**Second Project Specification**

**This file describes what I did for the second project.  
The information for Project A, B can be found in ‘../ProjectABSpec’. The information for the first project can be found in ‘../FirstProject\_SRS/RequirementsElicitationAndSpecification’. SRS of the first project that Vivian made can be found in ‘../References/VivianSRSProject1’.**

You have been made responsible for developing the Implementation Planning and Initial Design for a major software project. You have access to a fairly detailed requirements document and if necessary a client representative (and any publicly available information that might help you).

In the first iteration of the Implementation Planning (which is all that you need to do for this assignment) you will gather whatever information you need, develop parts of a detailed implementation planning document, and develop a detailed plan for how you would test the software developed under your plan.

**Teams of Two**

You will work with the **same partner** as in Assignment 1. Contact your partner now!

But, and this is important, your job is to plan the development and testing for the project that your partner developed the requirements for.

In other words, you are swapping projects now. If your partner worked on the requirements for the train station system in Assignment 1, you will work here in Assignment 2 on the implementation plans for the train station system **based on the requirements document that your partner developed in Assignment 1**.

**How teams will work**

The assignment is an *individual* assignment. Each student will submit their own results from their own implementation planning (as described below). But in doing your planning you may need to speak with your client representative. That person is the other person on your team (so again, client representatives have swapped projects). In fact, the great bulk of what you need should be contained in the SRS that your partner developed in assignment 1, so there is likely to be much less need to consult the client representative in this assignment.

**What you need**

Your partner should provide you, as soon as possible, with the SRS document that they developed in Assignment 1 (and of course, you should likewise, provide them with the SRS document that you developed). This simulates a common situation in software development -- someone has spent the time preparing the requirements documentation, and now you need to take that document and plan the implementation.

Note that what each of you should supply to the other is the formal requirements documentation that you developed, not the additional reports to the boss and or to Carl that talked about the process and about your further plans.

**What you must submit**

You will be required to submit one document which should not exceed twenty A4 pages and must contain the following:

1. Your name and student ID, and the name and student ID of your partner.
2. A vision statement for the project on which you are doing the implementation planning (one paragraph -- this should be your own presentation of your vision for the project, not just a copy of the one your partner used in Assignment 1).
3. System Design Document
4. Data Definitions
5. Analysis and Design Class Diagram (1 page)
6. One or more State Diagrams for the more interesting objects in your design
7. Requirements Traceability Matrix
8. List of design assumptions (if any)
9. Test specifications (~10 pages)
10. A report to James and Carl describing any difficulties you had in working with the provided SRS and what you did to overcome them.

Be extremely well-organised, and make sure your submitted work looks like a professional document.

Remarks on some of the artefacts:

System Design Document: This will include the basic architecture of the system and the high level strategic decisions. You need to include a description of the:

* system architecture
* storage/persistent data strategy
* noteworthy trade-offs and choices
* any concurrent processes and how they will be coordinated
* and a package diagram showing the subsystems you will use

Data Definitions: Create a table showing what data will need to be stored in your system. For each item give the name of the field/attribute/variable, its type, its meaning in the problem domain expressed in natural language, and an example of valid data.

Analysis and Design Class Diagram: You need to do an initial design of your system -- what basic objects should it have? And what are the methods associated with those objects? You will represent your design decisions in a class diagram. In a full plan you need to make sure any classes or methods in any sequence diagrams have been included in the class diagram -- it might help you to draw some sequence diagrams to help you to decide what your class diagram should contain. Method signatures should be given. The diagram must include, as appropriate:

* classes
* attributes
* associations
* inheritance and/or aggregation (if applicable)
* multiplicities

State Diagrams: You are required to consider the relevant states of each object in your system and to submit state diagrams for those that have interesting states or complex behaviour. One way to measure if a state is interesting is to consider whether you need to test that state before performing a particular action or if the state changes after an action is performed. What is interesting will depend on the application.

Requirements Traceability Matrix (RTM): Set up an RTM with the following columns:

* Requirement-ID (from SRS)
* Use Cases
* Classes
* Methods
* Packages
* Build Number (kept blank at this stage)

There should be one row for each requirement. For this deliverable, just fill in the first five columns, since the last column (and usually a couple more after that which I've already deleted) are concerned with the design of the system.

List of Assumptions: This will help the marker to understand why you have done certain things. Please review the assumptions carefully before submission. (But note: A poor assumption should not be used as an excuse for poor design decisions.)

Test Specifications should contain the following:

1. Test-case specifications, made up of test-case identifiers, test data (input specifications and output specifications)
2. Test plans, including for example a test schedule, testing resources required, testing milestones and test deliverables.

Acceptable documentation for Test Case Specifications would include:

* Test Case Identifier
* Test description
* Input specifications
* Output specifications
* Test plans, covering scheduling and resourcing of all testing processes

Test plans can be more open format and should provide a description of how you would organise the actual testing of the Test Case Specifications that you've identified.

More details about test plans are available on a separate page nearby.

**Other documents to help you**

Separately from this document, there are the pages which describe the two projects. They are nearby with Assignment 1.

Also check the Resources Section. There are templates and other documents there to help you.

And of course, you can, and should, read more broadly to develop your ideas both about good software engineering practice, and about the areas of your specific projects.

**Need further help or clarification?**

First, read all of what is here very carefully. The specification above tells you a great deal, but needs to be read and interpreted carefully (in common with all technical documents).

If you still feel a need to ask about anything, do please do so. Talk to Mike after a comp2050 lecture he attends, or if he really hasn't been there for a few lectures email him. He always likes to talk to you (but do be sure that you've thought hard about whatever you're asking about before talking to him).

**And finally...**

Enjoy the challenge this assignment presents. There is a fair bit of work in getting a coherent set of documents together to make your submission, but it is an important, and interesting, experience.