

# ClubConnect

Rahul Daftari, Ben Delcid, Katrina Nemes, Abigail  
Sasdelli, Josh Surette

# What is the App?

- A social media app for BU clubs
- Students can scan through a database of active clubs
  - See full descriptions and contact info when tapped
  - Added to your favorites page when held down
- Club President or E-Boards can make a profile
- Provide:
  - Username
  - Name
  - Email
  - Description
- Can come back and edit later



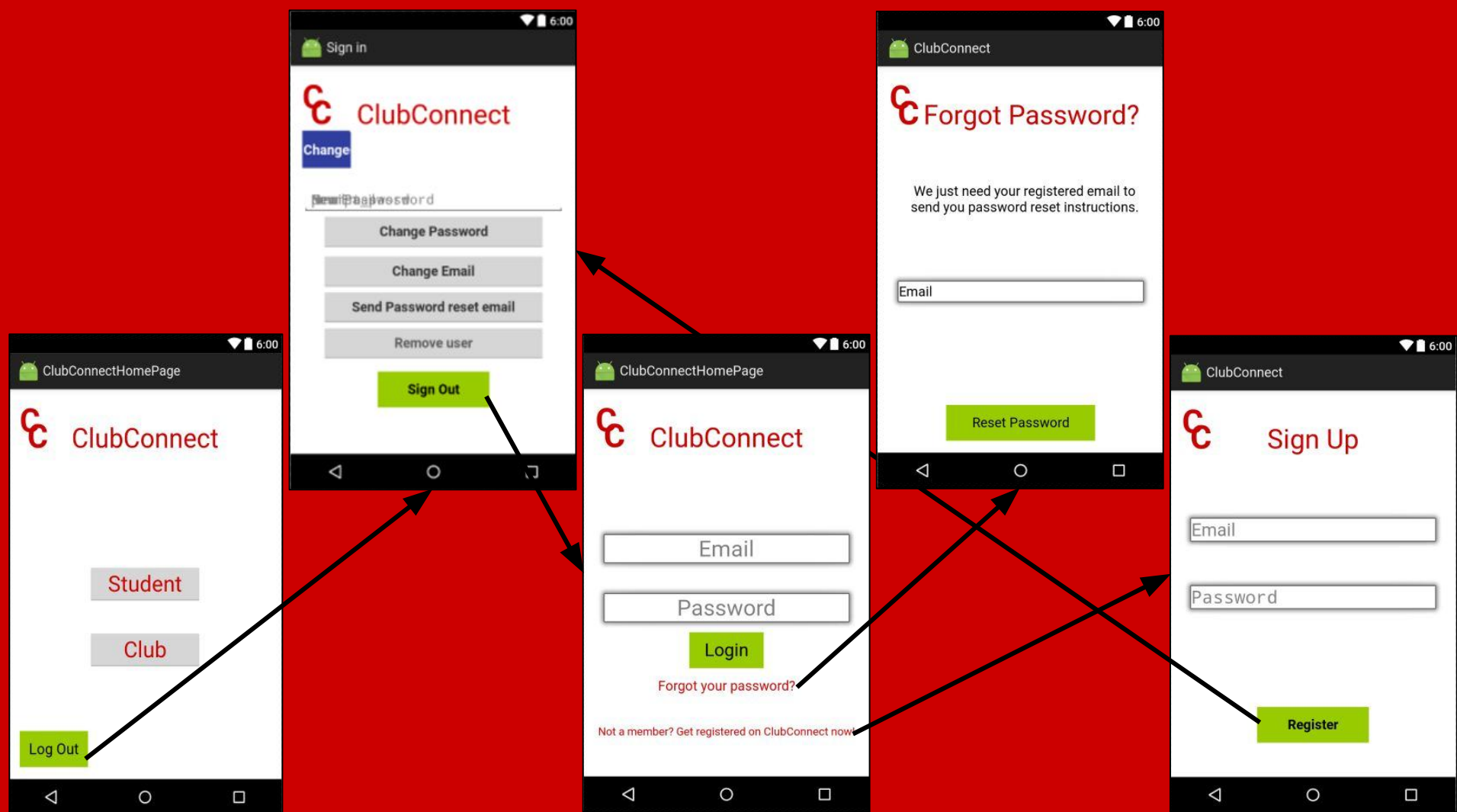
# Why Build This App?

- ClubConnect seeks to fill a pressing need for BU students trying to engage in student activities on-campus.
- OrgSync, the current student activities website is not optimized for students, and both students and club leaders have complained about the interface.
- We are creating an accessible and easy-to-use app that puts powers into the students' hands to define their own club experience.
- ClubConnect centralizes club information that, up until now, has been scattered across layers of OrgSync pages.



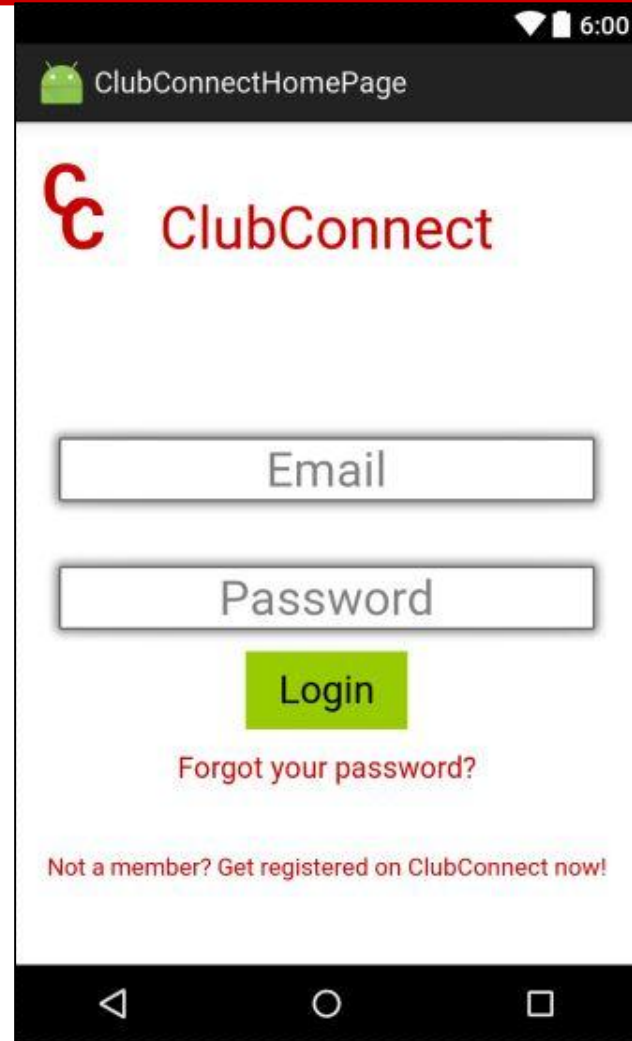


# Step 1: Authentication



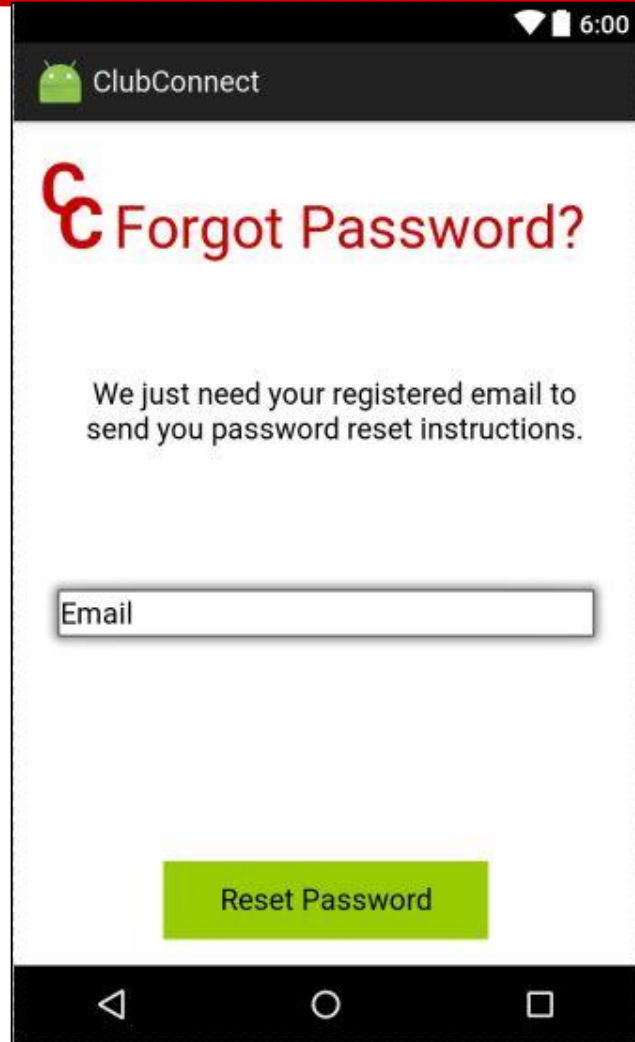
# Authentication

- Home page when you open app
- Directly links to Firebase authentication; stores user data remotely
- Provides security for the database



# Reset Password

- User inputs email, then presses “Reset Password” button
- An email with password reset instructions is sent to the user’s email



ClubConnect

## Forgot Password?

We just need your registered email to send you password reset instructions.

Reset Password

# Registration

- Students must sign up to both access ClubConnect's active club database and create/edit new clubs
- Enter email and password to register
- Password must be six characters long
- Email and password are remotely stored onto Firebase

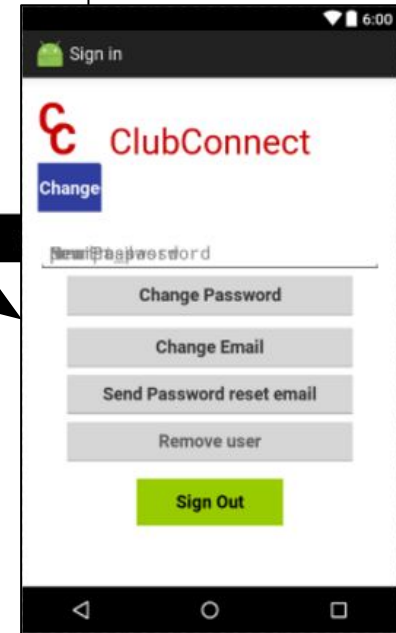
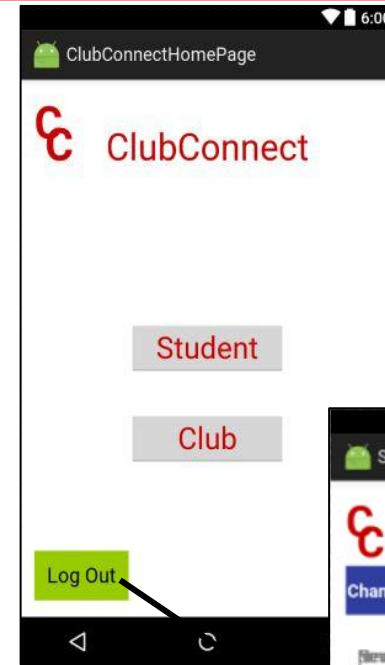


The image shows a mobile application interface for ClubConnect. At the top, there is a black status bar with a white Wi-Fi icon, a battery icon, and the time 6:00. Below this is a dark grey header bar with the ClubConnect logo (a green Android robot) and the text "ClubConnect". The main content area is white and features a red "CC" logo on the left and the text "Sign Up" in red on the right. Below the "Sign Up" text are two input fields: "Email" and "Password". At the bottom of the form is a green button with the text "Register". The bottom of the screen shows a black navigation bar with three white icons: a back arrow, a circle, and a square.



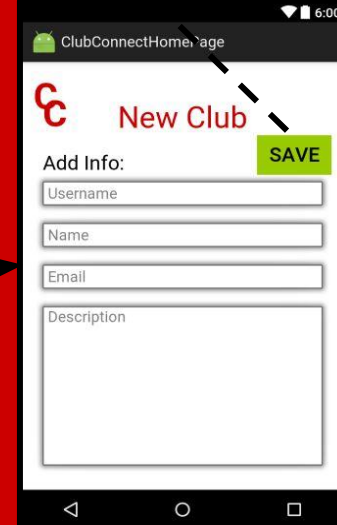
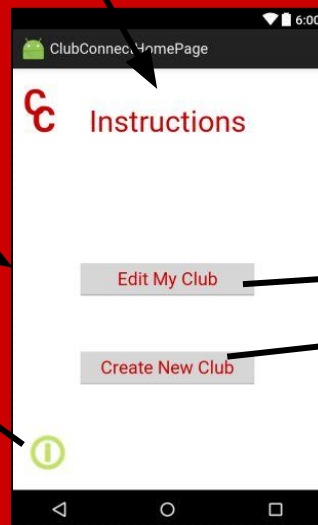
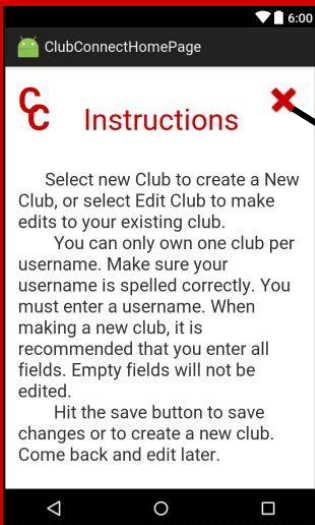
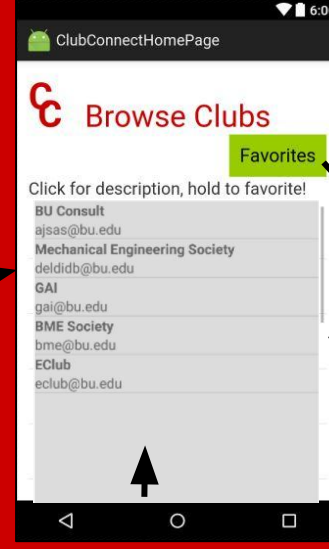
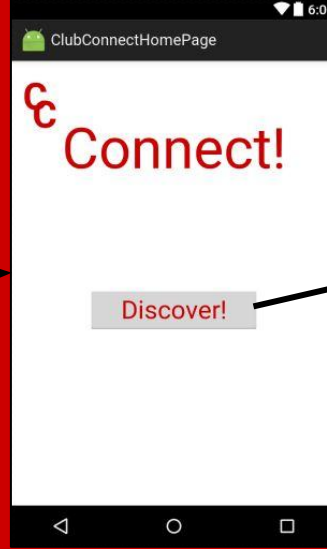
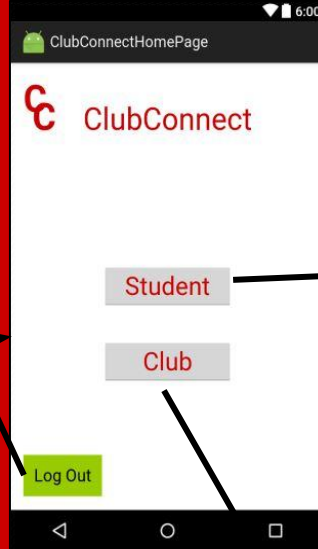
# Other User Functions

- Press “Log Out” button to access other user functions
  - Change Password
  - Change Email
  - Send Password Reset Email
  - Remove User
  - Sign Out
- Enables users to reset/alter their information in firebase



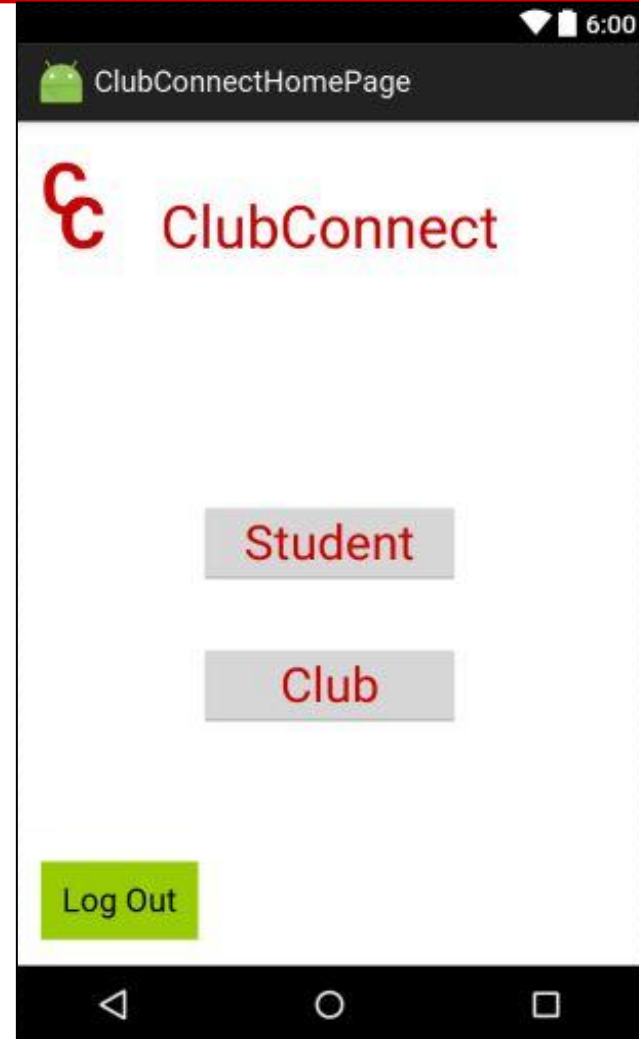


## Step 2: Setting Up Clubs



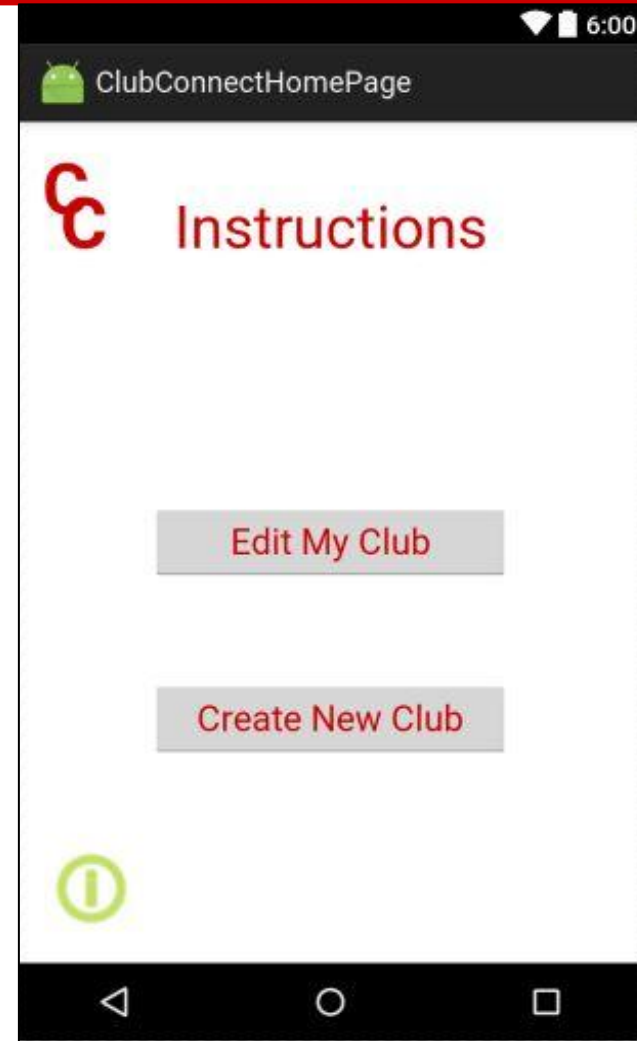
# Initial Screen: Who Is the User?

- Users select buttons that indicate their role
- Each button leads to a different pathway
  - Students will be able to search and interact with database
  - Club tab is for eboard members to create and edit clubs
- Exit via logout button
  - Allows you to change info if necessary



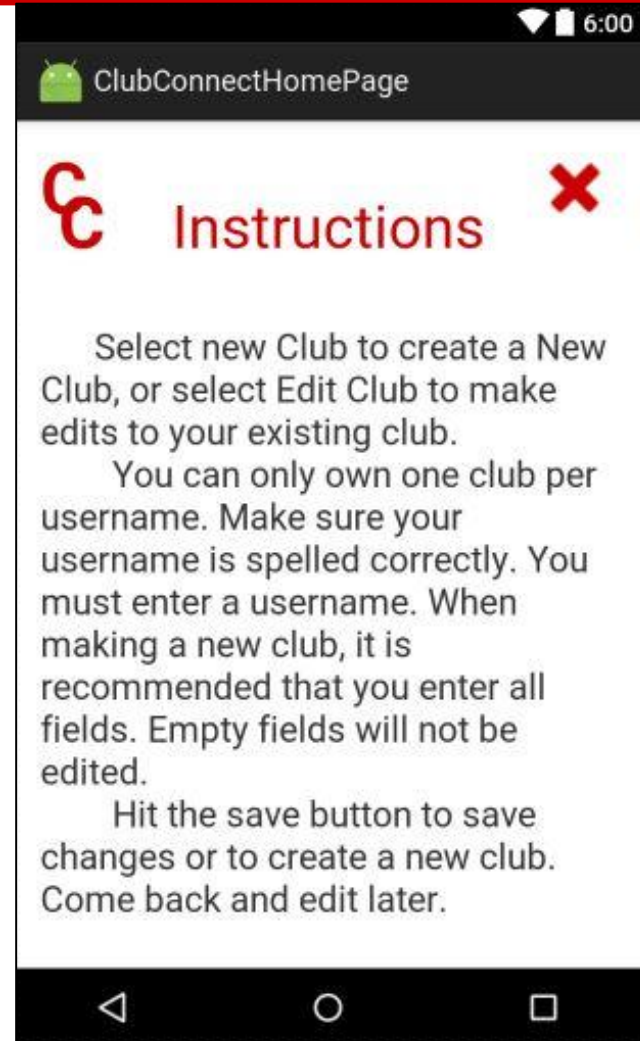
# First Step for Club Users

- Simple Buttons
- Select to go to different screens:
  - Edit existing club; Security added so we only update filled in areas
  - Create new club; each club linked to unique username so you can only own one club
- Info button directs to new page with instructions



# “How To” for Club Creation

- Provides instructions for creating or editing clubs



# New Club Creation

- “Hint” fields in TextEdits to guide users
- All inputs linked to username
- One club per username
- All data written to Firebase remotely and updated in realtime (see next!)



The screenshot shows the 'ClubConnectHomePage' of an Android application. At the top, there's a status bar with a Wi-Fi icon, a battery icon, and the time '6:00'. Below the status bar, the app's title 'ClubConnectHomePage' is displayed next to a green Android robot icon. The main content area features a red 'CC' logo and the title 'New Club' in red. To the right of the title is a green 'SAVE' button. Below the title, the text 'Add Info:' is followed by four input fields: 'Username', 'Name', 'Email', and 'Description'. The 'Description' field is a larger text area. At the bottom of the screen, there are three navigation icons: a back arrow, a circle, and a square.

ClubConnectHomePage

CC New Club

SAVE

Add Info:

Username

Name

Email

Description

# Firestore for New Club

- Data organization to store clubs at unique username nodes
- Adds security for club edits
- Relevant info about email, name and description stored at nodes
- Updates real-time and persists offline to be sure users are able to **CONNECT ALWAYS**

fir-practice-7403d

└─ fir-practice-7403d

└─ ajsas

└─ Description: "We meet everyday about all things Densmore! #SE

└─ Email: "dds@bu.edu

└─ Name: "Douglas Densmore Societ

└─ delcidb

└─ Description: "We work to bring engineering solutions to thirc

└─ Email: "ewb-bu@bu.edu

└─ Name: "Engineers Without Border

└─ densmored

└─ Description: "Internationally Genetically Engineered Machine

└─ Email: "igem@bu.edu

└─ Name: "iGem'

└─ knemes

└─ Description: "We actually do nothing but go to parties and ha

└─ Email: "parties@bu.edu

└─ Name: "Off Campus Club





# Club Editing

- Can enter/edit same fields as with new creation
- Don't have to edit all fields
- Changed fields updated in Firebase
- Designed to give user a dynamic experience, and
- Allow them to update their club with Firebase real-time to keep users updated



The screenshot shows a mobile application interface for editing a club. At the top, the status bar displays the time as 6:00. Below the status bar, the app's title bar shows the Android icon and the text "ClubConnectHomePage". The main content area features a red "CC" logo on the left and the title "Edit Club" in red on the right. A green "SAVE" button is positioned to the right of the "Edit Info:" label. Below the label, there are four input fields: "Username", "Name", "Email", and "Description". The "Description" field is a larger text area. At the bottom of the screen, there is a navigation bar with three icons: a back arrow, a circle, and a square.

# Firestore for Edited Club

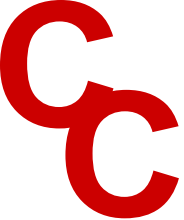
```
fir-practice-7403d
└-- fir-practice-7403d
    └-- ajsas
        ├── Description: "We meet everyday about all things Densmore! #SE
        ├── Email: "dds@bu.edu
        └-- Name: "Douglas Densmore Societ
```



```
fir-practice-7403d
└-- fir-practice-7403d
    └-- ajsas
        ├── Description: "Douglas Densmore Rules!!! #SEGFAULTS"
        ├── Email: "dds@bu.edu
        └-- Name: "Douglas Densmore Societ
```

- When fields are changed from the app, they are also edited in the Firestore
- Only updates entered fields for information security

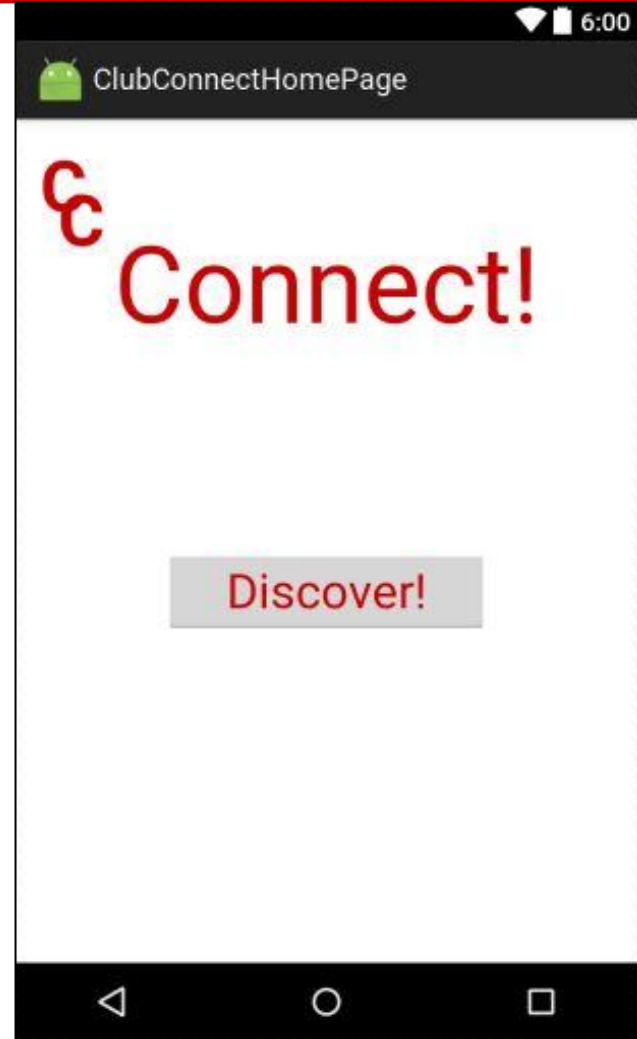




## Step 3: Acting as a Student

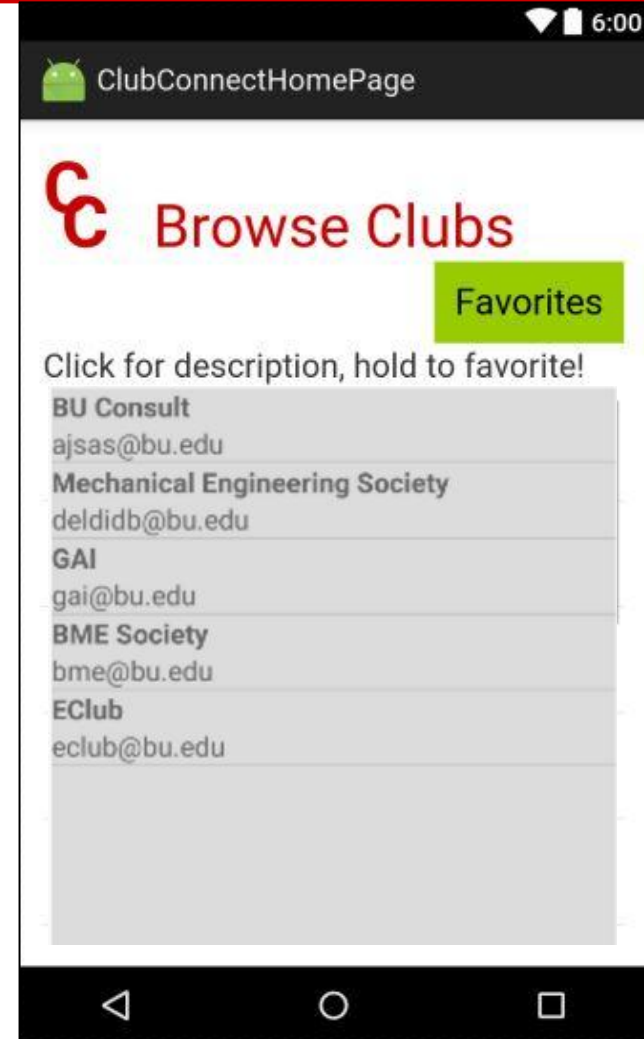
# What if the User is a Student

- Intermediate screen indicated user will enter new portion of the app to discover new clubs
- Used to inform user that they will be able to connect through this avenue of the app



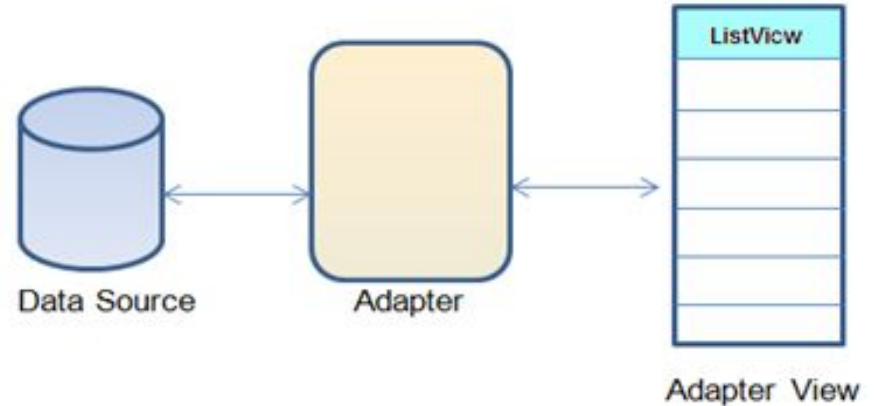
# Club Browse Interface

- Scroll through a “newsfeed”
- Find the description of all clubs listed in the app
- Clubs from Firebase are synced in realtime to the ListView for optimal search and connect feature
- Tap on list item to display description
- Hold on list item to add to favorites and locally store



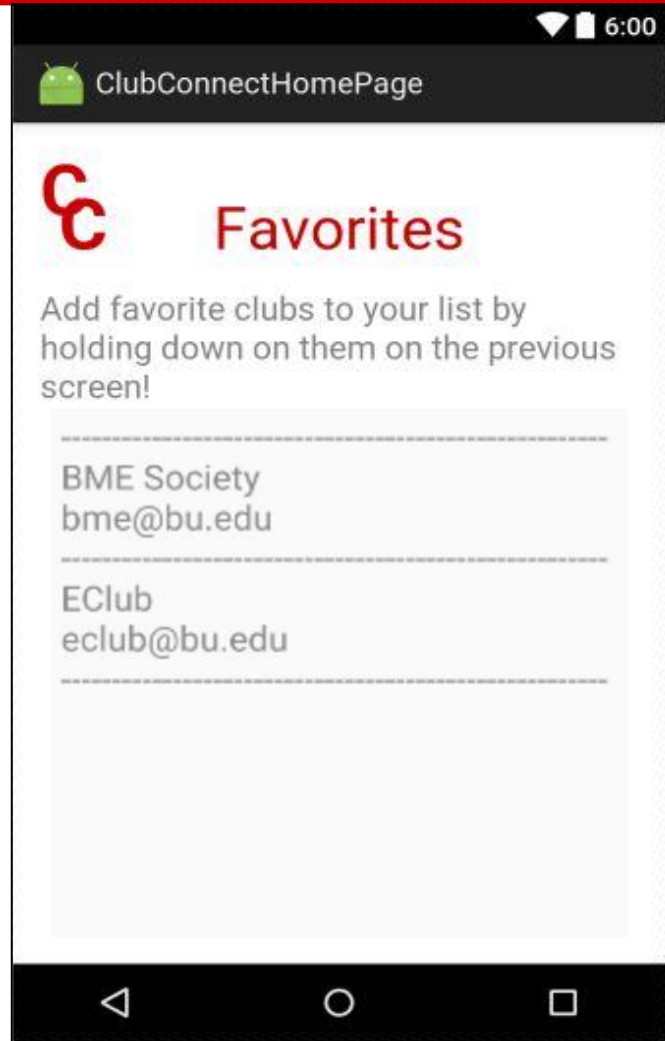
# Populating ListView

- Adapter object serves as medium
- Translates database to format in List
- Adapter stores data as a JSON, with getter functions to return data such as email, name, and description for later display



# Favorites

- Activity that displays locally stored favorites
- Shows Name and email for club
- Email hyperlinks
- Can scroll through list
- Allows for each user to view their personal unique favorites on their own phone



# Overall Data Transfer

**Club Writing/Editing → Firebase Database → Adapter → Student ListView →  
Local Favorites Storage**





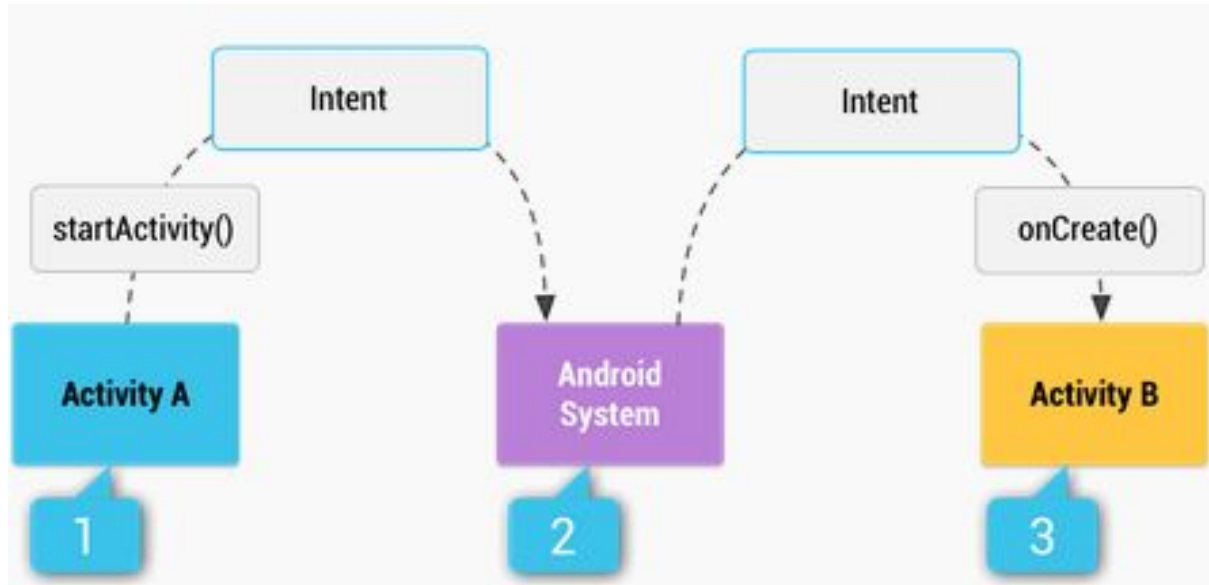
# Future Additional Functionality

- Ability to remove favorites or clubs
- Interactive Calendar view
- Keyword Search Capability by implementing SQLite storage
- Add image icons for clubs to identify clubs



# Populating Favorites Page

- When item is held, it stores the object in a an ArrayList
- Pass the ArrayList to the favorites page via intents
- Use getter functions to get and display data in the favorites page



# Technological Challenges Faced

- Firebase not good for searching features; had to scrap that feature (NoSQL)
- Combining authentication later on into project
- Consistency between screens
- Error-checking user input



# Timeline for App Development

Week 1

Come up with original idea. Define roles.

Week 2

Complete first round of UI. Layout data organization and storage. Get familiar with FireBase.

Week 3

Edit UI based on meeting plans. Link screens. Begin database.

Week 4

Complete UI. Link database to UI. Work on authentication.

Week 5

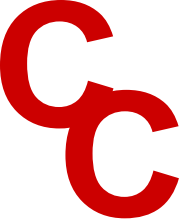
Complete authentication. Debug. Explore additional functionality if time allows.



# Roles

- Josh - Project Lead, backend
- Abbey - UI development, documentation
- Ben - Interfacing, backend
- Rahul - Backend
- Katrina - Backend, documentation
- Lots of crosstalk between roles





# Thank You!

<https://github.com/delcidb/EC327-App>

## Image References

Android Intents Diagram. Digital image. *Stackoverflow*. N.p., 1 Apr. 2014. Web.

Firebase Adapter. Digital image. *Careerride.com*. CareerRide, 2016. Web.