The design for Meet-Meet is made up of several individual components. The first component will be a Graphical User Interface (GUI) that the user will interact with. The second component will be the Android code (written in c#) that will run on a user’s device. Finally, we will create the backend database to store user data for the long term. To help us meet these goals we will be using a program called Xamarin to compile c# code and provide a GUI builder. We will also be attempting to use the free service tier of Amazon Web Services (AWS) to host our server and database. Our application must be able to access the user’s location and use that information to match them with other users in the same area. Thus, it is essential to the idea that we are able to efficiently find a user’s location, as well as easily finding users within the same area. Furthermore, we need to find a way to group users together and allow the grouped entity to interact with other user/group profiles.

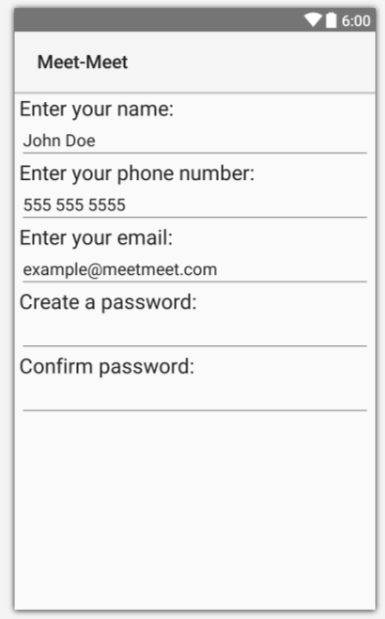
The most important aspect of Meet-Meet is the location based grouping. This will be the backbone of our application and it is imperative that it works properly. Our application will present users with the profiles of others in the area. This will allow the user to view the interests listed on the other profiles and will let them contact the other user through the app. Contact will either be done as a messaging service through the app itself, or it will merely push a notification to the other user’s device and if that user agrees, the application will distribute the phone numbers of the two users to each other (In this scenario, no information will change hands until both users have consented). Once a group has been formed, one user must be designated as the leader, this will either happen in an automatic, random way through the app, or the users may designate a leader themselves. Once a group has been formed and a leader has been designated, that new group will act in the same manner as an individual, and the application will allow the group to enter the grouping “pool” again and make more matches.

The information of a user profile will be stored in a database hosted on the AWS free tier of service. The database will also store the current location of a user which will be retrieved from the user’s Android device using the getLastLocation() method of the Google Play services API. We are unsure how the matching of locations will occur, but it will likely be determined by database queries using a specific radius around the user’s current location.

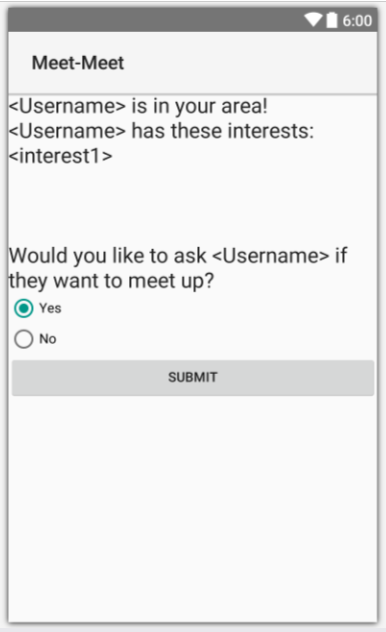
At our current time, we will be attempting to use Amazon Web Service’s (AWS) free service tier to host our server side operations. We will use this to store user information in a database which we will pull from to suit our needs. The design of the database will be basic, as none of us have much experience with formal database coding. We will store custom information about a user including their name, phone number, username/email, a random number specific ID bestowed to the user in order to be used for sorting their password hash, and finally, a hashed version of their password. In addition to custom information, we will require the user to choose one or more options from pre-determined lists in the following categories: Interests and Languages. Finally, we must also store the current user rating, which will be composed of two parts: the positive and negative votes that the user has received in the past.

C:\Users\Chris\Downloads\Untitled Diagram(1).png

Example user database table



Example user sign up gui



Example match gui