CS6310 – Software Architecture & Design
Assignment #9 [200 points]: Mass Transit Simulation – Team Implementation & Demo (v3)
Fall Term 2018 – Prof. Mark Moss

Submission

- This assignment must be completed as part of an approved group.
- One member of each team must submit:
  1. A VirtualBox (preferred) or VMware image that contains a pre-configured, ready-to-run version of your application named ***team_<team-number>.ova***, such as ***team_19.ova***;
  2. All source code (Java and otherwise) and links to external packages and other resources used to create your application in a file named ***source_code.zip***;
  3. The updated "as-is" UML Class and Sequence Diagrams, and any other accompanying documentation, in a file named ***design_docs.pdf***; and,
  4. An MP4 video (or other reasonably comparable format) with a ***viewing time of 10 minutes*** or less named ***mts_video.mp4*** that presents an overview of your application.
- For this project, you should select reasonably named files, and parts (2) and (3) must be submitted via Canvas. The virtual machines and videos for parts (1) and (4) can be very large, so a link to an external but accessible storage site (e.g. Google Drive) should be submitted via Canvas for those deliverables.
- Alternate (non-VM based) submissions for part (1) must be pre-approved by the Instructor or TAs. Also, submitting a link to an established code repository – for example, Georgia Tech's GitHub – for part (2) must be pre-approved by the evaluating TA.

- Submit your answers via Canvas.
- You must notify us via a private post on Piazza BEFORE the Due Date if you are encountering difficulty submitting your project. You will not be penalized for situations where Canvas is encountering significant technical problems. However, you must alert us before the Due Date – not well after the fact.

Scenario

The clients are excited about your design proposals as we enter the final phase of the project. During this phase, you will: (1) finalize, implement and deploy your team's design; (2) ensure your design documents are consistent with the finished application; and, (3) prepare a short video presentation for the clients and evaluators that demonstrates how your application functions.

Disclaimer

This scenario has been developed solely for this course. Any similarities or differences between this scenario and any of the programs at Georgia Tech programs are purely coincidental.

Deliverables

This assignment requires you to submit the following items:

1. Functioning Application [100 points]: You must submit a working application within a virtual machine image that satisfies the client's requirements as presented throughout all of the

assignments, and as clarified further through all associated Office Hours and Piazza posts. Your system must incorporate all of the design requests as described in all of the previous assignments and below. Your application must be submitted as a VirtualBox (strongly preferred) or VMware virtual machine image.

You have administrative control over the VM, and this submission method allows your team to have maximum flexibility for selecting, implementing and configuring different packages (e.g. GUIs, databases, etc.) in developing a solution for this problem, while minimizing the time needed by the graders to evaluate your (widely varying) solutions. You are allowed to use an alternate submission method if (and only if) you have received prior approval from your TA.

From a grading perspective, the points for implementation are distributed as follows:

- 20 points for providing a well-designed Graphical User Interface – the amount of points earned will be determined by the normal clarity and "ease-of-use" factors, as well as the extent to which the interface supports the normal "flow" of operations during each term.
- 20 points for each of the other design proposal categories: Bus Changes, Passenger Exchanges, Calculating and Displaying System Costs (Efficiency), and supporting Replays

2. Source Code [25 points]: You must also provide copies of the actual source code that your team has developed, along with references to all of the external packages, libraries, frameworks, services and systems used to develop your application. Your overall coding quality will be evaluated, to include factors such as good use of abstraction of modularity, reasonable documentation and descriptive class, variable and method names, etc.

3. Design Documentation [50 points]: You have already developed numerous design documents: Class Diagrams, Sequence Diagrams, Object Diagrams, and possibly other forms such as State Charts and/or Collaboration Diagrams. For this assignment, you must select and provide the most appropriate UML-compliant Structural and Behavioral Diagrams that describe the final design of your application. There are no more pending, additional or possible future requirements after this assignment, so your final design documents should be as accurate and consistent with the final implementation of your application as possible.

   Please clearly designate which version of UML you will be using – either 2.0 (preferable, and the latest OMG-accepted version: https://www.iso.org/standard/52854.html) or 1.4 (the latest ISO-accepted version). There are significant differences between the versions, so your diagrams must be consistent with the standard you've designated. The design documents can be submitted as different docs, but must be named clearly and accurately.

   From a grading perspective, the points for documentation are distributed as follows:

   - 20 points for your UML Structural Diagrams, which should include (at a minimum):
     - A Class Diagram
     - A Deployment (or similar) Diagram that shows how your system is actually implemented, especially in integrating external packages, services and/or systems

- 20 points for your UML Behavioral Diagrams which should include (at a minimum):
  - Sequence, State Machine (or similar) Diagrams for all functionality added per your design proposals (not required for pre-existing functionality)
  - Use Case Diagrams for all functionality added per your design proposals
- 10 points for the overall quality and presentation of your design documentation, including the addition of various diagrams that add significant value for the clients in understanding and maintaining your system into the future. *Do not simply create lots of diagrams to attempt to collect points – this is definitely a "quality over quantity" issue.* Ensure that new diagrams and/or documents are well formatted and add significant value.

4. Demonstration Video [25 points]: Your team must submit a video presentation that demonstrates the prototype that your team has developed. You can either submit a YouTube link (preferred) or an MP4 attachment in Canvas. If submitting a video attachment, make sure that the size doesn't exceed 360MB, which should be approximately 10 minutes for a 720p quality video. Also, we realize that not everyone wants to "make it to the big screen": faces are welcome, but there's no requirement for your faces to be displayed during the video. A screen capture of your system in action, along with a clear, audible English-language audio track, will be sufficient.

   Your video should be organized to give a clear demonstration of your system in action, including a well-considered test case/scenario that allows you to show how your interface and supporting components operate; and, how your overall system meets the client's requirements. And your video should highlight your design modifications. There's no need to spend significant amounts of time demonstrating the capabilities that already existed in the system – focus on the aspects of your system that have been changed, upgraded, etc.

Writing Style Guidelines

The style guidelines can be found on the course Udacity site, and at:
https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cs6310/assignments/writing.html

The deliverables should be submitted in the appropriate formats (ZIP, JAR, PDF) with file names such as ***source_code.zip***, ***design_docs.pdf*** and ***mts_video.mp4***. You should also use a similarly clear and simple name structure if you need to submit a file that we haven't explicitly listed here. Ensure that all files are clear and legible; points will likely be deducted for unreadable submissions.

Client's Final Problem Description

In earlier assignments, you proposed modifications to an existing application that the clients use to simulate the operation of a mass transit system. In this assignment, you are tasked to: (1) coordinate with your team members to develop a consolidated design proposal; (2) implement your consolidated design; and, (3) demonstrate those changes to the clients. We provide extra details below, which basically build on the design proposals you were asked to provide in earlier assignments.

User Interface and Error Checking: Your application must provide a reasonable Graphical User Interface (GUI) to support your application.

- You are welcome to develop a web-based or application-based GUI. An extremely complex GUI is not required (and not the intent for this course) – a simple GUI that provides the required functionality will be sufficient.
- The clients will greatly appreciate any efforts to make the GUI user-friendly, and to help prevent errors when entering data, configuring the simulation settings or executing a simulation session. For example, robust warnings and error messages will be very helpful.
- You are still permitted to have a Command-Line Interface (CLI) with your application to support development, maintenance, troubleshooting and execution functionality. Your GUI, however, must be able support the major functionality required for a user to conduct a simulation. Your CLI is allowed to provide "behind the scenes" support for the GUI as needed.
- We are giving your team fairly wide latitude in selecting a framework to support the development of your user interface, given that you are responsible for installing and configuring that framework on the course VM (or an alternate pre-approved platform) for submission.
- Your application must support the commands implemented in the existing prototype, in addition to the changes listed below. This is absolutely critical to our ability to evaluate your application thoroughly.

Updated Scenario Configuration Commands

Significant portions of the client's Assignment 6 requirements changes were about passenger exchanges at stops. To support these new requirements, the clients will now provide two separate files at the beginning of a simulation run. The first file will be the normal scenario configuration file that addresses the creation of stops, routes, buses and initial events as used since Assignment 2 and the introduction of the first prototype. The second file will contain the probability parameters needed to model the passenger arrival and departure behaviors as described in Assignment 6. Both files will be listed in a simple text-based, Comma Separated Values (CSV) format, and there might be spaces included between the parameter values and the comma delimiters.

For the first file, the commands are unchanged with one exception:

- **`add_bus, <ID>, <Initial Route>, <Location>, <Initial Passenger Capacity>, <Initial Speed>`**

This command creates a bus object with identifier <ID> that begins traveling along <Initial Route>, starting at index <Location> within the <Initial Route> list. All buses now begin carrying zero riders. Also, the bus can hold at most <Initial Passenger Capacity> passengers, and travels along the route at the given <Initial Speed> as measured in statute miles per hour.

The parameters have been updated to reflect the new requirements. For simplicity, all buses will begin with zero passengers, so the <Initial Passengers> parameter has been removed. The route, passenger capacity and speed parameters have all been renamed with the descriptor "Initial" added to reflect the ability of the simulation user to change these parameter values during the

simulation run. Otherwise, the <Initial Route>, <Initial Passenger Capacity> and <Initial Speed> parameters all serve the same purpose as in the earlier versions of the program.

Also note that the fuel-related attributes <Initial Fuel> and <Fuel Capacity> are not needed per the requirements, and have been removed. Similarly, there's no need to create a depot stop (or the associated refueling behavior) for this version of the MTS application, so the **`add_depot`** instruction will also be removed from the first file.

The second file will include the probability distribution limits for passengers arriving at and departing from stops, and getting on and off of buses. The second file format will consist of one line for each stop, and each line will include the probability parameters that stop in the following order:

- **`<Stop ID>, <Riders Arrive High>, <Riders Arrive Low>, <Riders Off High>, <Riders Off Low>, <Riders On High>, <Riders On Low>, <Riders Depart High>, <Riders Depart Low>`**

This separation of the data into separate files is a very reasonable approach for managing the simulations. With the exceptions of the stops and stop locations, the information that is used to configure the buses and routes is distinctly different from the data that describes the passenger's behaviors at different times and in different circumstances. This two-file approach will allow the simulation users to more easily experiment to observe how different bus and route configurations are affected by changing rider dynamics.

Closing Comments & Suggestions

We (the OMSCS 6310 Team) will conduct Office Hours where you will be permitted to ask us questions in order to further clarify the client's intent, etc. Also, the TA who will evaluate your final submission will be pre-assigned to your team. You can communicate with them if you have questions related to application design, implementation, deployment, etc.

Quick Reminder on Collaborating with Others

Since this is a group project, you may (and should) communicate freely with all of your group members. However, your group is not allowed to communicate with any other groups while working on this project, including outside personnel or consultants. Please use Piazza for your questions and/or comments, and post publicly whenever it is appropriate. If your questions or comments contain information that specifically provides an answer for some part of the assignment, then please make your post private (your group members and OMSCS6310 TAs/Instructors only) first, and we (the OMSCS 6310 Team) will review it and decide if it is suitable to be shared with the larger class.

Best of luck on to you this assignment, and please contact us if you have questions or concerns.
Mark