they have to struggle to find the time and means to travel to their appointments. Considering that around three times more pregnant women being treated for opioid use disorder are from rural areas, rather than urban areas, transportation is a relevant issue for this demographic<sup>10</sup>.

Having an intensive schedule that is already difficult to manage on its own can make the added commitments required in a recovery program nearly impossible to accommodate. While women in opioid recovery are encouraged to find new social groups and to dissociate themselves from those who supported their addiction, this can become an infeasible task if they have no leisure time. Additionally, time constraints make it difficult for these women to properly seek out key health or pregnancy information specific to their recovery. This leaves these women with a lack of social support and information access much needed for a successful recovery.

Our client, Dr. Emery Eaves, is a medical anthropologist at Northern Arizona University whose research focuses on the opioid epidemic among pregnant women in Arizona. Through recovery programs and support groups in Phoenix, Dr. Eaves finds pregnant women with opioid use disorder to interview; using this research to come to a better understanding of these women's situations and to identify their biggest recovery obstacles. Given the previously discussed circumstances of a typical pregnant woman in recovery, it is unsurprising that Dr. Eaves has found the following to be the biggest obstacles in a pregnant woman's recovery:

- **Attending appointments and obligations** - As previously mentioned, the number of weekly Methadone clinic, prenatal care, and counseling appointments can add up to be nearly impossible to accommodate in a typical woman's schedule. Women living in rural areas would find this particularly burdensome, as they would have no choice but to drive for hours each week to get to their appointments.
- **Lack of social support** - If pregnant women in recovery barely have enough time to attend the required appointments and other obligations they have, finding time to make

---

<sup>8</sup> Center for Substance Abuse Treatment. Substance Abuse: Clinical Issues in Intensive Outpatient Treatment. Rockville (MD): Substance Abuse and Mental Health Services Administration (US); 2006. (Treatment Improvement Protocol (TIP) Series, No. 47.) Chapter 4. Services in Intensive Outpatient Treatment Programs.

<sup>9</sup> What to Expect Editors. (2019, May 21). Prenatal Appointments.

<sup>10</sup> Jumah. Rural, Pregnant, and Opioid dependent: a Systematic Review. Substance Abuse: Research and Treatment 2016;10(S1) 35–41 doi: 10.4137/SaRt.S34547.

new friends and socialize can become out of the question. Due to the stigma surrounding pregnant women with opioid addictions, it's especially difficult to find people who are supportive of their recovery process. This can leave these women feeling isolated and unsupported in their recovery.

- **Lack of information access** - Similar to a lack of social support, not having any leisure time makes it more difficult for a pregnant woman in recovery to seek out key health and pregnancy information specific to their recovery. This leads to a lack of knowledge of certain recovery strategies that these women should adhere to, or medical advice that could help make the recovery process easier.

To help alleviate the challenges of traditional recovery programs, we have envisioned a solution centered around a mobile application, which we call iMATter, that can assist pregnant women with opioid use disorder by taking aspects of the recovery process out of the physical space and into a secure, virtual environment. This application will provide a safe space for pregnant women in recovery to talk to each other while allowing them to remain completely anonymous. It aims to directly address the three main recovery obstacles outlined above by bringing these women together for social support and allowing them to learn and ask questions about pregnancy and recovery in a judgement free space. Specifically, iMATter will provide users with the following:

- **Virtual learning space** - users can watch videos, take learning modules, and expand their knowledge on their pregnancy and recovery.
- **Chatroom** - Users will be placed into chatroom groups based on their due months. Users in these groups will be able to talk to one another, discuss their experiences, and give and receive advice in a chat room. This provides an outlet for these women to talk about their burdens with those who can understand and relate on a personal level and is a source of social support for women who feel isolated in their recovery experience.
- **Calendar** - a place where users can keep track of appointments, and other events may be added. Calendar events can be configured so the user is reminded of their occurrence.
- **Notifications** - iMATter will send notifications to remind users of appointments or medication consumption times that were added to their calendar, alert them that new information was added to the application and engage them in motivational activities.
- **Threaded forum** - this will serve as an information desk, where users may go to ask a provider recovery or medical-based questions. This will reduce the need for women to make special trips to prenatal care clinics to get their questions answered, saving them time in their already hectic daily schedule and decreases the number of appointments they need to attend.

- **Reward system** - As an incentive for research purposes as well as to keep women on the recovery track, users will be able to accumulate points through certain actions in the application and redeem them for a gift card.

Our application will provide pregnant women in recovery with resources and a social support system that is accessible from anywhere, at any time. With our application, finding a new social group is no longer nearly impossible and women in recovery will be able to access educational materials and get recovery questions answered without taking time out of their day to travel to a clinic. iMATter will optimize the treatment process and help more pregnant women be successful in their recovery.

## 2.0 Implementation Overview

Working with Dr. Eaves, we have formulated a blueprint for iMATter, which will be a mobile application that can assist pregnant women with opioid use disorder by providing a secure, virtual environment that supports aspects of their recovery process.

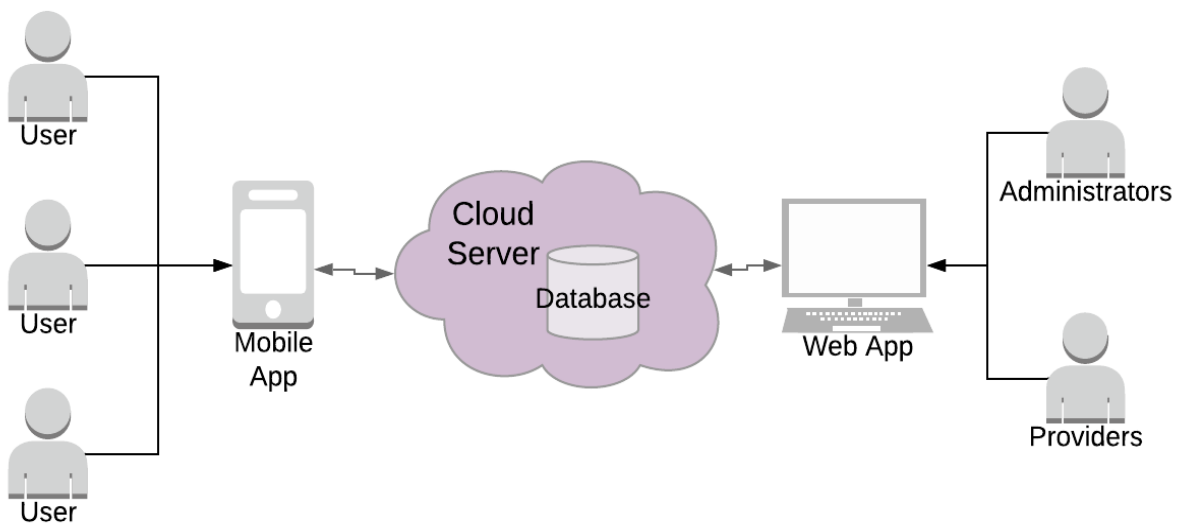


Figure 2.0 - System Diagram

Our general solution can be seen in Figure 2.0. This solution contains two major components, a mobile application for users and a web application for administrators and providers, which will share information through a shared server connected to a Firebase database. The mobile application will provide pregnant women in recovery a judgement free space to obtain information about recovery and where they can receive social support from other women in the same position, all while remaining completely anonymous. This application will provide customized resources based on pregnancy to the users, a way for users to talk to one another, and provide learning opportunities that users may not easily access otherwise. The web application for administrators will provide a place for them to maintain and manage users and

content. Administrators will have the ability to create and distribute learning modules and surveys to the users of the mobile application.

## **2.1 Tools and Technologies**

To implement our solution for iMATter, we will use the tools and technologies listed below.

### **Ionic Framework: Overall Architecture Framework**

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. We chose this framework for both our web and mobile application development as we only need one codebase for each application that can then be used on any device, rather than a codebase for each operating system. As well, considering our technical feasibility analysis, we know Ionic will be able to accomplish our requirements.

### **Firebase Firestore Database: Data Storage and Synchronization**

Firebase Firestore is a scalable NoSQL cloud database. We chose this database technology as it can be used for mobile and web applications and can sync data across the applications using realtime listeners. This database is also supported by Ionic, which is another reason we decided to use Firestore.

### **Chart.js: Graph Visualization for Analytics**

Chart.js is a free open-source JavaScript library that allows for data visualization with flexibility in graphs. It also allows for great scale granularity and good graphics performance. Chart.js is maintained by the community and is one of the most popular JS charting libraries on GitHub and is easy to use. In order to create visualization of the findings through our mobile application, we needed a tool that provides simple data input, which Chart.js provides. Using this tool, we can produce the graphs for our click path analysis of the mobile application, which will give our client a better idea of how users are interacting with the application.

For our applications to provide a viable solution using these technologies, we need to be able to have them accomplish a set of key responsibilities and features.

## **2.2 Key Responsibilities and Features**

The responsibilities and features differ based on the type of user. All three users will have different experiences using their application.

Administrators and providers will only be using the web application, but for different functionalities. Administrators, at a high level, will be using the web application to be able to do the following:

- Manage mobile application user, providers, and other administrators accounts, including creating, editing, and deleting the accounts.

- Manage forum posts/comments and chat logs posted on the application, meaning having the ability to delete forum posts/comments or chats sent.
- Create and add learning modules and surveys to be administered to the mobile application.
- Send out surveys to the mobile application for set periods of time.
- View click-path analytics of the user application.

Providers are only using this application for answering information desk forum questions. At a high level, providers will be using the web application to:

- Answer questions posted in the information desk forum.
- Create a profile that contains a username, avatar, and short bio for users of the mobile application to view.

The mobile application provides social support, informational resources, and other tools to help pregnant women in opioid recovery, which at a high level include:

- Interact with a chat room based on the cohort they are placed in, meaning they can send chats and view chats from other users that are also in their cohort.
- Interact with a calendar that uses local storage to store user input dates and receive notifications to remind them of the event.
- Ask questions in the information desk, as well as comment on their own posts or other posts.
- Answer surveys.
- Take learning modules, which can include watching a video, listening to an audio, and answering questions.
- View information about their current pregnancy status.
- Receive notifications from newly added surveys and learning modules.

With these key features and responsibilities outlined, we can create our overall architecture of iMATter, which will be discussed in the next section.



### 3.0 Architectural Overview

To build iMATter, we need our web and mobile applications to be able to share data with each other through a shared cloud database.

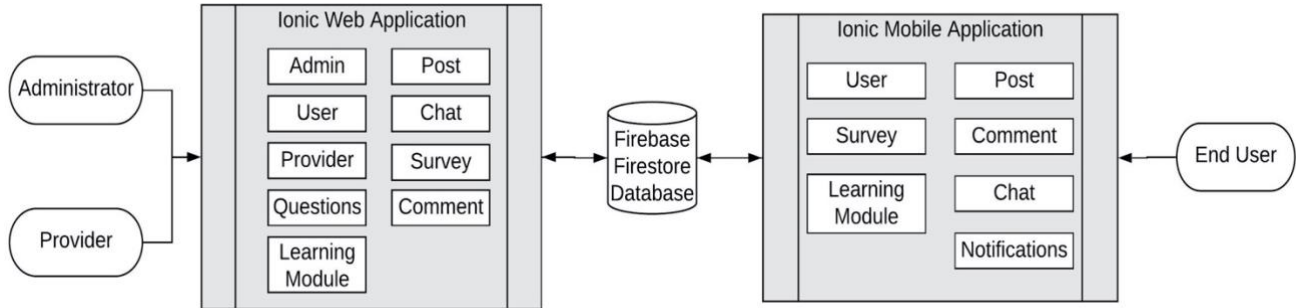


Figure 3.0 - Architectural Diagram

As illustrated in Figure 3.0, our database is at the center of our application. Web application users will be heavily using the database via the web application user interface to view, create, and edit content that is being displayed in the mobile application. On the other hand, mobile application users will be pushing information to the database whenever they engage with the application, whether that is through learning modules, the forum, or the chat room.

#### 3.1 Communication Mechanisms and Data Flow

As seen in Figure 3.0, using Firebase Firestore database, the web and mobile applications will be able to share data in real time. These two components will be sharing data for multiple different application functionalities. For example, the general flow of mobile app user creation will begin with the administrator creating a blank user, which is done by pressing a button that will provide a randomly generated code. This code is then stored in the database and the administrator may then give the code to the user. When the user goes to sign up in the mobile application, they will first be prompted into providing the required code. If the code is correctly entered, the user is brought to the signup page and asked to enter their required signup data. Upon submitting this, this data will be added to the database attached to code that was entered and will then show on the web application to the administrator where mobile application users are listed. This is the general concept for most app to app communication. An admin may create a survey, which will then be stored in the database so the mobile application can display it. Once a user takes the survey, the data will be sent back to the admin webpage after being stored in the database.

Now that we have established a high-level detail of the workings of iMATter, we will discuss our application's components in more detail.

### 4.0 Module and Interface Descriptions

As previously stated, iMATter consists of two applications, the mobile and web. We have broken this down into three main modules: the web application for administrators, the web application for providers, and the mobile application for users. In this section, we will discuss the classes and interfaces that make up each module, as well as why they are necessary components and how they will be used. To begin, we will discuss the administrator web application module.

## 4.1 Administrator Web Application Module

The administrator web application module provides administrators of iMATter with a user interface to view and manage content being displayed in the mobile application. This includes managing mobile application user accounts, provider accounts, creating surveys and learning modules, viewing mobile application chat and forum activity, and more. We envision the administrator web application having the components illustrated in Figure 4.1 below.

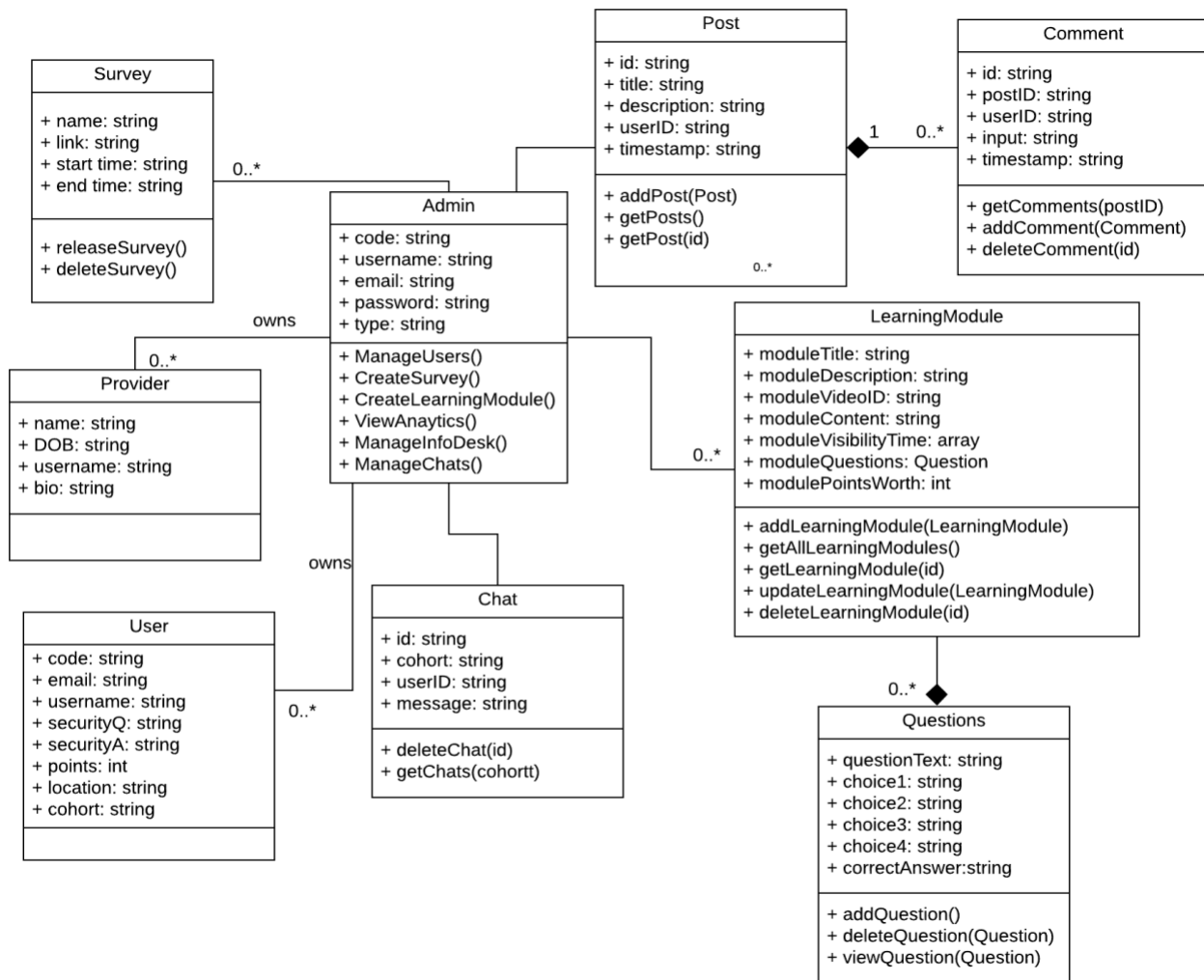


Figure 4.1 - Administrator Web Application Module UML

To better explain the functionalities of the administrator web application module, the details of each component are listed below:

#### 4.1.1 Admin:

We define an administrator as someone that has access to viewing all user created data and managing data. The admin type will be able to interact with multiple components. Admins will be able to create, edit, and delete all user types. Admins can create survey objects which will be viewable by not only themselves but also mobile application users. Learning module objects are also created by admins, which are also viewable by both admins and users. Admins can manage user created data, including post, comment, and chat objects that are created by any user type. Admins will also be able to view analytic data which is created by users on the mobile application. The following explains the fields that an admin will have:

- **code:** The code is an identifier as well as an access code, which will allow a new admin to sign up.
- **username:** Admins will have the ability to use the information desk, so they have a username which will be displayed.
- **email:** An email is used for account login.
- **password:** A password is necessary for logging into the application (not viewable by other admins).
- **type:** Type defines that this user is an admin.

In order for administrators to be able to manage the mobile application, admins will have the ability to do the following:

- **ManageUsers():** Admins can create all user types (admins, providers, and users), view information for all types, edit information, and delete individual users.
- **CreateSurvey():** Surveys will be sent to the mobile application from the web application by administrators. This allows admins to create surveys by inserting a link.
- **CreateLearningModule():** This will allow admins to create learning modules that will be sent out to the mobile application.
- **ViewAnalytics():** This will allow admins to view click path data from the mobile application.
- **ManageInfoDesk():** Admins can manage the information desk by answering questions, as well as being able to delete any posts or comments.
- **ManageChats():** Admins can view chat logs from each cohort and delete messages if needed.

#### 4.1.2 User:

We define a user as a pregnant woman in recovery for opioid use disorder that has been given access to this mobile application. Since users do not use the web application at all, only admins will be interacting with the user component. Administrators will be able to view a list of users and certain user fields. Admins can delete users, as well as edit certain fields of each user. The following explains the fields that administrators will be able to see:

- **code:** The code is an identifier of the user that was added. This will always stay the same.
- **email:** Administrators can see what email users have for communication purposes or for the event of a user losing their email.
- **username:** Administrators can see username so they may know who is posting to chat rooms and forums.
- **securityQ:** Administrators can see what security question a user chose so they may ask the user to answer it if the user calls having lost their email.
- **securityA:** Administrators can see answer to user security questions so they can see if the user answers the question correctly or not to know if they can give the user their email.
- **points:** Administrators can see what point totals users are at so they know if they can give them a reward.
- **location:** Administrators can see user locations, if the user has provided one.
- **cohort:** As users are placed into different groups and administrators can view chat logs for each cohort, administrators can see what cohort each user is in.

#### 4.1.3 Provider

We define providers as volunteers and medical professionals who are given access to the web application in order to assist the users by answering questions in the information desk. The administrators will be able to view a list of providers and some of their fields, as well as have the ability to delete providers. The administrators will be able to see:

- **name:** The name of the providers will be viewable by the administrators.
- **DOB:** The providers will give their date of birth, and the administrators will be able to see this.
- **username:** Administrators can view the providers usernames, so they know who answered the user's questions.

- **bio:** Some information about themselves that a provider wishes to share with other users.

#### 4.1.4 Survey

A survey object is created by an admin entering a link to a Qualtrics survey, a name for the survey to be displayed to the users, and a time period for the survey to be displayed. Surveys will be for administrative use to collect user information directly from the users. Admins also have the ability to delete surveys they have created if needed. The following fields will be available to the administrator when creating a survey:

- **name:** The name of the survey which will be shown to the user on the mobile application.
- **link:** The link to the survey provided by the administrator.
- **startTime:** This will be where the administrator will specify the time when the survey will be available to the user in the mobile application.
- **endTime:** This will be where the administrator will specify the expired time when the survey will not be available any longer to the user.

The functionality will be provided for by the following functions:

- **releaseSurvey():** This will add the survey to the database, making it available for the user to see.
- **deleteSurvey():** This will remove the survey from the database, making it no longer visible for the user to see.

#### 4.1.5 LearningModule

A learning module is a page containing informational media such as videos followed by questions for users to answer regarding the information presented. In the web application, the learning module is responsible for taking in administrator input from its fields and saving the input in the database. The learning module component in the web application gives administrators the ability to create, edit, and delete learning modules that appear for the mobile application user to view. Therefore, the admin component will be interacting with the fields and functionality of this component. The questions component will also be utilized in the learning module, as one of the learning module's fields is storing an array of questions. These questions can be added and deleted by the administrator within each learning module. The learning module's fields are as follows:

- **moduleTitle:** The title of the learning module, which will be displayed to the mobile application user.

- **moduleDescription:** A brief description of what the learning module is about, to help give the mobile application user an overview of the module.
- **moduleVideoID:** The video identification code of a YouTube video that is to be embedded in the learning module. Found in the URL of a YouTube video, and necessary for embedding the video.
- **moduleContent:** Instructions or information that the administrator wishes to include in the learning module, displayed before the other learning content.
- **moduleVisibilityTime:** The week(s) of pregnancy that the learning module should be displayed to mobile application users.
- **moduleQuestions:** The questions that will be displayed after the informational media and answered by mobile application users.
- **modulePointsWorth:** Since we will be implementing a reward system that gives points for completing learning modules, this stores the number of points the learning module is worth.

Given that the learning module has the functionality to create, edit, and delete learning modules, the public methods for the learning module component are the following:

- **addLearningModule(LearningModule):** Takes the data inputted in the fields listed above and creates a new LearningModule object, then stores it in the database as a new learning module.
- **getAllLearningModules():** Retrieves all the saved learning modules from the database.
- **getLearningModule(id):** Retrieves a specific learning module from the database, given the ID of the learning module that is to be retrieved.
- **updateLearningModule(LearningModule):** Takes any edited data from an existing learning module inputted in the fields listed above and updates the learning module in the database.
- **deleteLearningModule(id):** Deletes a specific learning module from the database, given the ID of the learning module to be deleted.

#### 4.1.6 Questions:

Questions are objects that are created by the administrator and added to learning modules. They contain questions on learning module materials and answer selections that will be displayed to mobile application users. The purpose of this component is to provide a quiz-like feature in the mobile application. Questions are exclusively used in learning modules and

cannot be created anywhere else. Therefore, the question component only interacts with the admin and learning module components. The following fields will be available to the administrator when creating a question:

- **questionText:** This will be the question itself that will be used to the user.
- **choice 1** The first answer option for the user to choose from.
- **choice 2:** The second answer option for the user to choose from.
- **choice 3:** The third answer option for the user to choose from.
- **choice 4:** The fourth answer option for the user to choose from.
- **correctAnswer:** The correct answer option for this question.

To provide the functionality to create, delete, and view questions, the functions for the questions component are as follows:

- **addQuestion():** Appends the new question object to the learning module array of questions.
- **deleteQuestion(Question):** Given a specific question to delete, this method deletes that question from a learning module.
- **viewQuestion(Question):** Given an existing question, this method displays that question, its choices, and the correct answer.

#### 4.1.7 Post:

A post is an object that is created when a user enters a question in the information desk forum. Its main contents are a title and description. A post object also saves user information of the user that created the post object, which is then used for showing who created the post by displaying a picture, username, and timestamp. It is viewable by admin. An admin also has the ability to delete a post, which will also delete all comments attached to that post. A post requires the following fields:

- **id:** This allows for post identification for adding comments to one post, deleting posts, and viewing posts.
- **title:** Similar to most forum formats, a post will require a user to enter a title.
- **description:** This is the user input detailing their question(s).
- **userID:** Each post displays what user posted it (unless anonymous).
- **timestamp:** This is the time displayed of when the user posted the question.

As admins will have the ability to view all posts, comment on posts, and delete posts or comments, the methods needed are the following:

- **deletePost(Post)**: This takes in a post object and deletes the data in the database that is attached to a post's id
- **getPost()**: This grabs all of the posts from the database so they can be listed by title.
- **getPost(id)**: This gets an individual post using the id of the post that was clicked.

#### 4.1.8 Comment:

A comment is what is created when an admin enters a comment on a post in the information desk forum. When a comment object is created, it saves the admin's data. With this data, a comment is then displayed with a picture identifying the admin as an administrator with a username as well. Comments are created and added on individual posts. All comments by any user type are viewable by admins. Any comment can also be deleted by an admin. A comment requires the following fields:

- **id**: This allows for comment identification.
- **postID**: For storing comments so they are only tied to the post commented on, the id from the post is saved.
- **input**: This is the user, admin, or provider input text that will be displayed when posted.
- **userID**: Each comment displays what user, admin, or provider posted it (unless anonymous).
- **timestamp**: This is the time of when comment was posted that is displayed.

Admins will have the ability to view, add, and delete comments. The following are the methods required to do so:

- **getComments(postID)**: Retrieves comments from database for an individual post.
- **addComment(Comment)**: Adds a comment to the comment table in the database.
- **deleteComment(id)**: Deletes a specific comment based on the comment id entered.

#### 4.1.9 Chat:

A chat is an object created by a user but is viewable by an admin for management purposes. A chat saves a userID and cohort as fields, so admins can view chats based on what cohort they want to see. As well, they will see who sent the chat and at what time. Admins have the ability to monitor chats by cohort and delete chats if they feel necessary. A chat object requires the following fields:

- **id**: This is the unique id used to identify a chat



- **userID**: The userID allows the chat to be connected to the user that posted it.
- **cohort**: This defines where the message is to be displayed.
- **message**: This is the user input message.

For admins to be able to properly manage the chat rooms, they require the following methods to interact with chat objects:

- **getChats(cohort)**: Chats are retrieved and displayed based on the cohort the admin inputs.
- **deleteChat(id)**: deletes an individual chat based on the id passed through.

Now that we have described the administrator module and how administrators will be able to interact with the web application, we will now describe the provider side of the web application.

## 4.2 Provider Web Application Module

The provider web application module is a user interface that allows providers to log into iMATter to view and respond to information desk forum posts. Providers are also able to edit their profile information, which is displayed publicly for other iMATter users to see. Figure 4.2 below illustrates the components that the provider web application module will be comprised of.

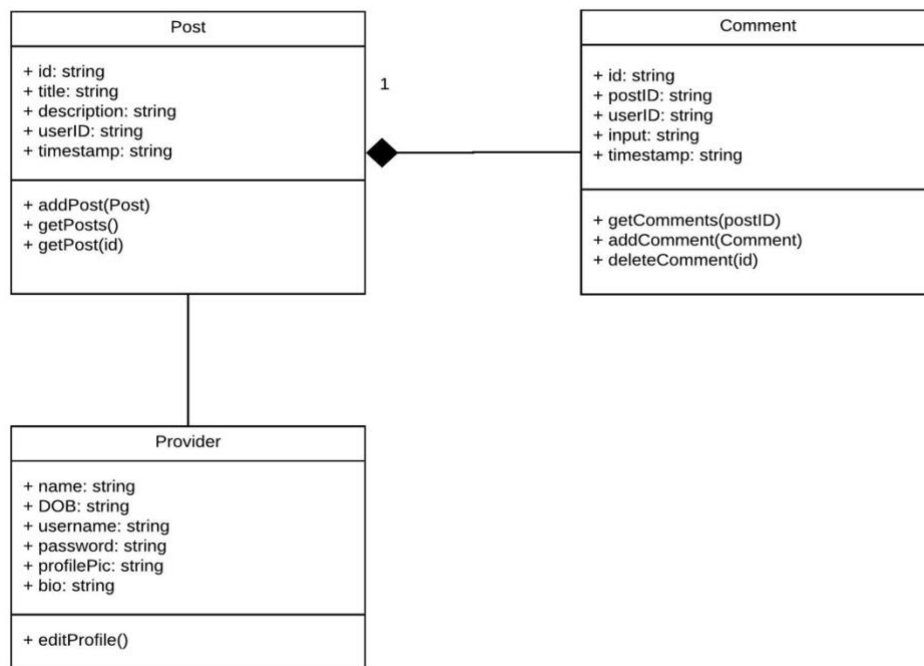


Figure 4.2 - Provider Web Application Module UML

To better explain the functionality of the provider web application module, the details of each component are as follows:

#### 4.2.1 Provider

The provider is tasked with answering users' questions in the information desk forum. They also have the ability to edit their profile and can change their information such as their bio, email, password. The provider interacts with users through the information desk by commenting on posts. The following will describe the fields that the providers have:

- **name:** The providers will provide their first and last name so they can be identified.
- **DOB:** The providers will have their date of birth stored in the database.
- **username:** As the providers will be using the information desk, they will need a username to be displayed.
- **password:** The providers will need a password so they can log into the web application.
- **profilePic:** The providers will have a profile picture so that can be viewed on their profile page.
- **bio:** On the providers profile page, they can add a bio where they introduce themselves.

The providers can perform these actions:

- **editProfile():** Providers can edit their profile and change their bio, email, and password.

#### 4.2.2 Post:

A post is an object that is created when a user enters a question in the information desk forum. Its main contents are a title and description. A post object also saves user information of the user that created the post object, which is then used for showing who created the post by displaying a picture, username, and timestamp. Through this, users are able to interact with the providers. Providers may only view this object, not create it. Providers can see a list of post titles on their homepage and can click a title to be brought to the specific post where they will see all of the fields displayed. A post requires the following fields:

- **id:** This allows for post identification for adding comments to one post, deleting posts, and viewing posts.
- **title:** Similar to most forum formats, a post will require a user to enter a title.
- **description:** This is the user input detailing their question(s).
- **userID:** Each post displays what user posted it (unless anonymous).
- **timestamp:** This is the time displayed of when the user posted to question.

Providers will only have the ability to view posts, so the following methods are required:

- **getPosts():** This grabs all of the posts from the database so they can be listed by title.
- **getPost(id):** This gets an individual post using the id of the post that was clicked.

#### 4.2.3 Comment:

A comment is what is created when an admin enters a comment on a post in the information desk forum. When a comment object is created, it saves the provider's data. With this data, a comment is then displayed with a picture identifying the provider as a provider with a username as well. This picture can be clicked by any user type to be brought to a viewable profile. Comments are created and added on individual posts. All comments by any user, admin, or other providers are viewable by providers. Providers are able to interact with users, admins, and other providers with comments. A comment requires the following fields:

- **id:** This allows for comment identification.
- **postID:** For storing comments so they are only tied to the post commented on, the id from the post is saved.
- **input:** This is the user, admin, or provider input text that will be displayed when posted.
- **userID:** Each comment displays what user, admin, or provider posted it (unless anonymous).
- **timestamp:** This is the time of when comment was posted that is displayed.

Providers will have the ability to view, and add comments to posts, which will require the following methods:

- **getComments(postID):** Retrieves comments from database for an individual post.
- **addComment(Comment):** Adds a comment to the comment table in the database.

### 4.3 Mobile Application Module

The mobile application is what users (pregnant women in opioid recovery) will be using. This application will allow them to interact with other users and have a place to learn and keep track of events related to their recovery. Figure 4.3 below shows a UML diagram for the mobile application implementation of iMATter.

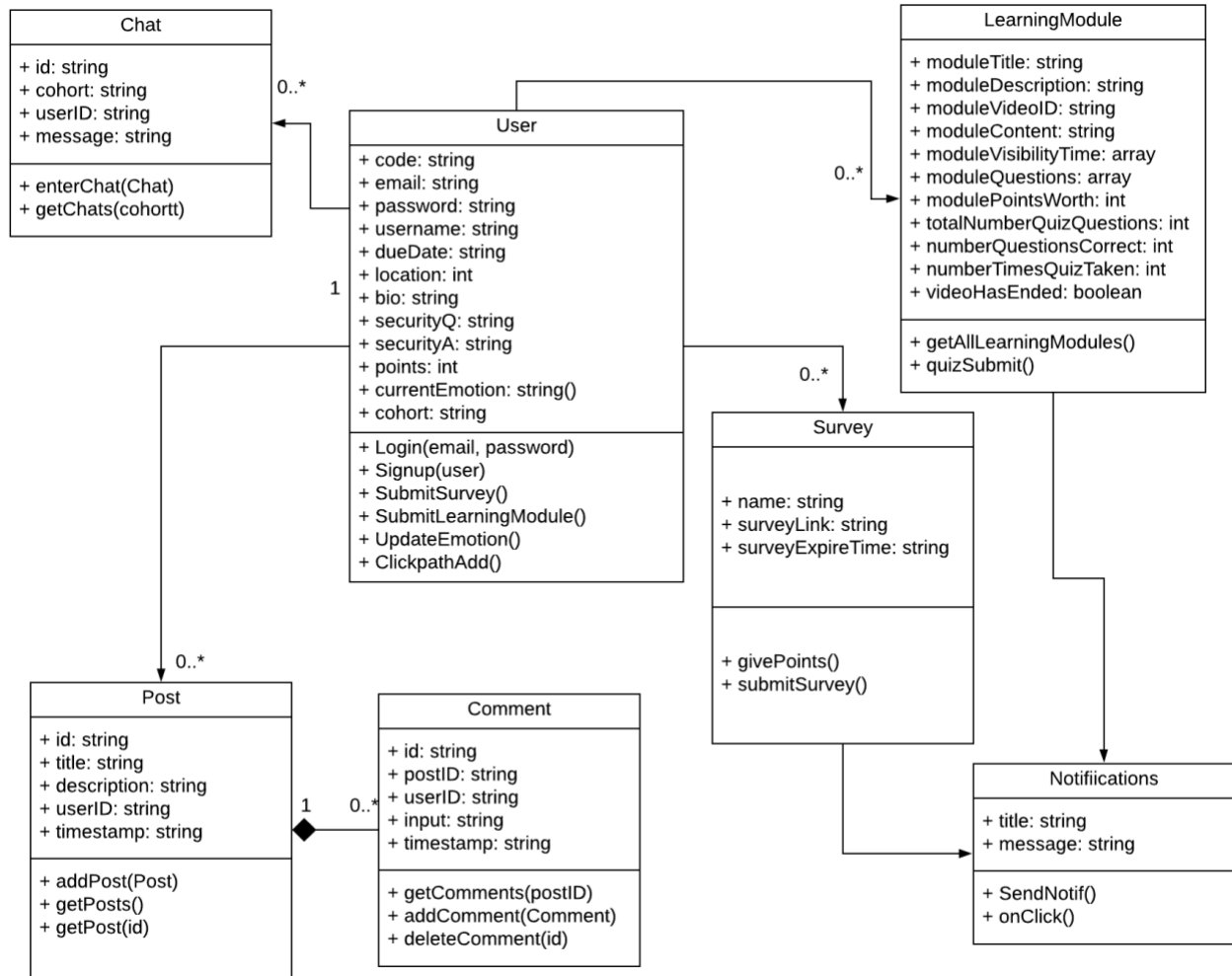


Figure 4.3- Mobile Application Module UML

A more detailed explanation of the functionalities of the mobile application implementation is as follows:

### 4.3.1 User

We define a user as a pregnant woman in opioid recovery that has been given access to this mobile application. They are the users that will be interacting with the classes shown in Figure 4.3. A user will be able to sign up for the mobile application after receiving a code from an administrator. Once they are signed up, they will have the ability to login to the application. Once logged in, users can interact with multiple different components. Users can update their emotion by clicking a button, which depending on the emotion will trigger different protocols. Users can interact with the information desk forum, which allows them to create post objects or comment objects by commenting on posts. Another functionality users have is creating chat objects in the chatroom, which will be sent to and visible to only other users that are also in their cohort. Users can interact with the learning module component by watching videos

provided and answering questions. Users can interact with survey objects by clicking a button that will bring them to a provided URL. Surveys and learning modules will also have a Notification component, which will notify users of when new ones are added to the application. To go into more detail about the fields of users:

- **code:** Before downloading the application, a user should have spoken to an administrator and received a code, which is a six-character alphanumeric string. Entering this code when prompted will allow access to sign up and serve as an identifier for administrators to use.
- **email:** Email is saved to the database for login purposes.
- **password:** This is used as a form of security, so only those with a password that matches what is saved under the email entered may enter the application.
- **username:** A pseudonym is used as a form of in-app identification for users in order to keep this application anonymous.
- **dueDate:** Users are placed in a cohort based on what month they are expecting to deliver, so this is used to place users in the appropriate cohort. Users will also have customized functionalities based on what week they are in their pregnancy, so this is going to be used for customizing what the user sees.
- **location:** This is an optional field, but if given, listed resources for recovery can be given based on zip code locations. Otherwise, national numbers will be provided.
- **bio:** Users may describe themselves for other users to see.
- **securityQ:** In the case of losing email and password used for login, users will have a security question they must choose to answer.
- **securityA:** This is the answer to the security question, it will be viewable by administrators so they may provide users with their email if the user gives the administrator the correct answer.
- **points:** Points are rewarded from different participation in the application, it will serve as a way for users to receive benefits from using this application, such as gift cards.
- **currentEmotion:** Users can log what emotion they are currently feeling, which will be shared with their cohort.
- **cohort:** This is created based on what dueMonth the user enters, used to decide which chatroom to show user.

For the operations users will be able to perform:

- **Login(email, password):** Users can login into the application with their email and password
- **Signup(User):** Users can submit user information in a signup form that is presented after they enter their code.
- **SubmitSurvey():** Users can click a “Completed” button to show they completed the survey they took and get points for it.
- **SubmitLearningModule():** Users can submit a learning module after completion which will check for accuracy of answers or completion and will give points to the user.
- **UpdateEmotion():** Users can press an emoticon that will update how they are currently feeling in the database. Certain emotions will trigger different protocols, such as a negative emotion will cause a pop-up to come up and remind users of resources, learning modules, and the chat room.
- **ClickpathAdd():** when users click on different buttons to navigate to other pages on the app the click of where they go will be added to the database.

#### 4.3.2 Survey

A survey consists of a Qualtrics link and a time period that the survey will be displayed to users. Users can take the survey, then earn points for finishing it. The survey object will display a button which when a user presses, will open the link inside an in app browser. The purpose of this component is to provide mobile application users with a way to access surveys that administrators wish for them to regularly complete. A survey will require the following fields:

- **name:** This will be the title displayed given by the creator of the survey.
- **surveyLink:** This will be a hyperlink that will take the user to the actual survey provided by a survey service inside the mobile application.
- **surveyExpireTime:** This will be the time the survey will expire and not be available, and it will be displayed for the user.

A survey will be provided functionality from the following functions:

- **givePoints():** Will give points to the user for taking and completing the survey.
- **submitSurvey():** Will indicate that the survey has been by the user and remove it from the list of available surveys.

### 4.3.3 LearningModule

A learning module is a page containing informational media such as videos followed by questions for users to answer regarding the information presented. The learning module component in the mobile application is responsible for retrieving learning module fields from the database and displaying them for the user to see. It also checks the user's week of pregnancy against the module visibility set by the administrator to ensure that users are only viewing learning modules they are meant to. In order to allow the user to view learning modules and answer questions, the user component will interact with the learning module component. The learning module is also responsible for handling checking user's quiz question answers for correctness and displaying their quiz scores. Its fields are as follows:

- **moduleTitle:** The title of the learning module.
- **moduleDescription:** A brief description of what the learning module is about, to help give the user an overview of the module.
- **moduleVideoID:** The video identification code of a YouTube video that is embedded in the learning module.
- **moduleContent:** Relevant instructions or information about the learning module that is displayed before the informational content.
- **moduleVisibilityTime:** The week(s) of pregnancy that the learning module should be displayed to users.
- **moduleQuestions:** The questions that are displayed after the informational media and answered by users.
- **modulePointsWorth:** Since we will be implementing a reward system that gives points for completing learning modules, this stores the number of points the learning module is worth.
- **totalNumberQuizQuestions:** The number of quiz questions in a learning module.
- **numberQuestionsCorrect:** The number of quiz questions a user answers correctly. Used to display the quiz score to the user.
- **numberTimesQuizTaken:** The number of times the quiz has been submitted. Used to make sure the user does not exceed the quiz submission limit.
- **videoHasEnded:** A boolean value that keeps track of whether the learning module's video has finished. Used to determine whether quiz questions should be shown to the user.

Given that the mobile application user only needs to be able to view learning modules and answer quiz questions, there are only two methods needed for the mobile application learning module component:

- **getAllLearningModules():** Retrieves all the saved learning modules from the database.
- **quizSubmit():** Collects the user's selected answer and checks to see if they are correct.

#### 4.3.4 Notifications

Notifications are a vital tool for engaging mobile application users with the application by reminding them of appointments or alerting them of new posts or learning materials in the application. Users will be able to receive notifications when new learning modules are added, surveys are added, a chat has been sent to the chat room, or calendar events are occurring. Notifications will have the following fields:

- **title:** The title of the notification, which typically appears in bold on a pop-up banner.
- **message:** The message that is being communicated to users.

Given the simplistic nature of notifications, this component will only need the following methods:

- **SendNotif():** The action that actually sends out the notification at a given time or under certain circumstances.
- **onClick():** This designates what should happen after users tap on the notification.

#### 4.3.5 Post:

A post is an object that is created when a user enters a question in the information desk forum. Its main contents are a title and description. A post object also saves user information of the user that created the post object, which is then used for showing who created the post by displaying a picture, username, and timestamp. It is viewable by admins, providers, and other users. Users will see a list of post titles and can click on to see the additional fields of a post as well as comments that have been made on that specific post. A post requires the following fields:

- **id:** This allows for post identification for adding comments to one post, deleting posts, and viewing posts.
- **title:** Similar to most forum formats, a post will require a user to enter a title.
- **description:** This is the user input detailing their question(s).
- **userID:** Each post displays what user posted it (unless anonymous).
- **timestamp:** This is the time displayed of when the user posted to question.



As users will have the ability to view all posts and comment on posts, the methods needed are the following:

- **addPost(Post):** This takes in a post object and stores the data in the database that is input.
- **getPosts():** This grabs all of the posts from the database so they can be listed by title.
- **getPost(id):** This gets an individual post using the id of the post that was clicked.

#### 4.3.6 Comment:

A comment is what is created when an admin enters a comment on a post in the information desk forum. When a comment object is created, it saves the user's data. With this data, a comment is then displayed with a picture identifying the user with a username as well. This picture can be clicked by any user type to be brought to a viewable profile. Comments are created and added on individual posts. It is viewable by admins, providers, and other users. A comment requires the following fields:

- **id:** This allows for comment identification.
- **postID:** For storing comments so they are only tied to the post commented on, the id from the post is saved.
- **input:** This is the user, admin, or provider input text that will be displayed when posted.
- **userID:** Each comment displays what user, admin, or provider posted it (unless anonymous).
- **timestamp:** This is the time of when comment was posted that is displayed.

Users will have the ability to view and add comments to a specific post. The following are the methods required to do so:

- **getComments(postID):** Retrieves comments from database for an individual post.
- **addComment(Comment):** Adds a comment to the comment table in the database.

#### 4.3.7 Chat:

A chat is an object created by a user. A chat saves a userID and cohort as fields, so users will only see chats from their cohort. The ID is used to display user information with the chat, such as a profile picture and username. Users may send chats, which will require the following fields to send:

- **id:** This is the unique id used to identify a chat
- **userID:** The userID allows the chat to be connected to the user that posted it.

- **cohort**: This defines where the message is to be displayed.
- **message**: This is the user input message.

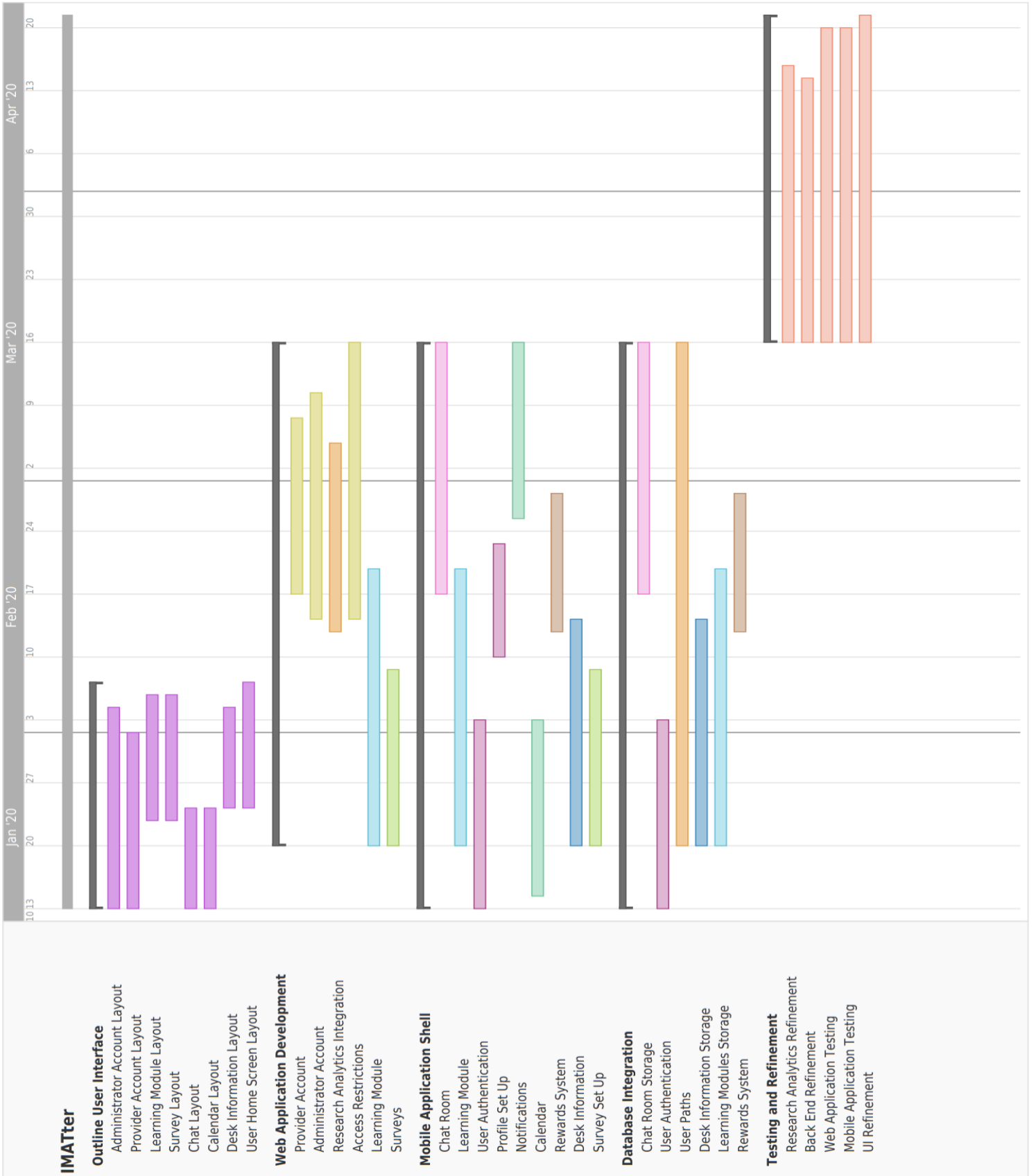
For users to be able to interact with the chat room, the following methods are required:

- **getChats(cohort)**: Chats are retrieved and displayed based on the cohort the user is placed in at signup.
- **EnterChat(Chat)**: Users can enter a message in the chatroom that will be stored in the database and then show up in the chatroom for their cohort.

## **5.0 Implementation Plan**

As a team we have set up an implementation plan, which we will follow for the remainder of the semester. Our plan has been segmented into five sections: Outline User Interface, Development Web Application, Mobile Application Shell, Database Integration, and Testing and Refinement. As a novel end-user focused product aimed at largely non-technical users, careful and early design of GUIs and user flows is critical. Therefore, we prioritize close collaboration with our client on these early on, while simultaneously developing basic internal elements and increasing our understanding of the tasks. We have developed our implementation plan by considering the requirements of our mobile application, web application, storing user information, our flow, and improvements that can be done. Each of these sections have been assigned smaller tasks and will be separated among all our team members. The following figure created is a visual representation of our implementation plan.

Figure 5.0 - Implementation Plan Gantt Chart



## 5.1 Outline User Interface

*January - February 2020*

During this phase we aim to have an outline of our user interface, which will serve as a guideline to account flow. This task is fundamental for providing a user experience that satisfies the use cases and requirements we have come up with. This task includes having a detailed plan on user workflow and mapping out where each feature of our application should be placed:

- General structure of web and mobile Applications
- Flow worked out
- Drawn out interfaces
- Understanding of web and mobile application interfaces
- Outline skeleton of web application
- Outline skeleton of mobile application

As shown in Figure 5.0, some major phases in this section include:

- Admin Account Layout
  - We will have the general outline of how the admin account for the web application should look and where everything will be placed.
- Learning Module Layout
  - In this phase, we take a look at how the admin will create learning modules and how users will be able to see these learning modules.
- User Home Screen Layout
  - We will be able to outline the user interface for the mobile application and have a better understanding of the flow the user will experience.

## 5.2 Web Application Development

*January - March 2020*

We will develop the main features for our web application. The user interface will be cleaned out to be more suitable as a final product. The web application will have the administrative capabilities implemented and be able to communicate with the mobile application.

As shown in Figure 5.0, some major phases in this section include:

- Provider Account

- We must add capabilities that will make up the provider account. For example, forum tools and a profile set up for the account.
- Access Restrictions
  - Each type of web account has different access to the chat room, information desk, and the ability to add users. Access restrictions for administrators and providers will be developed during this phase.
- Learning Module
  - We will create the process for administrators to add learning modules and be able to have this implementation on the web application for mobile users to see the final learning module

### **5.3 Mobile Application Shell**

*January - March 2020*

Concurrently with the web application, we will be developing the mobile application. This will implement the major functionalities expected from our mobile application features and will serve as a solid application with a few refinements later on. The expected functionalities for this basic mobile application include:

- User Interface
- Local Communications
- Web Communications
- Main Features

As shown in Figure 5.0, some major phases in this section include:

- Chat Room
  - Creating a messaging system for cohorts and communication to web application
- Information Desk
  - The information desk is a forum that needs to have the capability of adding questions that all cohorts can view. This forum needs to be able to communicate with the web application along with the database.
- Rewards System
  - This rewards system will be communicating between the user and the web application. Each module will be implementing points to users and we need to have a set up for points.

## 5.4 Database Integration

*January - March 2020*

Parallel in creating all accounts, models and features, everything will be stored in the database for future usage. Some usage of the database will be:

- Storing information
- Accessing the Database

As shown in Figure 5.0, some major phases in this section include:

- User Authentication
  - All user information will be stored in the database in order for other accounts for example the administrator can access. This will also help with creating new users and or updating user's information and profile set up.
- User Paths
  - In order for administrators to have access in the research analytics component of their account we must be able to store user paths. User paths will be stored in the database to be used by Chart.js for charts.
- Rewards System
  - We will store the users points in the database and access those points for the user to redeem and for the administrator to access them.

## 5.5 Testing and Refinement

*March 2020 - April 2020*

With the expectation of running into obstacles and needing to make adjustments to our application, we have planned more than a month for testing and refinement of our software solution. We will be running through our use cases once more to make sure we have covered all our scenarios and requirements. As we expect our mobile and web applications to be fully implemented by this time, we will be testing the products and refining them iteratively based on comments received from our sponsors.

As shown in Figure 5.0, some major phases in this section include:

- User Interface Refinement
  - By this time, our features have all been implemented and during this phase we will make the mobile and web applications have more appeal. We will edit the display of certain icons and or even change some aspects of the design based creating better user experience.

- Web Application Testing
  - We will come up with tests for the web application by running through our use cases and make sure all the capabilities it needs to run smoothly.
- Mobile Application Testing
  - We will come up with tests for the web application by running through our use cases, interaction between the web application are in functioning and make sure all the capabilities it needs to run smoothly.

## **6.0 Conclusion**

Millions of Americans struggle with opioid addictions, and over 130 people die from opioid overdoses every single day. This national crisis is costing the US billions of dollars each year for healthcare, lost productivity, and addiction treatments, and is only getting worse. Not only are overall rates of opioid addiction increasing, but the rates of pregnant women who suffer from opioid use disorders are growing as well. Pregnant women with opioid addictions who want to start on the path to recovery are placed in comprehensive treatment programs. Due to the intensity of the requirements of the treatment

Our iMATter product is aimed at revolutionizing opioid treatment by leveraging modern web and mobile technologies to make effective treatment accessible to thousands of women, many of which currently have no practical options. Specifically, pregnant women in opioid recovery who live in rural areas. Our product will help pregnant women in recovery receive social support, a way to learn about recovery and pregnancy, and resources to help aid their recovery.

Through the research of our client Dr. Emery Eaves, we have identified that an overwhelming schedule, lack of social support, and lack of information access that pregnant women in recovery have to deal with are major obstacles for a successful recovery. With our mobile application, we aim to help these women feel heard and safe, as well as provide a way for them to talk about exactly what they are going through without being judged. Overall, the mobile application will include the following main features:

- **A chat room**, which will be able to help women in recovery interact with other women that are experiencing similar situations. This feature will help them receive and give support and advice. As women are grouped into cohorts based on the same month they are due, they will be able to better understand what one another may be going through at that time in their pregnancies.
- **A learning center**, which will help pregnant women in recovery learn more about their pregnancy and recovery. They will have a way to find verifiable information in a single place.

- **An information desk**, which will provide a safe and reliable source of information from medical professionals who are trained to help inform people in opioid recovery. Our users will be able to ask any questions in this part of the mobile application, as well as see other questions that have already been asked and what answers were provided. This will allow for women to ask questions they may feel scared or embarrassed to ask anyone otherwise, as they will have the option to remain completely anonymous.

In this document, we have presented a detailed design for our mobile and web application to show an outline of how we will be building our software. Our architectural overview details the connections between the three major modules of our applications and defines a more in-depth look at our methods. Our administrator web application module serves as a user interface for our administrators and manages all content from user accounts to creating learning modules for users. The provider web application is an interface for our providers that allows them to edit their profile information, as well as view and respond to user forum posts. Finally, our mobile application module provides a user interface for our mobile users. This module will allow for social interaction between users, provide a learning environment, and keep track of events related to their recovery. We have also constructed an outline that details our steps to implement our software.

We believe our mobile application will provide a revolutionary method of support for the recovery process for many pregnant women with opioid use disorder in recovery, especially women who are struggling in under-served rural areas. We have currently been able to create and implement our user interface designs for our mobile and web application, start progress on our web accounts, and create the majority of the functionalities for our main features, such as the information desk forum, chat room, learning modules, and surveys. The major steps that have been taken are reflecting back to our use cases and making sure as a team we are following them. Our next step is to add notification functionality to the mobile application and begin implementing some of our other tasks, such as a point system. Our team is very optimistic and extremely excited with the progress we have been making in regard to being on track with our implementation plan and completing project requirements. The iMATter application will provide pregnant women in recovery with a safe environment, social support, and educational material to help guide them to a successful recovery.