**Design Discussion**

Johnathan Cao

Weimei Sun

Joon Park

Richard Fields

**Design 1**

Diagram

Description automatically generated

Our first design is highly modular, which provides a good basis for extensibility. However, the complexity is much higher than what is needed for this application. Also, the design is too tightly coupled to UI elements. There are several classes that can be converted into methods of another class, such EditCurrentJob, etc.

**Design 2**

Diagram

Description automatically generated

The second UML design is simple, abstract, and clear. All the classes and operations have reasonable and workable logic. This design meets all the requirements for the assignment, and it is a viable design for the application. One improvement is that the multiplicities between the Application, ComparisonSettings, and JobDetails classes need to be switched.

**Design 3**

Diagram

Description automatically generated with medium confidence

The design is very detailed, and it attempts to outline every interaction mentioned in the requirements. One improvement is that the design contains a “User” class that is not necessary since the assumption is that there will be a single user. Additionally, the design is very complex, and the number of interactions and classes can be confusing. There are also several classes that can be converted into methods within other classes here.

**Design 4**

Diagram

Description automatically generated

This UML is a simple representation with four classes. The attributes are abstracted, and only relevant interfaces are accessed through public methods. The multiplicity clearly shows the relationship of multiple job objects for 1 job manager class. Some improvements can be made to further simplify the design. The job\_list class is redundant and should be moved into a list/array attribute of the manager class as was done in Design 2. Additionally, the startup class (entry point) being a separate class is not necessary. It can be moved withing the job\_manager class as a member function.

**Team Design**

Diagram

Description automatically generated

We coalesced quickly around Design 2 and 4 and spent much of our time discussing the key differences between these two. We decided that, given the size of the project, simplicity in design is something that we want to prioritize. Design 2 and 4 both provided simple and easy-to-understand designs which satisfy all the requirements of the specification. All group members had some variation of a Job class which tracked all the attributes of a job (title, company, location, etc.). One interesting point of the discussion was that we all had different views on what an entrypoint to the system should look like. Weimei brought up a great point that some of us had defined entire classes for requirements which could be satisfied with just an operation/method.

After discussing the pros and cons of each design and deciding on simplicity as the priority, we decided to start with Design 2. It has the simplest form and does not include any unnecessary classes or attributes. The only update made from Design 2 to the Team Design is the multiplicities. The multiplicities between the ComparisonSettings and Application classes are now 1 to 1 which reflects a single set of comparison settings for the Application object. We also updated the multiplicities between the JobDetails and Application classes as 1 to 1..\* reflecting we can have 0 or more JobDetails objects for every Application object.

**Summary**

This process has helped us learn more about the features and usefulness a UML diagram can provide as a design tool during the development process. Sharing and discussing the designs with the team has helped all of us gain a better understanding of how to use UML more effectively as a language and platform agnostic tool. We discussed the structure of an Android application as well, and how this structure can have an impact on the various aspects of the design.

In addition to learning more about UML, we were able to use this time to build team report and develop our inter-team communication. We discussed how we can collaborate throughout this project virtually as we are working out of 3 different time zones and 4 different countries. It was also a good opportunity to understand everyone’s life situation and priorities, as these closely mirror what life is like in a professional working environment. All group members were motivated to find a working style which suits the entire team, and we made sure all group members felt their contributions were valued and everyone had a chance to input their thoughts. We also briefly discussed each other’s backgrounds and interests. Continuing this discussion will help guide our team in understanding each member’s strengths and weaknesses so that we can more effectively split up tasks within the group.

This design exercise also provided a good opportunity to go over teamwork logistics moving forward throughout the project. We discussed various strategies for distributing workload so that all group members would have an opportunity to earn points/credit for individual contributions. We also had some preliminary discussions on how our github workflow can operate moving forward.