IN2013, Week 3 – Object Oriented Design: adding non-problem domains, design with off-the shelf components

Dr Peter T. Popov

11th October 2018

Question 1 (design class diagram). Consider BAPERS analysis class diagram provided in the Appendix. Refine the class diagram to produce a design class model adding details:

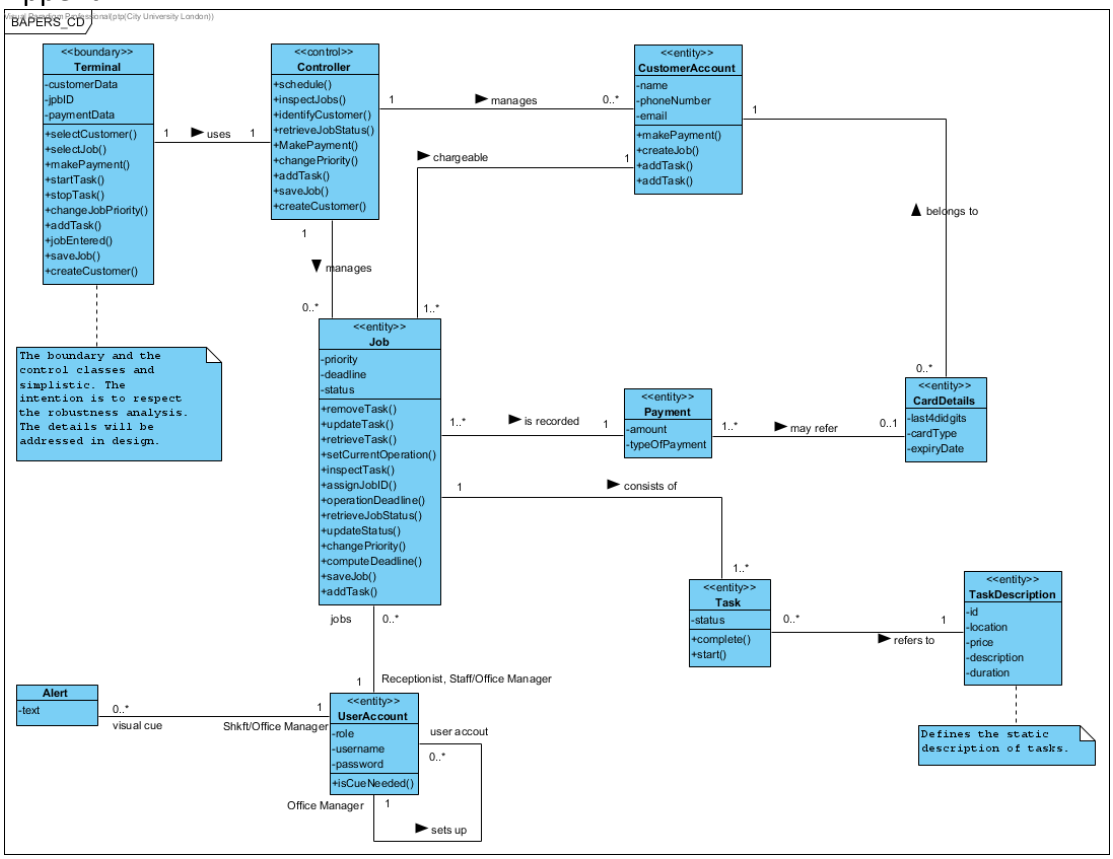
- to the class attributes and operations (methods).
- Revisit the associations and apply aggregation/composition as necessary.
- Make sure that constructors are defined for all classes.

Additional assignment (to develop independently at home): Non **problem domain** (GUI, DB).

- Add a set of boundary classes and develop them into fully fledged classes using a set of 'visual controls' (refer to what you did in Java module building user interfaces). Integrate the boundary classes with the problem domain classes they are related to, e.g. Job class will be associated with a boundary class JobForm, aggregation would be appropriate here. You may decide to replace the single control class used in the analysis class diagram with a series of control classes to split the logic of the applications between smaller control classes.
- For the DB domain consider using an interface, say DBConnectivity, with a minimal set of operations (e.g. connect(), write(), read(), closeConnection() and an implementation class, which provides a realization of this class, similarly to the example discussed in class). Model the fact that most/all entity classes are *persistent* and need to store/retrieve data from the DB via the interface DBConnectivity.

Please note that in the .vpp project BAPERS_2018_Tutorial3.vpp, which I created for this tutorial I created a ***Design Model*** for you and copied the analysis class diagram (without packages) to a new class diagram under the Design model. This copy belongs to a different "namespace". You should add design details by refining the class diagram under the Design Model. These changes (in the Design Model) will have no consequences for the analysis class diagram(s) which will be retailed in the Analysis model despite the modifications you apply to the diagrams in the Design Model. Thus, we can keep both - the analysis and the design class diagrams under the same .vpp project.

Appendix



Last updated: 10th October 2018