# #FriendFinder

Tyler Allen, Karen Dana

September 5, 2014

## 1 Overview

The goal of this capstone is to develop a social networking application for the Android mobile platform. The application will allow users to connect with other people and plan events. Users will be able to find people by name or through mutual interests. The application will allow users to invite other users to events they are planning. The application will display friends and nearby users. This will provide students with a fun and easy-to-use social networking application.

The application will allow users to register an account with a user name and password. Account information will be stored in a database on a back end server. Users will authenticate using their user names and passwords every time they launch the application. User account authentication will be handled by the server to help ensure secure log in. User account credentials will be encrypted before they are sent from the application to the server.

The application will provide an interface for the user to create a *user profile*. User profiles will provide a way for users to identify other users that they may already know, or that they may wish to become friends with by exposing some of the personal information, such as their name and university, to other users. Every user will create a user profile when they log into the application for the first time. A user is not required to enter information for all input fields in the user profile creation interface. This information will be stored in the back-end database. Input fields for user profiles may include their name, university, course schedules, and an identifying picture of their choosing. The list of potential user input fields may be expanded in the future. Users will be free to modify/delete information from their user profile at any time. Users will have access to a search feature that allows them to find other users. The search feature will be implemented using queries to the database.

Users may include their class schedule as part of their profile information. A user's profile will display a "busy" status if they have a class when the profile is accessed. A user's schedule will only be visible to other users if they choose to make their schedule publicly accessible. A calendar Application Programming Interface (API) will be used to create the events and store the user's schedule.

The application will contain three other entities existing alongside users: *groups*, *likes*, and *events. Groups* are entities that contain zero or more other *groups*, and zero or more users. The current plan is to create *default groups* that users may join. For example, a user could join the "Western Carolina University" group if they attend Western Carolina University. Users will also be able to create custom groups that other users may join. A user's *group membership* will be visible from their profile as identifying information. A *like* is anything in which the user has an interest. For now, users will be able to choose from a list of predefined *likes. Likes* will show up on a user's profile, and users will be able to search for users who have a particular *like.* Users are permitted to create *events* within the application. They can invite other users and entire groups to attend the *event.* Every *event* will have an *event page* containing the name, date, time, length, location of the *event*, as well as a list of attending users. The current plan is to have the application display a notification on an attending user's phone when an *event* is about to start. Users will be able to see a scrolling list of information about groups and events. The search feature will be capable of searching for groups and users who have specific likes.

The application will allow users to share their locations with their friends and chosen users. Users will be able to identify nearby users within a predefined radius and treat them as a *group.* Scheduled events will contain a map that displays a map marker at the event location. The current plan is to use a Global Positioning System (GPS) service to locate users. The application will use a mapping API, such as Google Maps, to display user or event locations.

The application will be targeted at Android Version 4.0. This version was chosen because the majority (85.7%) of Android devices are running Android Version 4.0 and above according to Android Developer statistics generated on August 12$^{th}$, 2014 [3]. The current plan is to use an Structured Query Language (SQL) database as the backend database and access the database from another high level language a database connector API such as Java Database Connectivity (JDBC). The software architecture model for communication between the mobile application and the server will be based on the client-server model. A data encryption API will be used to encrypt user credentials when they are transferred from the application to the server as a way to increase user security.

## 2 Problem Statement

This project contains a number of challenging aspects. A user authentication system involves sending encrypted user credentials from the application to the server, querying the database to check the validity of the credentials, and maintaining an authenticated connection between the server and the application. An encryption API will need to be learned and applied to authenticate the user credentials. An Android application interfacing with a server will require large amounts of dynamic content to be generated by the application in order to provide current content; this can be difficult due to frequent changes and inconsistent behavior in the Android API. The application includes features that locate devices within a certain range using GPS; this can be difficult to accomplish in a timely and resource efficient manner and will require a large amount of prior research. The server will also need to sync with all authenticated users periodically to keep information about groups and events up to date without causing unnecessary web traffic and draining the battery life of devices running the application.

## 3 Requirements Specification

- The application should provide the ability to create a user profile. We will investigate using facebook authentication for this app because it is popular and provides an easy method of user registration [1].
- The application will allow a user to join *groups*, add *likes* to their profile, and broadcast/receive events based on these *likes* and *groups*. Users will be able to create *likes/groups*.
- The application will allow users to search for other users, *likes*, and *groups* available. Users will also be able to build their schedule. User schedules will allow the application to deactivate itself when the user is 'busy' according to their schedule.
- Users will be able to have a 'friends' list, a special group which allows viewing of additional profile information such as the user's class schedule. This will also allow the user to broadcast their location

to their friends. GPS is notoriously known for using large amounts of power, draining the battery in a short amount of time. We will be investigating methods for efficiently making use of the GPS transmitter as well as using other technologies in place of GPS in order to conserve battery power [2] [4].

- Users will have a scrolling list containing event and group information. The server should provide a stable, consistent database of user credentials, profiles, and other information.
- The server will be able to provide information about a user without revealing information marked private (i.e. extract 'busy' status from a user's schedule and deliver it to other users, without revealing the actual details of the schedule).
- There will be a user authentication mechanism; the user credentials will be encrypted by an encryption API.

## 4 Testing Plan

Testing will be completed as the application is developed. We will create regression/integration tests using Android TestSuite - a derivative of JUnit. An adequate server-side unit testing framework will be selected based on the final language decision; presently, there is no concrete language decision for the server while the concerns about performance, usability, and database and API availability are weighed. Testing will also include an evaluation of the user experience on Android devices of multiple form factors and screen sizes. Ideally, a large number of registered users can be simulated to test usability. The server will be located on Polaris. Test Android devices will include Samsung Galaxy Neo, an LG G2, a Google Nexus 7 (2012), and any other devices that can be procured during the course of this project. All testing devices will run Android 4.0 or higher.

## 5 Schedule of Completion

**Tuesday, September 16, 2014**

The planning stage for the capstone will be completed; this includes the layouts for the application and Unified Modelling Language (UML) diagrams for the entire 'Friend Finder' project.

**Tuesday, September 30, 2014**

The skeleton of the application will be complete. The skeleton contains the page layouts, including the graphics, styling, and non event-driven buttons. Options to enter text and edit user profiles and events will have been completed. User input validation will be completed as well. The normalized database schema will be completed, setting the layout for the information stored in the database by the application.

**Tuesday, October 21, 2014**

User authentication and encryption will be completed. The database schema will be applied to the database and the database will be set up for queries and modification. There will be several different default *likes* that users can choose to add to their profile, which will then be updated in the database.

**Tuesday, November 4, 2014**

The information gathered from user profiles that is stored in the database can now be queried for *likes*; a user can choose a *like*, and the database will return a list of the current users who also have that same interest. Users will also be able to create groups of different people that will be saved in the back-end database. Default groups will have been completed.

**Tuesday, November 18, 2014**

Users will now be able to create events and invite their friends or groups to the event. The user profile creation interface will be complete. When the user is in class, it will display that they are busy on the user's profile.

**Tuesday, December 2, 2014**

When a user is invited to an event, the application will generate a notification for the user. If the user has entered their class schedule and is currently in class, their notifications will be silenced until they are free; if they have their options set to not receive notifications, no notification will be created. We will also have completed a map feature which shows nearby users who have the hidden mode disabled in their settings; hidden mode allows a user to accept invites and create events without being visible on the map to other users.

# References

[1] Konstantinos Mourtzoukos and Ioannis T. Christou. Experiences running a prototype location-aware mobile social networking system. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '10, pages 362–365, New York, NY, USA, 2010. ACM.

[2] Wolfgang Narzt. A generic context-based architecture for energy-efficient localization on mobile devices. In *Proceedings of International Conference on Advances in Mobile Computing &#38; Multimedia*, MoMM '13, pages 33:33–33:42, New York, NY, USA, 2013. ACM.

[3] Android developers dashboards. Web, 2014. `https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net` Accessed: 2014-09-05.

[4] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 315–330, New York, NY, USA, 2010. ACM.