SENG3011
Intial Management Report

Group Number: 02

Version	1.7
Print Date	19/03/2013 10:26
Release Date	27/03/2013
Release State	Final
Approval State	Pending
Approved by	Group02
Prepared by	Group02
Reviewed by	Group02
Confidentiality Category	Confidential

Document Revision Control

Version	Date	Authors	Summary of Changes
v1.0	20/03/2013	Group02	Added in Introduction
v1.1	24/03/2013	Group02	Added in Architectural Diagram
v1.2	25/03/2013	Group02	Added in Use Cases
v1.3	25/03/2013	Group02	Added in Sequence Diagrams
v1.4	25/03/2013	Group02	Added in Model Architectural Diagram
v1.5	26/03/2013	Group02	Added in Project Plan
v1.6	26/03/2013	Group02	Layout for Use Cases changed
v1.7	27/03/2013	Group02	Added in Justification of Chose Language and Envrionment
v2.1	9/04/2013	Group02	Changed Use Cases
v2.2	15/04/2013	Group02	Added in new Use Cases

Contents

List of Figures

1   Introduction

This report is designed to discuss our initial prospects for the project and what we have in planned for our final prototype.

Within this report we will illustrate our Use Cases, Architecture and Sequence Diagrams, explain our chosen language and provide a project plan to which we hope to follow to complete our prototype.

We will be designing our report carefully following the Requirements List which has been provided to us:

1. Reading a correctly formatted Sirca orders file (1 day only)
2. Choosing an appropriate algorithmic trading strategy and setting its different parameters
3. Generating algorithmic orders for 1 particular day
4. Evaluating algorithmic trades and providing feedback to the user
5. Generating a strategy performance report
6. GUI functions to control and use the Use Cases to load and execute orders
7. GUI functions to visualise market data (spread, volume and depth)

We will also aim to meet the Quality Requirements:

1. Speed of execution
2. Usability of the GUI
3. Quality of the visualisation
4. Quality of the strategic performance report

2 Use Cases

From discussion between our members we were able to establish the following Use Cases:
1. Reading Correctly Formatted Sirca File
2. Choosing an Algorithmic Trading Strategy
3. Generating Trades for the Day
4. Evaluating Algorithmic Trades
5. Generating Strategy Report

Use Case 1: Reading Correctly Formatted Sirca File

Actors	User
Triggers	The user starts the program.
Preconditions	The user has a correctly formatted Sirca file.
Postconditions	The system will be populated with orders
Normal Flow	The system starts up, reads in the CSV file and prepares the order book of trades.

Use Case 2: Choosing an Algorithmic Trading Strategy

Actors	User
Triggers	User decides on a trade strategy to try.
Preconditions	User has a decided quantity and strategy to trade with.
Postconditions	System generates new orders based on specified quantity.
Normal Flow	User indicates trading strategy, system accepts trade strategy, generates the appropriate bids and ask orders and sends it to the trade engine.

Use Case 3: Generating Trades for the Day

Actors	User
Triggers	User has chosen trading strategy and sent it to engine.
Preconditions	Algorithmic trades are ready to process and a set of orders have already been read in.
Postconditions	All algorithmic trades are executed and ready for evaluation.
Normal Flow	User tells system to continue with selected trades. System executes the algorithmic trades and prepares for evaluation.

Use Case 4: Evaluating Algorithmic Trades

Actors	User
Triggers	User wishes to determine the effectiveness of the strategy
Preconditions	Algorithmic trades have been processed.
Postconditions	The user is given feedback based on the algorithmic trades.
Normal Flow	User asks for an evaluation, system finds the percentage gained or lost.

Use Case 5: Generating Strategy Performance Report

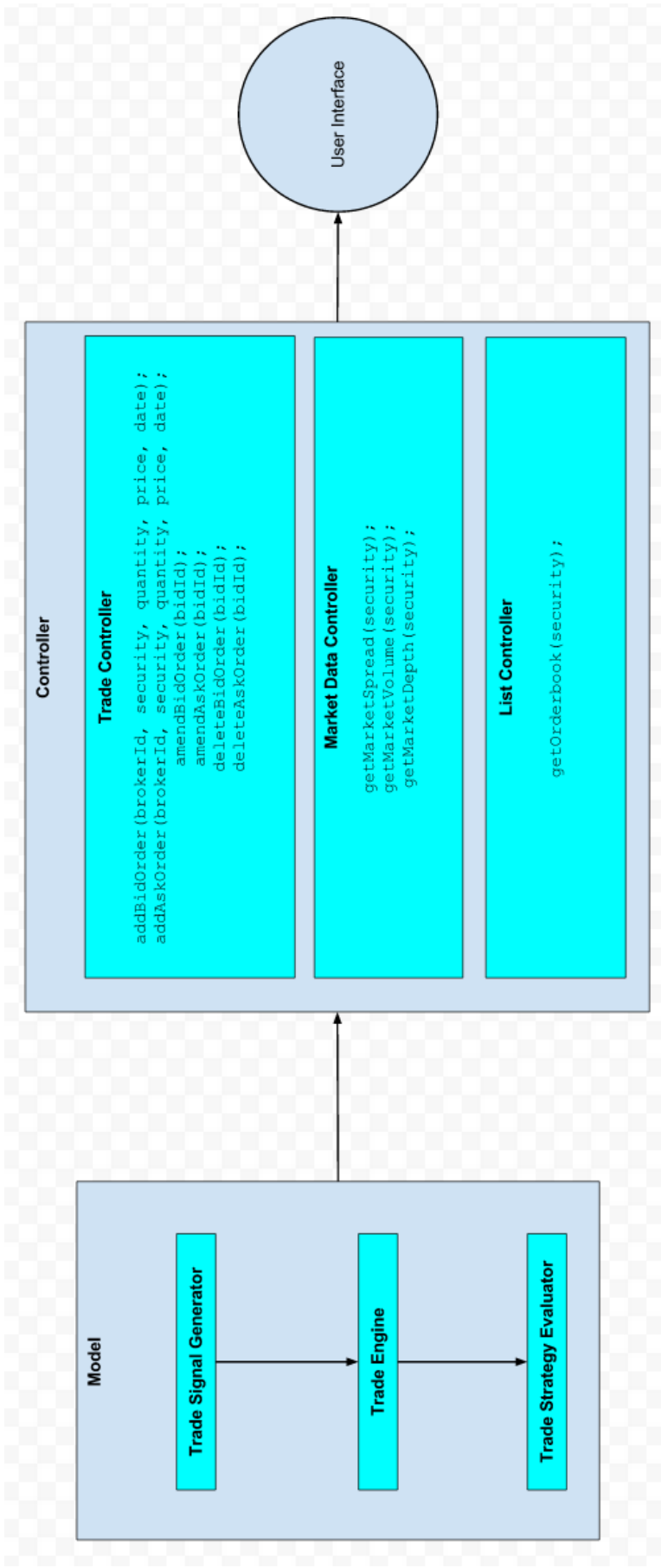| Actors	User
Triggers	User wants to see a performance report
Preconditions	A strategy has been executed.
Postconditions	A report will be generated for the user.
Normal Flow	User asks system to generate strategy performance report. System provides a performance report to the user.

3 Architecture Diagram

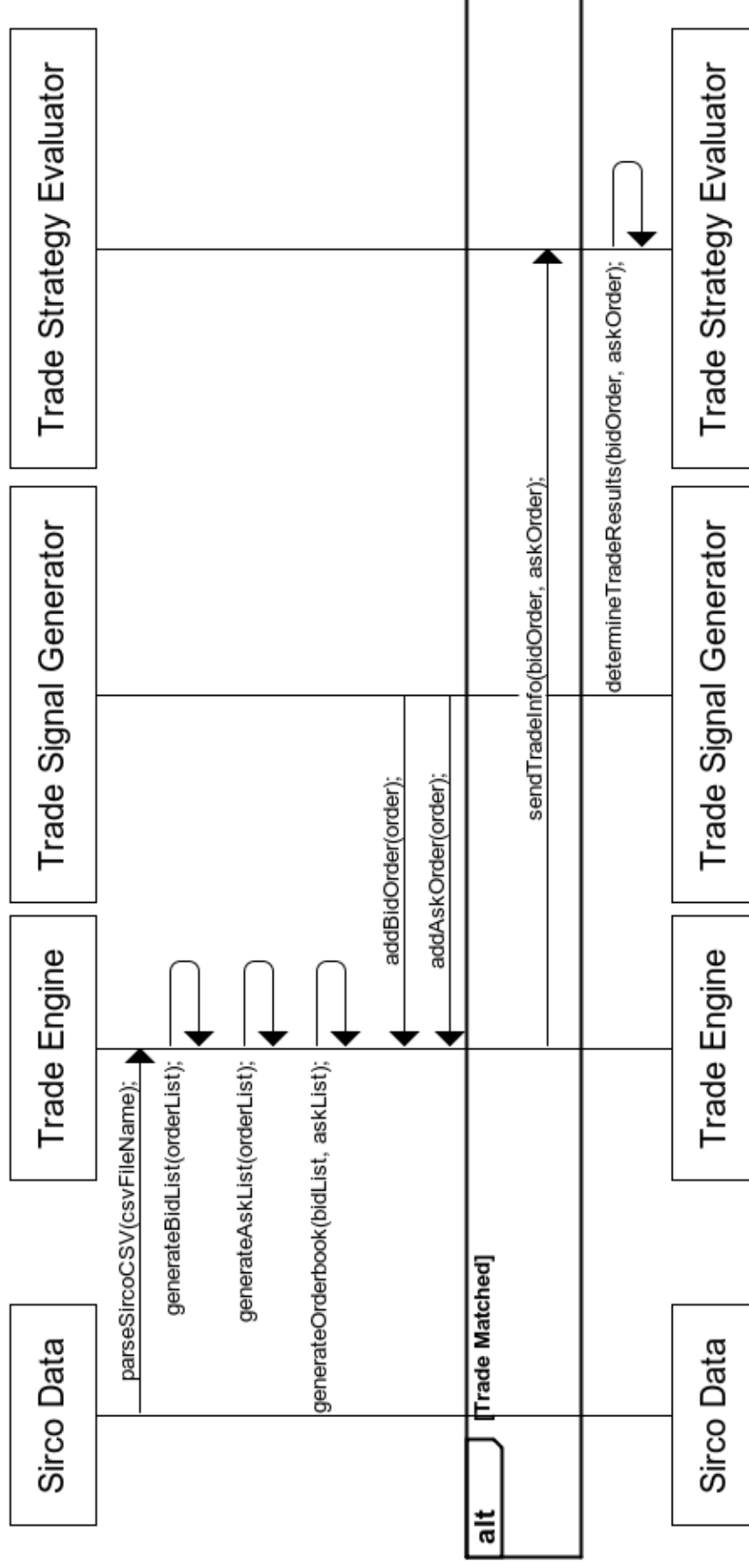Figure 1: Architectural Diagram.

Architectural Diagram (Model)

Sirco Data	Trade Engine	Trade Signal Generator	Trade Strategy Evaluator

parseSircoCSV(csvFileName);

generateBidList(orderList);

generateAskList(orderList);

generateOrderbook(bidList, askList);

addBidOrder(order);

addAskOrder(order);

alt [Trade Matched]

sendTradeInfo(bidOrder, askOrder);

determineTradeResults(bidOrder, askOrder);

Sirco Data	Trade Engine	Trade Signal Generator	Trade Strategy Evaluator

Figure 2: Model Architectural Diagram.

4 Sequence Diagram

The following are the Sequence Diagrams created created in parallel to the Use Cases created by our team.
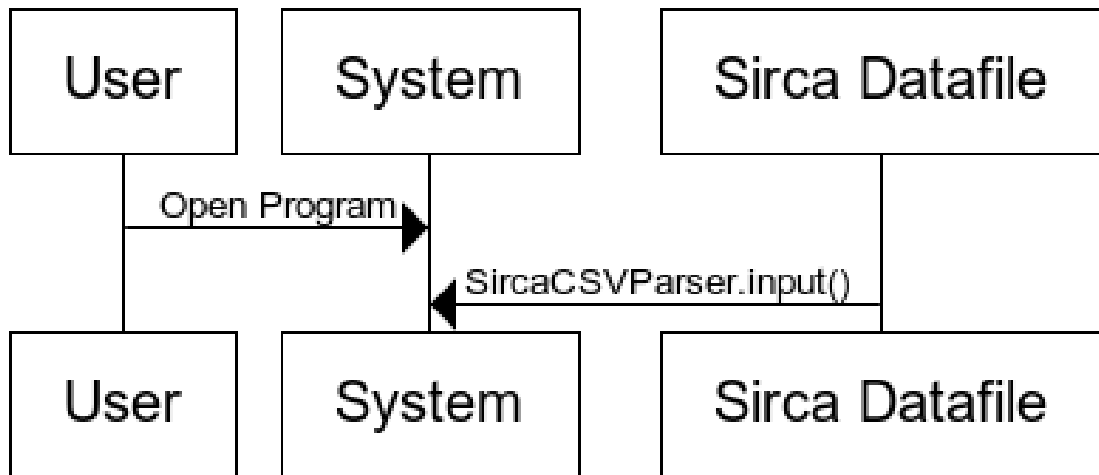
Figure 3: Use Case 1: Reading correctly formatted Sirca files
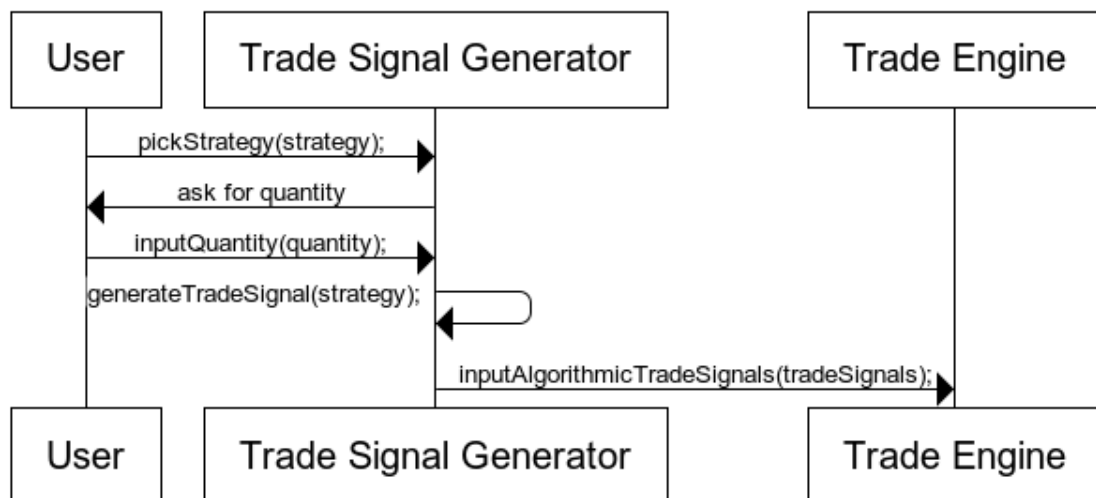

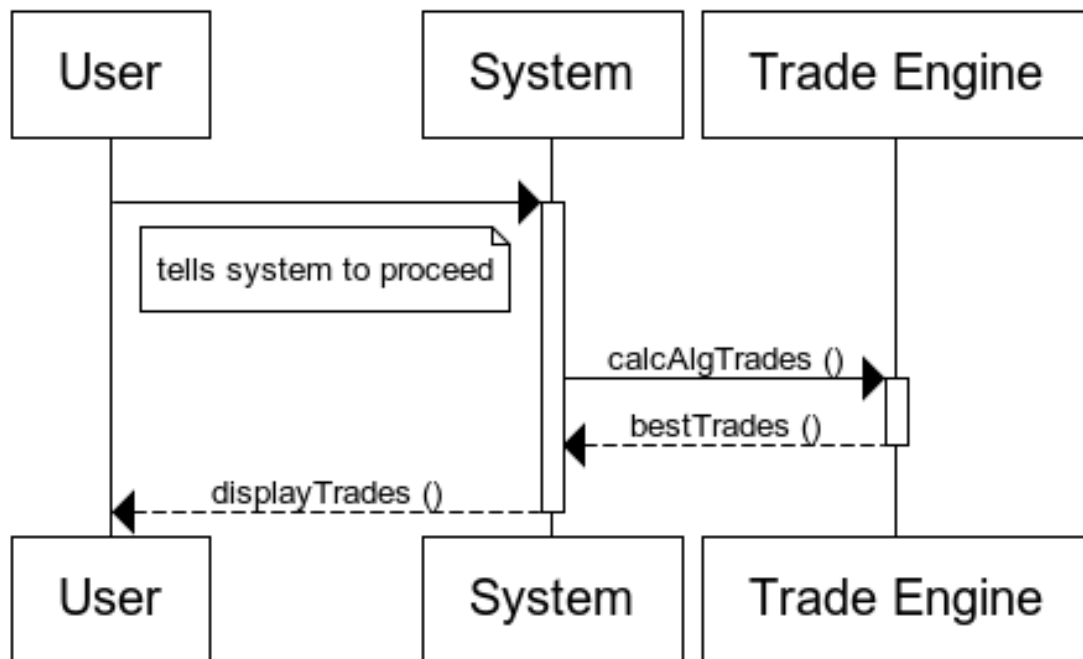Figure 4: Use Case 2: Choosing an Algorithmic Trading Strategy

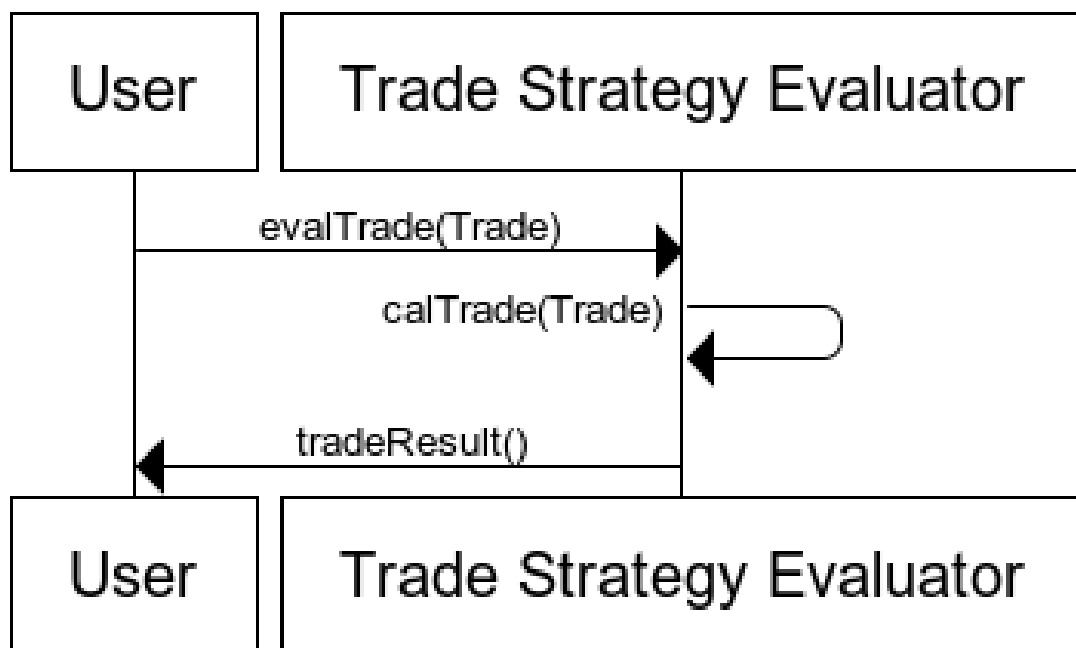Figure 5: Use Case 3: Generating Trades for the Day
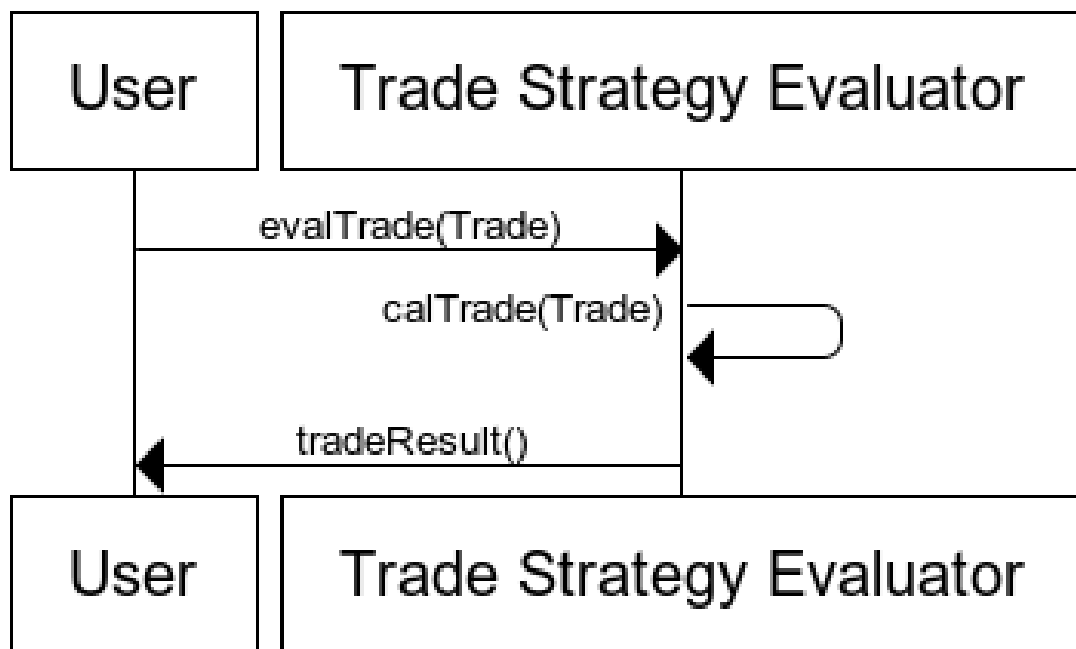

Figure 6: Use Case 4: Evaluating Algorithmic Trades

Figure 7: Use Case 5: Generating Strategy Report

5 Justification of chosen Language and Environment

Development Environment:

Our language of choice is Java. Our decision to use Java is because of the extensive amount of open source resources available for Java. Two resources we have already planned to use are OpenCSV to read in the Sirco data and Swing to write our Graphical User Interface. Additionally, we are more confident with Java in larger sized projects. Java also has an extensive amount of external resources and documentation.

Collaboration and version control:

We have decided to make use of Git and Github to maintain version control and collaborate our progress. We chose Git as it is a powerful version control tool and Github provides an informative interface to show past commits and if required, separate branches of our project.

Environment:

Everyone will be developing in Eclipse as it is quite a versatile and powerful open source IDE. It is also extensively integrated with JUnit and will reduce some of the work required to set up a testing environment for our system.

6 Project Plan

Communication:

1. We will meet up every Tuesday, Wednesday and Friday for face to face updates and progress checking.

2. For most other times we will collaborate through our Facebook group page and on Skype if conference calls are needed.

3. Google docs and other cloud software like LucidChart will be used for collaboration on report writing and diagrams.

Development Tools:

1. Eclipse is our IDE of choice.
2. Our code will be in Java 6.
3. Open source libraries like OpenCSV and Swing will be used.

Version Control:

We are using Git to keep our projects up to date.

Roles:

Sohaib Mushtaq - Developer, Tester
Shanku Roy - Developer, Quality Assurance
Michael Vuong - Developer, LaTeX/Report Generator
Albert Wang - Developer, Team Lead