

## Final Project:

Due: End of the Fall semester, 2019; the exact date will be announced later

In this project, the final one for this semester, you and your partner (you will work in pairs, again) will create a larger, more involved Android application. You will utilize most of what we have learned this semester, and perhaps *impress everyone* by considerably extending your Android abilities to fully realize your selected project.

You and your partner should think about an interesting app to create and develop. I will include some very general requirements for the project, laying down some basic expectations, and it will be up to you and your partner to select and implement an interesting app. To start, here are some ideas for the final project:

1. **Ride sharing app.** Some students at UGA do not have cars on campus, but still need to go shopping or run errands, which is not easy, or perhaps not even possible by riding a bus. Or, perhaps somebody does not want to drive to attend a dinner gathering at a friend's home. It would be nice to have an app to request a ride to your destination. We could assume that giving a ride to someone would earn you some ride points, and you could then convert your ride points into an actual ride that someone would give you at a later time. Membership would be free, and by signing up, you would get an initial amount of ride points to start with. Registration/Login/Logout must be necessary.
2. **Messaging app.** It would be cool to have a "private" messaging app, which you could share with a number of friends. The messaging app could provide groups, or topics/rooms for conversations. Additionally, you could incorporate a mapping service to locate your friends when interacting with them. Google's Firebase could be used to provide the basis for communication. Registration/Login/Logout must be necessary.
3. **Roommates shopping app.** If you live with a roommate, you may have to purchase items, which are shared by all roommates. It would be useful to have a common place for recording shopping needs for shared items (everyone needs paper towels ☺), and then recording shopping expenses (items bought, with dates and costs). This could be used later to compute the equally shared cost of such expenses. One of the roommates going to a store could check the current shopping list and then purchase and record some of the needed items. Google's Firebase could be used to provide the basis for this app. Registration/Login/Logout must be necessary.
4. **Voting app.** It is often the case that a small organization (e.g., UGA ACM Chapter) needs to resolve an internal issue by way of a popular vote. An app admin creates a vote on an issue, creates its description, a deadline for casting a vote, and then notifies all other users (members of the organization) that their vote on the issue is needed. Subsequently, the users cast their votes and the system records their votes, provided they are before the deadline. Finally, the app admin gets the vote result and notifies all users about the result. Google's Firebase could be used to provide the basis for this app. Registration/Login/Logout must be necessary.
5. **An app of your choice.** I will be happy to accept other app ideas, provided they include an external source of data/information and fulfill some relatively real (but modest) need. A simple *multi-user* game is permitted.

**Any app you create must be *original* and not based on your own (if you have developed any apps in the past, either for Android or iOS) or any other prior apps.**

**Furthermore, you will be required to:**

1. by 11/08/2019, submit on eLC a written project idea and have it approved,
2. by 11/22/2019, provide a written update on your progress (also on eLC), and
3. by 12/03/2019, create a short report and demo your app, likely on 12/05/2019.

## Additional requirements

- Each team should have 2 students, both of whom are registered in CSCI 4060, or both in CSCI 6060 or 4060 Honors. Teams from the previous project may continue working on this project. Although the 6060 or 4060 Honors teams will not have additional requirements, their projects are expected to be more advanced and the implementations more robust. All project ideas will have to be submitted for approval by 11/08/2019 on eLC.
- **IMPORTANT:** Both students on each team are expected to contribute equally to the project. You should meet and work on the project together, preferably engaging in pair programming. You will be asked to submit a form with detailed contributions of both members and a log of meetings.
- Select a suitable name for your project. **You must use Android Studio version 3.5 or 3.4.2 (preferably, 3.5.1, the most recent one). When creating your project, you must select Phone and Tablet, and API 16: Android 4.1 (Jelly Bean). You must code your app in Java.**
- Basic application requirements:
  - The app must involve at least 2 users (you should test it with your partner, at least).
  - The app must use suitable components, including views, layouts and other widgets, as needed. The interface and navigation paths should be well designed and easy to use.
  - The user must register for use and log into the app to use its (external) functionality, but it can be just a simple user id/password-based authentication.
  - You must use an external data or information source. Google's Firebase is a relatively easy to use system, which may provide a lot of useful external functionality to your app.

If you would like to incorporate a Web service, [www.programmableweb.com](http://www.programmableweb.com) is an excellent source for finding interesting Web services to use. Many other RESTful services exist and are relatively easy to find. You should look for services that are free to use, but you may need to establish a membership and obtain an app key/id. As an example, if you are into movies and would like to work on a movie recommendation-like app, check out <https://www.programmableweb.com/api/movie-database-tmdb>.
  - You must use asynchronous calls to perform IO operations.
  - Your app must be sensitive to switching orientations, as needed. It must also save and restore its state, as necessary.
  - Whenever possible, you should use ConstraintLayout and you may not use RelativeLayout in your app.
- Test your app with your partner – the app should involve at least 2 users.
- Test your app in a Pixel XL, API 27 AVD and eliminate any errors. During the demo, you may use your app installed on a real Android device, if you want.
- Create a ZIP file with your entire project directory and submit it your zipped project to the Final **Project Assignment Submission** Folder on eLC. As before, if you don't have a fast Internet access at home, plan on submitting your project while on campus, as an expected size of your ZIP file may be quite large.
- Even though you submit the app, a demo will be required.
- Follow good Java coding style (suitable variable names, indentation, etc).
- Include suitable comments in JavaDoc style.

Additional explanations will be provided in class.