Firebase:

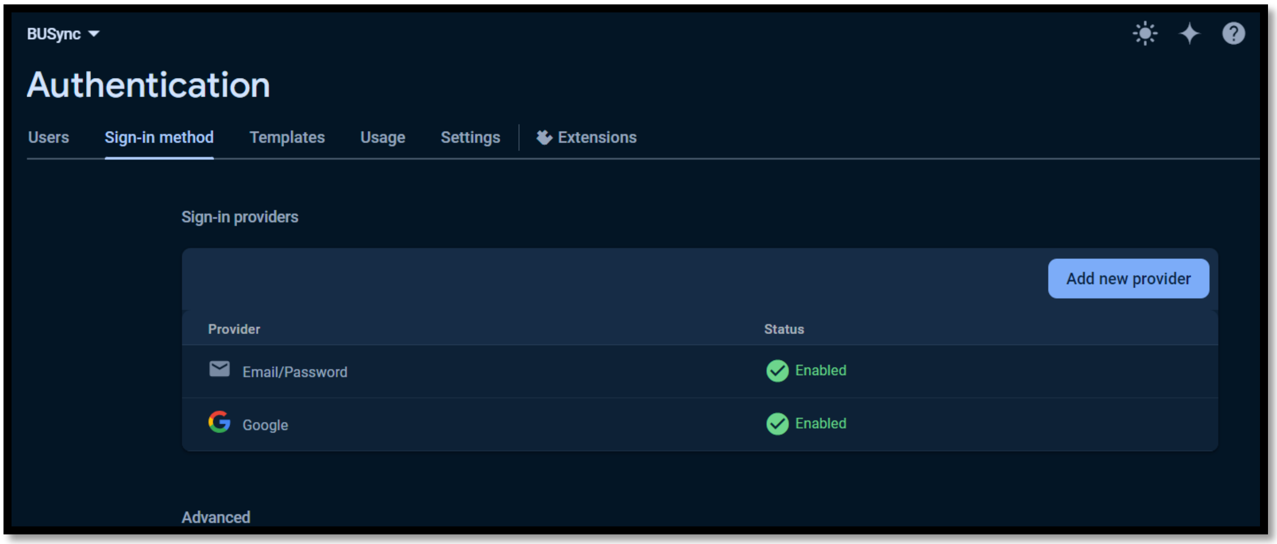
Running BUSync using your own Firebase production server

To run this application, you'll first need to create a Firebase account and create a new project named BUSync.

Reference for creating firebase account, and setups.
https://firebase.google.com/docs

1) Authentication

- Go to the authentication side menu
- Go to sign-in method and add configuration for Email/Password and Google Oauth credentials and any other sign-in provider you'd like to add to the project.

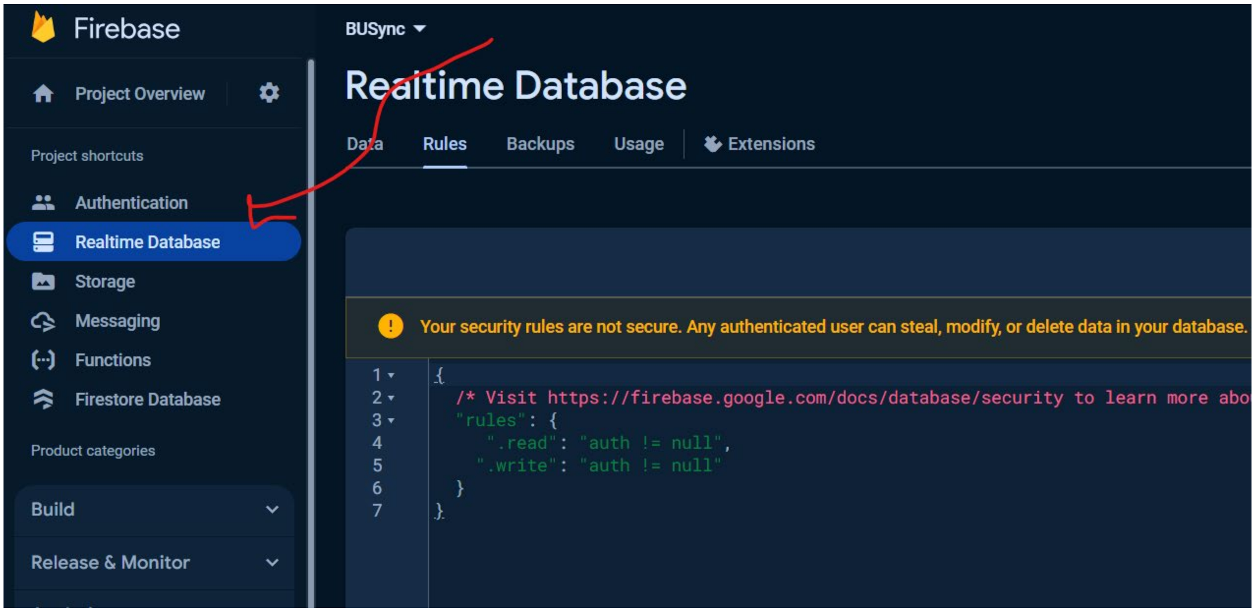


Sign-in providers

Provider	Status
Email/Password	✓ Enabled
Google	✓ Enabled

2) Realtime Database

- Create and Follow firebase instructions for creating a real-time database.



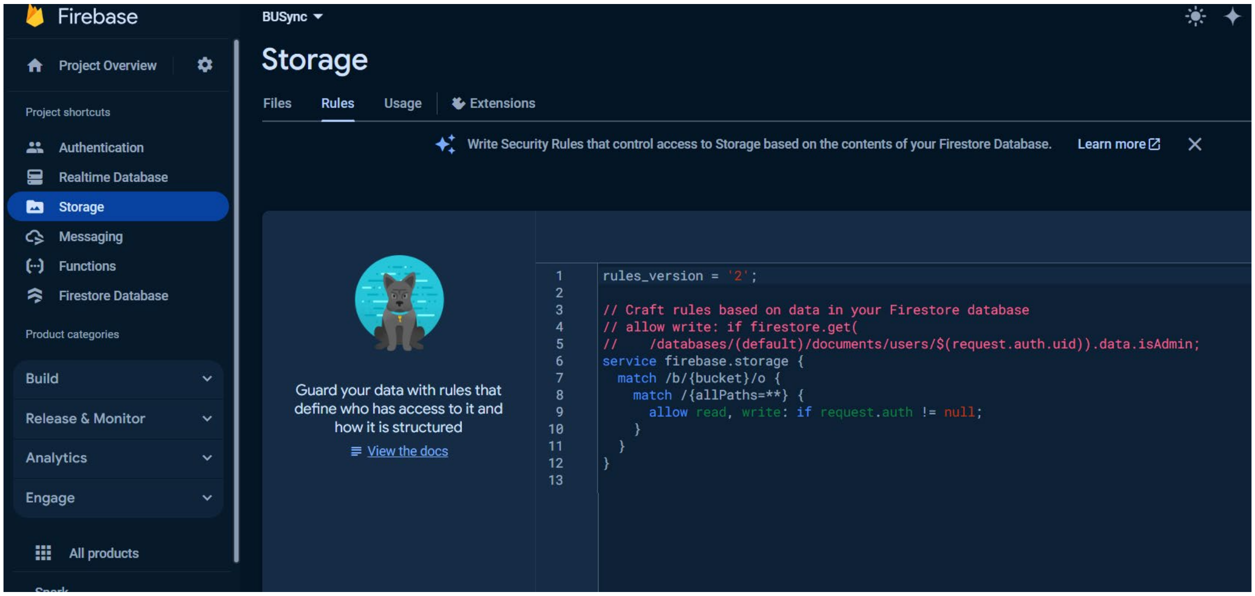
Your security rules are not secure. Any authenticated user can steal, modify, or delete data in your database.

```
{
  /* Visit https://firebase.google.com/docs/database/security to learn more abo
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

Make sure to change the rule to what is on this image to allow authenticated read and write for users.

3) Storage

- Create and follow Firebase instructions for creating a new storage



Write Security Rules that control access to Storage based on the contents of your Firestore Database.

Guard your data with rules that define who has access to it and how it is structured

```
rules_version = '2';

// Craft rules based on data in your Firestore database
// allow write: if firestore.get(
//    /databases/(default)/documents/users/$(request.auth.uid)).data.isAdmin;
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

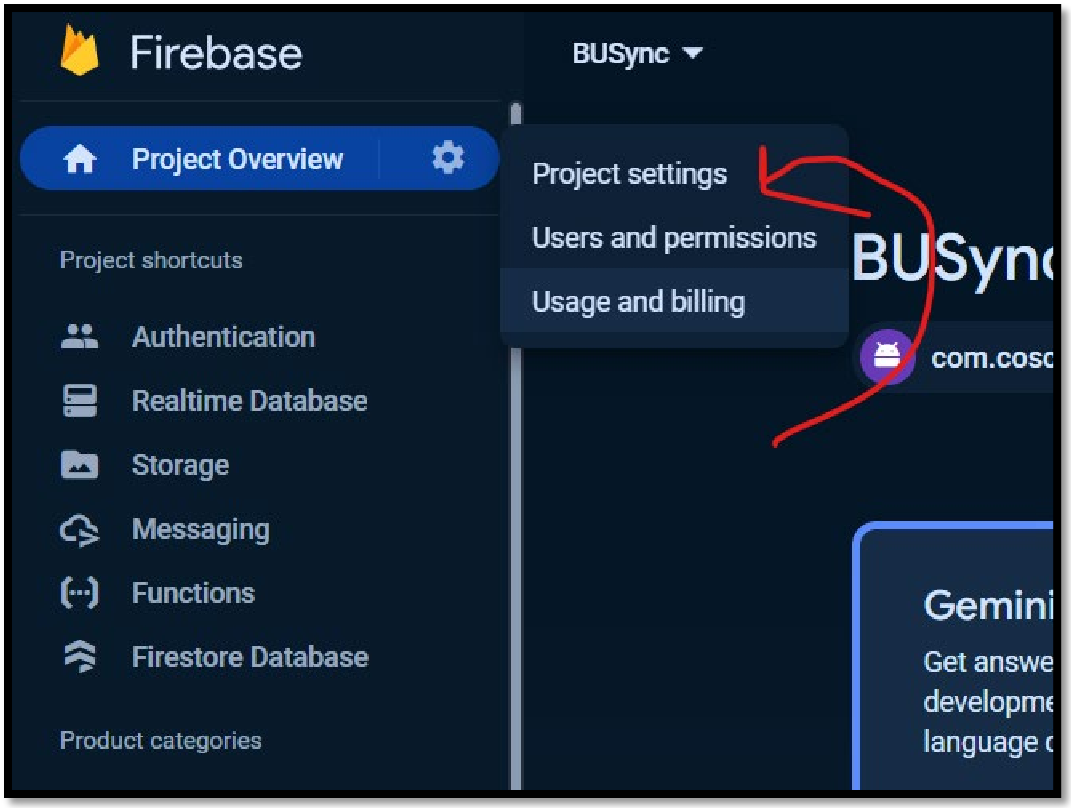
Make sure to change the rule to what is on this image to allow authenticated read and write for users.

4) After configuring your project

After creating your project, go to your project settings and add your SHA certificate fingerprint for both SHA-1 and SHA-256 certificates.

5) Download the google-services.json file and place it inside the root of the app folder at this directory: `andriod/app/google-services.json`

You can find the file by going to the settings of project overview



And then, scroll down and click on the android icon, before choosing the google-services.json.

App Code Layout:

The app has the following code sections:

App/java/package: com.cosc3p97project.busync

Controller: Contains two custom adapters to view messages

- Message Adapter
- Tabs Accessor Adapter – provides a means to view different tabs, Chats, Groups, Requests, etc.

Model: Contains two custom classes to hold contact information, and message information.

- Contacts
- Messages

View

- Fragment: Four Fragments
    - Chats Fragment: The following code implements functionality for private chats, utilizing real-time database interactions, populating a recycle view with user data fetched from fire-base real-time database.
    - Contacts Fragment: Implements a fragment for displaying user contacts, utilizing firebase real-time database, populates the recycle view with user data, names, statuses, and profile images.
    - Groups Fragment: This fragment is used for displaying the groups fetched from the database, populating the List view with group names.
    - Requests Fragment: Manages the chat requests, provides an option to accept or reject friend requests. Utilizes real-time firebase database as well and displays information using recycle view.

Chat Activity:

- Manages a chat interface between users, and multimedia files utilizing the firebase real-time database, and firestore storing methods. Handles all the chat functionalities, such as sending texts, receiving texts, uses recycle view to show chats. Also shows when the user was online.

Find Friends Activity:

- Uses recycle view to display a list of users available to fetch for sending friend requests or remove existing friends.

Group Chat Activity:

- Helps with sending messages within a specified group chat. Also allows users to send and receive text in real-time utilizing firebase real-time database. Utilizes scroll view to show text messages with custom UI layout.

Image Viewer Activity:

- Simple interface to allow users to view images.

Login Activity:

- Handles user authentication using Firebase, allows only Brock email users to login. On successful login, users are directed to the landing page. If not successful, a Toast message appears on screen.

Profile Activity:

- Handles user profiles data, such as username, status of the user. Handles sending and accepting chat requests between users and interactions like sending messages, accepting requests, and managing contacts.

Register Activity:

- Allows new users to sign up with our app, helps with creating new accounts. The app restricts outside emails other than brock emails. On successful registration users are leads to the landing page.

Settings Activity:

- In here, we handle the code for updating the settings information such as username, status and profile pictures. After changes are made, they are then redirected to main activity

Main Activity:

- This is basically the landing page, redirecting users to different Intents, such as, chat activity, group activity, logouts, and others. Additionally, the user's online status is managed, they are directed to login page if the user is not logged in.

App Descriptions:

Android API: Minimum SDK: 23, Target SDK: 34, Compile SDK: 34

Android Emulator Devices: Nexus 4 API 23, **Pixel** 5 API 23. These are devices we checked our app.

Descriptions:

- The app is pretty much UI friendly
- The app starts with a Login Screen.
- If not, the user has a login, you can register with your email.
- A registration pages
- Upon new registration a new page is shown to setup username, and Status.

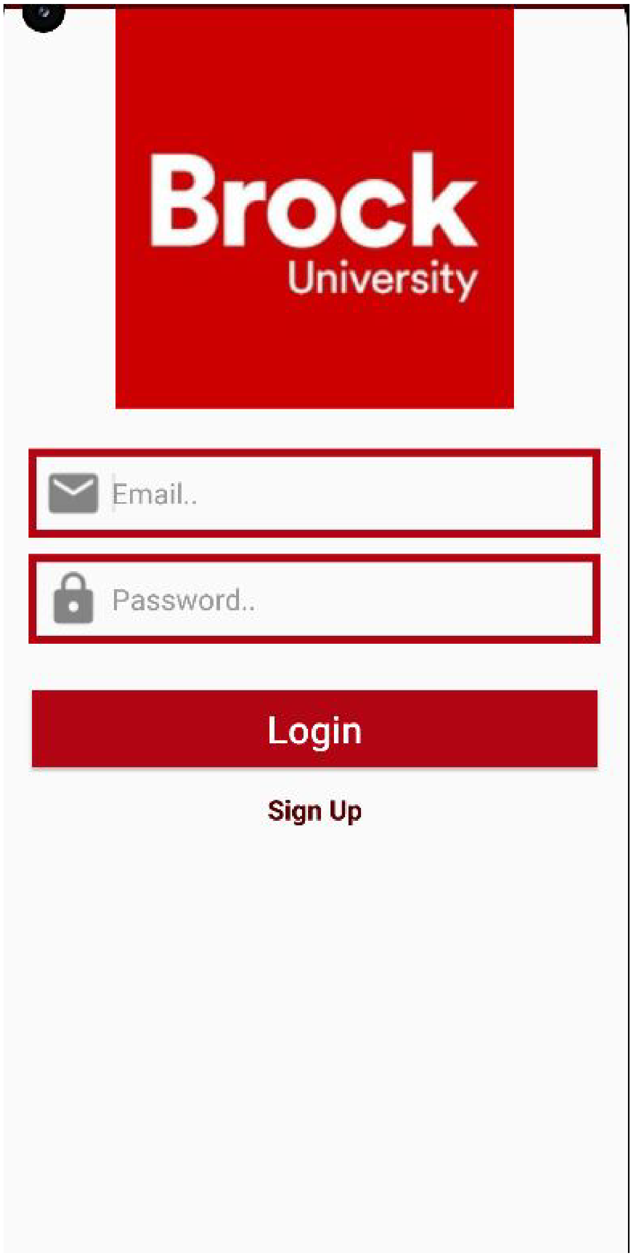
- After that we have a landing page, which has a recycle view, with 4 tabs
- CHATS, GROUPS, CONTACTS, REQUESTS Tab.
- There is a three-dot menu button on the UI, showing more functionality such as Find Friends, Create Group, Settings, Logout.
- Click on Find Friends to add existing people. Find friends adds any registered user. More like a directory.
- From Find Friends, you can send requests to users. It's up to users whether they will accept or reject.
- Upon adding friends, selection is available to send a message from the contacts menu.
- There is a group section, showing all the available public groups within the community. You can join in and out as well from there.
- Chat Functionality, you can send texts to added friends or to a group that you are interested in.
- Inside the chat room, User communicates with one another. Basic chat functionalities are as last message seen, multimedia adding, send text button, and a text box are provided. You can click on any text and delete the text for everyone, from your side, etc.
- Group Chat: anyone can send text in a group, and multimedia documents.
- You can create your own group by clicking on the menu button and Create Group. It will appear automatically in the GROUPS section.
- There is a settings option that lets you change the username, update status.
- There is also a logout option to logout of the app.
- The chat data is saved in real-time firebase, and media files are stored in fire store.
- Overall, the app is for communicating within the brock community. Reach out to friends, talk with them, create groups, have fun or study.

Limitations:

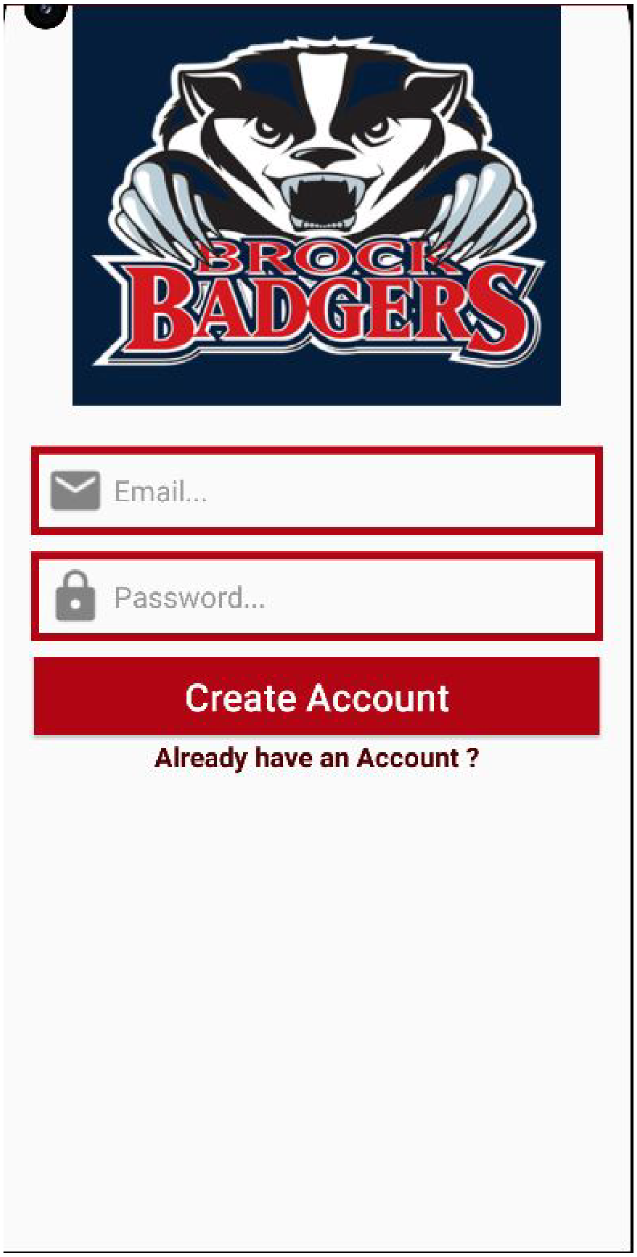
- Only brock people can sign in.
- If the email is not from Brock, sign up isn't available.
- Group chat is public, not private. Anyone can join.

UI Segments:

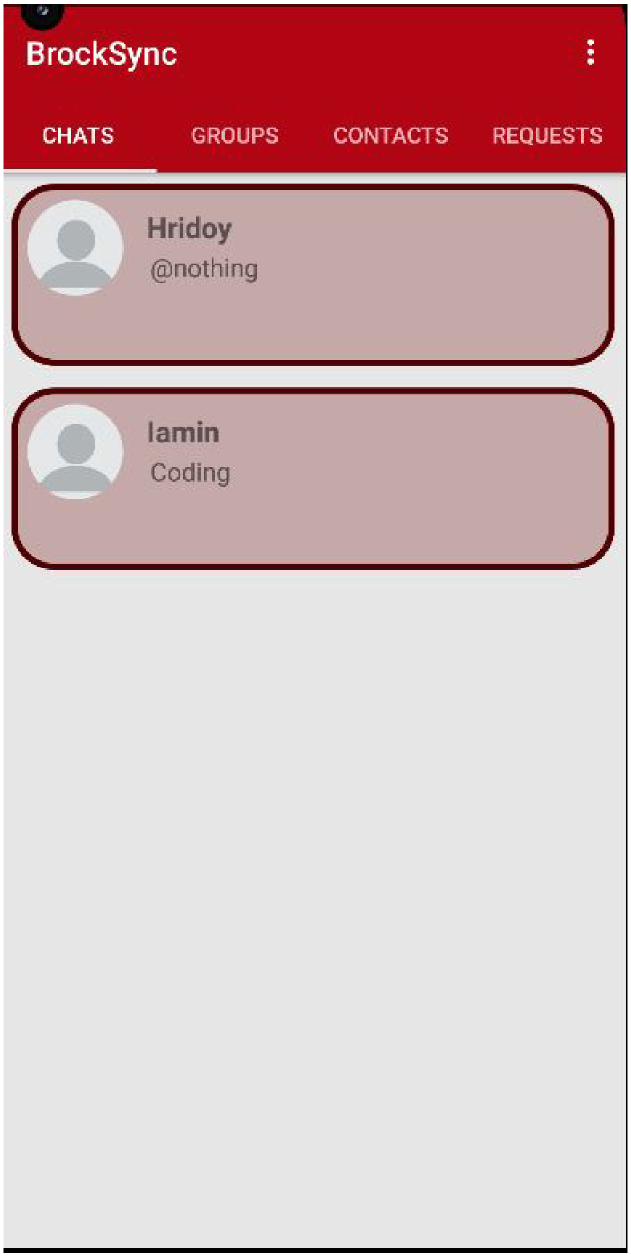
Login Screen:



Register Screen:



Landing Page:

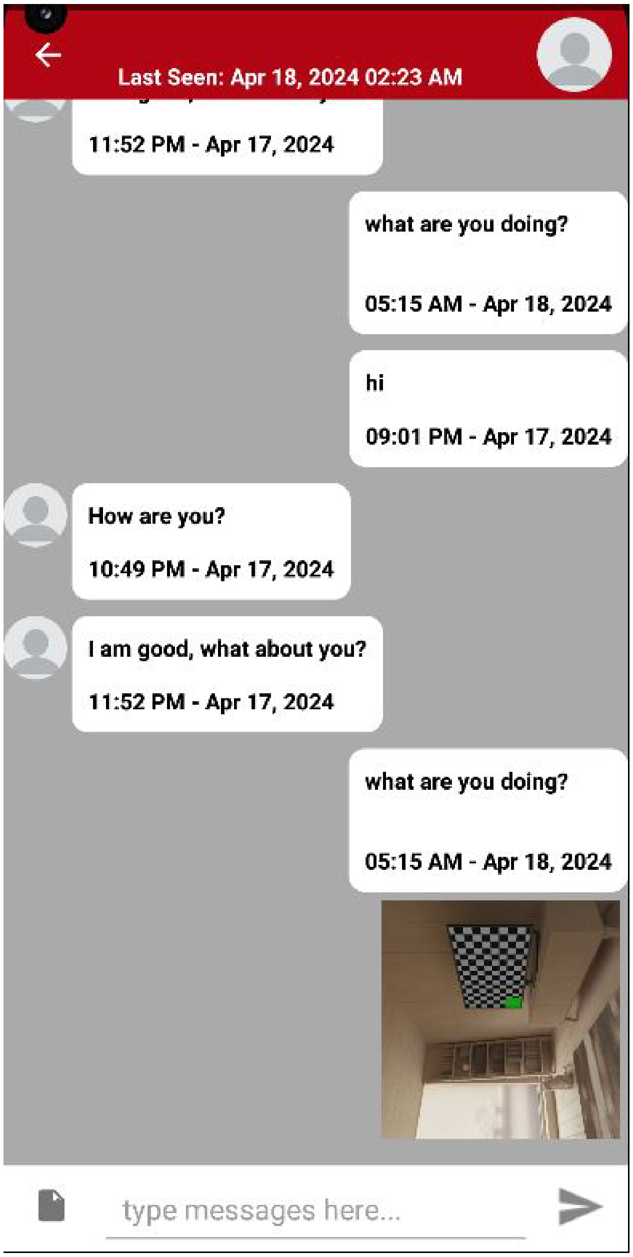


BrockSync

Hridoy
@nothing

Iamin
Coding

Chat Room View:



Last Seen: Apr 18, 2024 02:23 AM

11:52 PM - Apr 17, 2024

what are you doing?

05:15 AM - Apr 18, 2024

hi

09:01 PM - Apr 17, 2024

How are you?

10:49 PM - Apr 17, 2024

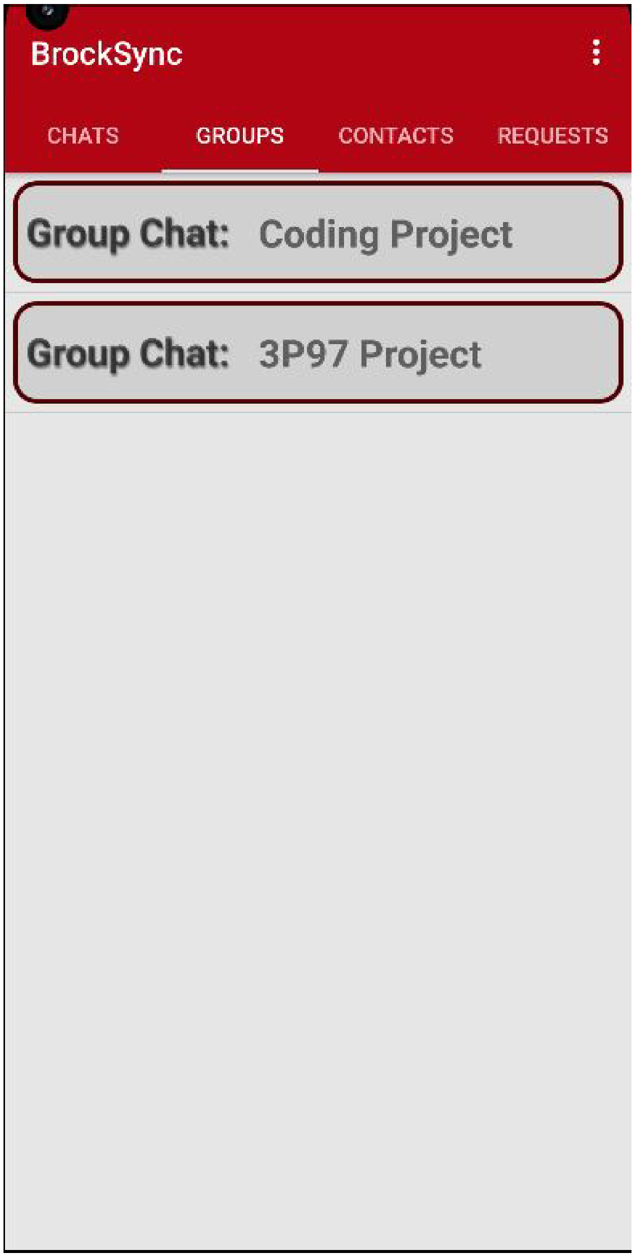
I am good, what about you?

11:52 PM - Apr 17, 2024

what are you doing?

05:15 AM - Apr 18, 2024

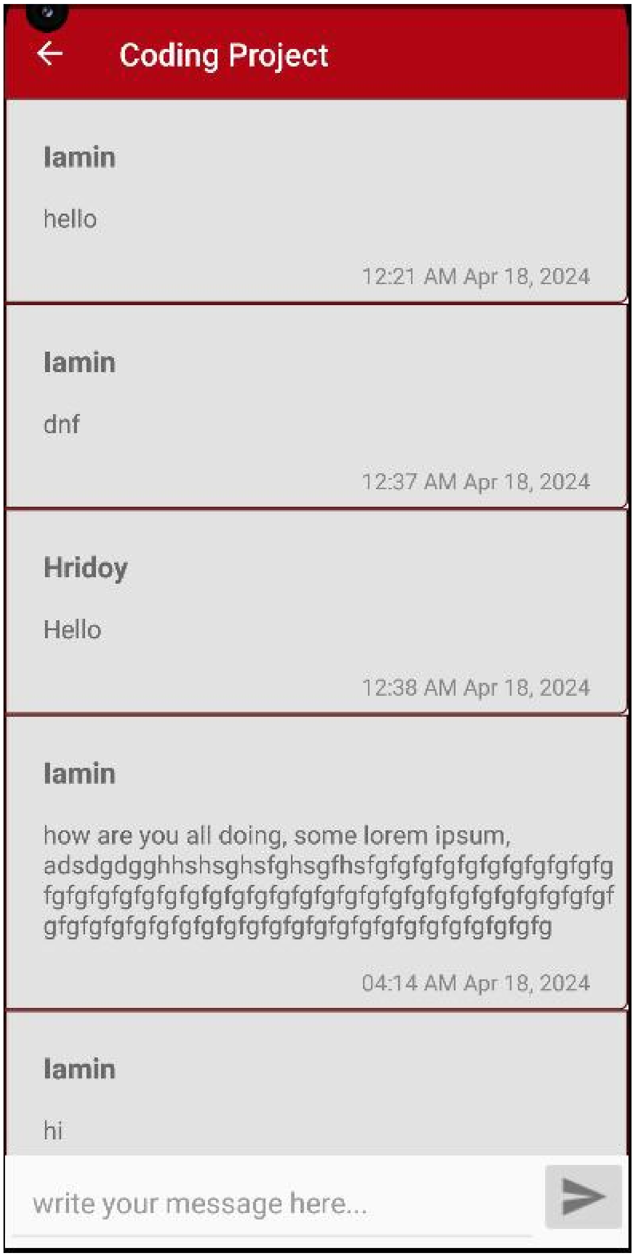
Groups:



Group Chat: Coding Project

Group Chat: 3P97 Project

Group Chat Room:



Iamin

hello

12:21 AM Apr 18, 2024

Iamin

dnf

12:37 AM Apr 18, 2024

Hridoy

Hello

12:38 AM Apr 18, 2024

Iamin

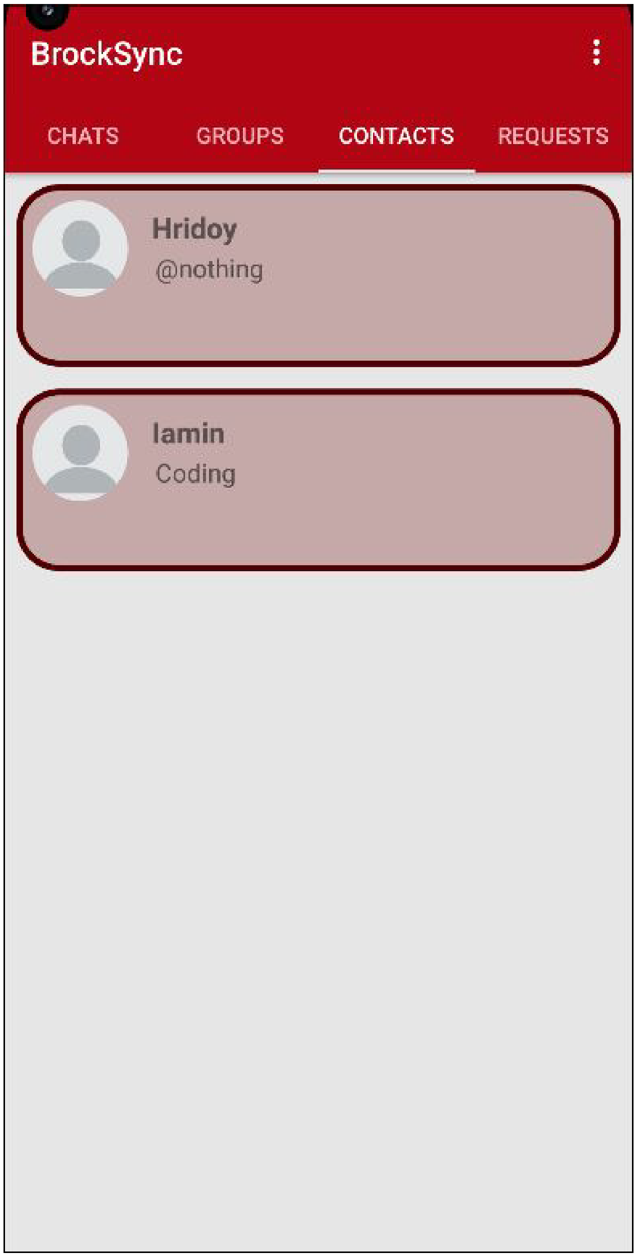
how are you all doing, some lorem ipsum, adsdgdgghhshsghsfghsgfhsfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfgfg

04:14 AM Apr 18, 2024

Iamin

hi

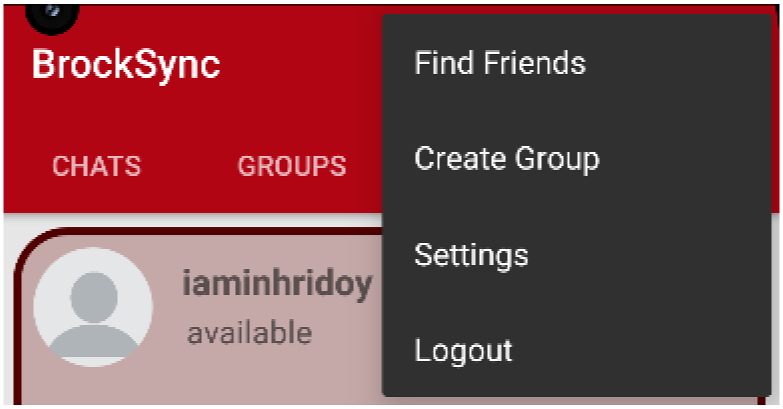
View Available Contacts:



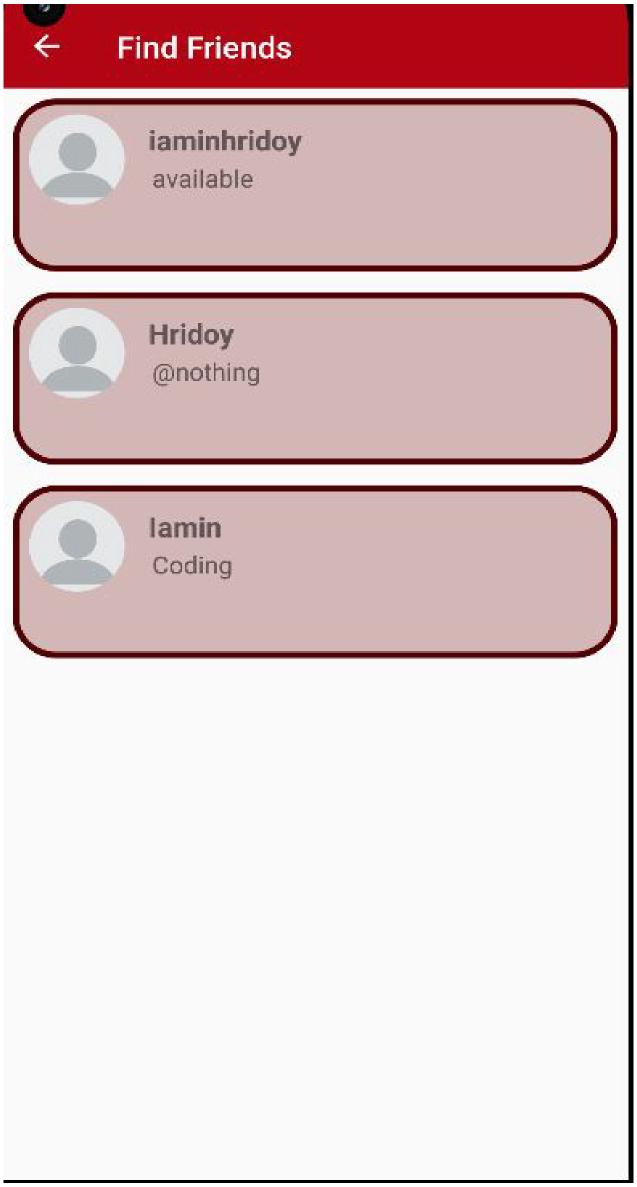
Hridoy
@nothing

Iamin
Coding

Find Friends: Send A friend Request: Click on the people you want to send a request to.



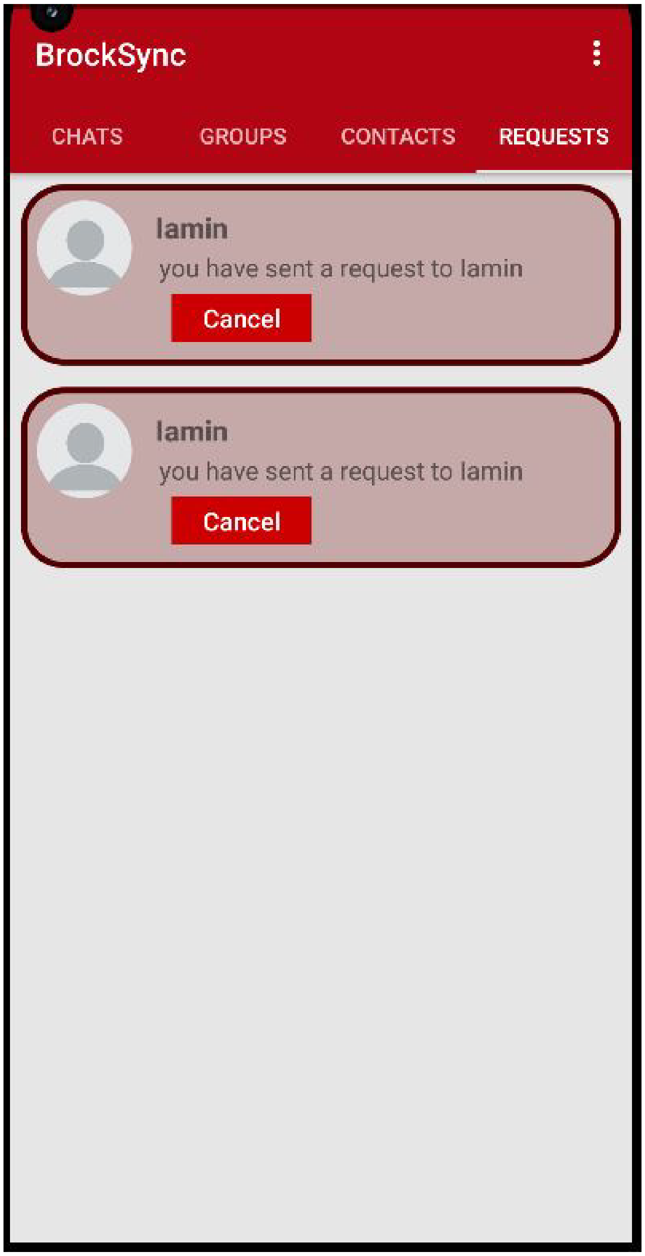
iaminhridoy
available



iaminhridoy
available

Hridoy
@nothing

Iamin
Coding



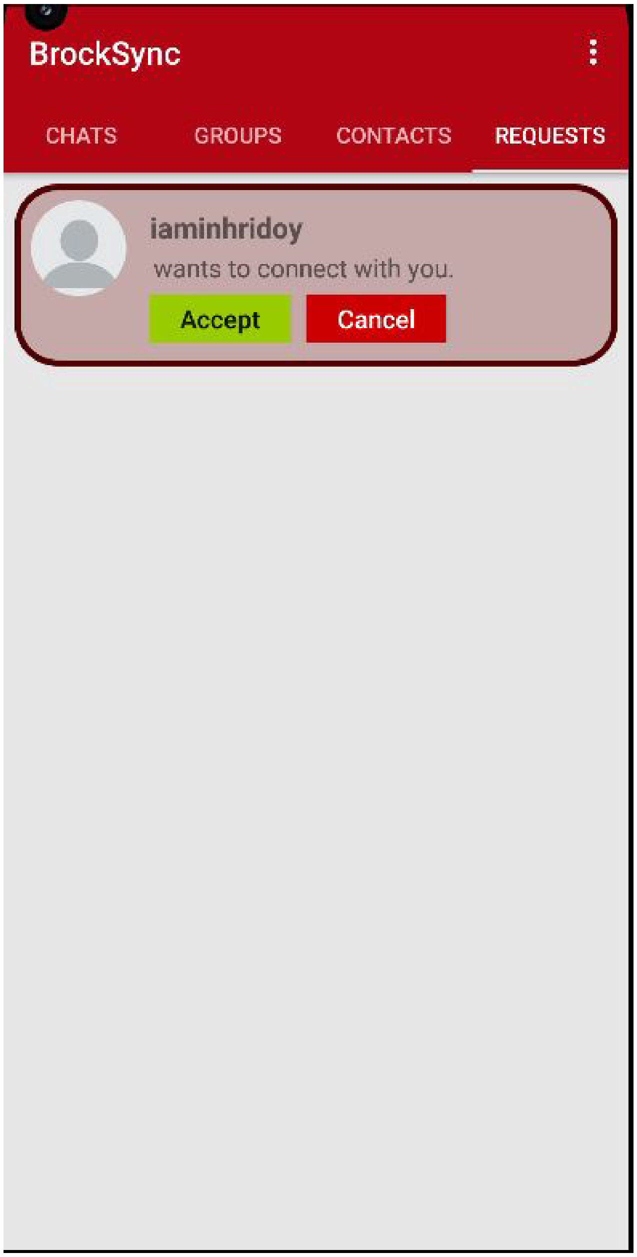
lamin

you have sent a request to lamin

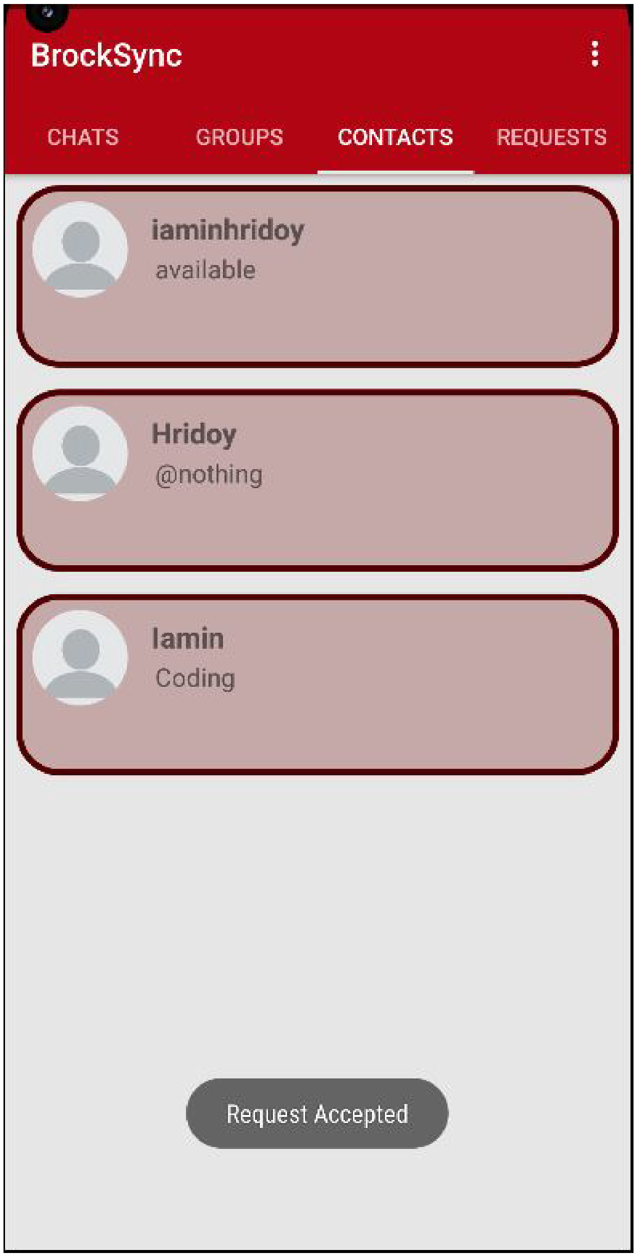
lamin

you have sent a request to lamin

Friend Request Received:



Friend Request Accepted: contacts list added the new contact.



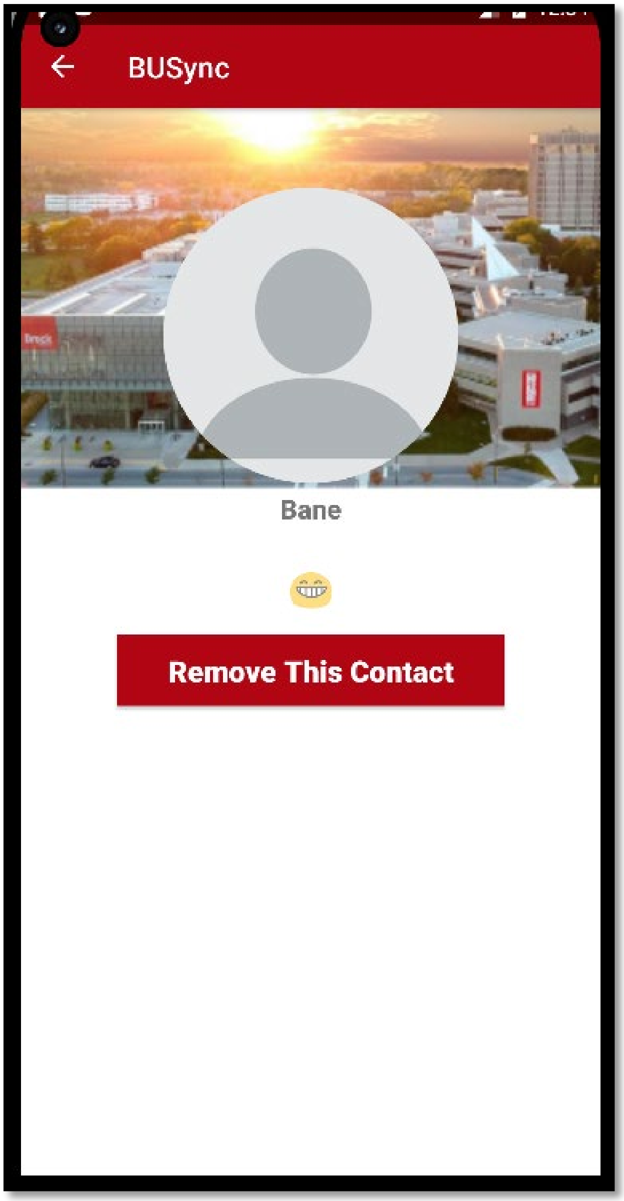
iaminhridoy
available

Hridoy
@nothing

Iamin
Coding

Request Accepted

Remove a friend from Find Friends: Click on Find Friends, click on the person you want to unfriend or add.



Bane

😁