

Group Project

JobCompare: Job Offer Comparison App

Deliverable 2–Inception and Elaboration

Deliverable 3–Construction

Deliverable 4–Transition

Background

A fellow classmate, George P. Burdell, is looking for a new job after graduation. As it can be complicated to compare job offers with benefits, in different locations, and other aspects beyond salary, he would like an app to help with this process and has asked for your assistance in creating a simple, single-user job offer comparison app.

Goals

- Develop an Android app that implements the job comparison app that George P. Burdell envisioned (and that your team already designed).
- Get experience with (a simplified version of) the Unified Software Process.

Requirements

See the [requirements in Assignment 5](#).

Deliverables

Important: this section lists all the remaining deliverables. Which one you need to do depends on the week and is clearly stated in the Canvas assignment and in the schedule.

- D1—~~Software design~~
[done]
- D2 – *Inception and Elaboration* (see the corresponding lessons for details):
[see Canvas / Syllabus for due date]
The template documents contain instructions and should be self-explanatory.

- Project plan (**ProjectPlan.md**), showing the project development process and roles (see [template](#)).
- Use-case model (**UseCaseModel.md**) (see [template](#)).
- Supplementary requirements (i.e., requirements that do not fit the use-case model, such as non-functional requirements). These can be provided as a simple list but have to be provided. Put these requirements in a file, in markdown format, called **ExtraRequirements.md**
- Design document (**DesignDocument.md**) (see [template](#)).
- Test plan without results (**TestPlan.md**) (see [template](#)).
- D3 – *Construction* (see the corresponding lesson for details):
[see Canvas / Syllabus for due date]
 - Possibly revised earlier documents (**ProjectPlan.md**, **UseCaseModel.md**, **ExtraRequirements.md**, **DesignDocument.md**, **and TestPlan.md**), based on your better understanding of the system.
 - An initial version of the app (think of this as an alpha/beta release).
 - Test plan **with initial results**. (These may be partial results).
 - User manual. Put this in an MD file called **UserManual.md**
- D4 – *Transition* (see the corresponding lesson for details):
[see Canvas / Syllabus for due date]
 - Possibly revised earlier documents, based on your better understanding of the system.
 - Test plan **with final results**.
 - The final version of the app.

Instructions

Documents

1. Go to the directory “GroupProject” that you created for Deliverable 1 in the **team repo** that we assigned to you. Hereafter, we will refer to this directory as `<dir>`. Create a directory `<dir>/Docs`. All the documents that you will produce for Deliverables 2-4 should be placed in this directory.
2. Any image that you want to include in the markdown documents should be [embedded](#) by [including the image files in your repository and using a relative link](#), such (e.g., `./images/image.png`) within the markdown. This will ensure that your image links work when the repository is locally cloned and through updates in later commits.

Application (Starting in Deliverable 3)

1. Go to `<dir>`.
2. The code of the app must be uploaded to the repository as a regular Android Studio project called `JobCompare6300`, **whose root is directory `<dir>/JobCompare6300`**. All classes should be in package `edu.gatech.seclass.jobcompare6300`, and all

the files needed to run your app must be uploaded to the repo, as you did for Assignment 4 (see the instructions there). As a safety check, we recommend that you clone your repo in a separate directory/workspace and compile and run your project from there, in a clean AVD. Cloning your repo on a different machine would be even better, in case you have the option of doing that.

3. The minimum API level for the app should be “API 28: Android 9.0 (Pie)”
4. **Your application state should persist between runs.** You can accomplish this by leveraging [Android SQLite](#) (recommended) or using plain files. Either way, please test your application in a clean-checkout state and with a new or cleared AVD, so as to be sure that it does not need to be run in a special environment. Please note that simply redeploying within Android Studio to an existing AVD will not clear any database.
5. In addition, to help with cases in which we may have problems compiling your app, make sure to **add an APK file for your app in directory** `<dir>/APK`. Note that we intend to grade your app as compiled from the provided code; **this APK is required in case of problems and does not replace having complete, compilable, and runnable code in your submitted commit.**

Submission

1. As usual, after submitting a deliverable by pushing your changes to the assigned team repository in GT GitHub, the project manager must post on Canvas the corresponding commit ID. **The group should check that this ID is correct and on time in Canvas, as the team is jointly responsible for the submission.**

Notes (important—make sure to read carefully)

1. For Section 3.1 of the design document (Class Diagram), you should simply use the design that your team created for Deliverable 1, possibly updated.
2. The project plan should cover your **entire** intended project development process, including the work done in Deliverable 1.
3. **Avoid jumping to coding before the Construction (D3) phase, and do not just focus on the code.** Documents (use case model, design document, ...) are valued just as much as the code, and sometimes more, so having a great app with a set of poorly prepared documents will likely result in a low grade.
4. If in a later deliverable you need to update any of the documents in an earlier deliverable based on your better understanding of the system, you can do so, as explicitly stated in the description of the deliverables. If you do so, please make sure to add a version number at the beginning of the document and provide a concise description of the update (any reasonable format would do).
5. To clarify the previous point, please note that this is not an invitation to procrastinate. Although we will grade the various artifacts submitted based on their latest version, **teams will be penalized if they simply submit “placeholder documents”**. In other words, we are perfectly fine with the documents evolving throughout the project and will

grade their latest versions, but **the required documents have to be present and reasonable in every deliverable.**

6. **Optional:** Once you satisfy all of the requirements, feel free to be creative and extend the functionality of the app if you feel like it and believe you can afford it. This is not required, of course.
7. **Optional:** Automated test cases are always appreciated, and ultimately useful for you, but it is fine to have a list of test cases that can be run manually (as long as for each test case you describe how to run it, what inputs to provide, and its expected output). Whether manual or automated, make sure that your test cases adequately cover the functionality of the app.