IS1220 - Final Project
Supplementary Info

1 Project Evaluation

The evaluation of the final project includes the following main steps:

- the project report will be read and its quality thoroughly evaluated
- Testing:
  - the documentation related to all test scenarios (i.e the mandatory test1 scenario corresponding to file `test1input.txt` and `test1output.txt`) as well as any supplementary test provided by a project's team will be read and evaluated
  - all test scenarios will be executed and their outcome verified (e.g. commands stored in `test1input.txt` will be executed and their outcome verifed against the content of file `test1output.txt`)
  - Junit tests will also be run and verified
- Javadoc verification: the Javadoc documentation of the project will be analysed and evaluated.
- Code analysis: various part of Java code the project consists of are going to be analysed and the code quality evaluated.

2 A roadmap to project implementation

Below you find a short roadmap outlining the main steps that a project's team should follow for working out the project implementation.

1. Analysis of the project text (group discussion)
   (a) writing of first version of the Introduction of the project report.
   (b) sketching of a preliminary version of Chapter 1 of the project report, highlighting the context and understanding of the problem.
2. Design (on paper, through UML class diagrams) of a first architecture for the project, taking into account the project specifications and trying to apply as much as possible the OPEN-CLOSED principle for code design. In this step you will deal with issues like: classes identification, relationship between different classes, etc...
   (a) first sketch of Chapter 2 of the project report. Chapter 2 will be devoted to describe the code architecture, the design choices and their realisation. Chapter 2 will contain a Chapter 2.x sub-section describing the UML class diagrams and will include a description of each design choice. Chapter 2 will be ended by a Chapter 2.z sub-section containing a discussion about all designing issues you faced, the solutions you have adopted highlighting possible controversies you might have gone through during the design phase.
3. Start working on your implementation (coding of classes conceived in previous phases). For this you should start from the JUnit tests of each class (Test Driven Development). This will give you a chance to fairly split your implementation tasks between the team's members. During this phase of implementation you should also:
   (a) Write a first version of Chapter 3 (Implementation) of the project report. Chapter 3 will contain a sub-section 3.x describing the implementation issues you have faced.
   (b) Write a first version of Chapter 4 (Junit tests) of the project report. Chapter 4 will contain a sub-section 4.x describing issues faced during the Junit testing (i.e. the tests that failed and how you fixed them).
4. Implement the CLUI. While doing so you should also complete the writing of Chapter 3 and 4.
5. Realisation of test scenario to be stored in file `test1input.txt`. This is a basic step that the project marker will go through in order to assess the quality of

your project's realisation.

remark: supplementary test scenarios stored in files `testXinput.txt` are very welcome and warmly advised. The implementation of a command for a prompt evaluation of each supplementary test scenario you realise is **strongly recommended**.

(a) writing of chapter 5 (of the project report) devoted to the description and analysis of each test scenario. This should contain a clear description of how to use your test scenarios, what results are they supposed to give and also contain a detailed discussion on any issue you faced related to running/realisation of test scenarios for the project.

6. Writing of Chapter 6 (Conclusion) of the project report. This will contain concluding remarks. summarising briefly your implementation, design choices, what you managed to do, what you did not manage to do, possible future work direction to extend/improve your project implementation