

DIABETES SOCIAL NETWORKING APP FOR AMERICAN INDIANS

A Paper  
Submitted to the Graduate Faculty  
of the  
North Dakota State University  
of Agriculture and Applied Science

By

Rachana Gurram

In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF SCIENCE

Major Department:  
Computer Science

March 2019

Fargo, North Dakota

North Dakota State University  
Graduate School

---

**Title**

Diabetes Social Networking App For American Indians

---

**By**

Rachana Gurram

---

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

**MASTER OF SCIENCE**

SUPERVISORY COMMITTEE:

Dr. Juan Li

---

Chair

Dr. Oksana Myronovych

---

Dr. Ranjit Prasad Godavarthy

---

Approved:

March 12, 2019

---

Date

Dr. Kendall E. Nygard

---

Department Chair

## **ABSTRACT**

Chronic disease effects health and quality of living largely. Social networking can initiate the onset of pathology and serve as direct cause for several chronic and non-communicable diseases. Technology has the potential to create true collaborative care among multiple providers in healthcare systems and social media can serve as virtual office. Social networks can be used to build relationships between patients and their families to extend health care and support beyond traditional points of service like hospitals, clinics to the home, schools, communities. Social networks can be used to reach patients and providers to gather information and leverage collaboration between them. Many patients who research about their health conditions are also likely in search of their fellow patients with same conditions, to discuss their health, gather knowledge and support before making their healthcare decisions. With all this advantages, social networks can be used as a gateway to chronic disease care.

## TABLE OF CONTENTS

ABSTRACT.....	iii
LIST OF FIGURES .....	vi
LIST OF ABBREVIATIONS.....	viii
1. INTRODUCTION .....	1
2. RELATED WORK.....	3
3. SYSTEM DESIGN.....	5
3.1. Structure .....	5
3.2. MVC Architecture.....	6
4. IMPLEMENTATION.....	8
4.1. User Interface .....	8
4.1.1. Ionic Framework .....	9
4.2. Database Design.....	10
4.2.1. Firebase.....	11
5. USE CASES.....	16
5.1. Authentication .....	16
5.2. Friends .....	17
5.2.1. Sending Friend Requests .....	18
5.2.2. Cancelling a Pending Friend Request.....	19
5.2.3. Friends List and Received Friend Requests .....	20
5.3. One to One Conversations.....	22
5.4. Group Conversations.....	23
5.5. Profile .....	25
5.5.1. Settings .....	27
5.6. Templates .....	28

6. CONCLUSION AND FUTURE WORK .....	30
6.1. Conclusion.....	30
6.2. Future Work .....	30
7. REFERENCES .....	31

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1: Use Case Diagram for Authentication .....	5
2: Use Case Diagram for Contacts .....	6
3: MVC Component Interactions.....	7
4: Structure of Social Networking App .....	8
5: Basic Profile Info Structure .....	11
6: Setting Endpoint for One-to-One Chat .....	12
7: Example of User End point Structure .....	12
8: Account Node Structure.....	13
9: Conversation Node Structure.....	14
10: Users Node Structure .....	14
11: Messages Node Structure.....	15
12: Registering an Account.....	16
13: Login Button Before and After Entering the Password.....	17
14: Searching for New Friends .....	18
15: Sending a Friend Request .....	19
16: Deleting a Pending Friend Request .....	20
17: Friends List Before and After Adding Friends .....	21
18: Friend Requests Received.....	21
19: One-to-One Conversation .....	22
20: Creating a Group and Adding Users.....	23
21: Messages and Notifications in a Group Chat.....	24
22: Basic Group Information .....	24
23: Basic Profile Information.....	25

24: Changing Password.....	26
25: Updating Profile Info .....	26
26: Other Settings .....	27
27: Logout or Delete the Account.....	28
28: Template for Email Verification.....	29
29: Template for Password Reset .....	29

## LIST OF ABBREVIATIONS

AI .....	American Indians.
UI .....	User Interface.
MVC .....	Model View Architecture.
HTML .....	Hyper Text Markup Language.
CSS .....	Cascading Style Sheets.
SQL.....	Structured Query Language.
JSON.....	JavaScript Object Notation.
HTTP.....	Hyper Text Transfer Protocol.



## 1. INTRODUCTION

Social media, online social networks and apps for smartphones and tablet devices are changing the way we communicate. Nowadays most of the adults engage in social networking to access, create, and share contents. Recently, the number of people using smartphones across the world is tremendously increasing. Mobile application stores like Google Play, iTunes store, etc. are providing us with thousands of social networking applications or alternative communication tools .

There are number of possible risk factors like lack of exercise leading to physical inactivity, smoking or consuming tobacco, alcohol, anxiety, high stress levels, excess cholesterol, obesity, blood pressure, etc. Most of these risk factors [1] can be prevented or cured faster by making educating people and provide them with better communication tools which support a healthy lifestyle.

In the past few years, people have shown much interest in making use if digital technologies in order to support health interventions, education and communication, because digital technologies have the capability to improve the level of surveillance compared to behaviors. Therefore, they have the potential to provide users with low-cost health interventions by transferring treatment and help in preventing a condition to self-care within the community.

With number of social networking sites and smartphone-based applications coming up, people tend to move away from time taking emails and phone calls. Users have shown tremendous interest in using social media as a major communication tool to interact with healthcare and public health systems. Since the social media platforms being very user-friendly, interactive and participatory in nature, it is convenient for the users to create and share any information in a viral fashion. They provide new techniques to enhance involvement and partnership with users and help

them change their behaviors and to fight against any unhealthy lifestyle. In today's world, there are millions of applications related healthcare systems available in the market, which allow users to record, track and analyze any conditions and their physical health data from time to time. They can obtain feedback and basic information about the condition they have, or the disease they're suffering from. They can also receive alerts or notifications as a reminder for them to take medication time to time or measure their blood glucose levels etc.

By creating online space in a social networking application, patients can interact with medical providers, interact with other patients or users to share their experiences. Bringing up frequent wellness programs incorporates social media to create an atmosphere of group people with shared goals, offers emotional and social support.

## 2. RELATED WORK

Social networking is useful in delivering healthcare to users, especially to people with chronic diseases. It supports in promoting healthy lifestyle and behaviors such as weight gain/loss, being physically active, having a healthy and balanced nutrition, etc. Social networking sites or applications are serving as a platform for patients to gather health related information, at almost free of cost. This makes the patients free from the constraints of a any traditional health care system. Social media such as Facebook, Twitter and other social networking sites [2] have already proved the power of social connectedness among individuals. Social media has a lot of influence on their behavior and lifestyle. Since beginning, delivering healthcare has been focused on individual patients rather than groups. A few well-known healthcare apps [3] are “PatientsLikeMe” and “Connected Living”. Applications like these help users to improve their health behaviors. Social networks have the potential to provide basic information and serve as a platform or community to patients and discuss their conditions with providers or other users. Otherwise, patients might struggle alone with their existing disease or a new one.

As per the website [4], “A survey of 2,125 PatientsLikeMe members in the U.S. says that adult social media users with health conditions embrace the idea of sharing their health data online if it helps clinicians improve care, helps other patients, or advances medical research.”

Be it a group conversation or one-to-one conversation, face-to-face communication is always named as the best mode of communication. However, with the advancement in technology, and an effort to involve younger patients, setting up virtual connections through social media is slowly being prioritized on top of the list. Social media platforms [5] are being standardized as more stable channel for communication and connection with patients and among themselves.

Major parties to be involved in a healthcare social application are patients, providers and nurses. Things that need to be considered while building the application are what tools to use, how to integrate this app into workflows, and how to involve patients and providers successfully.

As per the blog [6], “Social media offers advantages over traditional data sources, including real-time data availability, ease of access, and reduced cost. Social media allows us to ask, and answer, questions we never thought possible” said Prof. Michael Paul and Prof. Mark Dredze in their book called “Social Monitoring for Public Health”.

In this paper, we designed an ionic-based social networking application for diabetes management. Nobody understands what it is like to have an illness better than someone with the same illness. This social networking application provides diabetes patients with an opportunity to know more about their illness and get peer support from others with similar experiences. This level of engagement and communication with other patients or providers is important, especially for people living in rural areas and limited peers in network to communicate with.

Therefore, to promote a healthy lifestyle, we propose a social networking platform customized for AI patients. In this app, we use a Firebase real-time database to assist patients automatically find appropriate peer patients to build network and friendship and obtain peer support based on their health conditions and social behaviors.

### 3. SYSTEM DESIGN

Overall, the goal of the project is to build a cost-effective software application for diabetes healthcare management and social connectivity for AI patients. The project is divided into several modules, such as, ontology definition, personalized recommendation based on their physical condition, social relationships and geolocation, etc., personalized health risk prediction, social networking, etc. Social networking is initially developed as its own module which will be integrated with the knowledge base in the overall project. The possible stakeholders for this application can be the AI patients and the medical providers.

#### 3.1. Structure

The application is developed using JavaScript, Angular, HTML5 and CSS. The proposed social networking app is designed in such a way that client-side application works even if the connection to the real-time database is lost. All the data persists on the client-side, and if the data is modified, it gets updated as soon as the connection is established with the server-side. Below is the pictorial representation of how the app is structured.

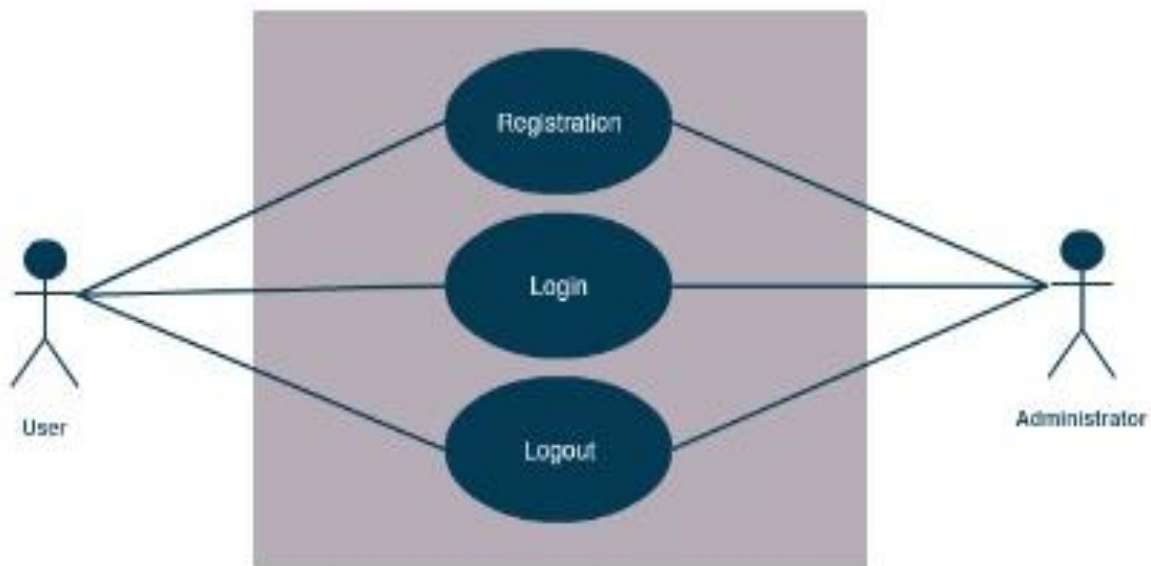


Figure 1: Use Case Diagram for Authentication

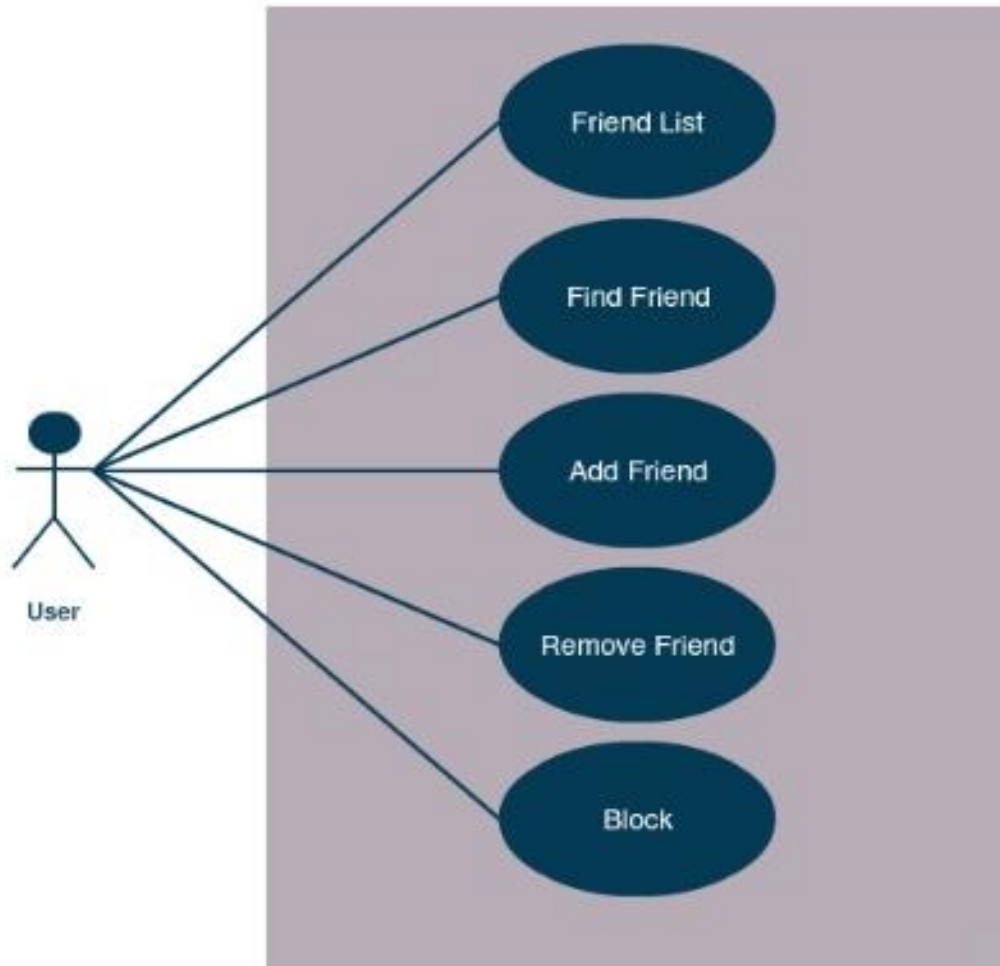


Figure 2: Use Case Diagram for Contacts

### 3.2. MVC Architecture

Model-View-Controller (MVC) is an architectural pattern used in software applications to implement some of the user interface systems. Since the social networking app requires a stronger software with a good life cycle, where maintaining the code is easy and reusable in future. Also, this concept of separating the business logic, database and domain from the user interface aids in maintainability and adding any new features to the application would not be a problem with this kind of architecture.

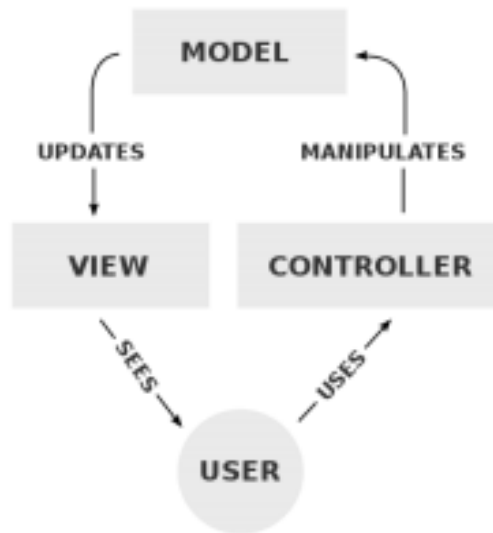


Figure 3: MVC Component Interactions

In this application, firebase is considered as a model where the data is stored as JSON objects with all the authentication, read and write permissions/rules. HTML is used to build the user interface of the application and view the pages. Finally, controller is the part where the javascript code obtains data dynamically, calculates and renders into the HTML content. Whenever user takes an action in the UI, controller calculates and updates the model. Once the model receives the updates, it passes the updated to view, showing the new data to the user.

## 4. IMPLEMENTATION

In this section, we present the implementation and technical details of the social networking application. Since this application is developed to be part of an overall comprehensive app which is developed based on ionic-framework, the same technology is used for this application as well. In the world of software development, there are number of new and incredible tools that help us build applications. Firebase is one such application. Firebase is a Google mobile platform that helps and supports to develop high-quality mobile and web applications very quickly. Therefore, for the backend purposes, Firebase is used for storing the data.

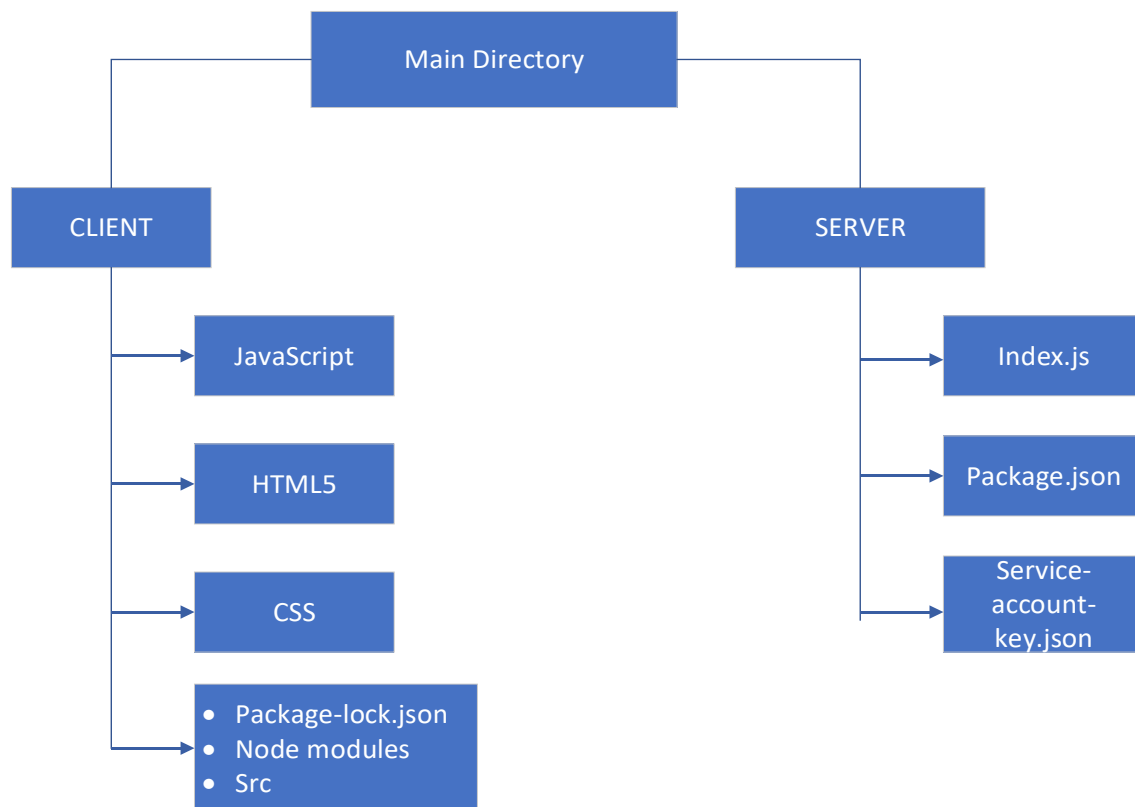


Figure 4: Structure of Social Networking App

### 4.1. User Interface

One of the real challenges for building the application is to create an elegant look and feel and ensure that the application offers to AI patients not just the functionality, but also how it



delivers complete user satisfaction. UI(User Interface) plays an important role in guiding the users through their journey and navigating through the application. Ionic framework is one such technology which helps us build an application that delivers unique product and successfully showcases it within the application.

#### **4.1.1. Ionic Framework**

Ionic is an open-source framework used to build mobile applications. Ionic applications are written in Javascript using HTML and CSS. Ionic framework is the UI framework considered for this application. It handles the UI interactions of applications and supports native mobile components and beautiful design. When developing a large application using JavaScript language, Ionic framework provides developer with the infrastructure. Ionic has tools and components allowing the developers to build applications with functional and elegant UI, with the native look and feel of the app.

Ionic framework can run on mobile devices with the native wrapper. Also, ionic is used for building hybrid mobile applications. It supports cross-platform application development, i.e., the apps can be compatible with IOS, Android, Windows, etc., thereby saving developers a lot of working time. Ionic also supports UWP for building Windows 10 apps.

Since the application is exclusively build for AI patients with diabetes, it is very important to remember that they might have limited knowledge or experience with software and mobile applications. Ionic framework has lots of components and themes which can be customized further. UI components and icons that ionic provides are self-explanatory thereby making it easier for AI patients to understand the app better and interact with the UI at ease.

## 4.2. Database Design

One of the most challenging part while developing the application is to data synchronization across all client devices every time the data is changed. Creating a SQL database, developing the backend, retrieving the data using typical HTTP requests would not fit in right with this application. Firebase is one such application which is real-time, offline, accessible from all client devices, and is scalable across multiple databases.

Firebase is a cloud-hosted database. Not only does Firebase real-time database allows syncing data across all client-side in real time, but also, the data continues to be available even if the application goes offline. According to the website [7], all Firebase real-time database data is stored as JSON objects. It is something like NoSQL database where all information is stored as JSON-objects and the structure looks like a JSON tree. Firebase real-time database, unlike a SQL database, has no tables or records. If new data is being added into the database, a new node will be created with a key-value. Whenever data is added to the JSON tree, it becomes a node in the existing JSON structure with an associated key. You can provide your own keys, such as account ID. For example, consider our chat application that allows users to store a basic profile information.

Although the database uses a JSON tree, data stored in the database can be represented as certain native types that correspond to available JSON types to help you write more maintainable code. So, ultimately, Firebase was an effective solution for developing a cloud database for this application. It integrates well with the ionic framework [8] and provides fast synchronization on all client devices.



Figure 5: Basic Profile Info Structure

#### 4.2.1. Firebase

Another challenging part of storing the data is deciding how we structure the JSON tree and how we store the JSON objects. With the Firebase real-time database [9], we will store each user's information at an endpoint, which is identified by their id. User's data will be stored at user's endpoint. With this JSON structure, grabbing user's information to fill in their profile will not be quick and easy.

##### 4.2.1.1. Defining the Nodes(Structure of the Application)

One of the complex parts of designing the app is deciding how the messages will be stored. Google's Firebase provides us with two options, the first one being real-time database where all messages are stored at a message node and the second option is Firestore, where the messages are stored as a message collection. With the JSON structure of the database of our application, all messages will have a unique id and associated author's id. For instance, consider how a typical

one-to-one conversation works. When user X begins a conversation with user Y, a new thread is created for the two of them and so on. `

Every user has their own id, which is a string. In order to model this kind of structure, whenever user X starts a conversation with user Y, you compare their two ids and to the lower id, append the greater id. This can be applied if the conversation is small. But the application might crash if it gets large as querying a person's conversation will require walking through the whole messages' node.

```
/**Function setup endpoint for one to one chat**/  
private String setOneToOneChat(String uid1, String uid2)  
{  
    //Check if user1's id is less than user2's  
    if(uid1 <uid2)  
    {  
        return uid1+uid2;  
    }  
    else  
    {  
        return uid2+uid1;  
    }  
}
```

Figure 6: Setting Endpoint for One-to-One Chat

For example, if person X's id is 'x' and person B's id is 'y', you create a thread between them with a node 'xy' and all their messages are stored at that node. In this situation, we can just query the endpoint of the conversation and load the messages between the two persons. With this structure, we always end up starting a new thread between any pair of users.

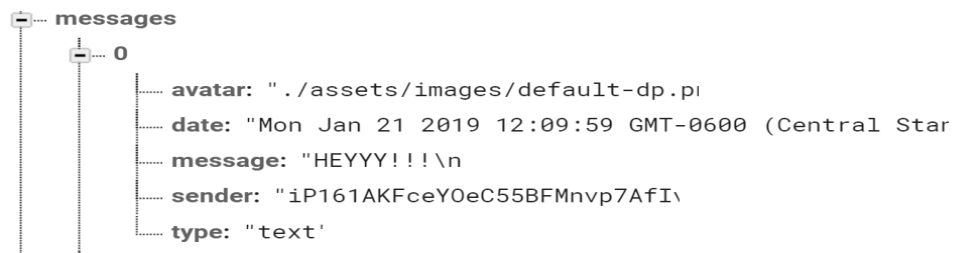


Figure 7: Example of User End point Structure

### 4.2.1.2. Account Node

Account node is created under the root of the database. Every account node has its unique ID. All the basic profile information related to the user is stored under this account node. For example, age, date of birth, email, user ID, etc. Every user has its own user ID. Below is an example of how accounts node looks in the JSON tree structure.

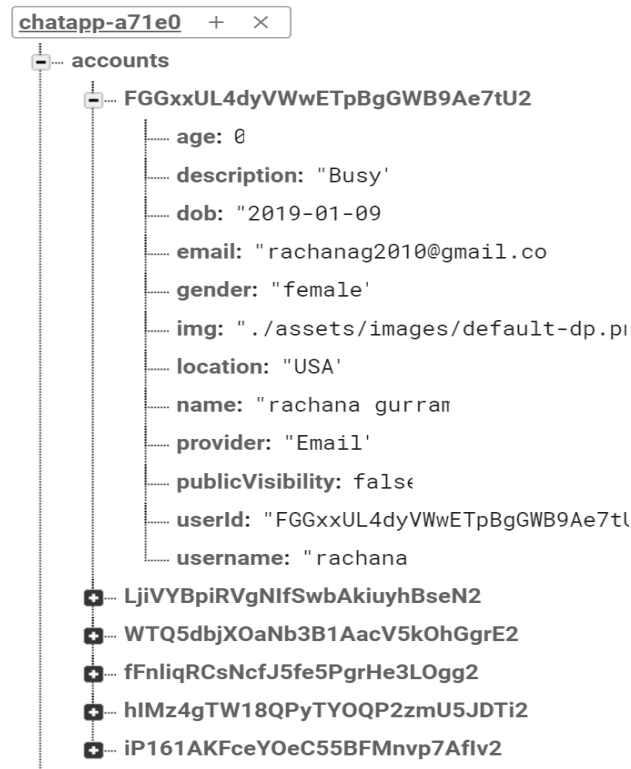


Figure 8: Account Node Structure

### 4.2.1.3. Conversation Nodes

Now that account node exists, like the account node, conversation node is also created under the root of the database. Every conversation has its unique ID. Information related to the conversation such as the date conversation started, messages, users, etc. will be stored under the conversation thread.

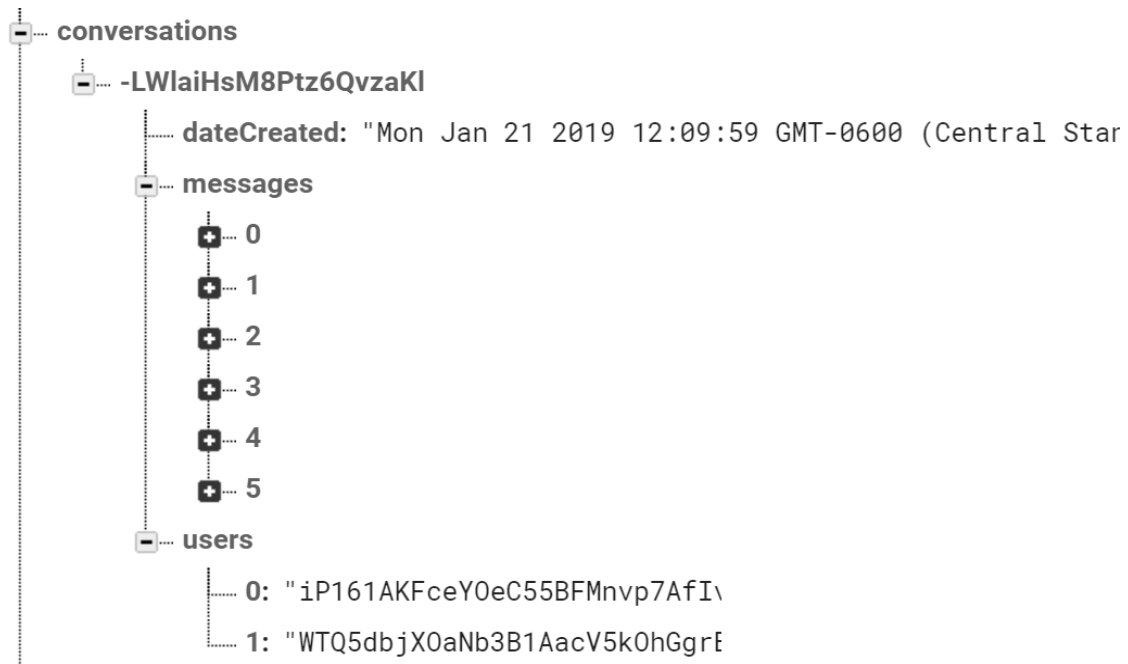


Figure 9: Conversation Node Structure

The conversations node has the child nodes each which holds the complete data of the corresponding conversation. The key nodes are date created, messages, and users involved in the conversation.

#### 4.2.1.4. User's Node

The user's node exists under the conversation node. Users node has the list of users who belong to a part of that conversation. This can also be used to send notifications when a new message has been sent to the group.

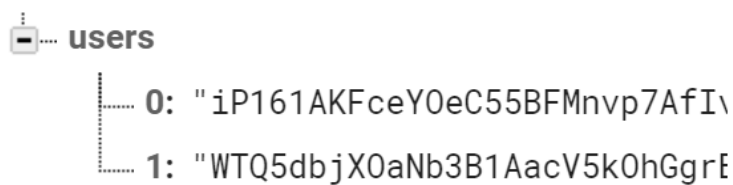


Figure 10: Users Node Structure

#### 4.2.1.5. Messages Node

Messages node is where we'll be hosting all the messages in that conversation. It'll be a list of messages, with its own index sorted by time. Each node in the message has information related to that message like avatar, date, message, sender, type, etc. For future purposes, we can include additional data like attachments, documents, media, etc.

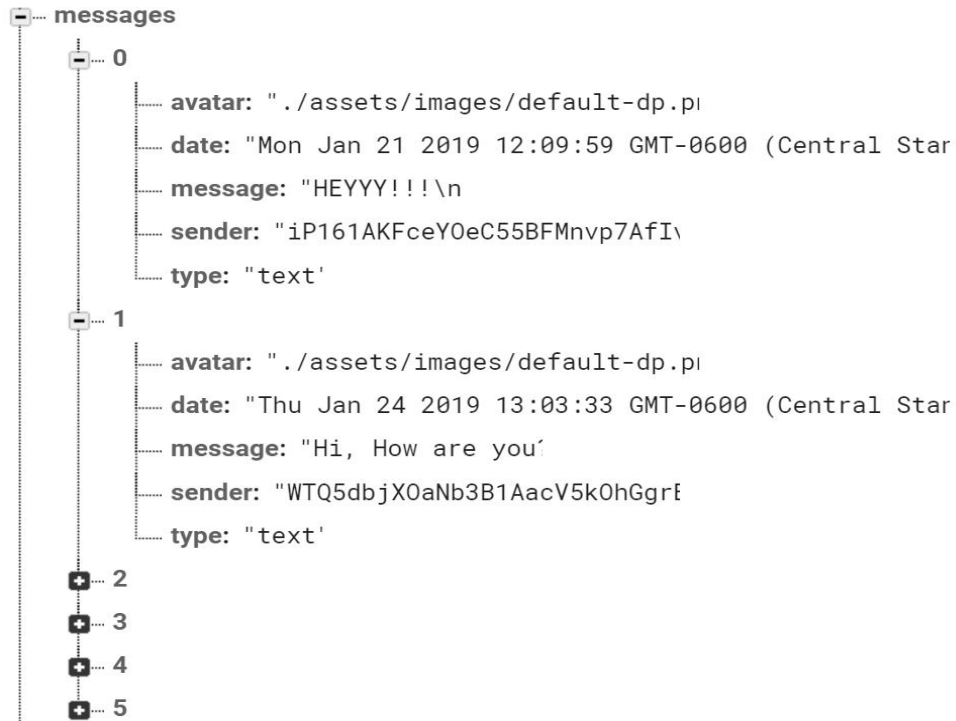


Figure 11: Messages Node Structure

## 5. USE CASES

Starting from the basics, one-to-one chat allows any user to chat with each other in real time without leaving the platform. With this social networking application, users can chat, negotiate, and seek information irrespective of their geographical boundaries. Here are a few use case scenarios that are identified while developing the application.

### 5.1. Authentication

A new user can register into the app via the interface. It requires a username of at least 5 characters long. Username and password cannot be identical. Initially, “Register Now” button is disabled until all prerequisites are satisfied. Once all the conditions are met, the “Register Now” button becomes enabled. User can create only one account per email.

The image displays two side-by-side screenshots of a mobile application's registration interface. Both screens feature a blue chat icon at the top center and the text "CREATE ACCOUNT" below it. The left screenshot shows the registration form with empty input fields for "Fullname", "Username", "Email Address", and "Password". The "Register Now" button is disabled (grey), and a "cancel" link is visible at the bottom. The right screenshot shows the same form with the fields filled with "Robert", "Robert", "robert@gmail.com", and a masked password (represented by dots). The "Register Now" button is now enabled (black), and the "cancel" link remains at the bottom. The status bar at the top of both screens shows "LTE" and the time "12:34".

Figure 12: Registering an Account



Initially, login button is disabled. Login button is enabled only if text entered in email address matches the email format and if the password entered is at least 5 characters long. If the email and password match to the values stored in database, then user is successfully logged into the account.

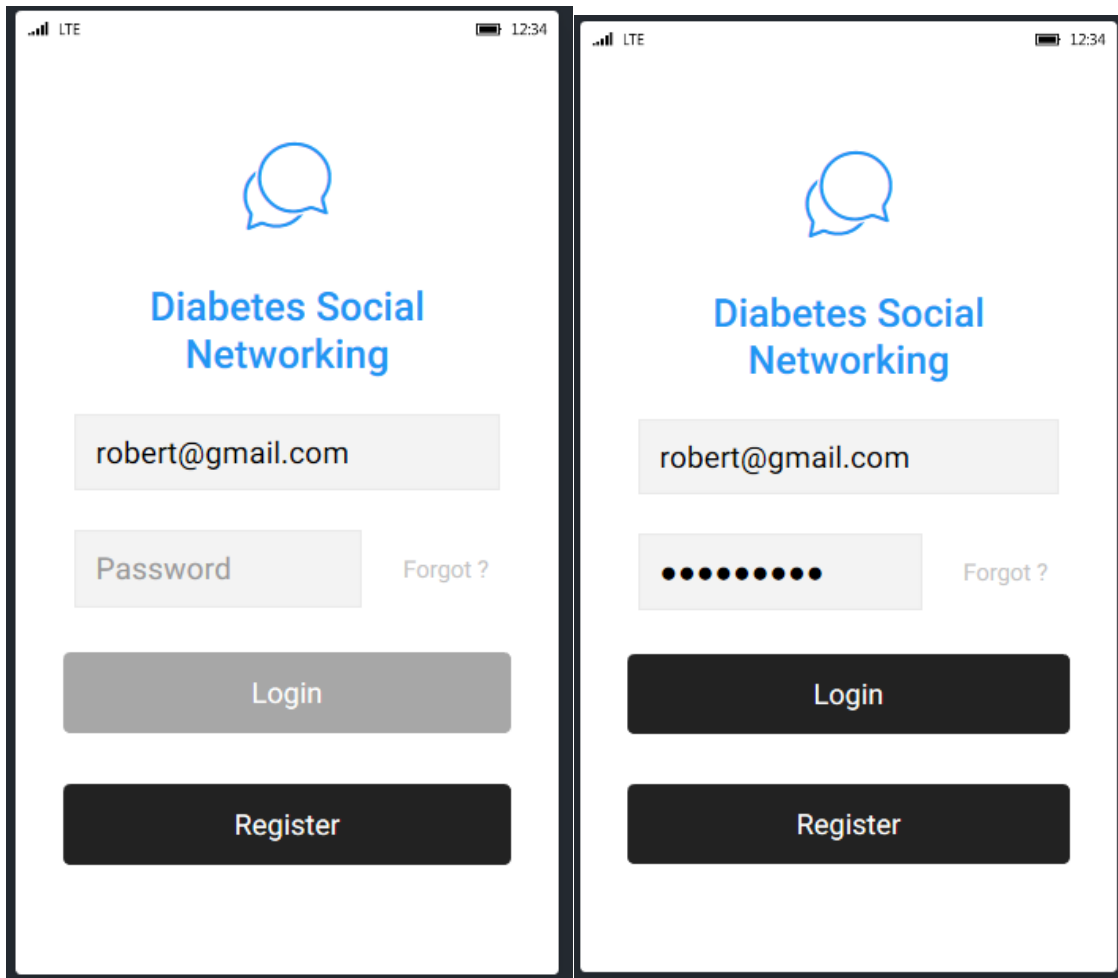


Figure 13: Login Button Before and After Entering the Password

## 5.2. Friends

User can always find new friends within the ecosystem and send or receive friend requests to or from the users. When user tries to send or cancel a friend request, he is always asked for a confirmation with a pop-up alert. If the friend request is not sent, there will always be a “add” button next to the username.

### 5.2.1. Sending Friend Requests

User can always find new friends within the ecosystem and send or receive friend requests to or from the users. When user tries to send or cancel a friend request, he is always asked for a confirmation with a pop-up alert. If the friend request is not sent, there will always be a “add” button next to the username.

If the profile visibility is set as private, then user cannot look up for other people in the ecosystem. Once the visibility is turned on and set as public, user is visible to all others within the ecosystem.

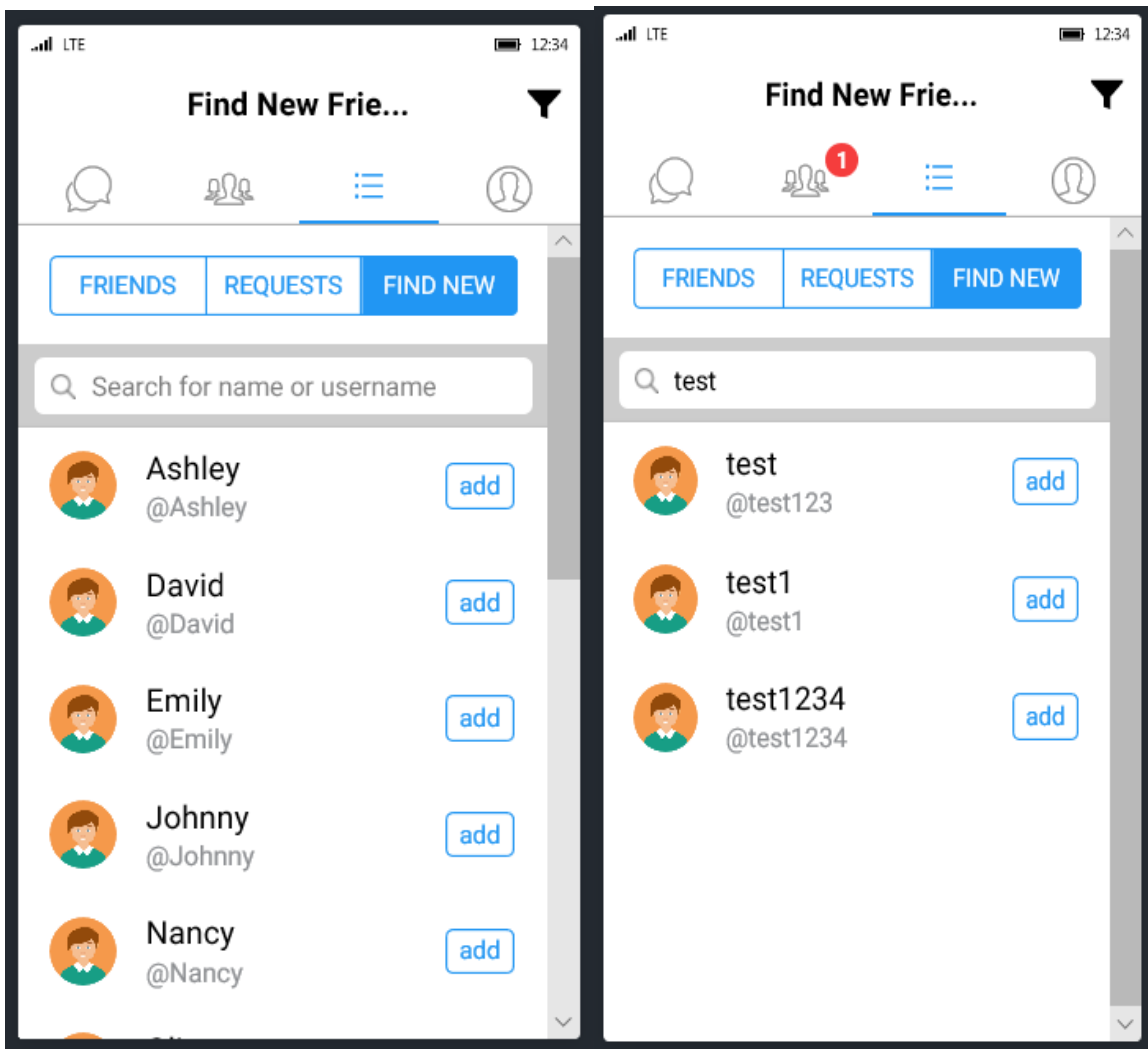


Figure 14: Searching for New Friends

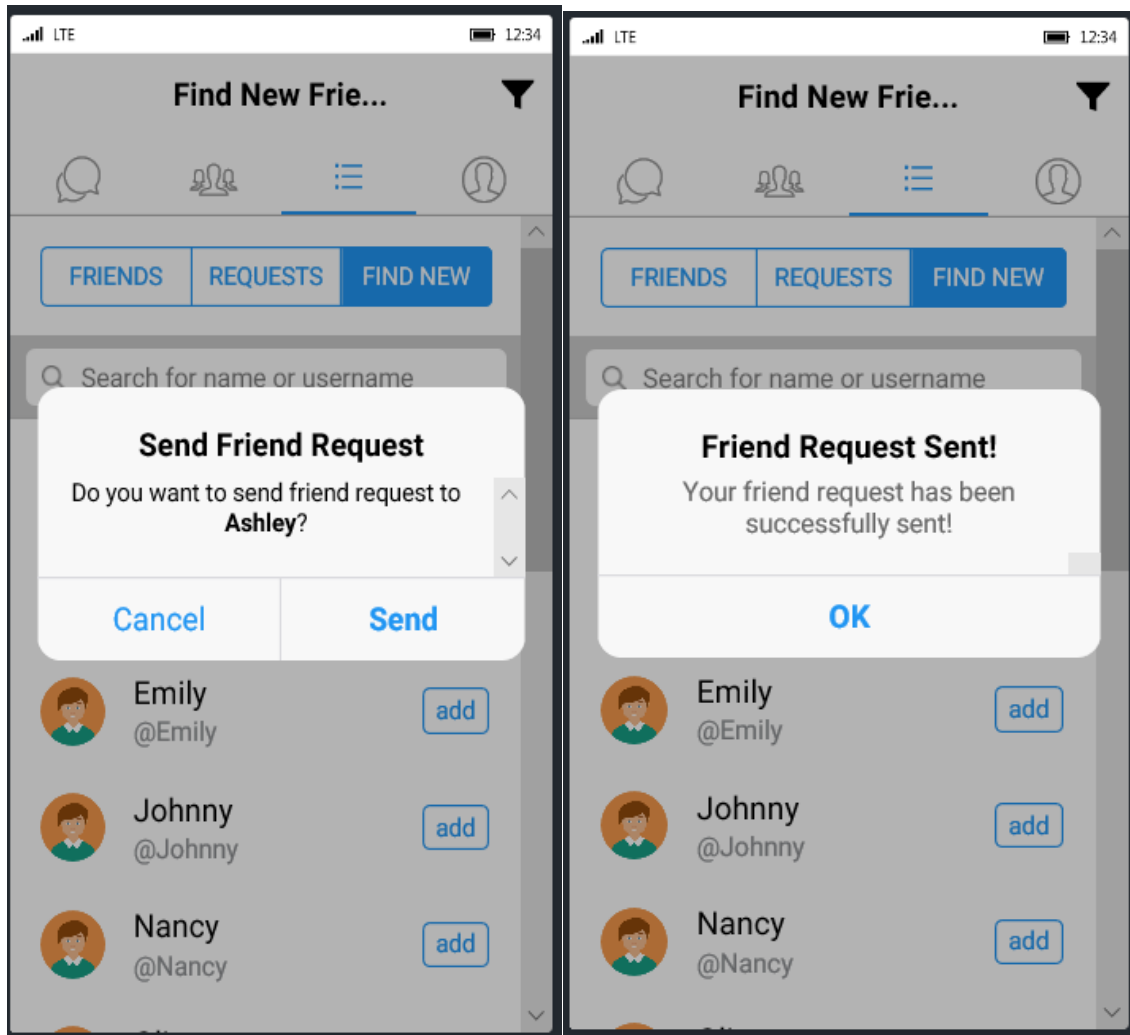


Figure 15: Sending a Friend Request

### 5.2.2. Cancelling a Pending Friend Request

If a friend request is already sent, and the other user has not accepted or declined it, then there is always a “cancel” button next to the username in the requests tab. User can cancel a previously sent friend request if he chooses to. Again, there will be a confirmation whether to delete the pending friend request or not.

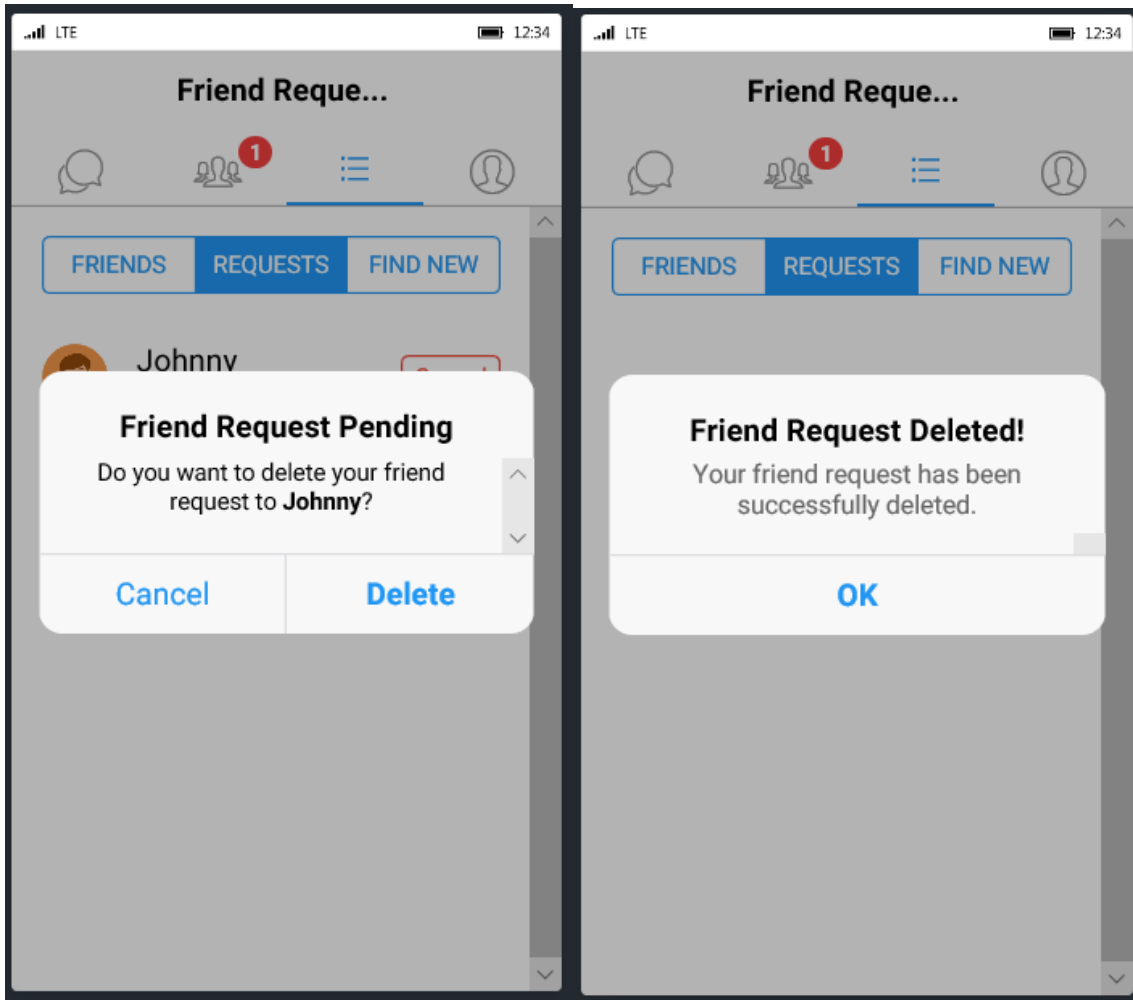


Figure 16: Deleting a Pending Friend Request

### 5.2.3. Friends List and Received Friend Requests

By navigating to the “friends” tab, user can find all his existing friends as a list. By clicking on any friend name would show his basic profile information, and “Send Message” button. IF you navigate to the “Requests” tab, user can see a list of all the friend requests received so far and which are pending.

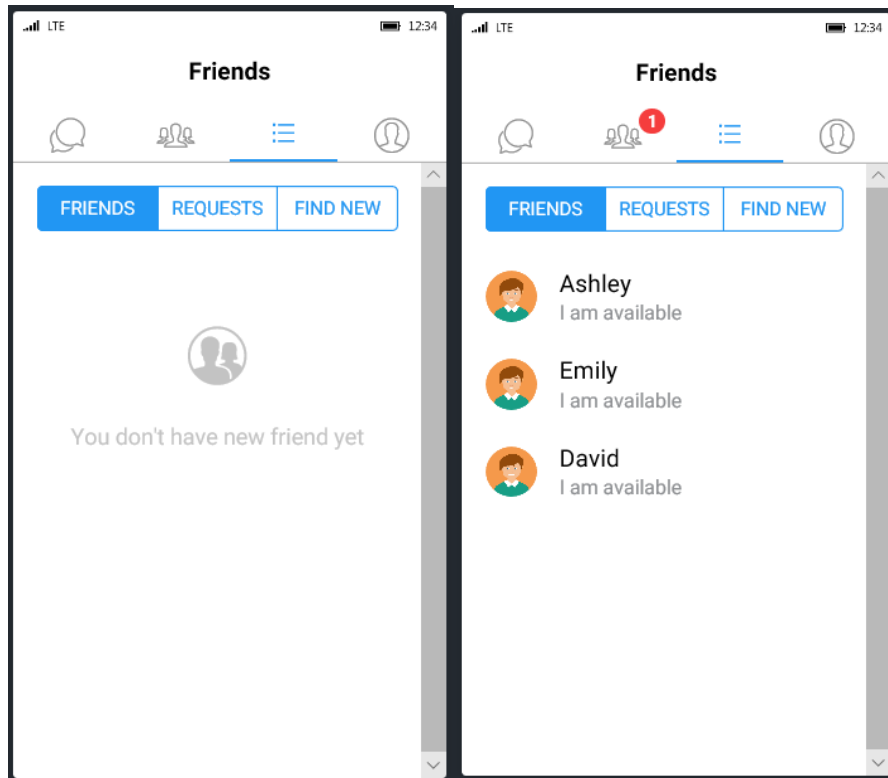


Figure 17: Friends List Before and After Adding Friends

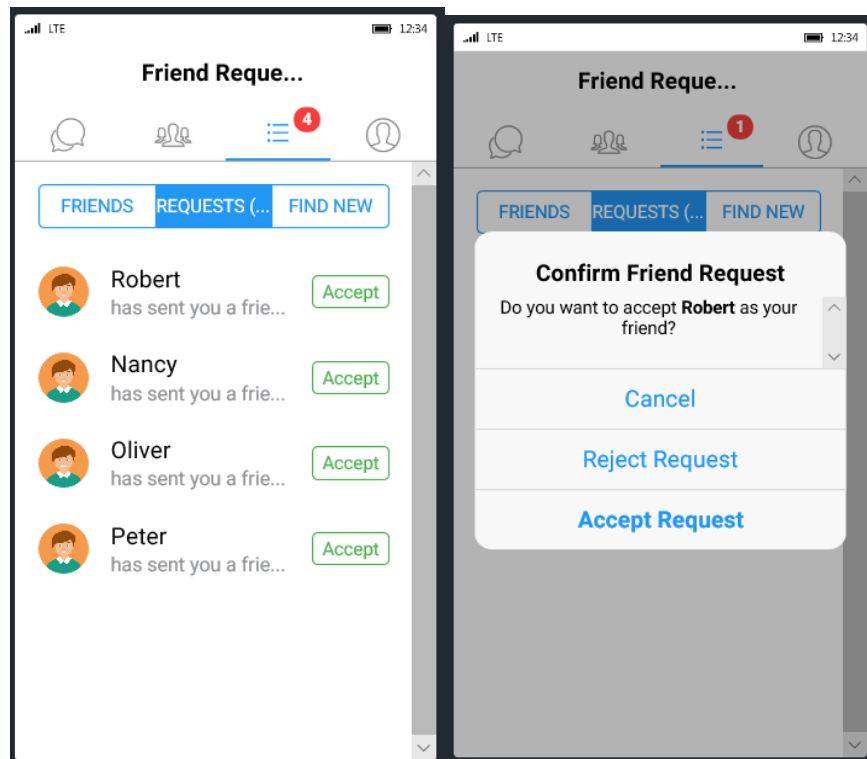


Figure 18: Friend Requests Received

### 5.3. One to One Conversations

A typical user is more likely to chat with other users very often. Therefore, chat being quicker alternative to phone calls and emails, it gives a higher probability of conversation to happen. A conversation can be viewed as an exchange of messages. For example, user1 might send a message to the user2 by displaying a dialog box. The user might reply to this message by clicking a button on the dialog box. Conversations can be long or short. If the message is unopened, you can see the count of unread messages on the badge of the icon. As you open messages, the count reduces. Also, you can perform a search on the users by typing any keyword in the search bar.

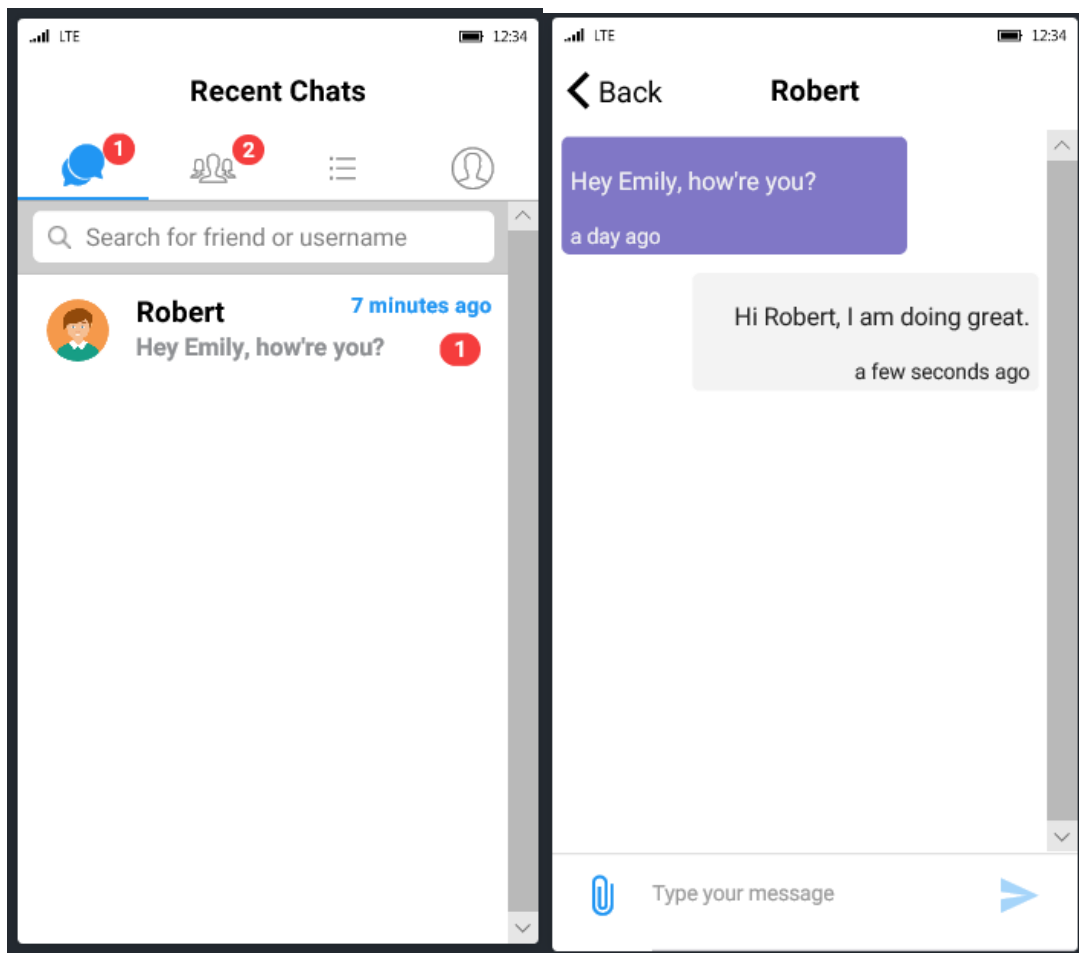


Figure 19: One-to-One Conversation

## 5.4. Group Conversations

Group conversation is something like one to many conversations. User can create a group, give it a name, description and add other users into the group. Once the group is created, the app generates a message saying the group is created and displays it on screen along with time.

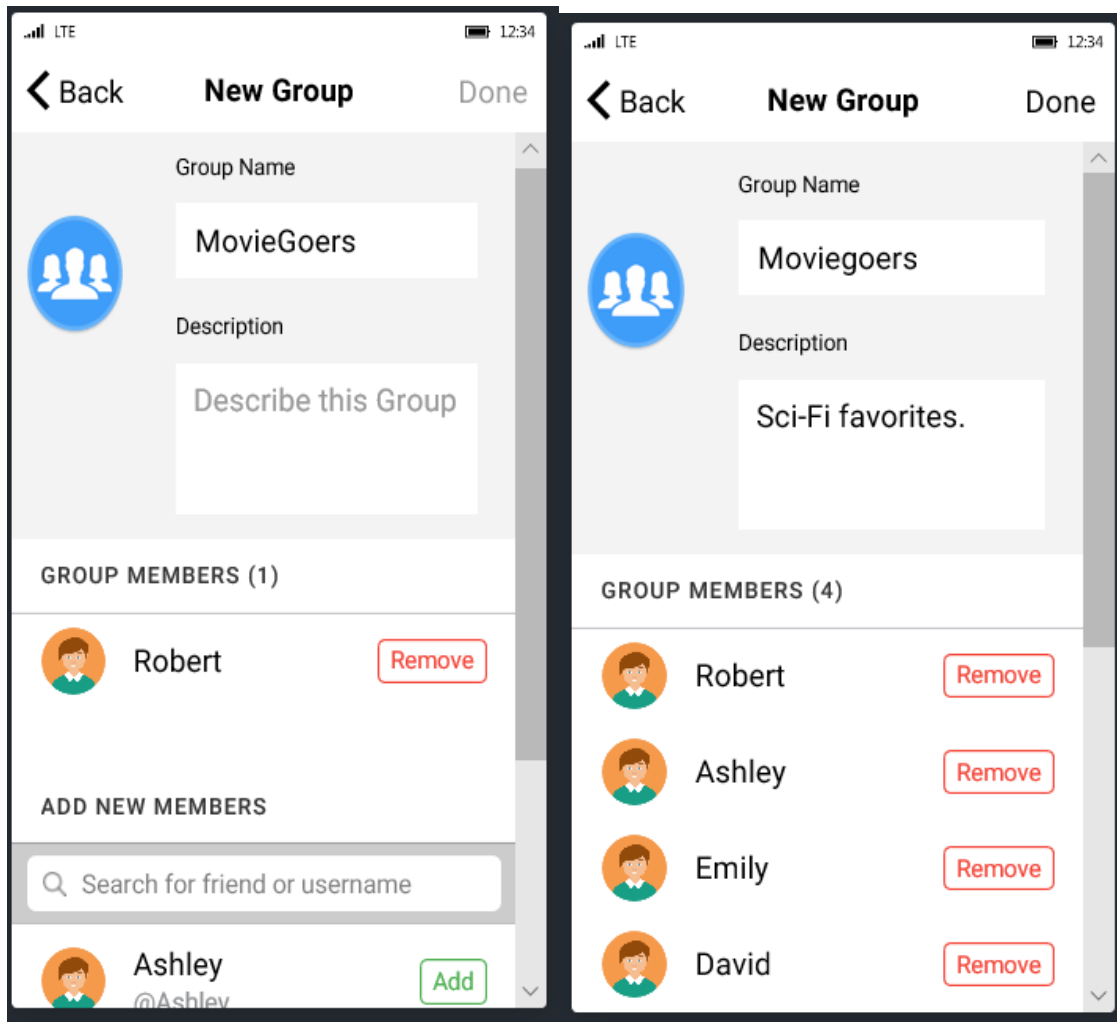


Figure 20: Creating a Group and Adding Users

As users keep sending messages in the group, it dynamically updated the number of unread messages on the group icon badge. Every message in a group conversation has a timestamp, just like the one to one conversation messages have.

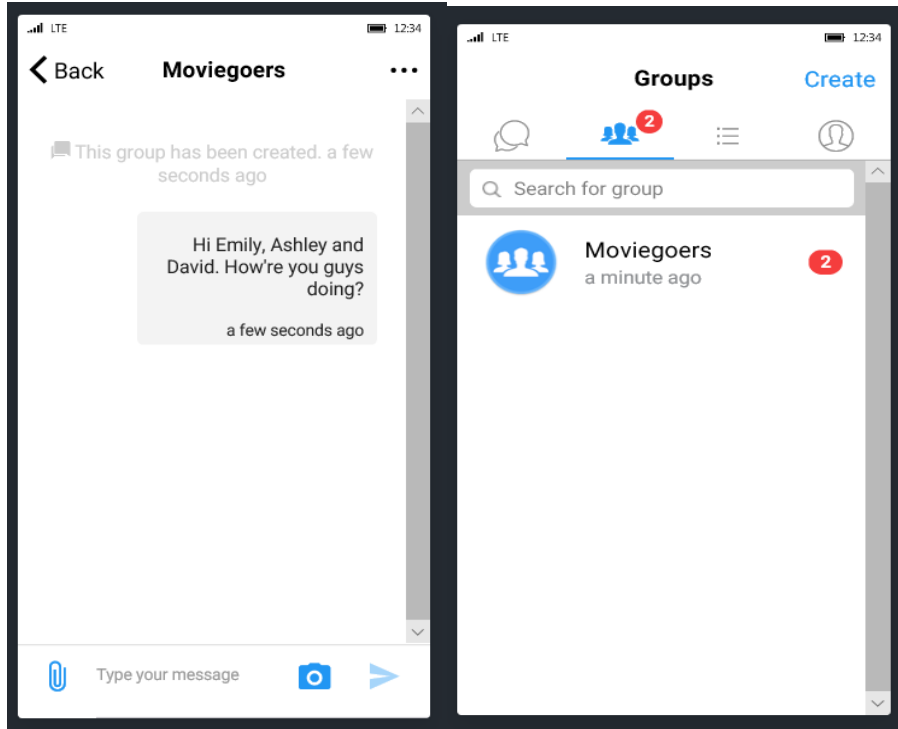


Figure 21: Messages and Notifications in a Group Chat

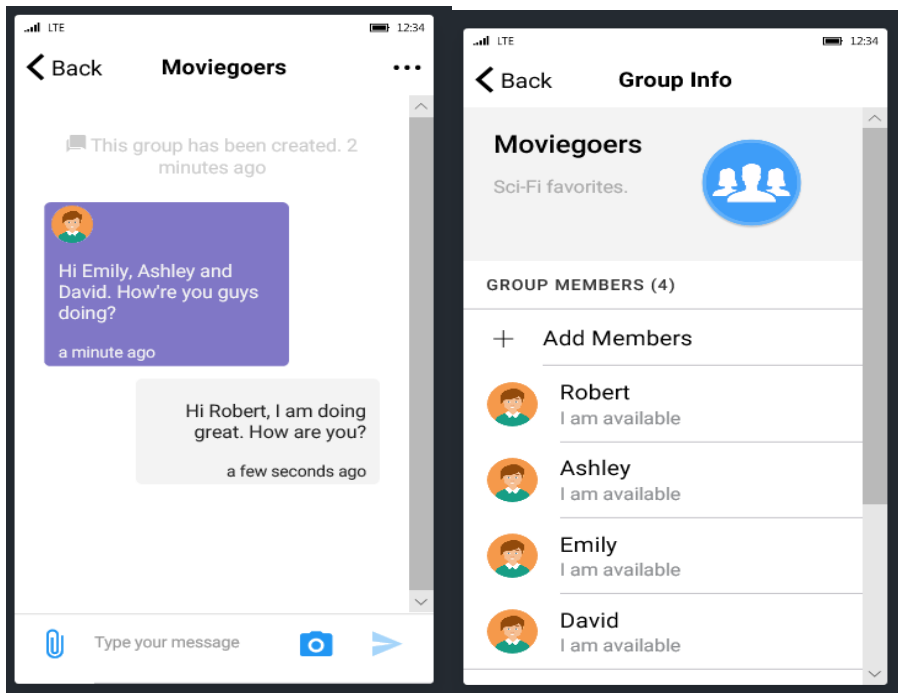


Figure 22: Basic Group Information



## 5.5. Profile

Profile is the section where all the basic information user has provided during registration is maintained. User can anytime navigate to profile section, update the details such as full name, username, bio, email, etc. and save the changes made to the profile information. There is an update password option within the app. User can enter the current password and create a new password through the UI. The basic user profile can also include information about the user's medical condition and which clinic or doctor they are consulting.

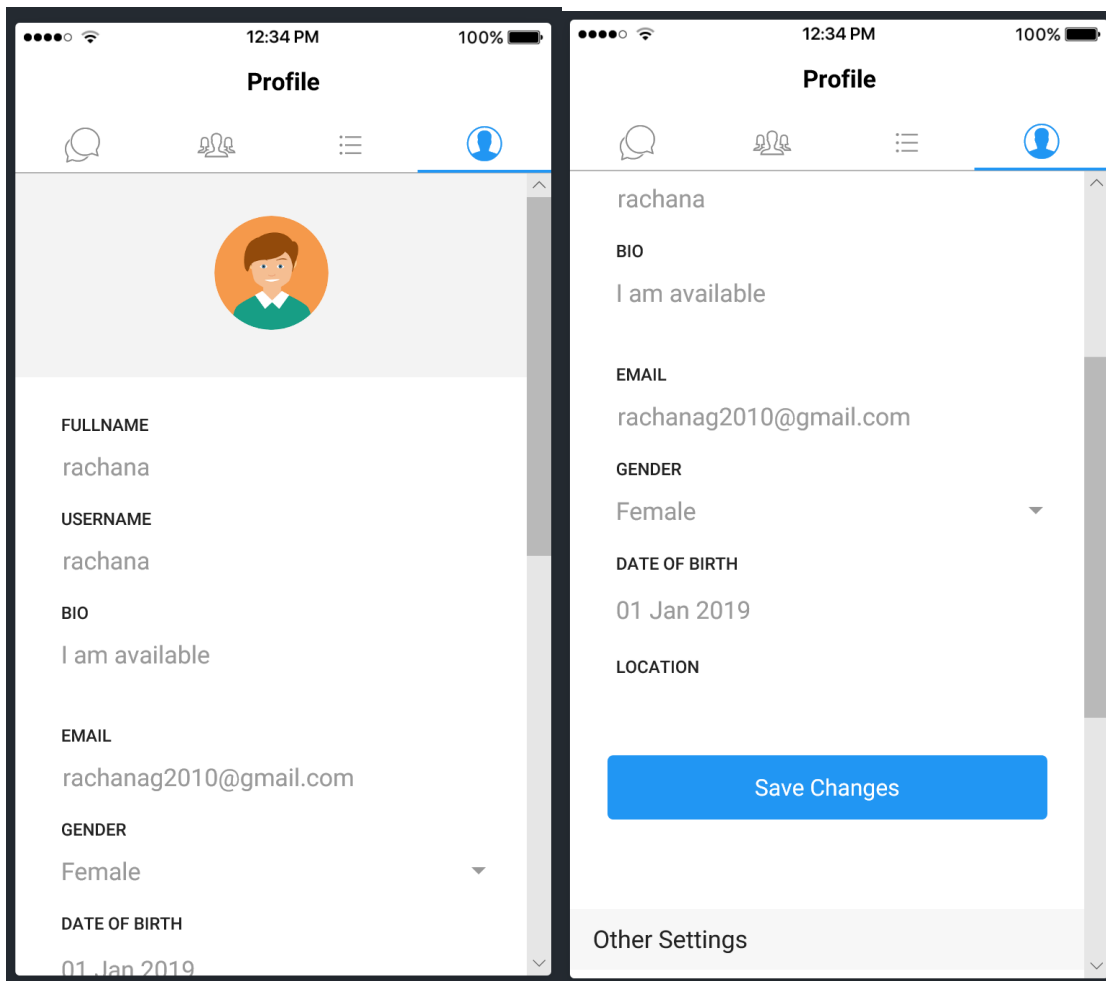


Figure 23: Basic Profile Information

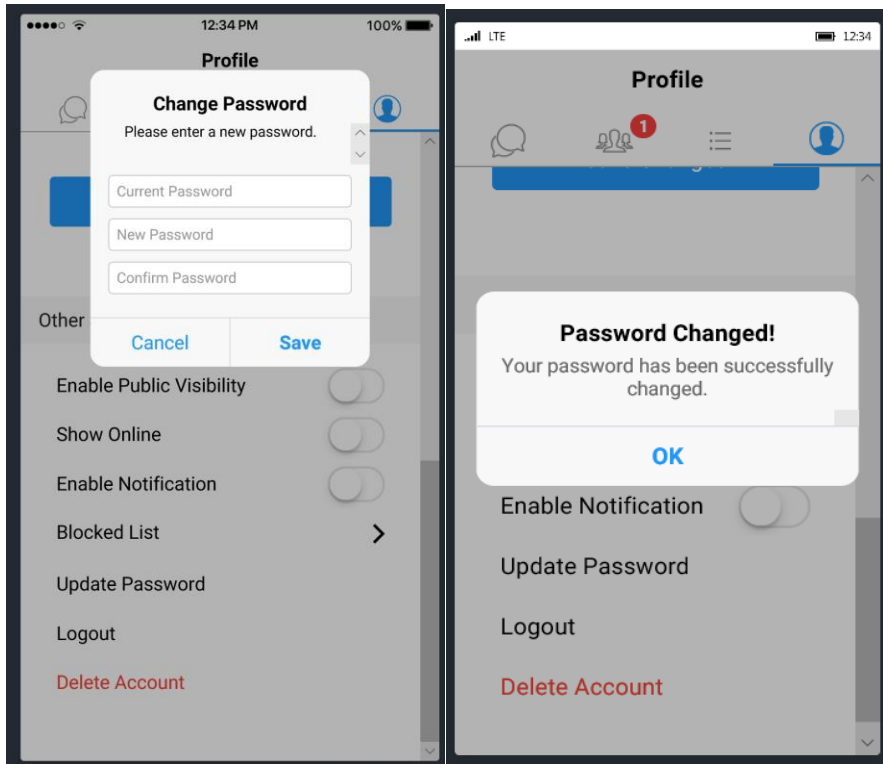


Figure 24: Changing Password

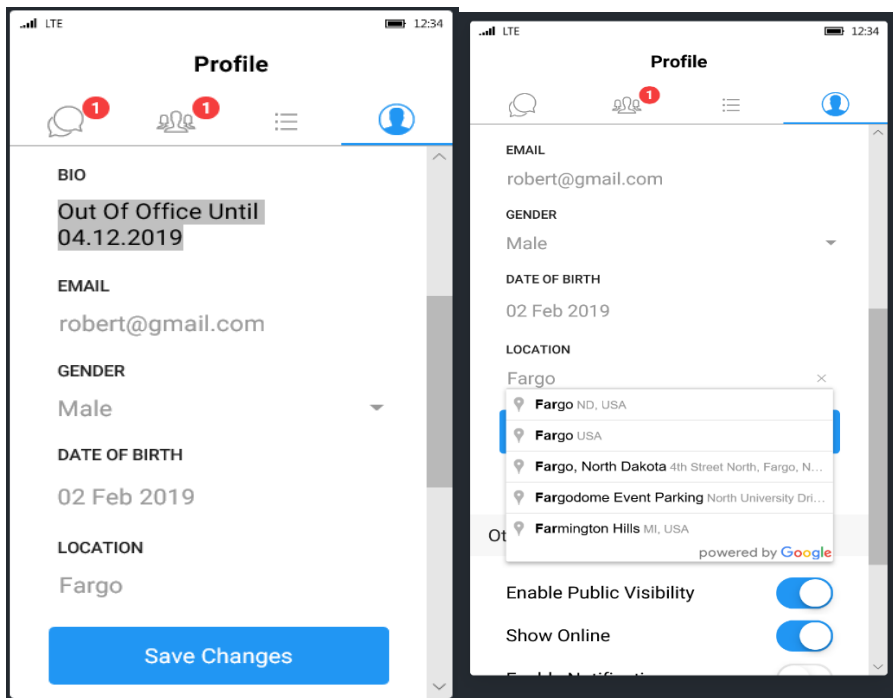


Figure 25: Updating Profile Info

### 5.5.1. Settings

There might be settings other than the basic profile information which user want to set the preferences. For example, controlling the visibility, if the user wants to set his profile information visible or invisible to other people, or show the profile status as online or offline, enable or disable push notifications, etc. There can be other privacy settings to help users protect their privacy. Ex, users can enable their information visible only to a group of people and hide it from others.

In this application, user can also update the password via the UI through other setting in profile if he chooses to. User can logout from the application through other settings. User is provided with a confirmation if he wants to logout or not. There is also a “delete account” feature, which lets the user to permanently delete his account from the database. User must register again if he wants to continue using the application.

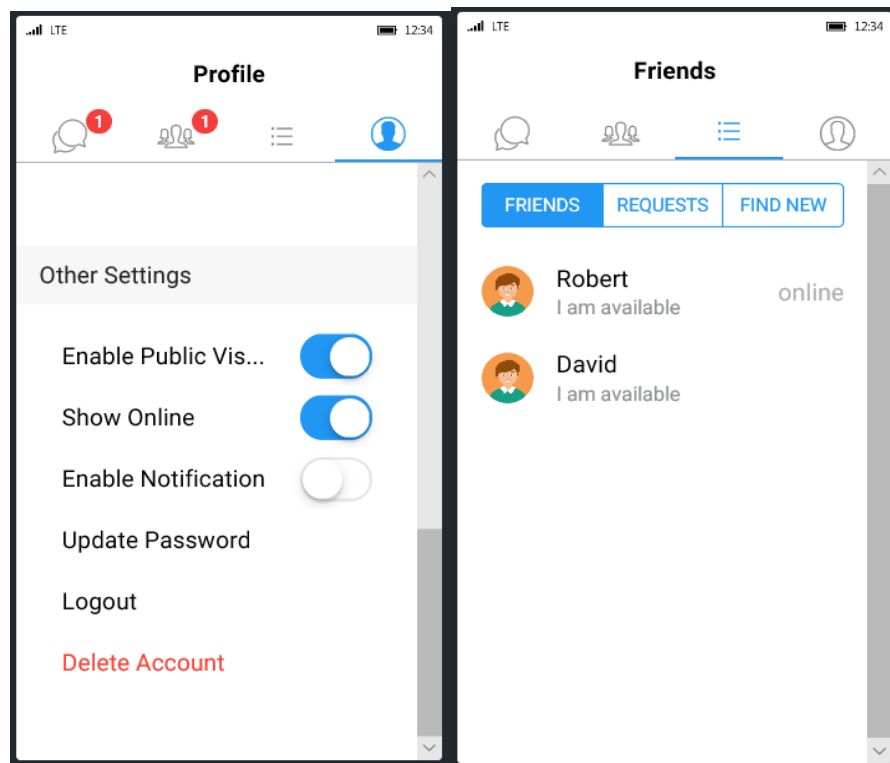


Figure 26: Other Settings

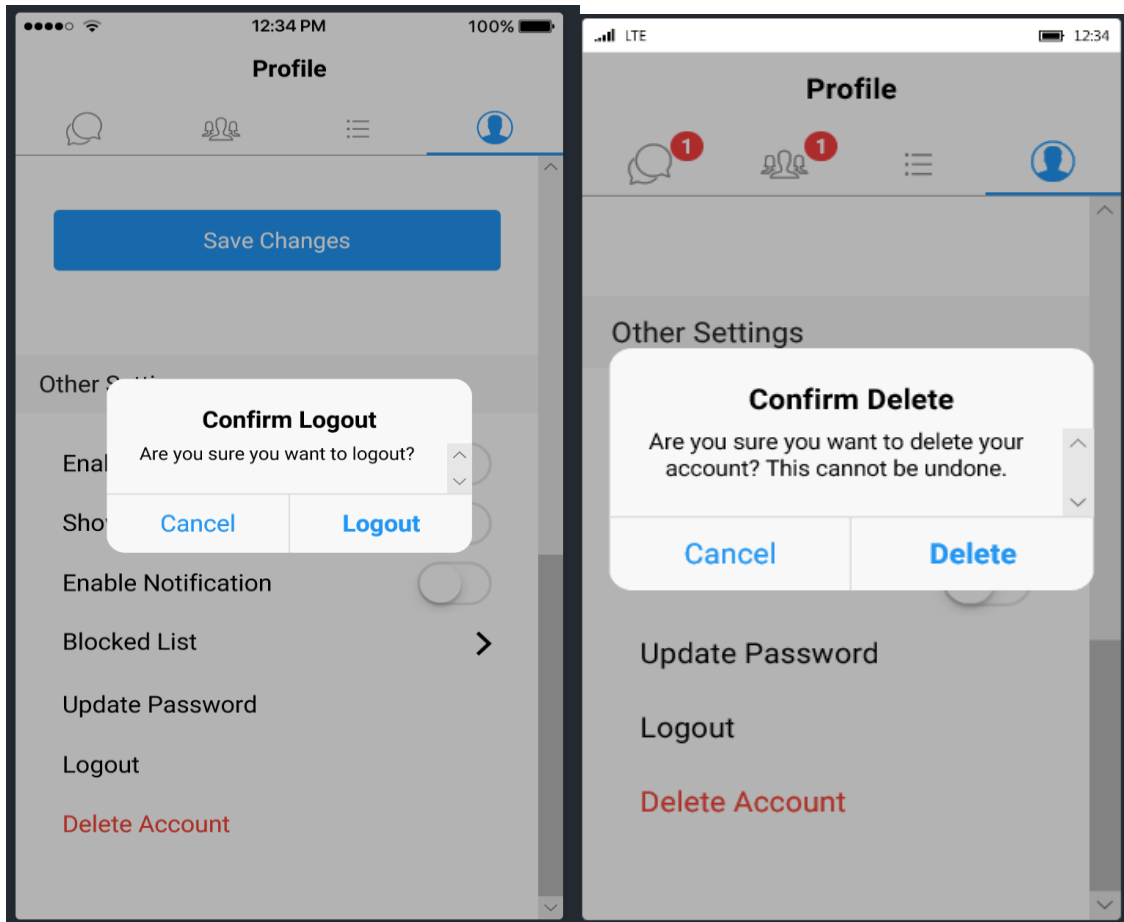


Figure 27: Logout or Delete the Account

## 5.6. Templates

There might be some user-management actions, such as updating a user's email address and resetting a user's password, which results in sending emails to the user. The emails received consists of links which recipients need to open to finish or cancel the user-management action.

In the Firebase account, we can create templates for sending an email for verifying the account, resetting the password, email address change, etc. This way everything is system-generated and there is no waste of time in handling emails or providing services/support to the user. Below are examples of templates for “Verifying Email Address” and “Password Reset” respectively.

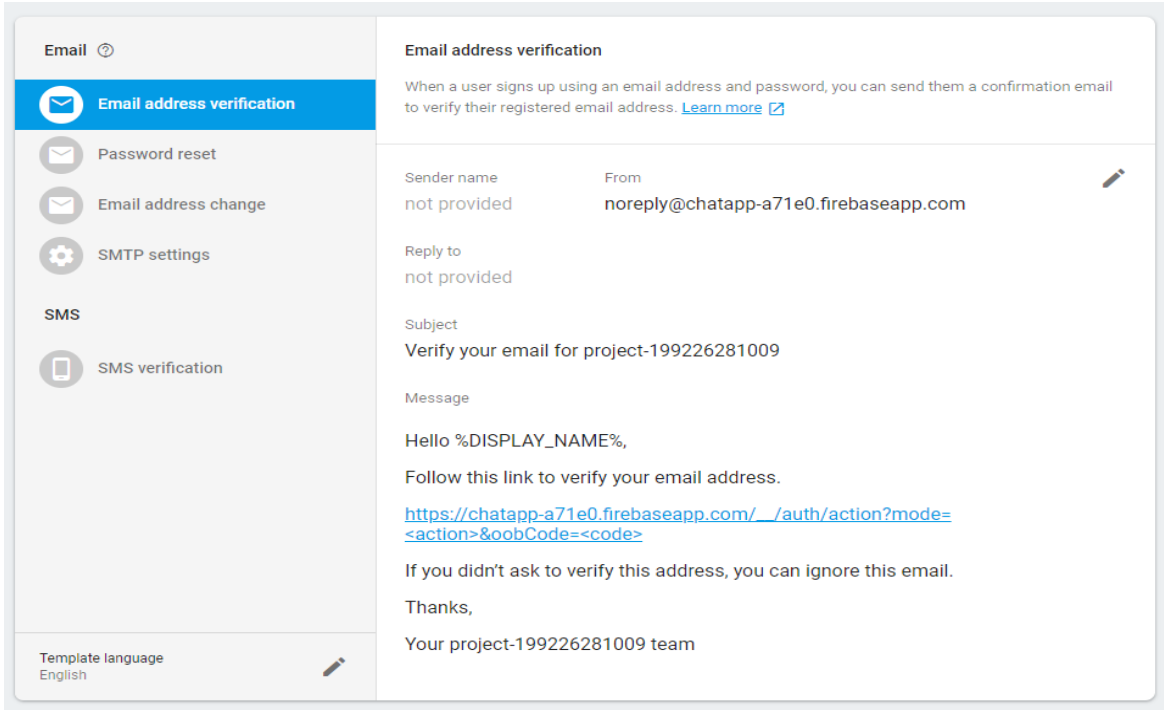


Figure 28: Template for Email Verification

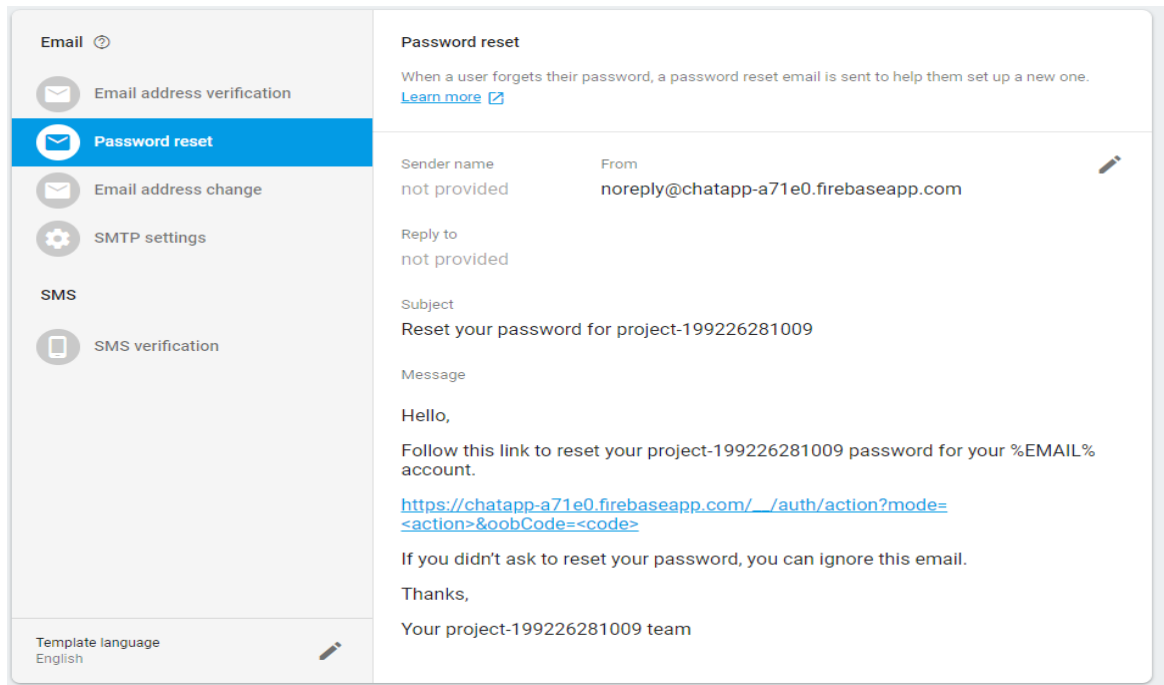


Figure 29: Template for Password Reset

## **6. CONCLUSION AND FUTURE WORK**

### **6.1. Conclusion**

Ultimately, this social networking app assists American Indian diabetic patients in their healthcare management and get peer support from the community. It serves as a good tool of communication between the patients and the providers and helps in maintaining a healthy lifestyle. Thus, mobile health gives individuals with chronic medical conditions an opportunity to be involved and engaged actively in their health care. Leveraging the social networking app, patients have the option to find other individuals with similar symptomology and medical condition history to confirm diagnoses, look for physician-recommended treatment and ask for any alternative therapies.

### **6.2. Future Work**

There is always some room for improvements or developments in any software application. As the community becomes increasingly connected through mobile device wirelessly and share information or private matters such as their health conditions via social networks, there might be new challenges and opportunities to share the information in a safe and timely manner. Since the American Indian participants might not be too much aware of Information Technology and may have fewer experiences with software applications to understand, analyze and set privacy rules. Therefore, applying semantics-based social matching as an intellectual framework to connect the AI patients can be considered as an advancement to this application.

## 7. REFERENCES

- [1] Siwicki, B. 2016. Are social networks the future of chronic disease care? *HealthcareITNews*.  
<https://www.healthcareitnews.com/news/are-social-networks-future-chronic-disease-care>.
- [2] John Snow, Inc. 2019. Chronic Disease Management. *John Snow, Inc. and JSI Research & Training Institute, Inc.*  
<https://www.jsi.com/JSIInternet/USHealth/techexpertise/display.cfm?tid=1000&id=90>
- [3] John, G. 2008. Analyzing the benefits of PatientsLikeMe social network site. *TheHealthCareBlog*.  
<http://thehealthcareblog.com/blog/2008/05/30/analyzing-the-benefits-of-patientslikeme-social-network-site>
- [4] Scanlon, L. 2019. PatientsLikeMe Survey Shows Vast Majority of People With Health Conditions Are Willing To Share Their Health Data. *PatientsLikeMe*.  
<https://news.patientslikeme.com/press-release/patientslikeme-survey-shows-vast-majority-people-health-conditions-are-willing-share-t>
- [5] Volpp, K. and Motha, N. 2018. Patient Engagement Survey: Social Networks to Improve Patient Health. *NEJM Catalyst*.  
<https://catalyst.nejm.org/survey-social-networks-patient-health>
- [6] Newberry, C. 2019. How to Use Social Media in Healthcare: A Guide for Health Professionals. *Hootsuite*.  
<https://blog.hootsuite.com/social-media-health-care>
- [7] Firebase. 2019. Structure Your Database. *Firebase*.  
<https://firebase.google.com/docs/database/ios/structure-data>

- [8] Raj, V. 2017. How to structure Firebase database for a scalable chat app. *Skript*.  
<https://www.skript.com/svr/how-to-structure-firebase-database-for-a-scalable-chat-app>
- [9] Simons, E. 2019. Mastering the Ionic Framework: Learn to Build & Deploy Native Speed HTML5 Based Apps. *Thinkster*.  
<https://thinkster.io/ionic-framework-tutorial>