

FINAL PROJECT

Social Media Application
-by MinerYang Dec.2020



Outline

Part1 Application introduction

Part2 App Design & Implementation

Part3 Further Expectation



Part1

- About My App
- Functionalities
- Use cases

ABOUT MY APP

With the rapid development of smartphone usage and increasingly requirements for people to share life and opinions via Internet, I came up with the idea that can I develop an ios based social media app like twitter and instagram for people to share photos and connect with others.

We can sign up as a new user, login, logout, view and edit profile, add a post, comment and like or unlike a post, follow and unfollow other users as well as searching users.



FUNCTIONALITIES

Enter Page

- Sign Up
- Login

Home Page

- Logout
- View Posts
- Comment post
- View comments
- Like / Unlike post
- Navigate to user profile



FUNCTIONALITIES

Searching Page

- Searching user
- Navigate to User Profile page

Add post Page

- Publish new post
- Navigate to Home Page


User Profile Page

- View user profile
- Edit self profile
- Follow or Unfollow others
- Navigate to Home Page

LOGIN & SIGNUP

When you fully entered the
information the button will
be enabled

3:51



123@gmail.com


•••••

Login

Don't have an account? [Sign Up](#)

4:11

[Back](#) Sign Up



Email

Fullname

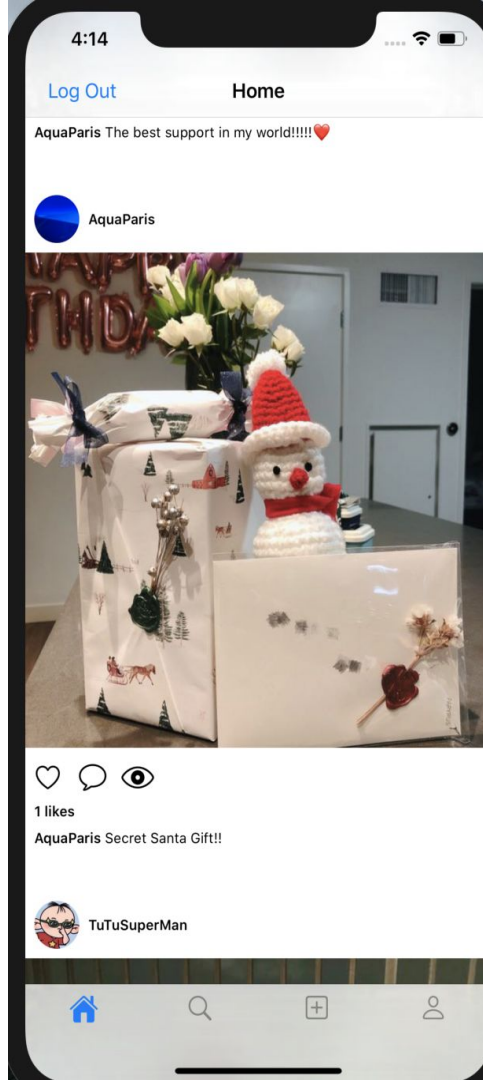
Username

Password

Sign Up

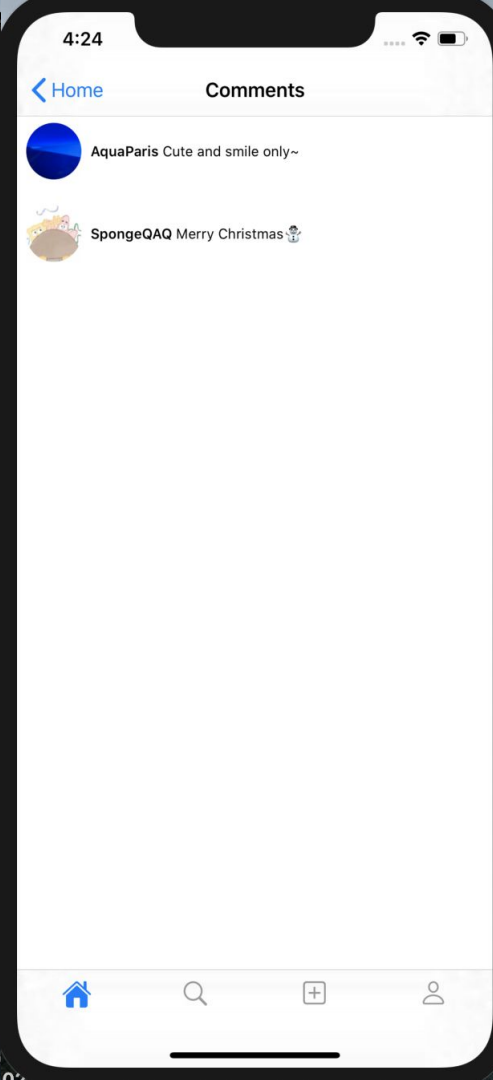
HOME PAGE

You can scroll up and down
to view posts
You can also like or unlike
a post, etc.



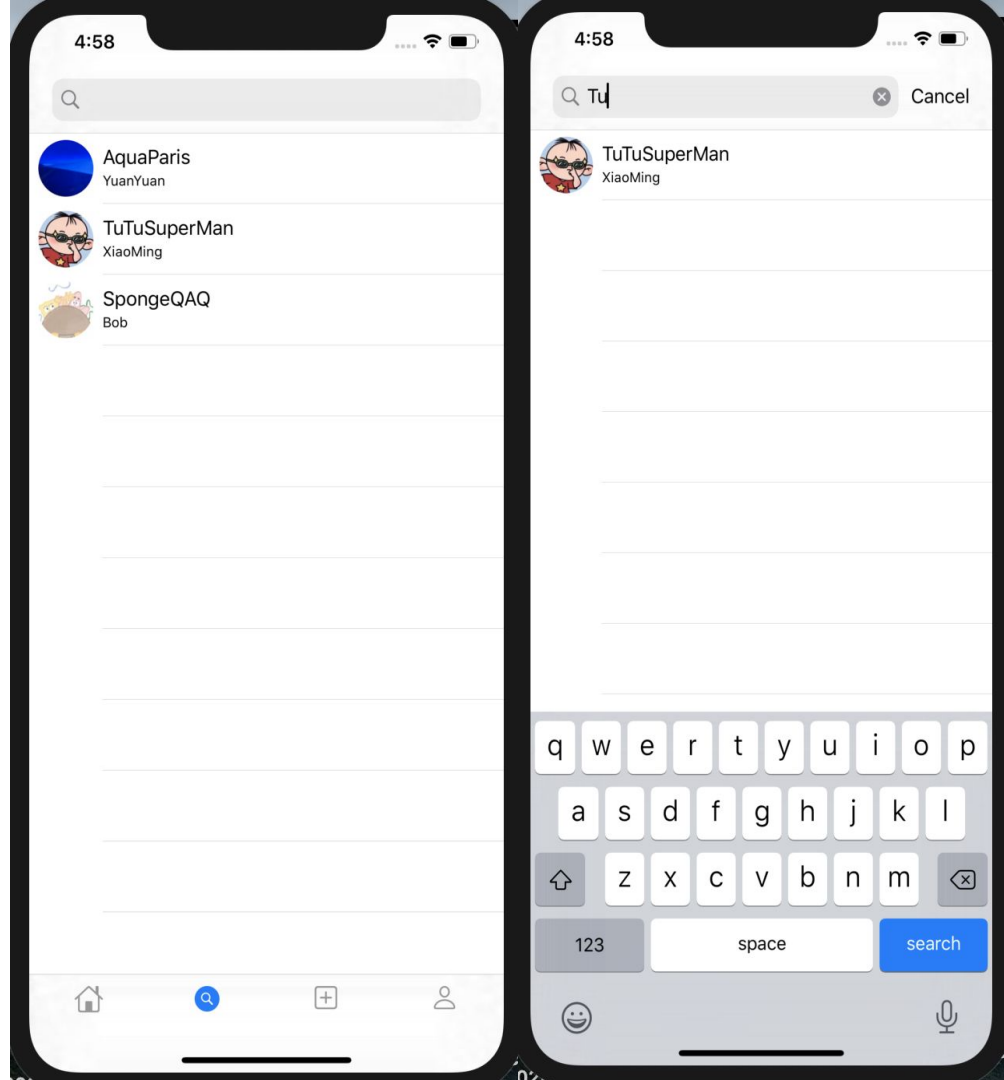
COMMENT

By clicking the comment and view button, you can comment a post and view all the comment of this post



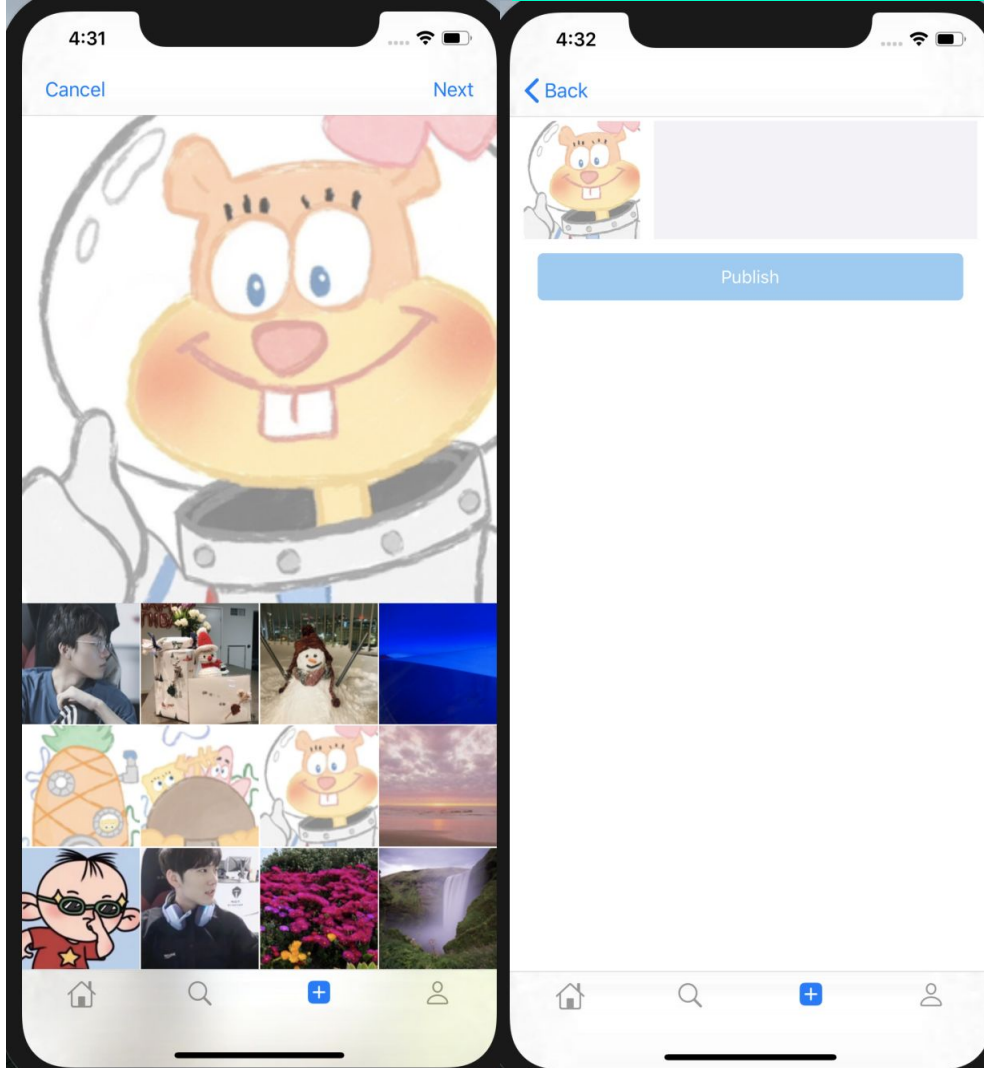
SEARCHING PAGE

By entering the username you can start searching and by clicking the profile image, it will leading you to the user page



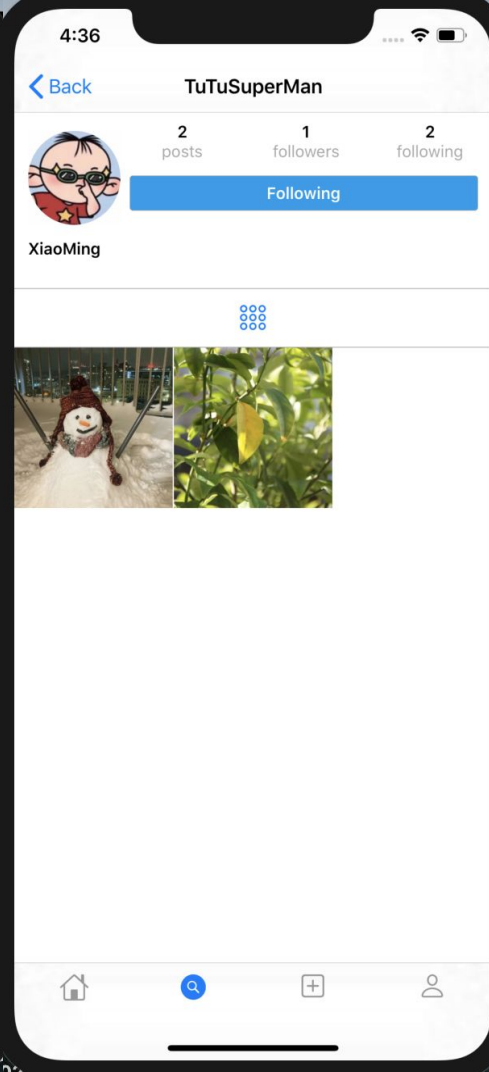
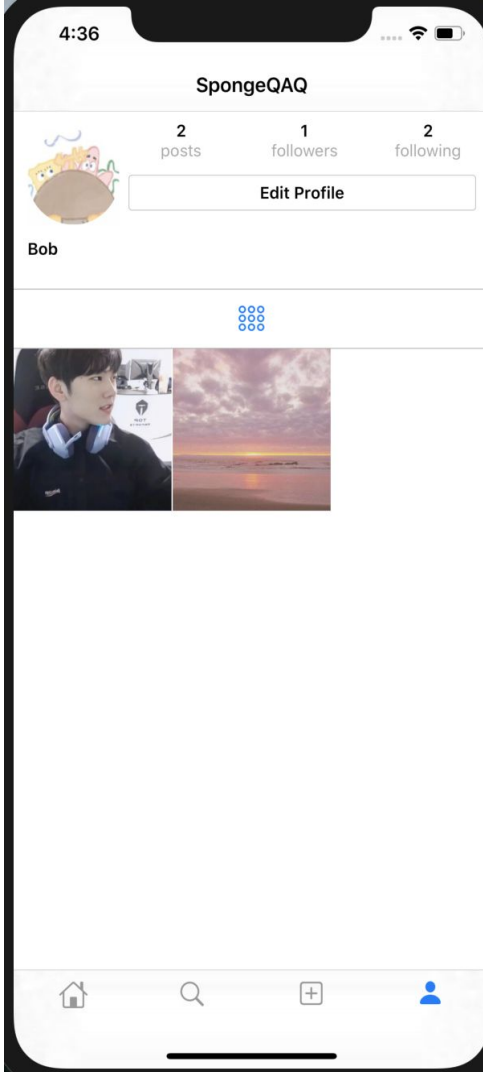
POST PAGE

By choosing the photo and
entering post description,
You can publish a new post



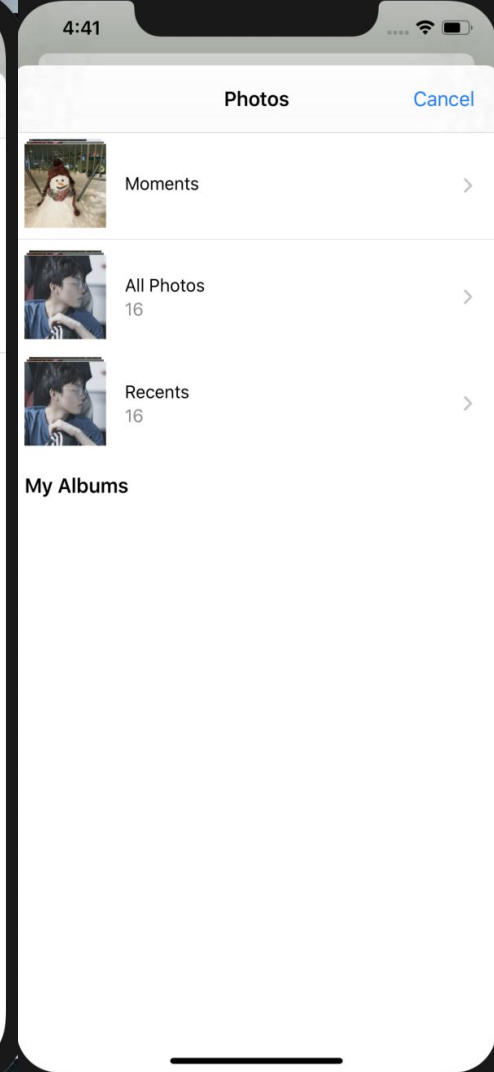
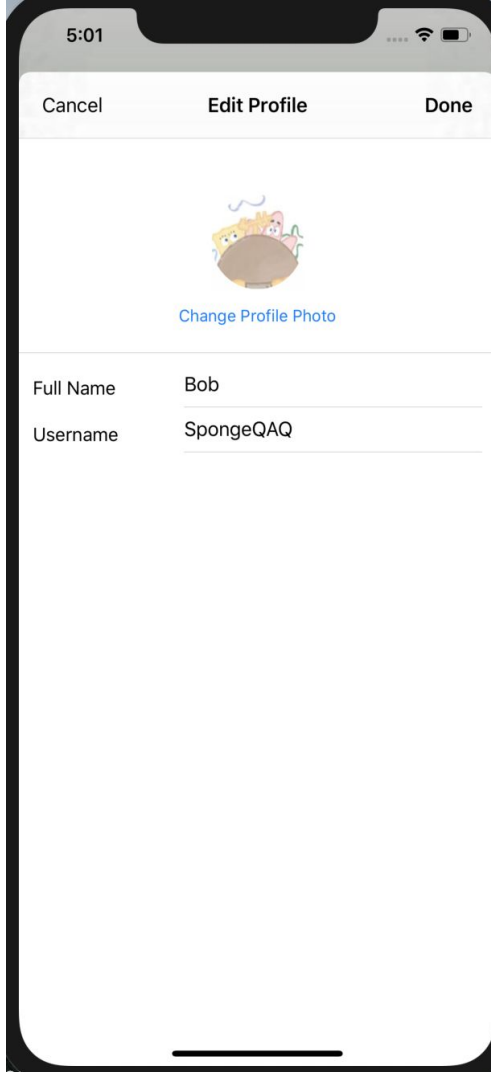
USER PROFILE PAGE

You can edit your own
profile in your page
you can follow or unfollow
others in other user's
profile page



EDIT USER PROFILE

By choosing the photo and
entering user name, you can
update your profile
You are not allowed to
change the full name





Part2

- Design Pattern and Structures
- Extension and Protocols
- ServerEnd - Firebase

DESIGN PATTERN & STRUCTURE

DESIGN PATTERN & STRUCTURES

MVC Design Pattern

Three roles:

- Model
- View
- Controller

- **Model**

The objects that hold the application data and define how to manipulate it. For example, in my application , one of the Model is the `User` struct, which you can find in `User.swift`.

- **View**

The objects that are in charge of the visual representation, basically, all the `UIView`-derived objects.

- **Controller**

It accesses the data from the model and displays it with the views, listens to events and manipulates the data as necessary.

USER

MODEL

User.swift

- id
- Username
- Full name
- Profile Image Url
- ★ Func Follow()
- ★ Func Unfollow()
- ★ Func CheckIsFollow()

VIEW

- **User Profile Header Cell**
UICollectionViewCell
- **Edit profile View**
UIView
- **User Post Cell**
UICollectionViewCell
- **Search User Cell**
UICollectionViewCell
UISearchBarDelegate

CONTROLLER

- **User Profile Controller**
UICollectionViewControll
er
- **Edit profile Controller**
UIViewController
UIImagePickerController
(delegate)
- **Search User Controller**
UITableViewController

POST

MODEL

Post.swift

- postid
- Creation Date
- Image Url
- Description
- Likes
- ★ Func adjustLikes()

VIEW

- **Select Photo Header Cell**
UICollectionViewCell
- **Select Photo Cell**
UICollectionViewCell

CONTROLLER

- **Select Image Controller**
UICollectionViewControll
er
UICollectionViewDelegate
FlowLayout
- **Upload Post Controller**
UIViewController
UITextViewDelegate

COMMENT

MODEL

Comment.swift

- Auto id
- User id
- Comment text
- Creation Date

VIEW

- **Comment Cell**
UICollectionViewCell
- **Edit Comment View**
UIView

CONTROLLER

- **Comment Controller**
UICollectionViewController
UICollectionViewDelegateFlowLayout
- **Edit Comment Controller**
UIViewController
UITextViewDelegate

OTHERS

VIEW

- **Home Profile Cell**

UICollectionViewCell

- **Login View**

UIView

- **SignUp View**

UIView

CONTROLLER

- **Home Profile Controller**

UICollectionViewController

UICollectionViewDelegateFlowLayout

- **Login View Controller**

UIViewController

- **SignUp View Controller**

UIViewController

UIImagePickerController (delegate)

EXTENSION & PROTOCOLS

Protocol

- **UserProfileHeaderDelegate**
 - `func handleEditFollowTapped()`
 - `func setUserStats()`
 - `func handleFollowersTapped()`
 - `func handleFollowingTapped()`
- **HomeCellDelegate**
 - `func handleUsernameTapped()`
 - `func handleLikeTapped()`
 - `func handleCommentTapped()`
 - `func handleViewCommentsTapped()`

Extension

- **UIView**
 - `func anchor()`
- **UIViewController**
 - `func alert()`
- **Database**
 - `func fetchUser()`
 - `func fetchPost()`

Custom Class

- **CustomImageView : UIImageView**
 - `func loadImage()`

SEVER-END
FIREBASE

FIREBASE

AUTHENTICATION

Store User information
(email & password) to
LogIn

- Identifier(email)
- Created Time
- Last Sign In Time
- User UID

REALTIME DATABASE

Store Regular data

- Users
- User-Posts
- User-Likes
- User-following
- User-followers
- Posts
- Posts-like
- comments

STORAGE

Store Image Data

- Post_Images
- Profile_Images

FIREBASE - AUTHENTICATION

Authentication

[Users](#)[Sign-in method](#)[Templates](#)[Usage](#)

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

[Get started](#)[Add user](#)

Identifier	Providers	Created	Signed In	User UID ↑
qk@gmail.com		Dec 13, 2020	Dec 18, 2020	0y3IgXquYWUAs2SEpC8ihDxYNG33
tutu@gmail.com		Dec 13, 2020	Dec 18, 2020	H3vHfcWTv6U65Gk2YCuG3UvJ7jT2
123@gmail.com		Dec 13, 2020	Dec 18, 2020	YUSNK8T2XwhfvJZzwH06Lde5HB...



Rows per page: 50



1 – 3 of 3



FIREBASE - AUTHENTICATION

`Auth.auth().createUser(email, password){(user, error) in ...}`

```
Auth.auth().createUser(withEmail: email, password: password) { (user, error) in
    //handle error
    if let error = error {
        //self.alert(message: error.localizedDescription)
        print("Failed to create user with error\(error.localizedDescription)")
        return
    }
    //
    guard let profileImg = self.PlusPhotoBtn.imageView?.image else{
        //self.alert(message: "imageView set failed")
        print("imageView set failed")
        return
    }
    // upload image
    guard let uploaddata = profileImg.jpegData(compressionQuality: 0.6) else {
        //self.alert(message: "upload image data failed")
        print("upload image data failed ")
        return
    }
    // place image in firebase database
    let filename = NSUUID().uuidString
    //print("image filename : \(filename)")
```

FIREBASE - AUTHENTICATION

```
Auth.auth().signIn(email, password){(user, error ) in ... }
```

```
print("Click Login Button")
Auth.auth().signIn(withEmail: emailTXT.text!, password: PswTXT.text!) { (user, error) in
    // handle error
    if let error = error {
        print("unable to Login with error:\(error.localizedDescription)")
        self.alert(message: error.localizedDescription)
        return
    }

    print("Successfully Login!")
    let mainST = UIStoryboard(name: "Main", bundle: Bundle.main)
    let VC = mainST.instantiateViewController(withIdentifier: "idTabBar")
    VC.modalPresentationStyle = .fullScreen
    self.present(VC, animated: true, completion: nil)
}
```

FIREBASE - REALTIME DATABASE

USERS

users

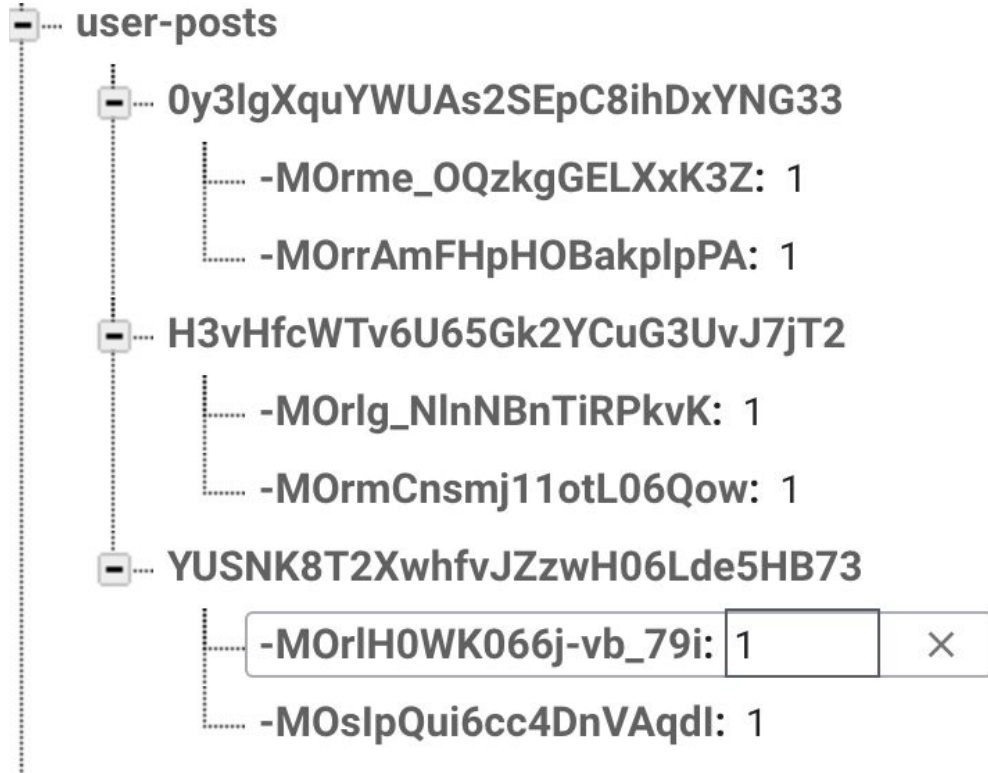


Structure

- User id
 - Full name
 - ProfileImage Url
 - User Name

FIREBASE - REALTIME DATABASE

USER-POSTS

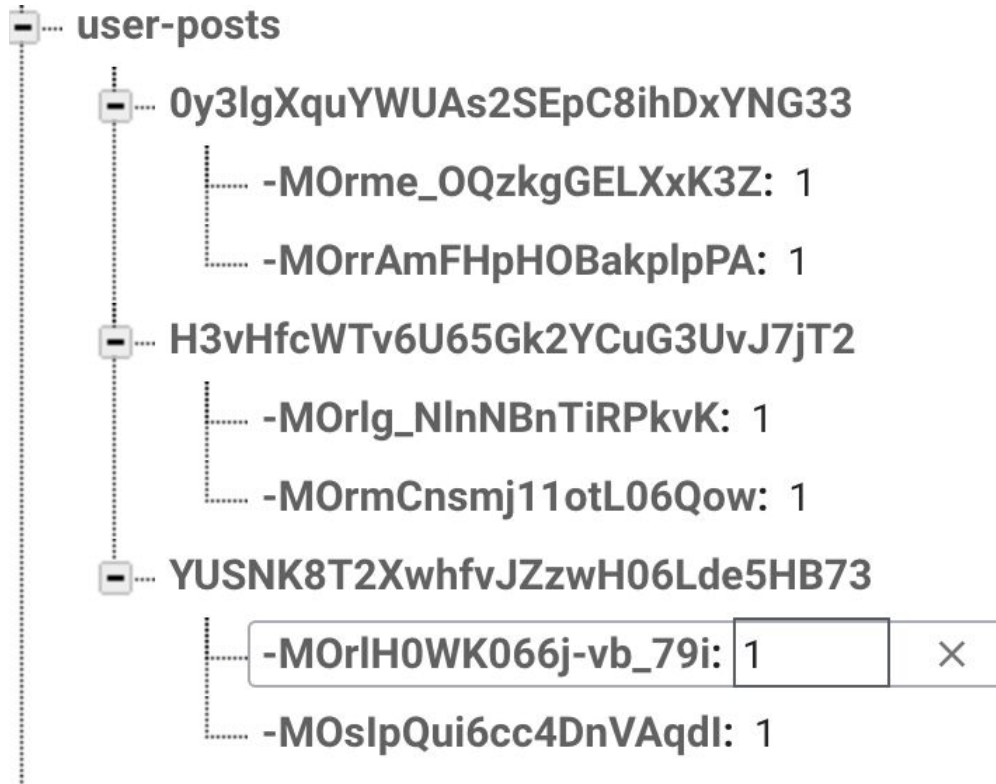


Structure

- User id
 - Post id

FIREBASE - REALTIME DATABASE

USER-LIKES

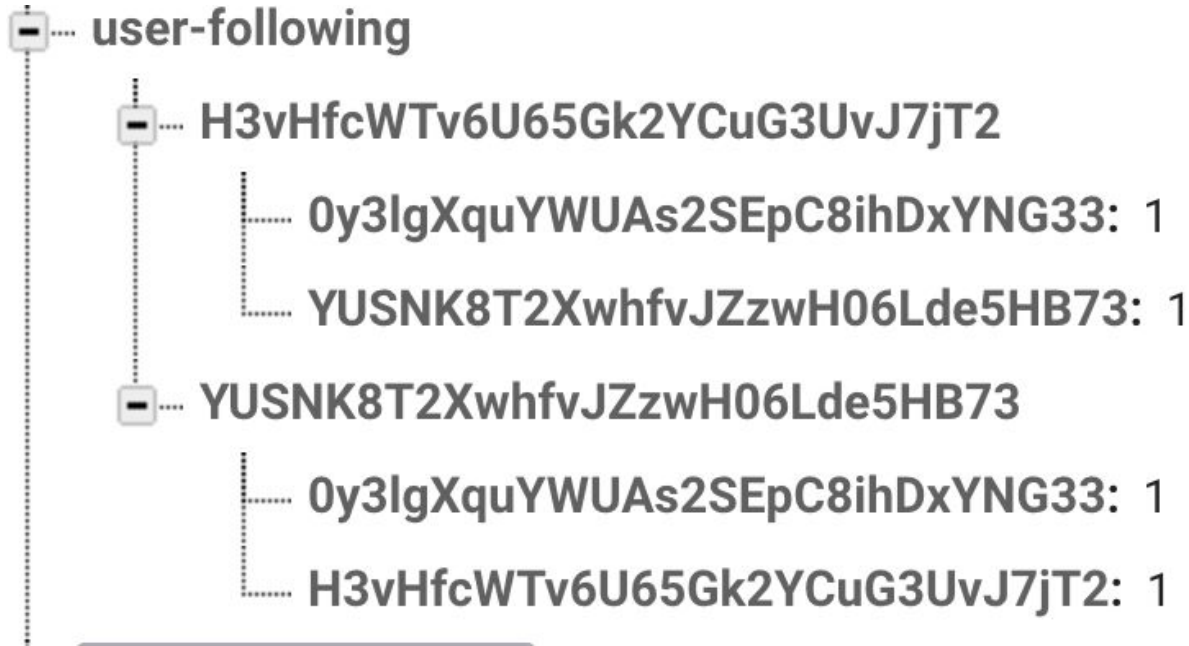


Structure

- User id
 - Post id

FIREBASE - REALTIME DATABASE

USER-FOLLOWING



Structure

- User id
 - User id (following)

FIREBASE - REALTIME DATABASE

USER-FOLLOWER

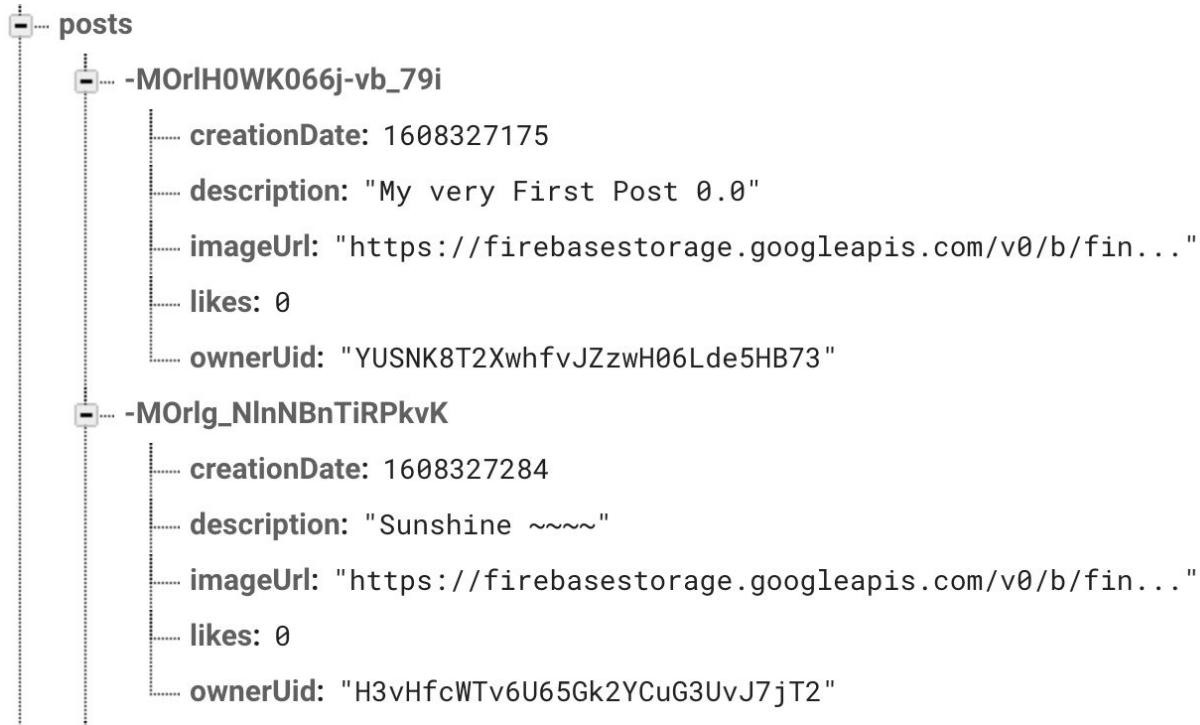


Structure

- User id
 - User id (follower)

FIREBASE - REALTIME DATABASE

POST



Structure

- Post id
 - Creation Date
 - Description
 - Image Url
 - OwnerUid(user id of who owns this post)

FIREBASE - REALTIME DATABASE

POST-LIKE



Structure

- Post id
 - User id(who liked this post)

FIREBASE - REALTIME DATABASE

COMMENTS



Structure

Post id

- Auto comment id
 - Comment Text
 - creationDate
 - uid(user id)

FIREBASE - REALTIME DATABASE

```
Database.database().reference.child("structureName").updateChildValue(value){(
error, red ) in ... }
```

```
guard let uid = user?.user.uid else { return }
let dic : [String : Any] = ["name": fullname,
    "username": username,
    "profileImgUrl": profileImgUrl]

let value = [uid : dic]

// SAVE INFO TO DATABASE
Database.database().reference().child("users").updateChildValues(value) {
    (error, ref) in
    //self.alert(message: "Successfully create user and save info!")
    print("Successfully create user and save info!")
}
```

FIREBASE - REALTIME DATABASE

```
Database.database().reference.child("structureName").observeSinglePost(of  
.value){(snapshot) in
```

```
    Snapshot -> <String, AnyObject>    ...
```

```
}
```

```
static func fetchPost(with postId: String, completion: @escaping(Post) -> ()) {  
  
    POSTS_REF.child(postId).observeSingleEvent(of: .value) { (snapshot) in  
        guard let dictionary = snapshot.value as? Dictionary<String, AnyObject> else { return }  
        guard let ownerId = dictionary["ownerId"] as? String else { return }  
  
        Database.fetchUser(with: ownerId, completion: { (user) in  
            let post = Post(postId: postId, user: user, dictionary: dictionary)  
            completion(post)  
        })  
    }  
}
```

FIREBASE - STORAGE

Storage

[Files](#)[Rules](#)[Usage](#)

[gs://finalprojectinsgram.appspot.com](#) > post_images

Upload file



<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	06A429CD-EB01-4E4A-A93D-1C4DC2DDDDCF	38.68 KB	application/octet-stream	Dec 15, 2020
<input type="checkbox"/>	08F1B143-0425-40DA-86ED-8548ECAB4DF0	83.32 KB	application/octet-stream	Dec 14, 2020
<input type="checkbox"/>	09AF56CF-B15D-4915-B6D8-0DFE245A1735	36.11 KB	application/octet-stream	Dec 15, 2020
<input type="checkbox"/>	1D531809-D955-4086-AE90-641F712E4CCC	83.32 KB	application/octet-stream	Dec 14, 2020
<input type="checkbox"/>	29FD7ADC-6BD9-4869-A585-79A82C741AFF	33.73 KB	application/octet-stream	Dec 18, 2020

FIREBASE - STORAGEEE

`Storage.storage().reference.child("structureName").putData(uploaddata){(metadata, error) in`

`Storage.storage().reference.child("structureName").downloadURL{url, error) in ...} ... }`

```
let filename = UUID().absoluteString
let postRef = Storage.storage().reference().child("post_images").child(filename)
postRef.putData(uploaddata, metadata: nil) { (metadata, error) in
    //error
    if let error = error {
        print(error.localizedDescription)
    }

    // image url
    //var imageurl = ""
    postRef.downloadURL { (url, error) in
        if let error = error{
            print(error.localizedDescription)
            return
        }
        guard let abstr = url?.absoluteString else{
            print("failed to get image url absolute string")
            return
        }
    }
}
```

CONCLUSION

&

FURTHER EXPECTATION

CONCLUSION

All Basic Functionalities in my proposal are all implemented with server (database) embedded to upload and retrieve data.

And Implemented the Advanced searching function as well.

WHAT TO IMPROVE

As a real-world social media app, it's better to have message and notification functions.

And might need to use Pagination to load views when the data is large in volume.

...



THANK YOU !