SE2030 Lab4: Class Diagram Lab Spring 2018

Overview: In this lab you will be creating high-level class diagrams for the entire application. Note that this is a team lab, but you should individually be able to justify EVERY design decision.

Learning Outcomes:

- Apply high-level class design concepts to create a syntactically correct UML class diagram
- Use EA to generate class diagrams

Instructions:

Using the sequence diagrams, use cases, acceptance tests, and transit project description, create a high-level UML class diagram for the entire application. All relationships between classes should be fully specified and justifiable (expect "why" questions from the professor and have justifiable answers ready). Keep in mind that **all** the functionality specified in the project description must be achievable by your design!

Your class diagram should specify ALL classes that are expected to be necessary for the application to work, and the "driver" class (with the main method) and Controllers for each window should be appropriately named. Include a dashed dependency line from the driver to the Controller tied to the fxml file launched by the driver if you use FXML stating "launches".

All variables that are anticipated should be specified as well as their visibility (public, private). You should include all public methods that are necessary for the functionality, although basic getters and setters can be omitted for brevity. Private helper methods are not required, but you are encouraged to specify them as you see fit.

It is more important for you to specify the details of the data structures than the GUI. In fact, the GUI classes should be only minimally specified. You should specify as many details of the controller class as is reasonable (it will likely be large – try to push as many responsibilities out of the controller and into your other classes as you can). Keep in mind that the diagram should be readable, so use good naming conventions and layout practices!

Deliverables: A pdf of your class diagram should be submitted. Be sure, in a comment on the PDF, to include your full names, the course code, the lab number and name, and a date. Do your best to fit your class diagrams neatly on a sequential set of pages (don't have a single class or a part of a class on its own pdf page). Lines should not cross over classes. Avoid crossing lines over other lines except where necessary. Note access levels for instance variables and methods. Include directions on associations, and do NOT include instance variables in class boxes where they are represented as associations.

Lab Checkoff: You must show an early class design with meaningful associations to the professor for feedback before you put a large amount of energy into specifying methods, instance variables, etc.

Due Date: Start of class, Tuesday of Week 5.

Grading:

GTFS Relationships Captured	20%
Data Structure Design	20%
Controller Design	20%
Syntax	20%
In class checkoff	10%
Following submission instructions	10%
TOTAL	**100%**