CMPE 277 Spring 2017

Term Project

*Last updated: 5/14/2017*

In this project you build a native mobile app (either Android or iOS ) with full-fledged social features. You app needs to interact with a server hosted in the cloud. If you choose Android, you must use Java; for iOS, you must use Objective-C and/or Swift.

**Project Features**

1. Sign up and authentication
   1. A user must be able to sign up with any email account he owns, and set password specific to the app.
   2. Email verification.
      1. The server must send a verification code (an integer of four digits) to the user through his email.
      2. The app must provide UI for the user to type in the verification code. Before the verification takes place, the user cannot use functional features of the app.
      3. Profile management
2. Profiles
   1. Every registered use has a profile.
   2. A profile needs to contain at least the following fields
      1. Screen name (must be unique)
      2. Email
      3. Profile file picture
      4. Location  (city level address is sufficient)
      5. Profession (Software Engineer, Student, etc.)
      6. About me (text to describe the user)
      7. Interests (travel, sports, camping, etc.)
   3. A user must be able to update his profile at any time.
   4. A user must be able to browse all profiles that are public or belong to his friends.
3. Settings
   1. Visibility: Profile can be set to one of the following two visibility levels
      1. FriendsOnly: besides myself, my profile is visible and searchable to all my friends. (Default)
      2. Public: visible and browsable by everyone.
   2. Notification option
      1. Whether to receive instant email notification when a user is not online and one of his friends or a person he has been following made a new post and sent him a private message.
4. Friends
   1. A user can add other users as friends, one at a time
      1. When browsing users with public visibility, one can add a selected user as a friend, which sends an invitation to the user. The friendship is not fully established or functioning until the other user accepts my invitation.
      2. One add any user as a friend by his email
         1. If the user with the given email does not exist,  the friend requester cannot tell, except that his request will show as not accepted yet. An invitation to join the app, however, is sent to the user.
      3. One must be able to see all his pending friend requests that are not accepted by others yet.
      4. One must also be able to see invitations sent to him and not accepted yet.
5. Post management
   1. A user must be able to create posts (similar to Twitter’s tweets, but with a higher size limit).
   2. Besides text, a post can also contain pictures.
   3. All Alice’s friends should be able to see all Alice’s posts.
   4. If a user’s profile is public, all his posts are also public, visible to everybody.
6. Follow
   1. One can follow any user whose profile is public.
   2. When you follow another user, any post the other makes becomes part of your your timeline.
7. Timeline
   1. Timeline provides a view for posts ordered by reverse chronological order, with the newest shows up first.
   2. Timeline should include the following posts
      1. All the current user’s posts
      2. All his friends’ posts
      3. All the posts from the users he has been following
   3. Timeline visibility
      1. Only you can see your own timeline.
      2. Your friends (or followers, if you profile is public) cannot see your full timeline, but can see all the posts you posted in reverse chronological order, rendered in a format similar to your timeline.
8. Private messaging

           Private messages are similar to LinkedIn’s InMail, an email system for members.

* 1. You can send private messages to any of your friend or anybody whose profile is public.
  2. Format of private message is similar to mail, with one single recipient, a subject, and message body.
  3. When you browse profiles and pick one to send a private message, the screen name should be automatically populated; you can also start sending a private message by typing in a screen name.
  4. When you receive a private message and are not online at this moment, you get a notification email with the content included, but the email address of the sender is hidden, as this is private messaging, not direct email.
  5. Within your app, you must be able to browse all your private messages, view any you select, and delete those you do not want to keep.

1. [Bonus feature] Push notification
   1. Wherever email notification applies, you can provide push notification as well.
   2. You want to give the user to configure whether to use email notification, or push notification, or both, in his setting page.
2. [Bonus feature] Albums
   1. You can create/update/delete named albums.
   2. Besides a name, each album has its own visibility, either private, or visible to friends.
      1. The default visibility for an album is friends
   3. In each album, you provide a scrollable view of thumbnail pictures.
   4. When a thumbnail is pressed, the full picture is shown in a full screen.
      1. You can swipe left/right to browse through (back and forth) pictures in the album, providing a carousel view.
   5. You are able to add/delete pictures within an album.

**Additional Requirements**

* This is a group assignment, even though you can be on your own group.
* Please add proper Java documents and **unit tests**.

**Grouping**

This project is group based, with group size up to four people. You need to get **explicit permission** from the TA or instructor if your group size is less than 3.

**Source Code Management**

You are encouraged to use a Source Control Management (SCM) system to manage your team’s source code. This can be a private Bitbucket repository, your local git or SVN repository, or any repository that is not publicly accessible to everyone. If you do not use SCM, you must clearly document the code changes in HISTORY.TXT, covering who made what changes at what time.

**Cheating Policy**

Your app must be built by yourself, and cannot be based on the code base of any existing app. If you used any code not written by yourself, it must be clearly documented in your README.TXT file, unless it is part of publicly available libraries. If your app is already used to serve the requirements of any other class, it will not be accepted by this class. In the case **any** form of cheating is confirmed, your will get an **F** grade for this class.

**Deliverables and Grading**

The project is worth 25 points in total with 2 bonus points possible. The actual *due dates* of the deliverables will be specified in *Canvas*.

**Project Presentation and Demo (5 points)**

To be presented in class.

* The presentation should cover introduction, high level design, and major features with screenshots. Page limit: 10;  time limit: 3 minutes. Must be Google Slides so that they can be shared with the instructor and TA ahead of time and presented from the same computer.
* You must also do a live demo. The guideline for how to do demos is to be added. Time limit: 4 minutes.
* Grading will be based on successfulness of the demo, the content and clarity of the slides, and the delivery of the presentation

The presentation slides must be submitted through Canvas as a PDF file.

**Project Report (4 points)**

The report needs to cover the following topics.

1. Motivation and introduction of your app
2. High level and component level design
3. Technology choices
4. Description of features with final **screenshots**
5. Testing plan executed and results
6. Lessons learned and possible future work

You are recommended **not to** exceed 18 pages, but you will **not be penalized** just because the report is too long or too short, as long as the level of coverage for the required topics is reasonable and clear. The report must be submitted through Canvas as a PDF file.

**Project App (16 points + 2 bonus points)**

Note: the instruction for submission is still *subject to change*.

1. You must submit all your source code / resource files through Canvas
2. You must submit your app binary (apk or ipa)
3. Features richness, complexity, and correctness are worth 12 points
4. User interface and user experience are worth 4 points
5. Include README.TXT file with the project name, the names, email IDs, and students IDs of the members, and build instructions
6. If you have a youtube video to demo your app (cannot exceed 5 minutes), please include the URL close to the top of README.TXT.
7. You may be asked by the TA to demonstrate a certain features live.
8. Include HISTORY.TXT, which is either the original log of the last 100 commits if you use SCM (e.g., if you use git, this can be generated by *git log -100*), or the full manually documented logs.