**Pure Connect: A basic social media app with**

**news updates for IOS**

**Submitted By**

**Israt Tasnim Esha**

Roll: 1907090

**Jobair Al Nahian**

Roll: 1907071

**Hasib Hasan**

Roll: 1907089

**Sonjoy Roy**

Roll: 1907073

**Md. Safwan Zaher Asif**

Roll: 1907066



**Submitted To:**

Most. Kaniz Fatema Isha

Lecturer

Dept. of Computer Science and Engineering

Khulna University of Engineering & Technology

Argha Chandra Dhar

Lecturer

Dept. of Computer Science and Engineering

Khulna University of Engineering & Technology

1. Introduction

* 1. Background

We're making an app to make keeping up with news and social communication simpler. Right now, there's a lot of information online, and it's hard for people to find the news they really care about. Our app wants to fix that.

The big issue we're tackling is that people get kinda overwhelmed by all the information out there. Our app plans to make things easy by putting social media and news together in one place.

Another good part is that our app encourages people to know more about what's going on. By making news easy to get and trustworthy, we hope users can make better choices and be a part of important talks. So, our project helps individuals and builds a community where everyone's in the know.

In short, our idea deals with too much info by creating a friendly app that puts social media and news together. We want to make staying in the loop simple, fun, and something everyone can do together on our app.

* 1. Objectives and Goals:
* Easy to Use: Make the app simple so people can use it easily. When they sign up, they should go straight to the timeline without any confusion.
* Share Moments Easily: Let users share their pictures and thoughts without any hassle. Making posts should be quick and straightforward. Users can search and delete post anytime.
* Connect with Others: Help users feel part of a community by letting them like and comment on each other's posts. Make it a fun place for socializing.
* Quick Access to Timeline: After signing in, users should immediately see what's happening in the timeline. No extra steps, just straight to the main social hub.
* Stay Updated with News: Keep users informed about the latest news by adding a special section with real-time updates. Blend social life with staying in the know..
* See Post Popularity: Show users how many likes and comments their posts get. It's fun to know when people enjoy what they share.

**2. Methodology :**

**2.1 SplashViewController.swift**

* **Responsibility:**
  + Displays a splash screen while the application is loading.
* **Methodology:**
  + Confirm that it serves as the initial view controller in the storyboard.
  + Ensure it displays a brand logo or any relevant information during the app loading.
  + Verify that it transitions to the appropriate view controller (SignInViewController) after loading.

**2.2 SignInViewController.swift**

* **Responsibility:**
  + Manages user authentication and allows users to sign in.
* **Methodology:**
  + Confirm that it includes text fields for entering the user's phone number and password.
  + Implement secure password handling, such as password hashing.
  + Verify that users can sign in using Firebase Authentication.
  + Ensure proper navigation to the HomeTabBarController upon successful sign-in.
  + Provide a mechanism for users to reset their passwords.

**2.3 SignUpViewController.swift**

* **Responsibility:**
  + Allows users to create a new account.
* **Methodology:**
  + Confirm that it includes text fields for entering necessary registration details (phone number, name, etc.).
  + Implement Firebase Authentication for creating a new user account.
  + Ensure the uniqueness of user accounts (e.g., checking if a phone number is already in use).
  + Verify that users are directed to the HomeTabBarController after successfully creating an account.

**2.4 HomeTabBarController.swift**

* **Responsibility:**
  + Manages the tabs of the application.
* **Methodology:**
  + Confirm that each tab corresponds to a specific view controller.
  + Ensure proper initialization and setup of each view controller.
  + Verify that any common functionality shared among tabs is appropriately managed.

**2.5 TimelineViewController.swift**

* **Responsibility:**
  + Displays a timeline of posts.
* **Methodology:**
  + Confirm that the timeline retrieves posts from Firebase Realtime Database.
  + Check that posts are displayed in reverse chronological order.
  + Verify the functionality for liking and unliking posts.
  + Confirm that the number of likes is displayed and that users can comment on posts.
  + User can delete the post from timeline.
  + Can search post by typing in the user name.

**2.6 CreatePostViewController.swift**

* **Responsibility:**
  + Allows users to create and upload new posts.
* **Methodology:**
  + Confirm that users can select images from the photo library.
  + Verify that selected images are successfully uploaded to Firebase Storage.
  + Ensure new posts are created in the Firebase Realtime Database with the associated details.

**2.7 AccountViewController.swift**

* **Responsibility:**
  + Manages user account information and provides options for user interaction.
* **Methodology:**
  + Confirm that user details (name, email, address, etc.) are displayed and editable.
  + Ensure that users can reset their passwords securely.
  + Verify that the user's profile image can be uploaded to and retrieved from Firebase Storage.
  + Confirm that user information is updated in the Firebase Realtime Database upon editing.

**2.8 CommentViewController.swift**

* **Responsibility:**
  + Manages the display and addition of comments for a specific post.
* **Methodology:**
  + Confirm that comments for a specific post are retrieved from Firebase Realtime Database.
  + Verify that users can add new comments to the post.
  + Confirm that comments are displayed with the owner's profile image and other relevant information.

**2.9 APICaller.swift:**

* **Responsibility**
  + Manages API calls for fetching top news articles from a specified endpoint.
* **Methods:**
  + **getTopStories(completion:)**: Parses the JSON response using **JSONDecoder**

**2.10 APIViewController.swift:**

* **Responsibility:**
  + Displays a list of top news articles and handles user interaction related to news articles.
* **Methods:**
  + **viewDidLoad**:
    - Initializes the view controller and sets up the UI.
    - Initiates the API call to fetch top news articles.
  + **fetchTopStories**:
    - Initiates the API call using **APICaller**, updating the model data.
    - Reloads the table view with the new data.
  + Table View Delegate and Data Source Methods
    - Configures and manages the table view displaying news articles.
    - Uses **NewsTableViewCell** for cell configuration.
    - Opens a **SFSafariViewController** to display the full news article when a cell is selected.

**2.11 NewsTableViewCell.swift:**

* **Responsibility:**
  + Configures and manages the appearance of individual cells displaying news articles.
* **Methods:**
  + **configure(with:)**:
    - Configures the cell with data from a **NewsTableViewCellViewModel**.
    - Sets the title and subtitle labels and loads the image from the provided URL.
  + **prepareForReuse**:
    - Resets the cell's content when it is about to be reused.

**2.12 Classes Used to represent User data, Post Data ,Comment data and Structure is used to represent Json Data.**

class SharedData {

    static let shared = SharedData()

    var userID: String?

    var userName: String?

}

Represents the user data .Which is used to verify the user and show the user name in different fields.

class CommentData {

    var ownerText: String

    var ownerNameText: String

    var timeText: String

    var commentText: String

}

Represents the comment data. Which is sent to firebase.

class PostData {

    var idText: String

    var ownerText: String

    var ownerNameText: String

    var timeText: String

    var likesText: String

    var descriptText: String

}

Represents the post related data which is sent to firebase.

struct APIResponse: Codable {

    let articles: [Article]

}

struct Article: Codable {

    let source: Source

    let title: String

    let description: String?

    let url: String?

    let urlToImage: String?

    let publishedAt: String

}

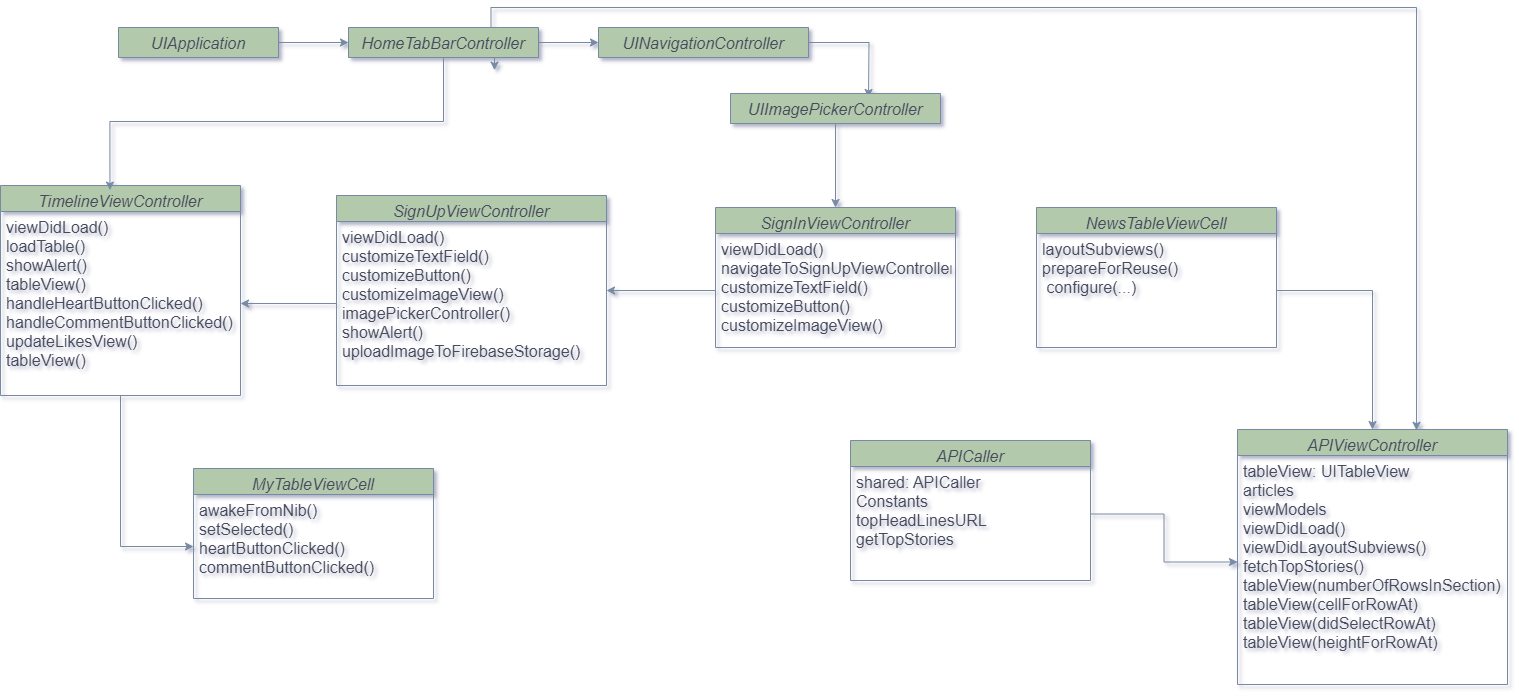
struct Source: Codable {

    let name: String

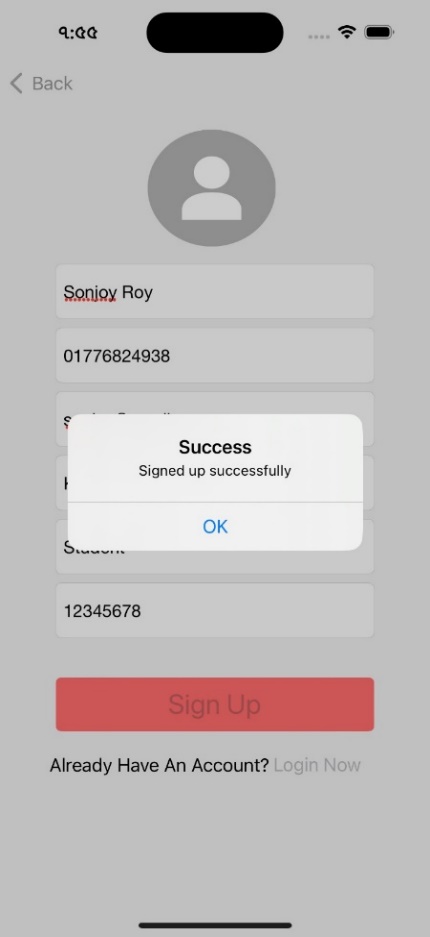
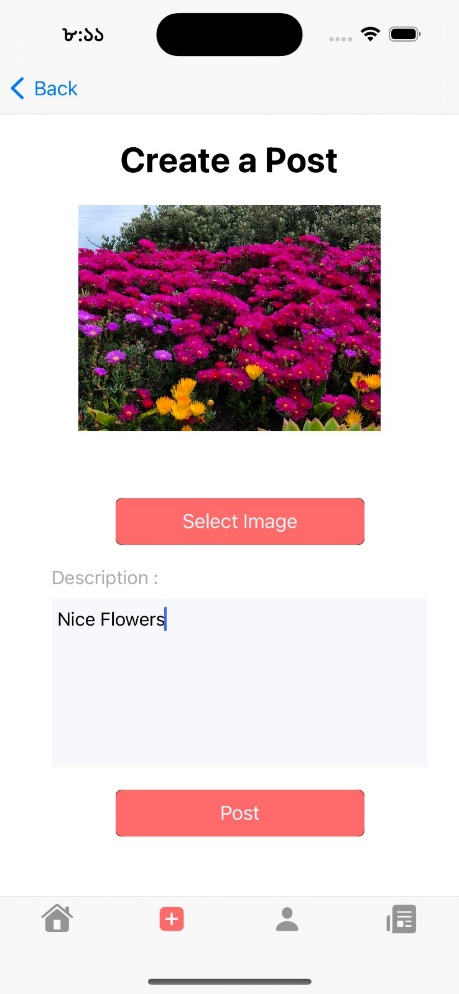
}

Represents the structure for Json parsing. Where there are arrays of Articles. In Articles there are fields like source , title, description, url etc. There is another source variable of name variable inside it.

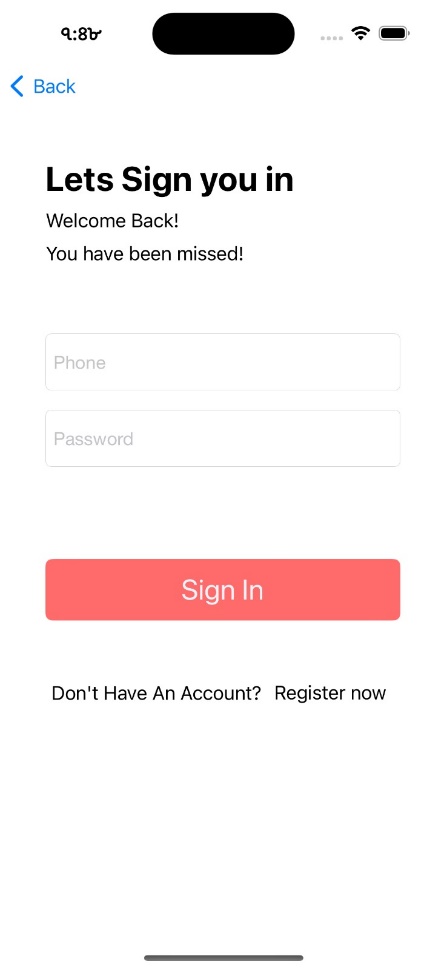
2.13 UML Class diagram of the Implementation



3. Result

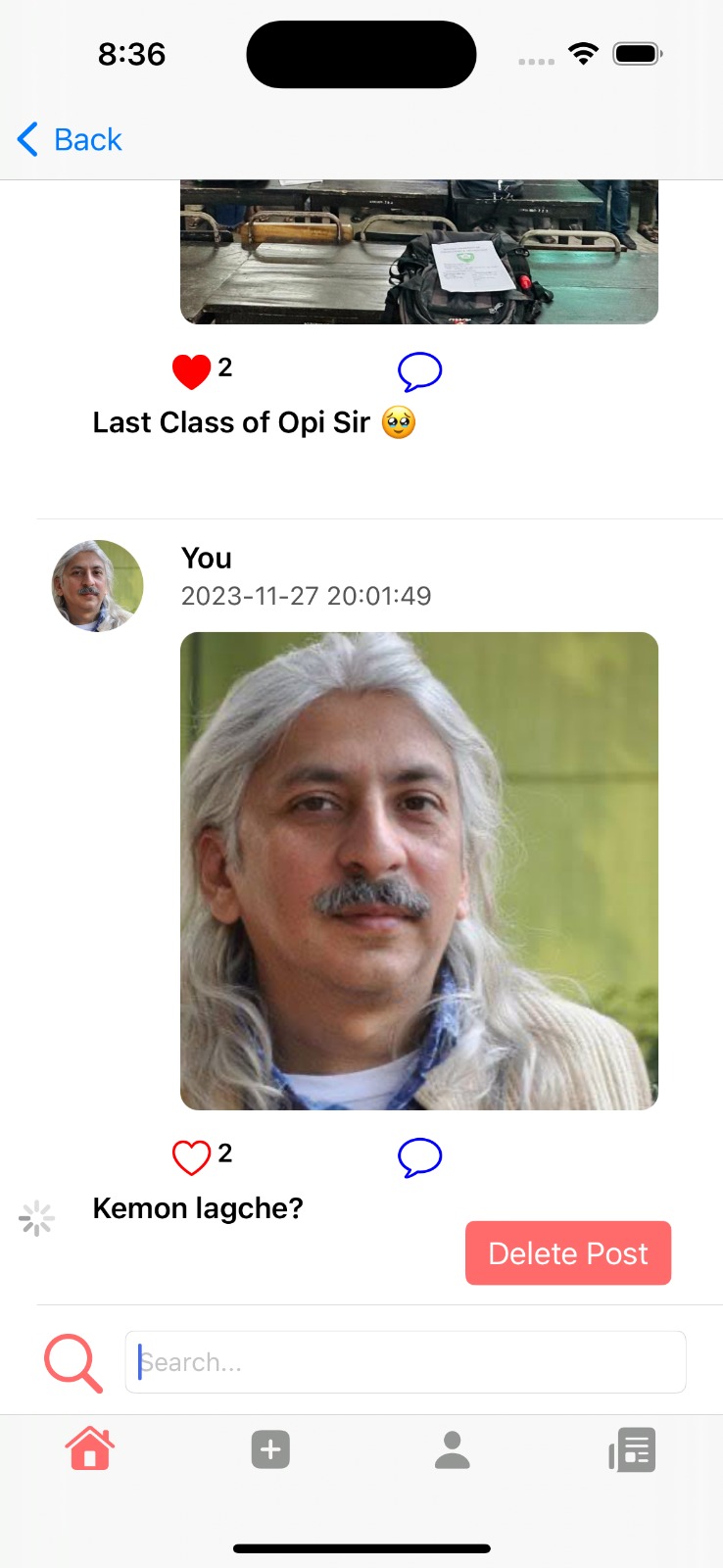
In this section we will discuss about the features of our application and will add corresponding features :  

**3.1 Splash Screen**: Behind the scenes, the app is loading and performing any necessary initialization tasks during the display of the splash screen. This ensures that when the main content of Pure Connect appears, it is ready for use.



**3.2 Sign In page:** Give your phone no. and password to log into pure connect or click register now for sign up. All the fields must be filled.



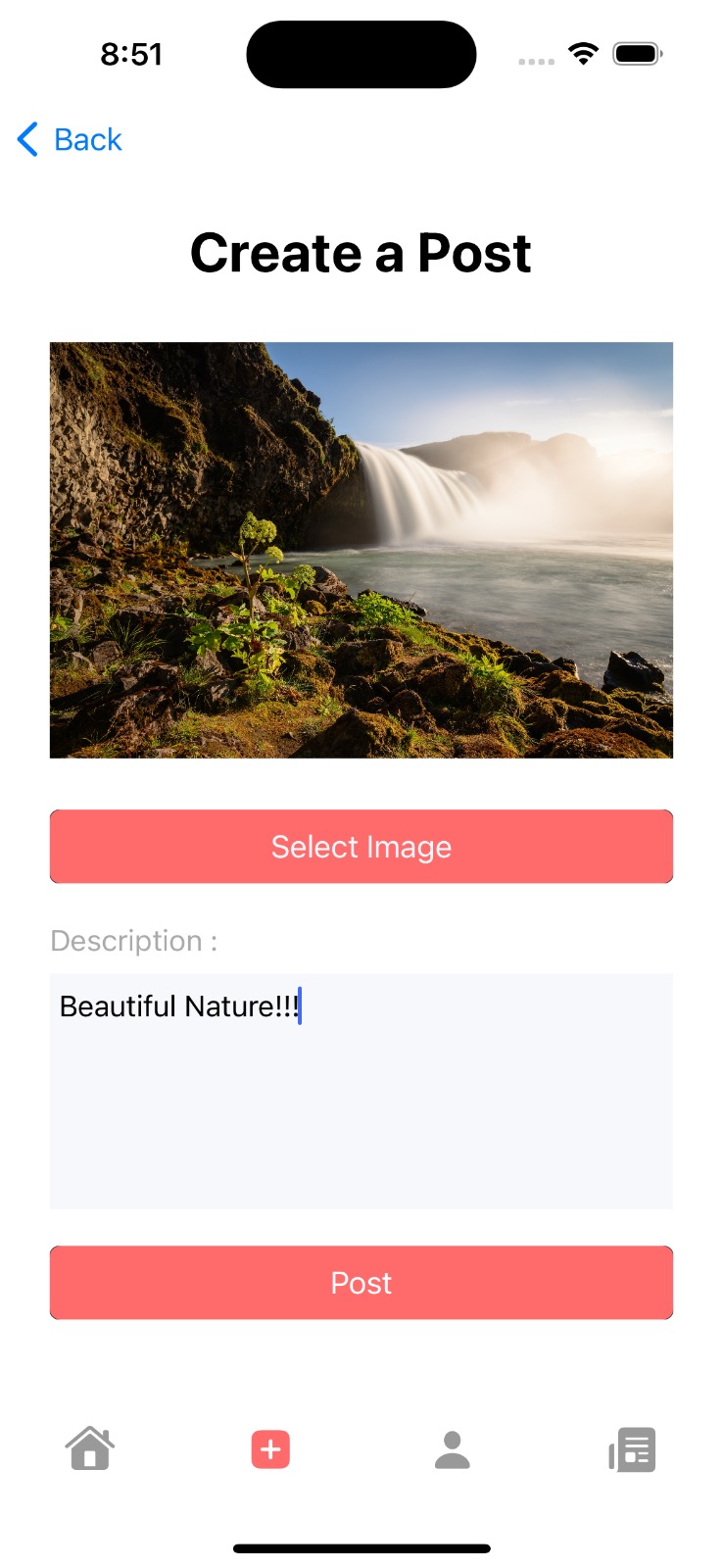


**3.4 Time Line view:** All the shared contents pop up here. Anybody can react one’s post and also comment on the post. Every post can have its description. Can see the number of reaction.

Here TableView used to represent the posts and the fields are palced in cells.  
Here Home tab bar used for navigating through pages easily making the app intuitive.

User can delete his post and also can search the user names to find his post.

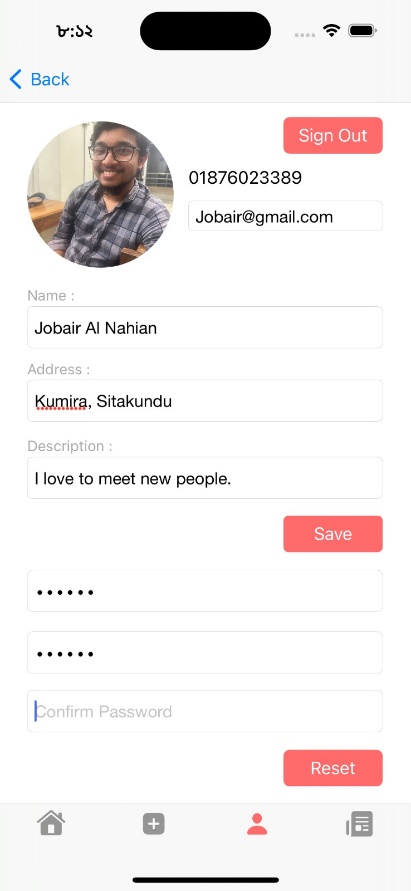
**3.3 Sign Up Page:** Fill the necessary details and register now. \* Marked text boxes are compulsory to fill. One can navigate to login page from here.



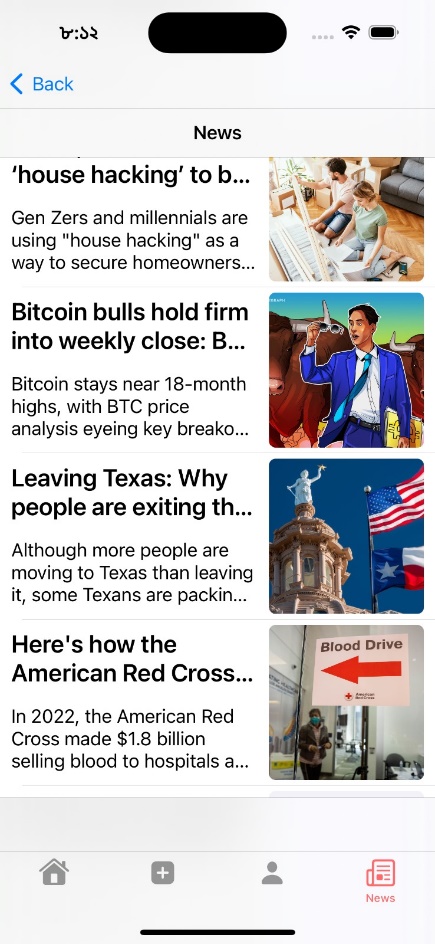


**3.5 Create post:** You can add a post by simply selecting a picture and adding some description corresponding to it. Description is must to fillup.

**3.6 Comments:** By clicking on comment icon one can watch every comment of the post and can add comment. Here the comment box are made by TableView and the contents are username time profile and the comments.Here the images are rounded for visually appealing UI.

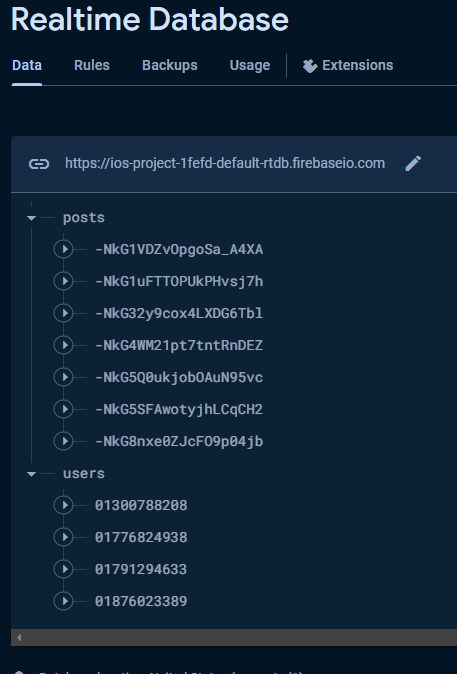


**3.7 Profile page** :One can see all the details about his deails and can Updating his profile and reset password . Also can signout from the app.

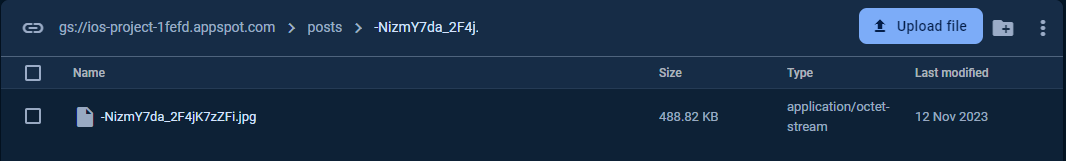


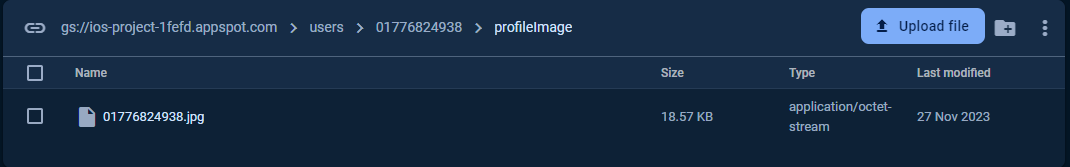
**3.7 News Section:** The News section dynamically retrieves and does Jason parsing and displays the news information by fetching data from the API link. Here table view used to show the cell contents one after another. Here navigation view was used to hold the view controller for smooth scrolling.

**3.8 Firebase Features:**



We used firebase real time to store user Information and update their information . Also for storing the data of Posts and Comments .





We used firebase real time to store user Information and update their information. Also for storing the data of Posts and Comments.

**4. Discussion:**

Developing Pure Connect had its challenges as we initially lacked our own Mac. However, we found creative solutions by using lab Macs and borrowing from our friends. Through collaboration and resourcefulness, we successfully completed the project. We could not add all the features that today’s Social Media app has. But with basic features Now, Pure Connect is ready for everyone to enjoy, thanks to the collective efforts of the team.

**5. Conclusion:**

We've made Pure Connect a friendly app where it's easy to share, connect, and stay updated. We aimed for a simple and enjoyable experience, and we're happy with what we've created. The app is all about bringing people together, sharing moments, and keeping everyone in the loop with what's happening.

**6. Future Plans:**

Looking ahead, we want to make Pure Connect even better:

* Make It More Intuitive: We're thinking of ways to make the app understand each user better, suggesting things they'd really like.
* Add More Features: We would try to bring messaging or groups, to make it even more fun and social.
* Keep the App Updated: Keeping an eye on new technologies. Maybe we'll try things like cool visual experiences or virtual reality to make using the app even more exciting.

**5. Reference:**

* [**https://developer.apple.com/documentation/uikit/uistoryboard**](https://developer.apple.com/documentation/uikit/uistoryboard)
* [**https://firebase.google.com/docs/ios/setup**](https://firebase.google.com/docs/ios/setup)
* **https://www.youtube.com/playlist?list=PLSzsOkUDsvdu5Mm67aBYs2YPu2OM4mFzt**