Final Project Phase III

1 Logistics

- Submission deadline (to your repo): Aug 4, 11:50pm.
- Software Release Demo: Aug 5, 2016 from 13:00 to 18:30. Please check out the course web site for the time slot assigned to your team.
- Peer evaluation forms: Aug 5, 2016 by 11:50pm (email the completed form to your instructor - details below).
- All deliverables (see below) must be submitted in the subfolder named PIII of your final project folder. In particular, the whole software must be committed under PIII/src folder.

2 Overview

In Phase III of the project you will complete the implementation of the shopping app.

3 Learning Goals

By the end of this phase, you should have:

- practised dealing with software requirements that change over the course of a software development project,
- worked closely with your teammates to re-evaluate and possibly update your design of a software system,
- produced a working Java application that implements your software design and corresponds to user requirements.

4 Software Release Demo

The purpose of the Software Release Demo is to demonstrate to the product owner (repesented by your TA) that you have a working Java GUI application. The application must be able to do the following:

1. Launch and start the main application. The main GUI window should contain the following elements:
   - A menu bar that offers the user a possibility to log in, register, log out
   - Once the user is logged in, based on the user type (shopper or administrator), the user should have available the necessary tabs to accomplish the intended goals.
   - The main content panel of the JFrame must show all available products to any user (logged in or not, administrator or shopper).
2. Allow a non logged in user to see all available products and prices
3. Allow an administrator to add a category, add a product, add a warehouse (center), maintain product quantities (add a product quantity to a specific warehouse), maintain the shipping graph (add direct routes between various cities indicating the length of each route), produce at least one sales reports (your choice, by category, range of dates, or shipping destination).
4. Allow a logged in shopper to add item/remove item/change item quantity in his shopping cart.

5. Allog a logged in shopper to check out his hsopping cart. The shopper must see the total value of his items and the shipping cost.
6. All those activities must be carried out using dialogs.

5 Feature List of this Phase

In this phase, the main focus is to make sure the features implemented in the Phase II work correctly with your GUI interface. A few additional features, already observed by many of you during Phase II as follows:

- The shopper must be abble to add mutiple items in the cart
- If an item can be fufilled from more than one warehouse, or various items are available in different warehouses, you must do the following:
  - Among the warehouses where the items in the cart are available, identify the closest (that is the cheapest) to the shopper
  - Fix the product quantities by transfering merchandise from other warehouses to the warehouse closest to the shopper.
- Compute the shipping cost by multiplying the distance (computed as the distance from the nearest warehouse to the shopper's city) by a constant factor.
- Please do ignore the differences in cost caused by the weight/size/quantity of the orders.
- Also we do assume that the merchandise that is transferred internally from one warehouse to another to the benefit of the shopper, is handled by the company and the cost is absorbed in the general operating costs of the business. So you may safely ignore all these concerns for this release.

6 Your Tasks

6.1 Task 1 - Modify the back end to accomodate the additional features

There are very few changes this time - all described above. Please implement them as required. Please update the CRC accordingly (if necessary).

6.2 Task 2 - Implement the front end

In this sprint, the focus is to implement the GUI front end. A demo of what is expected will be offered in class and it also be available during office hours.

6.3 Task 3 - API

In this sprint, the additions to the API are minimal - s single method has been added so we can test the functionality of new changes.

6.4 Task 4 - The software development process

Your team should meet regularly while working on the project. You are required to have two types of meetings: **planning** meetings and **status** meetings.
You need to have two planning meetings: one in the beginning of the project and one mid-way through the project phase. During a planning meeting, the team will (a) recap on the current state of the project (if mid-way meeting), (b) decide on a set of tasks the team will accomplish before the next planning meeting,

and (c) decide who will perform which tasks.

In addition to the planning meetings, the team will meet for weekly status meetings. (You may have more frequent status meetings if you wish.) During status meetings, each member will report on (a) what (s)he has accomplished since the last meeting, (b) what (s)he plans to accomplish before the next meeting, and (c) if there are any problems or obstacles that prevent him/her from making progress. To demonstrate the software development process the team followed, you need to maintain a plain text file called `meetings.txt`, where the team will record all meeting minutes. A sample of meeting minutes is posted on the course web site and on blackboard.

On the day of each meeting, commit this file into the directory for this phase in your team repository. The contents of this file must match the state of the rest of your repository! If a member repeatedly does not meet deadlines or misses meetings, contact your instructor immediately.

6.5 Peer evaluation forms

Peer evaluation is very important component of this project. For this part, you will evaluate your team members by completing the forms provided in the course web site. DO NOT COMMIT THOSE FORMS IN YOU REPO! They are confidential. That said, you are encouraged to share your opinion with your team members. Never nurse a problem - rather try to solve it. Keep in mind that honesty is the best policy!

Once you have completed the evaluation forms, please email them you your instructor by Friday, Aug 5 2016, 11:50pm.

The peer evaluation forms will be used to compute your grade for this component as follows:

- 4% comes from the rating that your team mates have assigned to you computed as 4∗((Sum of total points given to you)/24)
- 1% comes from the evaluation of your `meetings.txt` notes. Please note this component is identical to all group members.
- 1% comes from the answers to your questions on team dynamics - specifically - the clarity of explanation of your team dynamics feedback. Please note at this component we are looking for a well argued explanation of your ideas (one or two paragraphs per question will do).

7 Marking

All of these items will affect your grade:

- Peer evaluation (6% - see above)
- Software Release Demo (2%)
  - How clear is your presentation?
  - How much is your interface user friendly?
  - How robust is your implementation towards user input?
  - Can a user achieve its goal with ease?
- Coding style and quality (1%)
  - Simple and clear code, clear logic.
- Functionality and usability of the application (2%)
  - All functions from the feature list (starting from the `ProjectInfo` up to this document) must be implemented.

  - Stability of the application (e.g. should not crash on invalid input).
- Javadoc (1%)
  - The quality of the documentation of the code.
- Please note the whole phase III is worth 12% of your final grade.

8 Checklist

Have you ...

- committed your work to your team repository?
- committed `meetings.txt`?
- used `svn list` and `svn status` to verify that your changes were committed?
- completed the Software Release Demo on the assigned time slot?
- emailed peer evaluation form to your instructor?