CSC 207 Introduction to Software Design
Winter 2015 – Project Phase II

Logistics

- **Due date:** ~~9:00pm Tuesday 10 March 2015~~ 9:00pm Thursday 12 March 2015
- **Group size:** Four. In this phase of the project you and your Phase I partner are joined with another pair to form a team of four. You will work in this team in Phase II and Phase III of the project.

Overview

In Phase II of the project, you will merge your Phase I design with that of your new teammates. You will then implement a part of your new and improved design.

Learning Goals

By the end of this phase, you should have:

- worked closely with your teammates to produce a design of a software system
- produced a working application that implements a part of your software design

Task I — Software Design

You have two designs developed by two pairs during Phase I of the project. With your teammates, discuss the benefits and drawbacks of each pair's Phase I design. As a team, develop a new design by merging the two Phase I designs and adopting the best features of each one. Create a file `crc.pdf`, following the same format you used in Phase I, and commit this file to the directory `PII` of your (newly created) team repository.

Note that your design should cover **all** the features for the application, not just the ones you will implement in this Phase.

Task II — Implementing the Application Backend

Feature List for this Phase

Here are some of the features (slightly modified) from the original Feature List that you will implement for this Phase of the project.

- A User can search available flights by entering a departure date, and travel origin and destination. A flight info should include: (1) flight number, (2) departure date and time, (3) arrival date and time, (4) airline, (5) origin, (6) destination, (7) cost, and (8) travel time.
- A User can search available itineraries by entering a departure date, and travel origin and destination. An itinerary should include, per flight: (1) flight number, (2) departure date and time, (3) arrival date and time, (4) airline, (5) origin, and (6) destination, plus an overall itinerary cost and travel time.
- A User can display search results sorted by total travel time or by total cost.
- A User can view all stored client information.
- A User can upload personal and billing client information to be stored in the system in a file.
- A User can upload flights information into the system in a file.

The file format the user will use to upload personal and billing client information in a `csv` file is as follows:

```
LastName,FirstNames,Email,Address,CreditCardNumber,ExpiryDate
```

where the expiry date is stored in the format `YYYY-MM-DD`.

The file format the user will use to upload flight information in a `csv` file is as follows:

```
Number,DepartureDateTime,ArrivalDateTime,Airline,Origin,Destination,Price
```

where the date and time is stored in the format `YYYY-MM-DD HH:MM`.

Note that this subset of the original Feature List focuses on the "back-end" of the application. There is no Android, no fancy interaction with the user. However, the main parts of the application that deal with data storage and with the search algorithm are implemented in this stage of the project.

Class Driver

One of the main tasks in this phase is to implement your back-end design. It is entirely up to you to produce the object-oriented design of the application.

In order to test your code, we require that, in addition to your design, you submit a class named `Driver` that implements the methods specified in the `Driver.java` starter code. Notice that class `Driver` is required for testing purposes only! It should not contain any algorithms or do any interesting work. It should merely call appropriate methods in the classes that you designed and implemented. `Driver.java` must be in package `driver`.

The Software Development Process

Your team should meet regularly while working on the project. We have two types of meetings — planning meetings and status meetings.

For **planning meetings**, you need to meet twice: once in the beginning of the project and once 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.

For the **status meetings**, the team will meet at least once a week, in addition to the planning meetings. During these 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/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.[1] *On the day of each meeting*, commit this file into your team repository. The contents of this file must match the state of the rest of your repository!

The search algorithm

The solution to our task of finding all flight itineraries should look familiar: think of the algorithms for traversing a Graph.

To simplify your task, the marking scheme for this portion of your project work will be as follows:

- If your algorithm finds most, but not all itineraries that satisfy the search criteria, you will get 90% of the correctness marks for this part.
- If your algorithm finds all itineraries that satisfy the search criteria, you will get 100% of the correctness marks for this part.
- Of course, if your algorithm returns an invalid itinerary, runs into an infinite loop, etc., marks will be deducted accordingly.

[1] See lecture slides for some example meeting minutes.

The end of this project phase

At the end of this project phase, your team should have a working version of the application back-end that implements every feature on the above feature list. You should, of course, have Javadoc comments for all your code.

Task 3 — Team member and self evaluations

Any student who does not submit their evaluations on time will receive a mark of 0 on this phase of the project. The evaluations are due 48 hours after the project phase deadline.

You will be filling out and submitting a peer evaluation activity on CATME. This form will rate all team members, including yourself, on contributing to the team's work (contributing a sufficient amount of work, contributing work of good quality, being on time, helping teammates) and interacting with teammates (showing interest in teammates' ideas and contributions, asking teammates for feedback and using their suggestions to improve, making sure teammates stay informed and understand each other, providing encouragement and enthusiasm to the team).

These are meant to be private: each team member will submit these separately, and you are not required to show each other your forms. In the case of serious disagreement, or if you request it, we will hold a team meeting to discuss the results, but we will never reveal individual ratings.

Marking

All of these items may affect your grade:

- CRC Model
  - The modularity of the design, and the degree to which it is reusable and extensible.
  - The degree to which the design meets the requirements.
  - The use of OO concepts we study in class.
- The appropriate use of files and data structures.
- Correctness of the application: all required methods implemented correctly.
- Javadoc:
  - required for methods and instance and static variables
  - must have a period at the end of every sentence
  - must use @param and @return tags
  - must use good English
- Coding Style:
  - must follow Java naming conventions
  - indentation
  - consistency
  - white space
- Quality of the software development process:
  - the file `meetings.txt` must be committed according to the schedule
  - the contents of the repository and the state of the code must match the contents of the file `meetings.txt`
- Subversion commit history:
  - participation by all team members
  - frequent commits over an extended period of time
  - appropriate commit logs
- Peer evaluation
  - To view the evaluation criterion, see the CATME online evaluation form (`www.catme.org`).

Checklist

Have you...

- used your new team repository and not your individual repository and not your repository from phase I to submit your work?
- committed `crc.pdf`?
- committed **all** of your project files and directories, including `Driver.java`?
- committed `meetings.txt`?
- verified that your changes were committed using `svn list` and `svn status`?
- in the next 48 hours: submitted your team evaluation forms using CATME?