CS 495: Capstone Report 6
Tyler Allen
11/30/2014

The goals for this week were to implement a search feature and implement registration. We were also to allow users to edit their profiles when they are first created, but this was not completed this week due to a change in design.

Search has been implemented. Users can now search for all users, groups, and likes. The search function will search for any word or phrase, and will match entries containing the word or phrase. For example, searching for “Math” could match “Math Club”, as well as “The Math Club”. Implementing Search required multiple `ListAdapters` in order to support the different types. Later it may be possible to refactor this to work modularly with different types of searches, but could be complex due to the different types containing different elements. Implementing this feature required the extension of the client-side database (`ContentProvider`) to support queries for the multiple types, as well as for related matches. Selecting a User or Group will allow the user to view the profile page for the group or user. Currently, if the user selects their own profile from the search page, they will have the “edit profile” page instead of the typical “view” page. Groups will have the same option in the future when editing groups becomes available. A user’s likes will be available to edit when a good method for selecting likes is implemented, as the user must select from pre-existing likes currently. There was some difficulty in implementing the new tables in the `ContentProvider`. SQLite does not provide reliable debugging resources, causing small errors in table creation/insertion to produce the same error, regardless of the issue. The required data for these checks also required cross-checking with multiple tables to validate what information is required. Implementing multiple SQL queries for different synchronization methods did require a large amount of design to prevent the client from directly sending the query (this is exploitable), but also allow the queries flexible enough to allow for similar searches.

Registration has also been implemented. A user may enter their email, password, and full name, as well as select their gender from a list. The user may also select their birthday from a date list. Users are prompted to enter their password twice to verify the password was entered correctly. This implementation requires making a temporary client-side account until the server-side account can be verified. This account is deleted if the account already exists, and the user is given a message stating that it already exists. This required the implementation of the first client-prompted “insertion” into the server-side database. The implementation of registration was difficult due to the large amount of state that needs to be tracked to allow login to work effectively. For now, broadcast receivers are used to alert the registration page of a successful registration. This may be replaced with a more effective method using the `ContentProvider` if an efficient method is discovered. Synchronization when using broadcasts seems to be slow and unreliable.

Editing the user information is not available yet. It was decided that allowing a user skip to being logged in after registration causes too much redundant code to validate a large amount of state. In the future, after registration a user will be asked to log in as usual. Once logged in, a user will be able to edit their profile by visiting the “View Profile” button in the options on the “home” screen.

In addition, a spinning loading bar now displays when a synchronization with the server is in progress. This is to prevent state changes will a synchronization is in progress.

In summary, searching and registration work correctly. A loading bar has been added during synchronization to protect state. User and group information is mostly available when pages are visited.