CIRCLE APP DOCUMENTATION

New York University - iOS Programming
Spring 2020

Leena Loo
Ivanna Pena

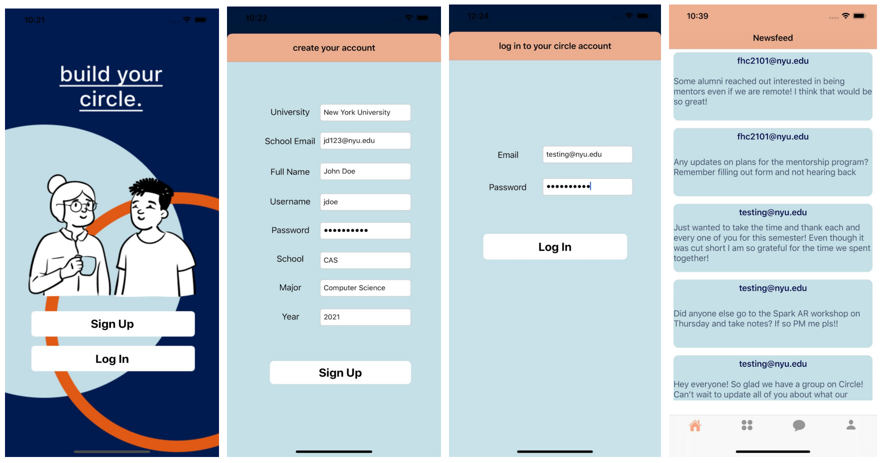
APP OVERVIEW

At NYU, there is a struggle to stay connected to our classmates, and have a sense of community due to our unique academic setting, campus, and culture. Throughout these unprecedented times, this has only been intensified, here at NYU, and in college communities across the country. Students are struggling to keep up with classes, find assistance with their work beyond zoom lectures, and reach out to their other classmates. Many are resorting to unique avenues of communication and community building to stay entertained and connected. Our solution is to make a social networking app to help our own NYU community, and other colleges, stay connected in a more casual way than current class discussion boards even after the pandemic ends.

This application will allow users to create accounts and profiles if they are students at an accredited university, and interact with other students strictly from their university. This will be done via forums, messaging and video chats categorized by different interests, classes, or majors. Within each interest group, there will be an option to randomly chat with other individuals in the group who are also online. Our first iteration of the app will be limited to just NYU students and will have more basic functionalities.

USER MANUAL

How to use Circle.



University	New York University
School Email	jd123@nyu.edu
Full Name	John Doe
Username	jdoe
Password	••••••••••
School	CAS
Major	Computer Science
Year	2021

Email	testing@nyu.edu
Password	••••••••••

fhc2101@nyu.edu
Some alumni reached out interested in being mentors even if we are remote! I think that would be so great!

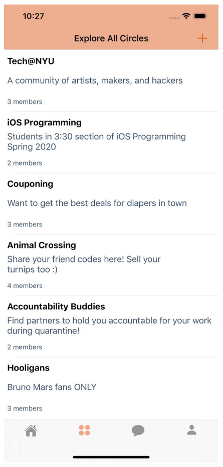
fhc2101@nyu.edu
Any updates on plans for the mentorship program? Remember filling out form and not hearing back

testing@nyu.edu
Just wanted to take the time and thank each and every one of you for this semester! Even though it was cut short I am so grateful for the time we spent together!

testing@nyu.edu
Did anyone else go to the Spark AR workshop on Thursday and take notes? If so PM me pls!!

testing@nyu.edu
Hey everyone! So glad we have a group on Circle! Can't wait to update all of you about what our

At launch, the user is presented with an animation of the Circle logo, followed by a transition into the **Welcome Screen** which displays prompts to **Sign Up** or **Log In**. To sign up, the user must fulfill all of the authentication requirements and fill all fields to proceed. Once logged in, the user is directed to the **Newsfeed** Screen, which displays posts from all groups that the current user is a part of.


10:27

Explore All Circles +

Tech@NYU
A community of artists, makers, and hackers
3 members

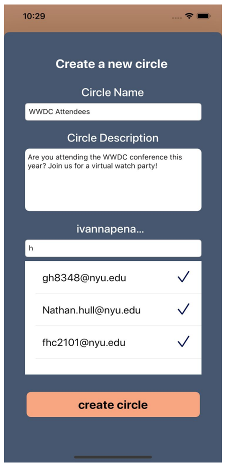
iOS Programming
Students in 3:30 section of iOS Programming Spring 2020
2 members

Couponing
Want to get the best deals for diapers in town
3 members

Animal Crossing
Share your friend codes here! Sell your turnips too :)
4 members

Accountability Buddies
Find partners to hold you accountable for your work during quarantine!
2 members

Hooligans
Bruno Mars fans ONLY
3 members


10:29

Create a new circle

Circle Name: WWDC Attendees

Circle Description: Are you attending the WWDC conference this year? Join us for a virtual watch party!

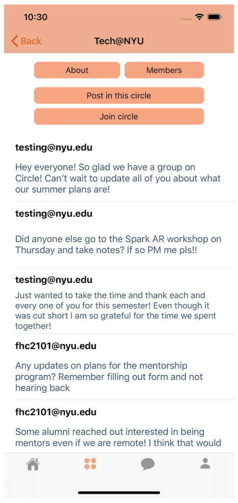
ivannapena...

h

gh8348@nyu.edu ✓
Nathan.hull@nyu.edu ✓
fhc2101@nyu.edu ✓

create circle

If the user clicks on the second icon in the tab bar, they will be taken to the **Groups page**, which shows **all groups** that have been created within the university, along with **descriptions** and the **number of members.** The + button in the navigation bar will allow a user to **create** their own group. On the **Create Group** Screen, the user can enter a new group **name**, **description**, and add any **existing users** by typing in their email and **selecting from the suggestions**. The create circle button will appear once all fields are full, and will then take you back to the **Groups** Screen. The user can click any of the groups to go to the corresponding screens for that particular group.


10:30

< Back Tech@NYU

About | Members | Post in this circle | Join circle

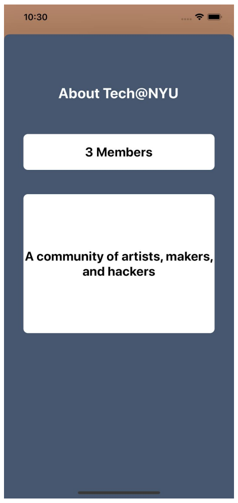
testing@nyu.edu
Hey everyone! So glad we have a group on Circle! Can't wait to update all of you about what our summer plans are!

testing@nyu.edu
Did anyone else go to the Spark AR workshop on Thursday and take notes? If so PM me pls!!

testing@nyu.edu
Just wanted to take the time and thank each and every one of you for this semester! Even though it was cut short I am so grateful for the time we spent together!

fhc2101@nyu.edu
Any updates on plans for the mentorship program? Remember filling out form and not hearing back

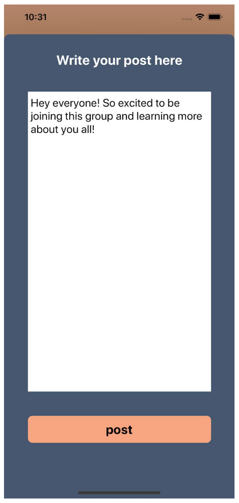
fhc2101@nyu.edu
Some alumni reached out interested in being mentors even if we are remote! I think that would


10:30

About Tech@NYU

3 Members

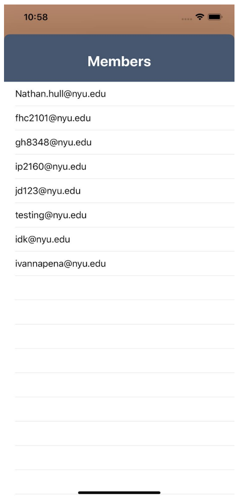
A community of artists, makers, and hackers


10:31

Write your post here

Hey everyone! So excited to be joining this group and learning more about you all!

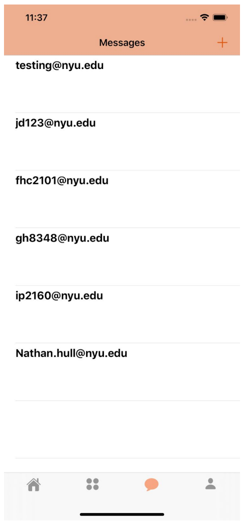
post


10:58

Members

Nathan.hull@nyu.edu
fhc2101@nyu.edu
gh8348@nyu.edu
ip2160@nyu.edu
jd123@nyu.edu
testing@nyu.edu
idk@nyu.edu
ivannapena@nyu.edu

Once a user is at a group's corresponding screen, they are given the option to **join** the circle if not already a member, to **post** in the group, and to **view** group details. The **About Screen** will show the **number of members** in the group, as well as the group's **description.** If a user chooses to post in the circle, they can write in the text input what they want to post, and it will then return and appear on the **Main Group Screen**. The **Members List Screen**, which displays the **current members** of the respective group. The user can then click on a user cell, to see the profile page of the selected user, which includes details about themselves and what other groups they are part of.


11:37

Messages

testing@nyu.edu

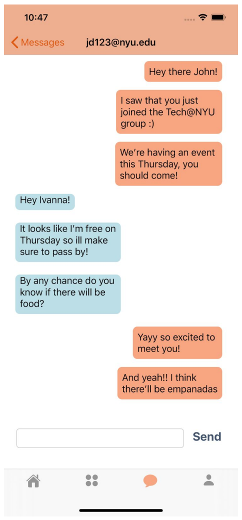
jd123@nyu.edu

fhc2101@nyu.edu

gh8348@nyu.edu

ip2160@nyu.edu

Nathan.hull@nyu.edu


10:47

Messages jd123@nyu.edu

Hey there John!

I saw that you just joined the Tech@NYU group :)

We're having an event this Thursday, you should come!

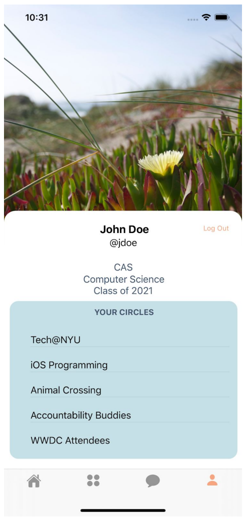
Hey Ivanna!

It looks like I'm free on Thursday so ill make sure to pass by!

By any chance do you know if there will be food?

Yayy so excited to meet you!

And yeah!! I think there'll be empanadas


10:31

John Doe

@jdoe

CAS
Computer Science
Class of 2021

YOUR CIRCLES

Tech@NYU

iOS Programming

Animal Crossing

Accountability Buddies

WWDC Attendees

The third icon on the tab bar allows a user to see all of the messaging conversations they are having with other users. From this list, a user can click in to see the past messages they have sent to each other, and continue to send messages using the text input box and send button. If a user wants to start a new conversation with somebody that they have not messaged before, they can press the plus icon in the navigation bar of the messages screen. This will take them to a screen that will allow them to select which user they want to start a conversation with from the list of existing users.

The fourth icon on the tab bar allows a user to see their own profile and account. If the user clicks on the picture area, they will be prompted to select a photo (can choose between taking a picture and camera roll) to set as their profile picture, which can be changed whenever they click on the area. The rest of their information from creating an account is displayed on this screen, along with a list of the circles that they have joined, which is updated when they join or create another group. They are able to log out of their account from this screen, which will take them back to the welcome screen and prompt them to create a new account or log in again.

TECHNICAL DOCUMENTATION

A brief overview of the implementation and an overview of the logic.

LOGIC

Animation and Welcome Screen: Upon launch, a user is presented with an animation of the Circle logo, accompanied by a sound. The animation was created using CoreGraphics, and the sound was implemented using AVFoundation.

Authentication:

Create Account: User authentication for Circle is supported by Firebase Authentication for Email and Password Sign-Ups. Due to the current school restriction for Circle users, ReGex is used on email fields to verify a valid NYU email. Other security checks such as password strength are also implemented. To create an account on our Firebase Database, the user must fill in all fields, after which, the user is created, and a login call is made to direct the user to their newsfeed.

Login: If a user has already created a profile, they can log in with their valid user credentials which are verified through Firebase Authentication after which, the user is directed to their newsfeed.

Tab Bar Controller: Upon success, the login/create account pages segue into a tab view controller hierarchy containing navigation controllers for these four things: Newsfeed, Groups, Messaging, and My Profile. The tab bar appears in all of these views, allowing a user to navigate the app.

Newsfeed: The Newsfeed is the initial view controller that the Tab Bar displays. It displays all of the messages from groups that the current user is a part of. The Newsfeed calls a function in the DataService called GetAllFeedMessages. From here, it uses the database reference to go into Firebase and loop through each group. If the current user's email is contained in the group, then it loops through the messages within that group and appends it to the message array. After GetAllFeedMessages finishes going through the groups, the messages are then each displayed in a newsfeed cell, with the user who posted the message and the message content itself.

Groups:

Create Group: The group object contains the group title, group description, and an array of group members. This view controller allows the user to input these fields in the text boxes, and click the create group button. Once pressed, the createGroups function is called, which creates a new group object key in the database with its respective values as children.

Groups Feed: This is the home page for a group and displays a list of messages. From here, you can access the About, Members and their profiles, and post a new message in the group. The messages are stored in Firebase under an individual group that it belongs to. Each message is stored with the user's email who posted it and the contents (text) of the post. We first retrieve each message and append the messages to an array in a loop and then in the view controller, display each message email and content in a table view cell, which is formatted in GroupVCCell.

Post: Stores the email of the current user as well as the text from text input into the database as a message object for the current group.

About Page: Displays the current group's description and counts the number of members in the member array.

Members: Loops through the members array and displays each element a memberTableView Cell in the table view. From here, you can click a member to view their profile.

Profiles:

Member Profile: This view controller will loop through the users in Firebase and if the user contains the email that was selected, then that user's information will be displayed in the member profile view. For the user's groups, it loops through each group's member array to find if the user's email is contained, and displays the group's title in the table view if it does contain their email

Profile: This view controller displays the current logged in user's email. It can be accessed from the rightmost icon on the tab bar. It will retrieve data from Firebase using the current user's user id. The user can also click on their profile picture to change it using a UIImagePicker popup, which will store the selected image

Messaging:

Create Chat: The Chat object contains the chat key and the two chat members. This view controller allows the user to input what existing user they want to start a private chat message with, and click the create chat button. Once pressed, the createChats function is called, which creates a new chat object key in the database with its respective members

Message Preview: A table view of each of the Chat objects, displaying the recipient user's email (retrieving the member array from a Chat object and selecting the index that did not contain the current user's email) and the most recent ChatMessage object's content using the messagePreviewTableViewCell. Upon selection of a cell, it will segue to the messaging screen and initialize the view controller with the current Chat object.

Messaging: The messaging view controller displays every ChatMessage object's contents within the parent Chat object through the table view and messagingTableViewCell by calling getallchatmessages in the data services to retrieve the data as an array of objects. The view controller checks each ChatMessage's senderId to see if it was a message that was sent by the current user or by the other user, which dictates how it appears on the screen (whether it is orange or blue and which side of the screen it appears on). The text input allows a user to type out a message and send using the button. This will call the function uploadpost function in data services and create a ChatMessage object within the current Chat object.

Services:

Data Service: This file is for managing our database references in one place. Since groups, users, and chats are all somewhat connected and referenced in the same functions, all functions relevant to retrieving data from Firebase are located here. These functions are called in all of our other view controller files.

Storage Service: This file is for managing our images, which are stored in Firebase Storage. All profile picture related functions for uploading and retrieving are defined here.

COCOAPODS AND APIS

- **Firebase/Analytics:** Recommended by Firebase during setup
- **Firebase/Auth:** Email/Password Authentication for Users
- **Firebase/Database:** Database for Users, Groups, and Chats
- **Firebase/Storage:** Storing User Profile Pictures
- **AlamofireImage:** Profile Picture Customization with access to camera and photo library
- **AVFoundation:** Playing Sound with Animation
- **CoreGraphics:** Animation on Logo

OUTSIDE ELEMENTS

- Chit-Chat Repo
  - Referenced for...
    - **Firebase** integration for Users and Groups
    - Implementation of **Group Posts** and **Group Lists**
    - **Profile picture** customization
- Making a Messaging App: Episode 4 (Swift 3 in Xcode)
  - Referenced for…
    - **Messaging Cell set up** for MessagingViewController Storyboard
- Soundsnap
  - Twinkle **sound** for animation
- OpenPeeps
  - Welcome screen people graphics
- All other graphics/designs were either from XCode or were made by us in Figma

MEMBER CONTRIBUTIONS

Ivanna Pena:

- Firebase Set Up
- Logo
- Code:
  - Groups Feed View Controller
  - Create Group View Controller
  - Post View Controller
  - Groups View Controller
  - Register/Login View Controllers + Authentication
  - Profile View Controller - Image Picker
  - Animation View Controller - Sound Implementation
  - Create Chat View Controller
  - Models (Message, Group, Chat, ChatMessage)

Leena Loo:

- Designs and Storyboards
- Code:
  - Newsfeed View Controller - getallfeedmessages function in dataservices; tableview; initialization of data
  - Profile View Controller
  - Groups View Controller - display groups in cells
  - Group Feed View Controller - follow a group + update info in firebase
  - About View Controller
  - Members View Controller
  - Member Profile View Controller
  - Animation View Controller

    - Register/Login View Controllers - firebase error checking and segues to segue if successful login and registration in firebase
    - Messaging Preview View Controller
    - Messaging View Controller

Anything not mentioned directly above was contributed to by **both of us**

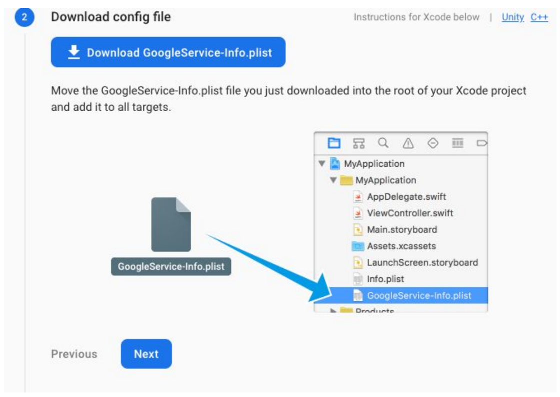
COMPILE AND RUN CIRCLE

Requirements

- Xcode version 11.4.1+
- Swift 5
- iPhone 11 Pro Max or higher
- iOS 13.0+

Installation

1. Install CocoaPods
2. Open Terminal and run **pod install** directly in circle/Circle folder.
3. Open the Circle.xcworkspace in Xcode.
4. Change the Bundle Identifier to your domain.
5. For Firebase to run, create a new project for your application.
6. Download GoogleService-Info.plist from the newly created Firebase project and replace it with the old one.



Download config file

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.

7. Enable Email/Password authentication
8. Create Realtime Database
9. Set Realtime Database rules to:

```
{
   "rules": {
         ".read": true,
         ".write": true
```

```
                    }
                }
```
8. Enable your Firebase Storage
9. Run Circle on your iOS Simulator for an iPhone 11 Pro Max

If Issues Occur With Firebase installation, use ours.

- Go to Firebase Link here
- Sign In With Our Test Email
  - **Email:** circle.ip807.ll3337@gmail.com
  - **Password:** Circle2020!
- Explore our Database
- Install CocoaPods
- Open Terminal and run **pod install** directly in circle/Circle folder.
- Open the Circle.xcworkspace in Xcode.
- Run Circle on your iOS Simulator for an iPhone 11 Pro Max

Test Account

To log in with an existing user to view all features implemented on our application, after the welcome screen, login with the credentials below:

Email: testing@nyu.edu

Password: LeenaLoo2!

Check out our GitHub here.

DATA DICTIONARY

- circle-ip807-ll3337
  - Groups
    - GroupID (autogenerated)
      - title: The name of the group (String)
      - description: The description of the group (String)
      - members: The emails of members that are part of the group (String Array)
      - messages
        - MessageID (autogenerated)
          - senderId: The email of the user that posted (String)
          - Content: The text of the post (String)
  - Users
    - UserID (autogenerated)
      - email: The user's email, needs "@nyu.edu" to register (String)
      - fullname: The user's full name (String)

                - major: The user's major (String)
                - password: The user's password, needs to fulfill all password requirements (String)
                    - At least six characters long
                - profileImageURL: URL of image to pull from Firebase Storage
                - provider: defaulted as the word 'password'
                - school: the user's school within the university (String)
                - university: the user's university, useful later if used by multiple schools (String)
                - username: the user's username (String) [did not use this]
                - year: the user's year of expected graduation (String)
    - Chats
        - ChatID (autogenerated)
            - members: The emails of members that are part of the chat (String Array)
            - messages
                - MessageID (autogenerated)
                    - content: The text of the message (String)
                    - senderId: The email of the user that messaged (String)

FUTURE WORK

- Debugging messaging (the cells dynamically change height when scrolling and constraints are overlapping); also need to make previews show the correct preview
- Show join circle button only if a user is not part of the circle, and only show post in circle button if user is part of the circle
- Adding more fields to our database
- Show profile pictures for users and groups in all of the view controllers
- Improving Newsfeed UI, by sending the user to the cells respective group on click
- Figure out how to get username instead of email
- User circles on Profile as a collection view and make Profile scrollable
- The option to sign in with Google
- Send email verification on sign up for security
- User Status Light, to determine if a user is currently on the app
- Implement video chat
- Adding security features and filters for posts, picture sending in messages
- Ability to report users
- Expanding to more schools than just NYU (will need to have different databases)