

## Project B

Due Date:

Design (5pm Fri. October 11)

Code (5pm Mon. October 28)

### 1. Introduction

This project is basically a maintenance project, where you are asked to extend the university registration system that you have developed for project A. A few new roles, constraints and requirements have been added to those from project A. Remember, in a software system based on a good modular design, small changes in the requirements should trigger changes in only a small number of modules.

### 2. New Requirements

The following are the new requirements:

1. Uniqueness of the ID numbers for all the users of the system (students, general staff and academic staff) as well as the uniqueness of the subject codes must be explicitly represented in the design (i.e. class diagram).
2. The system should be able to record marks for the subjects that a student is enrolled in. A new command "mark" is introduced for reporting the mark gained by a student for a subject. Obviously, this command should verify validity of the subject code as well as enrolment of the student in the subject. It should also validate marks (check for being within range). If any of these conditions is violated, a sensible error message must be sent to standard output.
3. There are two new commands "open" and "close" that open and close subjects for enrolment respectively. These two commands can only be used by a general staff.
4. Subjects are run as lectures and tutorials. Each student enrolled in the subject needs to be in one lecture and one tutorial group. The actual number of the lectures and tutorials are depending on the number of enrolments in the subject. For each subject, there is a maximum size for lecture and a maximum size for tutorials (that will be used to make the right number of lectures and tutorial classes for the subject based on the number of enrolments). Each lecture and tutorial class has its own day and time info. There are two new commands "lecture" and "tute" that can only be used by a general staff in order to put enrolled students into lectures and tutorials (automatically by going through all enrolled students for a subject, without the students intervention). A new lecture or tutorial is not created unless the previous

ones are full.

5. A student cannot enrol in a subject without passing all its prerequisites. A pass for a subject is a mark greater or equal to 50 for that subject.
6. All users need to identify themselves to the system using their ID number. A new command "login" is used for identification purposes (any syntactically correct ID number is considered as a valid user). In fact, a login starts a session and there can only be one session running at any given point of time. Thus, once a user logged into the system other users can not login until the current user logs out (see "logout" command).
7. A new command "logout" is used to indicate the end of a session started by a "login" command for a user.
8. The following commands are involving more than one actor:
  - Enrolling into a subject is initiated by a student (using "enrol" command) and it is pending approval by the academic staff who is teaching the subject (using the new command "approve").
  - Adding a new subject is initiated by an academic staff must be approved by two other academics (using the new command "approve").
9. The new command "approve" is only used by academic staff to finalize pending requests for adding new subjects or enrolling into a subject. Issuing this command with only a subject code as the parameter will results in approving a new subject (with the given code). Issuing it with two parameters (a subject code and a student ID) will results in approving a new enrolment.
10. There is a new command called "pending" that is only accessible by staff (academic or general). Without any parameter, this command lists all the pending requests for adding new subjects to the system. Having a subject code as a parameter, it lists all the pending enrolment requests for the given subject code.
11. An academic staff may also be enrolled as a part-time student. Part-time students can take up to half the number of subjects that can be taken by the full-time students.

Soon, a summary of all the new commands, including the format for the input file, and their accessibility to users will be given on the subject web page. This delay should not hinder your ability to start designing immediately.

### 3. Deliverables

The final deliverables are:

1. Design of the system (Due Oct 13) that includes:
  - A use-case diagram;
  - Collaboration and sequence diagrams (for each use case);
  - Class diagrams (both high level and detailed);
  - Design decision and assumptions should also be submitted.
2. Well documented working code that implements the design (Due Oct 28).

## 4. Submission

- *Hardcopy Design*

Your design should be submitted in hardcopy form to the submission box in the basement of SEECS (the same as for project A).

Also, you should fill out and attach, to the front of your design doc, a cover sheet (that you will soon be able to download from the 254 web page). *Submissions without cover sheets will not be marked.*

- *Implementation*

Your implementation should be submitted using the submit command:

```
submit 254 B *.java
```

(this assumes you are submitting from a directory with all your Java source files in it).

And do not forget to verify:

```
verify 254 B
```

Late submissions should be sent to `B.late` using the above commands. Late submissions will be penalised at 1 mark per day late.

Please note that while we will provide sample test cases, your submission will undergo other invisible tests and the tests you see, may well be not the ones we use for assessment. You should not view submit as a tool for testing the correctness of your solution ... testing is your job, not ours!

As was the case with project A, we should be able to run your project via the command:

```
java URS [input file]
```

## 5. Assessment

Project B will be marked out of 15, however the emphasis has now been shifted more towards design for this project.

1. *Design Document: 10 marks.*

You should review all the lecture slides that have shown examples of the diagrams we have asked for. This is what we are after. Soon some sample designs from project A will be posted on the 254 web page to show you some examples of good (and bad) design for project A.

2. *Implementation (and Correctness): 5 marks.*

The usual criteria will be used to judge this. The most important criteria is that your implementation obviously maps to your design and that the solution is correct. Documentation and Style will also be considered.

You should take these weightings to indicate that we are much more interested in your design and so you should spend considerable time developing it. It would be advisable to not start coding until you have thoroughly considered and drawn up your diagrams. You will be submitting your design early so changing your design once you start implementing it will be difficult.

## 6. Questions and Comments

As with project A, you should make a habit of checking cs.254 newsgroup. Also, the subject webpage will feature highlights and FAQ's from newsgroup postings. If the question is of value to at least one other person then you should mail it to cs.254. Questions will be answered as quickly as possible but your patience is much appreciated if some delay occurs.

Good luck!