

# Udacity Android Nanodegree Capstone

*Sepideh Miller, September 1, 2018*

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Login Screen](#)

[Selection Activity](#)

[Profile Activity](#)

[Chat Activity](#)

[Map Activity](#)

[Proximity Search Activity](#)

[Widget](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement Firebase authentication](#)

[Task 3: Create the Selection Activity](#)

[Task 4: Create the Profile Activity](#)

[Task 5: Create the Chat Activity](#)

[Task 6: Create the Map Activity](#)

[Task 7: Implement Proximity Search \(stretch goal\)](#)

**GitHub Username:** baghaii

# Alumni Connector

## Description

Alumni Connector is a tool for alumni of the Mississippi School for Mathematics and Science to stay connected with each other. This app offers a chat feature and a proximity search so people can see other alumni nearby.

## Intended User

The alumni of the Mississippi School for Mathematics and Science

## Features

This application will be solely written in the Java programming languages. It will do the following things:

- Save address information
- Provide chat feature.
- Use an API to determine lat/lon value of a location
- Possibly also uses the GPS to determine location of a user
- Calculate who is the closest

## User Interface Mocks

This was the color palette from [Material Palette](#).

#303F9F	#C5CAE9	#3F51B5	#FFFFFF
DARK PRIMARY COLOR	LIGHT PRIMARY COLOR	PRIMARY COLOR	TEXT / ICONS
#CDDC39	#212121	#757575	#BDBDBD
ACCENT COLOR	PRIMARY TEXT	SECONDARY TEXT	DIVIDER COLOR

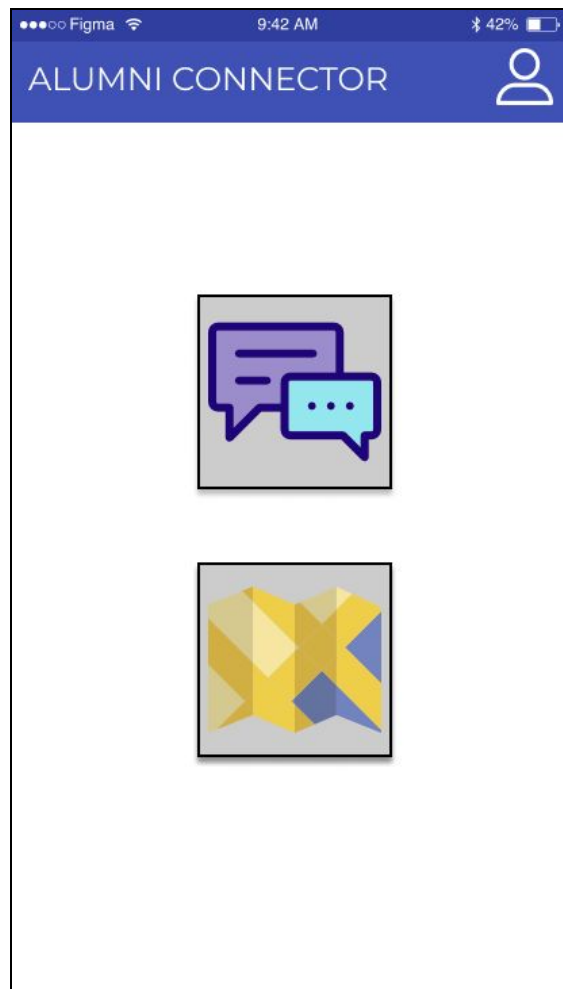
These mocks were created with [Figma](#).

### Login Screen



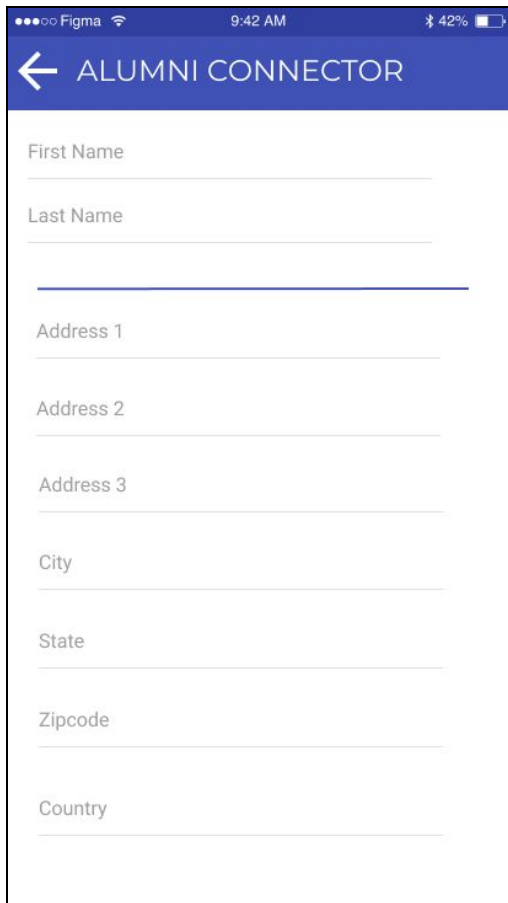
This is the login screen. I am considering using firebase to make this happen. The image is by Nhor Phai for Flaticon.

### Selection Activity



This allows people to choose whether to chat with each other or view the map. The icon on the menu bar will go to Profile Activity. These icons are also from Flaticon. The ImageView buttons and menu icons will all have contentDescriptions so they are accessible to all users.

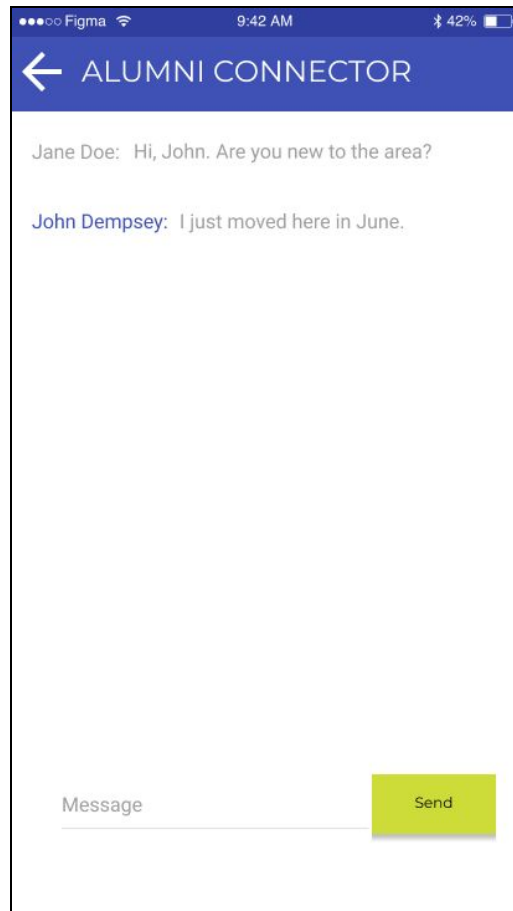
## Profile Activity



A mobile app mockup for the 'Profile Activity' screen. The status bar at the top shows 'Figma', signal strength, time '9:42 AM', and battery '42%'. The header is a dark blue bar with a white back arrow and the text 'ALUMNI CONNECTOR'. The main content area is white and contains several text input fields with labels: 'First Name', 'Last Name', 'Address 1', 'Address 2', 'Address 3', 'City', 'State', 'Zipcode', and 'Country'. A blue horizontal line is positioned below the 'Last Name' field.

The Profile Activity allows people to update their names and addresses. This should be the screen people are taken to after they log in for the first time. After the first time, they should go straight to the Selection Activity.

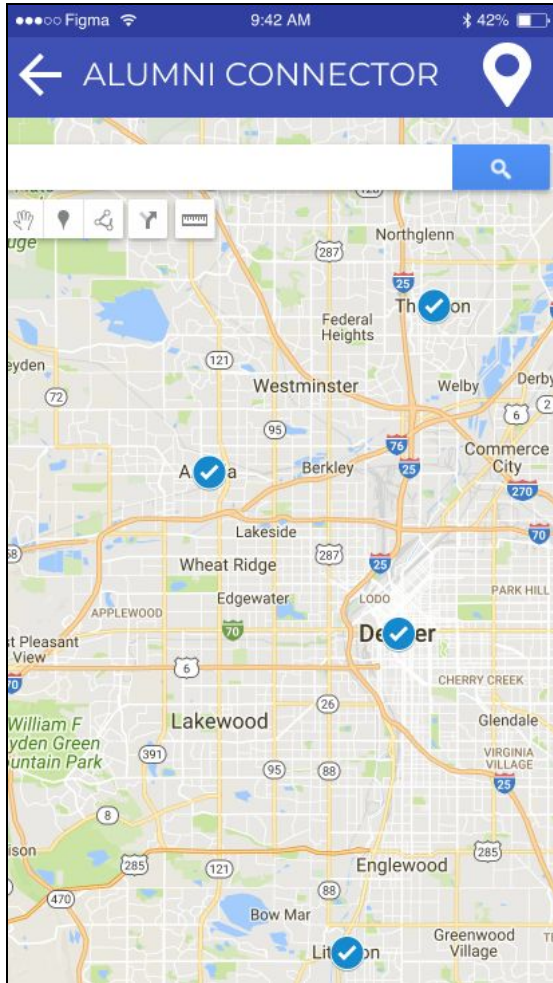
## Chat Activity



A mobile app mockup for the 'Chat Activity' screen. The status bar at the top shows 'Figma', signal strength, time '9:42 AM', and battery '42%'. The header is a dark blue bar with a white back arrow and the text 'ALUMNI CONNECTOR'. The main content area is white and displays a chat conversation. The first message is from 'Jane Doe' with the text 'Hi, John. Are you new to the area?'. The second message is from 'John Dempsey' with the text 'I just moved here in June.'. At the bottom, there is a text input field labeled 'Message' and a green 'Send' button.

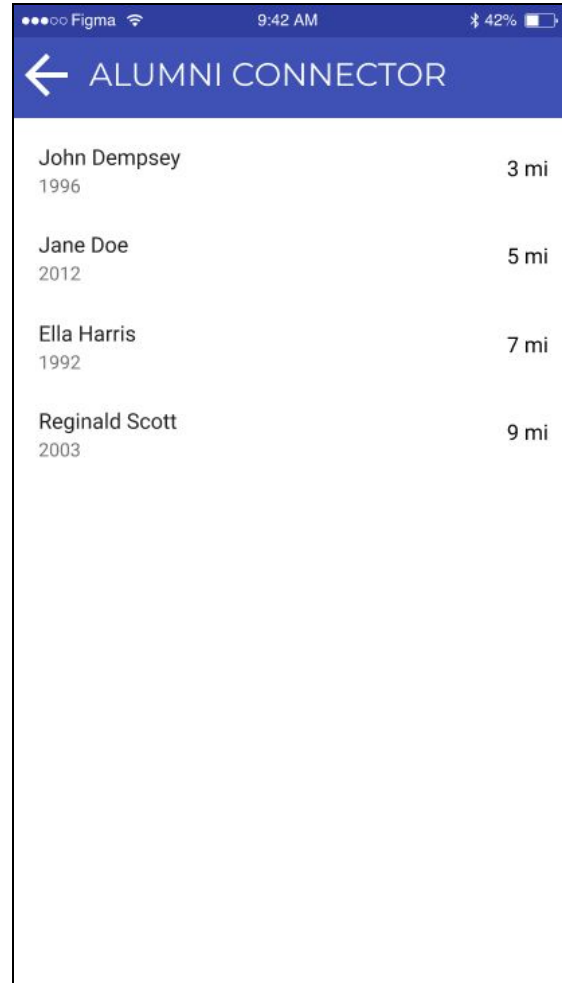
This Chat Activity allows people to see chat messages from other people and to join the conversation. The back button goes back to the Selection Activity.

## Map Activity



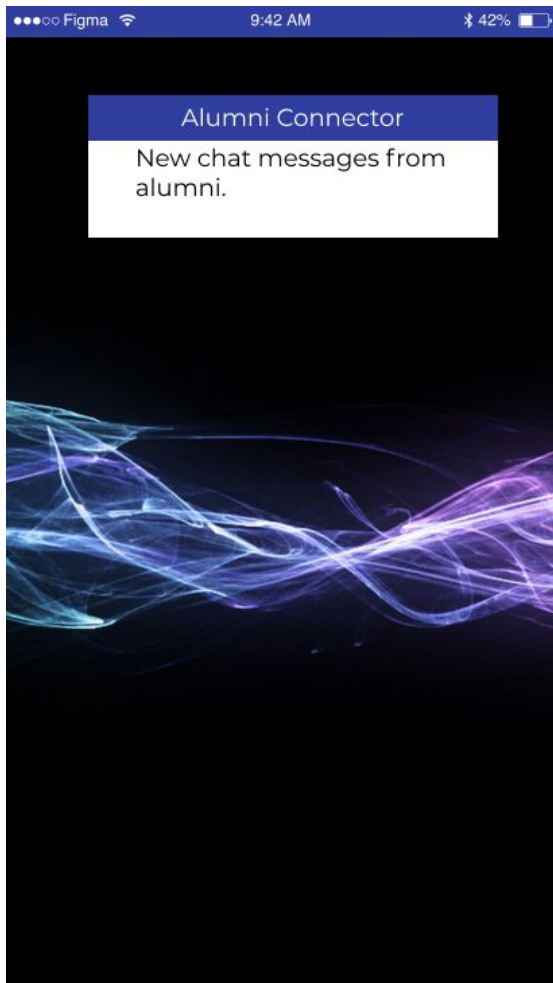
The Map Activity will allow people to see the location of others near their home address. The Location Icon on Toolbar is from Flaticon. An IntentService would be used to query the database for positions that need to be plotted on the map.

## Proximity Search Activity



The Proximity Search Activity will create a list of nearby alumni. That list will likely give you the distance to your ten nearest people. This activity may be a stretch goal. As a stretch goal, we will base this on a user's GPS location. Normally, we will base it on the location that they have stored in the database.

## Widget



The widget for this application will let alumni know if they have new chat messages from other alumni or not. Ideally something like this would probably be a notification. In this case, we are making it a widget. If we do get the proximity search activity complete, this widget can potentially have a list of the nearest alumni and how far away they are. The wallpaper used here comes from <https://wallpaper.mob.org/image/fon-46349.html>

## Key Considerations

How will your app handle data persistence?

I would use Firebase Realtime database because a number of phones would have to sync with a server to get the information that is in the database at any given time.

Describe any edge or corner cases in the UX.

Ideally the map screen will have two modes, one where it can determine "nearby" alumni based on the address that the user has put in the database and one where it can determine "nearby"

alumni based on your current location.

When the user lists nearby alumni and goes back to the map, the map may zoom out or in to clearly show some of the nearest alumni.

**Describe any libraries you'll be using and share your reasoning for including them.**

Though I have not specified the versions of the libraries below, I will use the stable release versions of the libraries.

To use the latest material design features, we will include:

`com.android.support:design`

To use the login features of firebase, we will include:

`com.firebaseui:firebase-ui`

To use a firebase database, we will use:

`com.firebaseui:firebase-ui-database`

To reduce boilerplate code, we will use:

`com.jakewharton:butterknife`

**Describe how you will implement Google Play Services or other external services.**

I plan to use [Firebase Authentication and Firebase Realtime Database](#). I will probably use `com.google.android.gms:play-services-maps` for my mapping, and I may use `com.google.android.gms:play-services-location` listed [here](#).

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

### Task 1: Project Setup

I would create a new Android Studio project and include the dependencies that I know I will need.

### Task 2: Implement Firebase authentication

- ☐ Learn how firebase authentication works.
- ☐ Set up the username and password screen.
- ☐ Store the API KEYS [reasonably](#).

### Task 3: Create the Selection Activity

- ☐ Use icons from FlatIcons
- ☐ Include a contentDescription on icons for accessibility
- ☐ Create Intent for the Profile Activity
- ☐ Create Intent for the Chat Activity
- ☐ Create Intent for the Map Activity

### Task 4: Create the Profile Activity

- ☐ Make sure we have all necessary fields
- ☐ Create the Firebase Realtime Database
- ☐ Communicate with the database correctly
- ☐ Avoid excessive database calls
- ☐ Make sure the back button works correctly

### Task 5: Create the Chat Activity

- ☐ Follow [this example](#) very closely
- ☐ Make it prettier if at all possible

### Task 6: Create the Map Activity

- ☐ Display the map on a screen
- ☐ Use either the address in the database or the user's location to center the map
- ☐ Choose a sane zoom level
- ☐ Plot points from the database onto the map
- ☐ If there is time, implement clustering for when the database is large and there are too many points



### **Task 7: Implement Proximity Search (stretch goal)**

- ❑ Use the points in the database to search for distances from our current position
- ❑ List the nearest points in a RecyclerView