

**SENG3011**  
**Initial 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 Envrioment

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Use Cases</b>	<b>2</b>
<b>3</b>	<b>Architecture Diagram</b>	<b>4</b>
<b>4</b>	<b>Sequence Diagram</b>	<b>6</b>
<b>5</b>	<b>Justification of chosen Language and Environment</b>	<b>10</b>
<b>6</b>	<b>Project Plan</b>	<b>10</b>

## List of Figures

1	Architectural Diagram. . . . .	4
2	Model Architectural Diagram. . . . .	5
3	Use Case 1: addBidOrder . . . . .	6
4	Use Case 2: addAskOrder . . . . .	6
5	Use Case 3: amendBidOrder . . . . .	7
6	Use Case 4: amendAskOrder . . . . .	8
7	Use Case 5: deleteBidOrder . . . . .	9
8	Use Case 6: deleteAskOrder . . . . .	10

# 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. Placing a Bid Order
2. Placing an Ask Order
3. Amending a Bid Order
4. Amending an Ask Order
5. Deleting a Bid Order
6. Deleting an Ask Order

### Use Case 1: Placing a bid Order

<b>Actors</b>	Broker
<b>Triggers</b>	The broker indicates they want to place a bid order.
<b>Preconditions</b>	The broker has selected a price and quantity for the specific security.
<b>Postconditions</b>	The order will be placed in the system. The broker will have a Order ID for the bid.
<b>Normal Flow</b>	Broker decides on a bid price and quantity for a specific security. System accepts bid with bid price, appends to bid list and Order Book. System returned generated order id to broker.

### Use Case 2: Placing an Ask Order

<b>Actors</b>	Broker
<b>Triggers</b>	The broker indicates they want to place an ask order.
<b>Preconditions</b>	The broker has selected a price and quantity for the specific security.
<b>Postconditions</b>	The order will be placed in the system. The broker will have a Order ID for the ask order.
<b>Normal Flow</b>	Broker decides on an ask price and quantity for a specific security. System accepts with ask price, appends to ask list and Order Book. System returned generated Order ID to broker.

### Use Case 3: Amending a Bid Order

<b>Actors</b>	Broker
<b>Triggers</b>	The broker indicates they want to amend a bid order.
<b>Preconditions</b>	The broker has indicated a change in the original price or quantity for the specific security. The bid Order ID will already be in the system.
<b>Postconditions</b>	The broker will have a new Order ID for amended Order.
<b>Normal Flow</b>	Broker decides on a change on price or quantity for original bid order. System accepts the new price/quantity. Cancels the old Order ID. Creates a new Order ID.

**Use Case 4: Amending an Ask Order**

<b>Actors</b>	Broker
<b>Triggers</b>	The broker wants to change the price or amount of their existing ask order.
<b>Preconditions</b>	The broker has an existing ask Order. The broker wants to change the ask price or ask amount.
<b>Postconditions</b>	The previous Order gets deleted. A new order is created with the new ask price and amount.
<b>Normal Flow</b>	The broker wants to update an existing order so the previous order is deleted and an new order is created. This prevents cheating as orders are prioritised by time, and a broker can take advantage of their placement in the queue to match a sell order.

**Use Case 5: Deleting a Bid Order**

<b>Actors</b>	Broker
<b>Triggers</b>	The broker indicates they want to remove a bid order.
<b>Preconditions</b>	The broker has an existing bid in the system. The broker knows the Order ID.
<b>Postconditions</b>	The Order will be removed from the system.
<b>Normal Flow</b>	Broker decides to remove their bid order from the system. System accepts the Order ID to remove. Bid list and order book are updated.

**Use Case 6: Deleting an Ask Order**

<b>Actors</b>	Broker
<b>Triggers</b>	The broker indicates they want to remove an ask order.
<b>Preconditions</b>	The broker has an existing ask in the system. The broker knows the Order ID.
<b>Postconditions</b>	The Order will be removed from the system.
<b>Normal Flow</b>	Broker decides to remove their ask order from the system. System accepts the Order ID to remove. Ask list and order book are updated.

3 Architecture Diagram

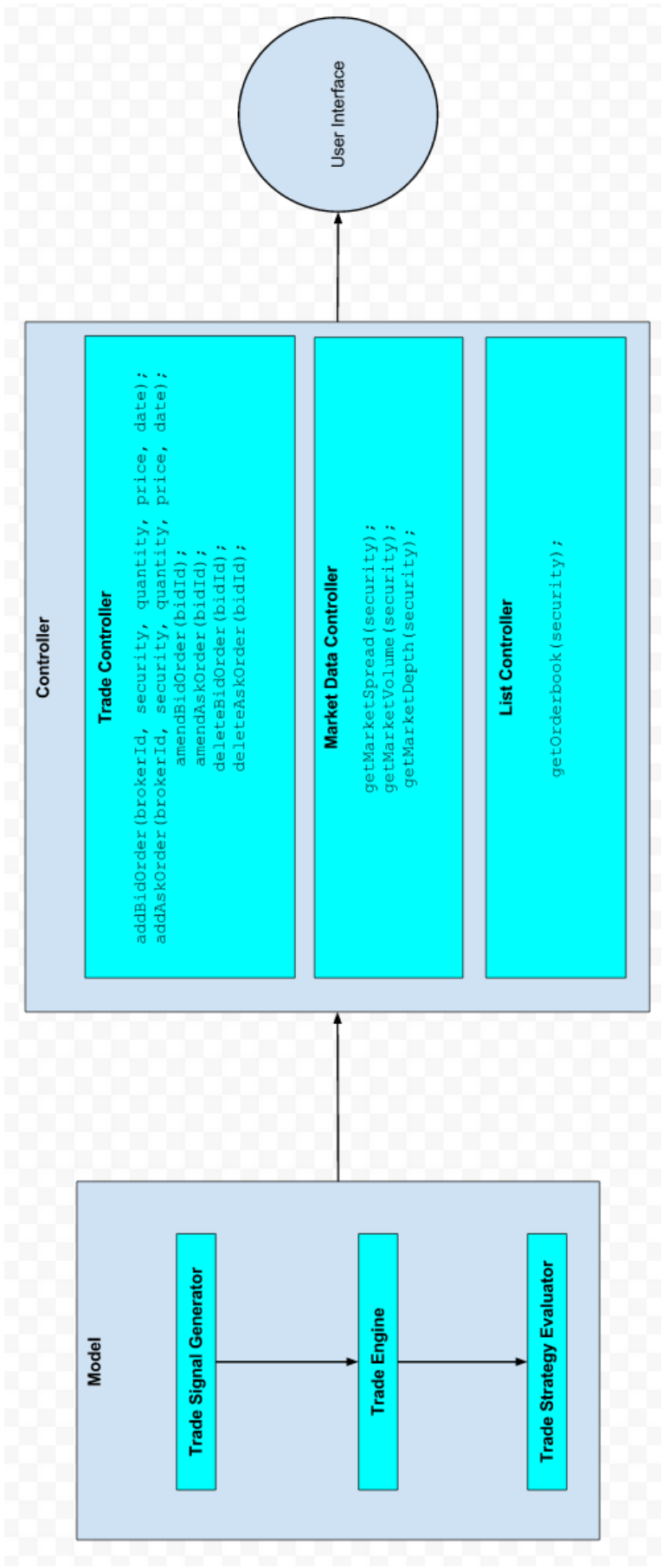


Figure 1: Architectural Diagram.



Architectural Diagram (Model)

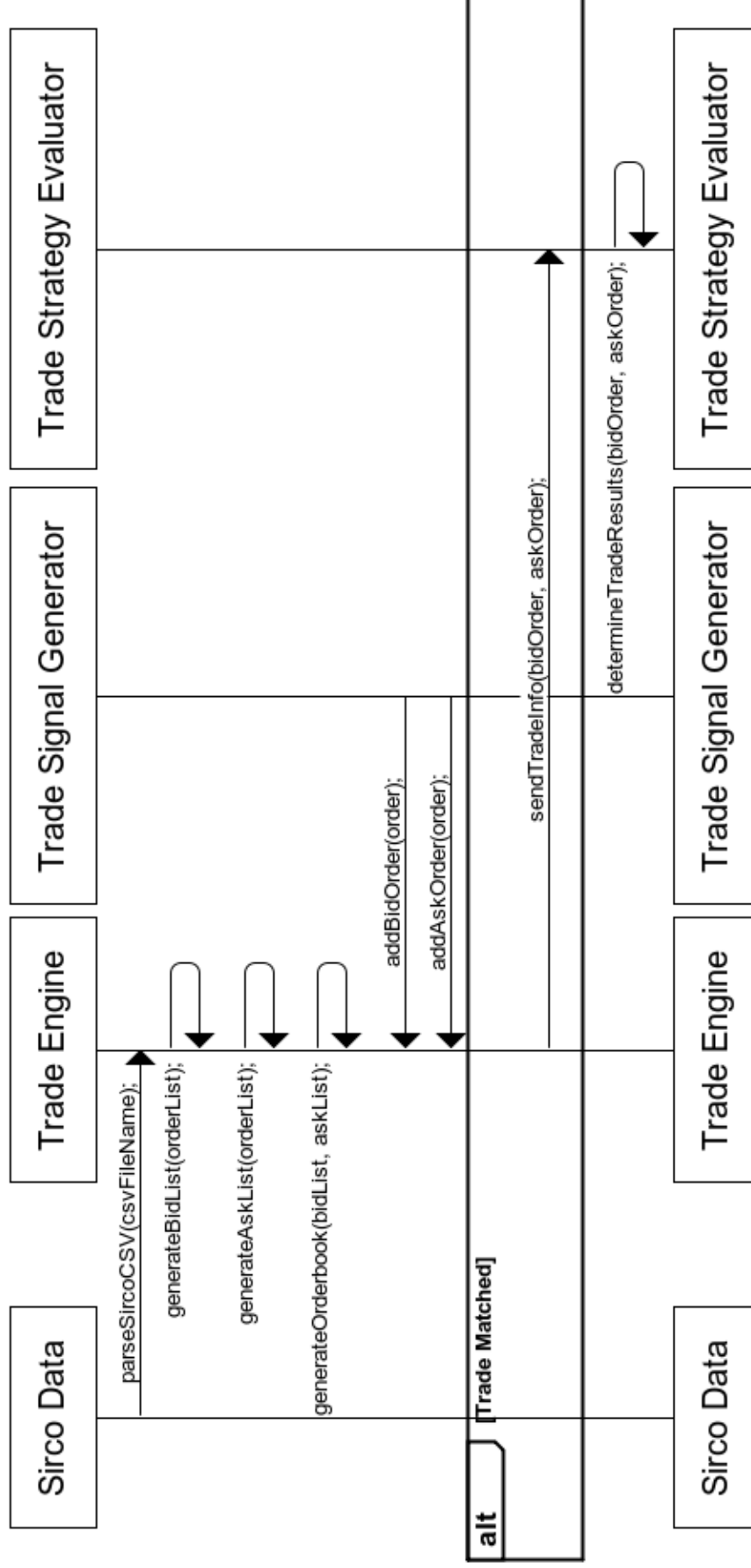


Figure 2: Model Architectural Diagram.

## 4 Sequence Diagram

The following are the Sequence Diagrams created in parallel to the Use Cases created by our team.

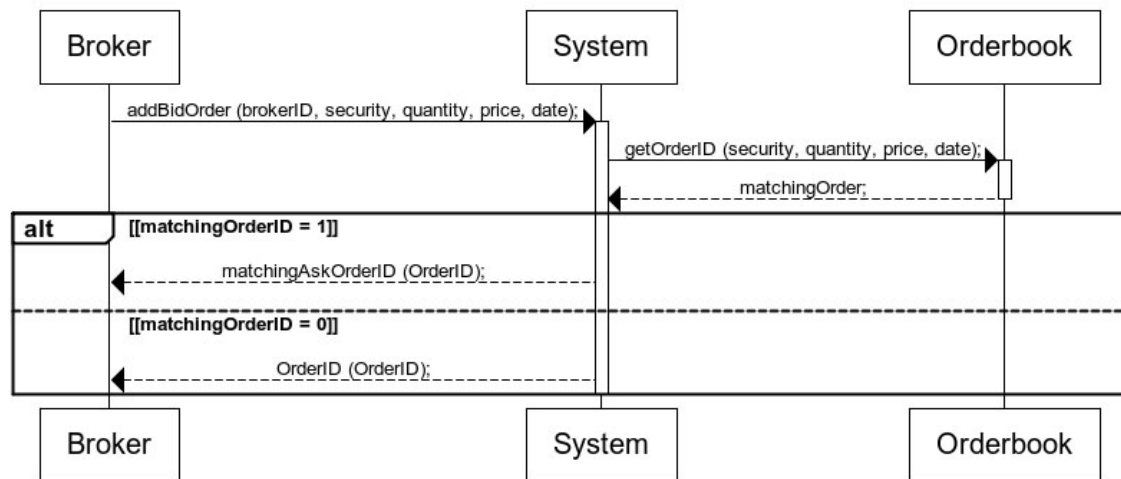


Figure 3: Use Case 1: addBidOrder

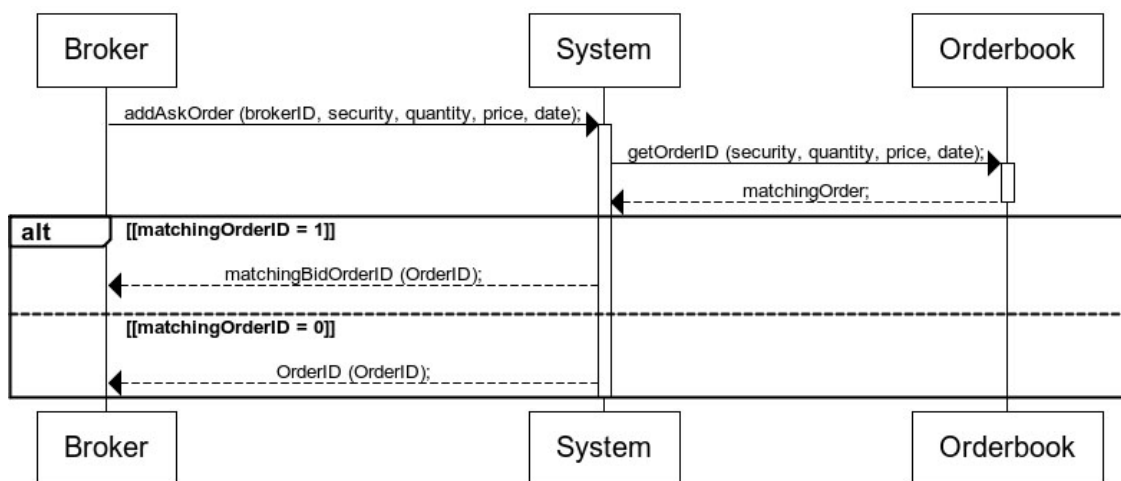


Figure 4: Use Case 2: addAskOrder

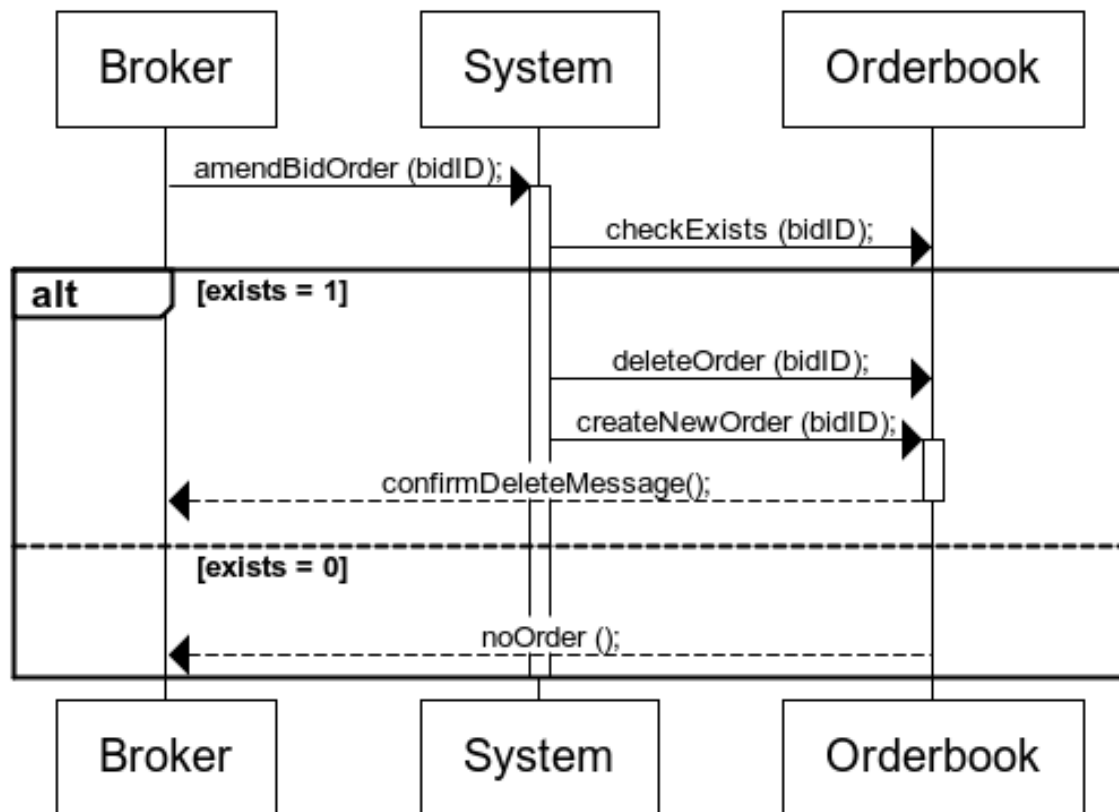


Figure 5: Use Case 3: amendBidOrder

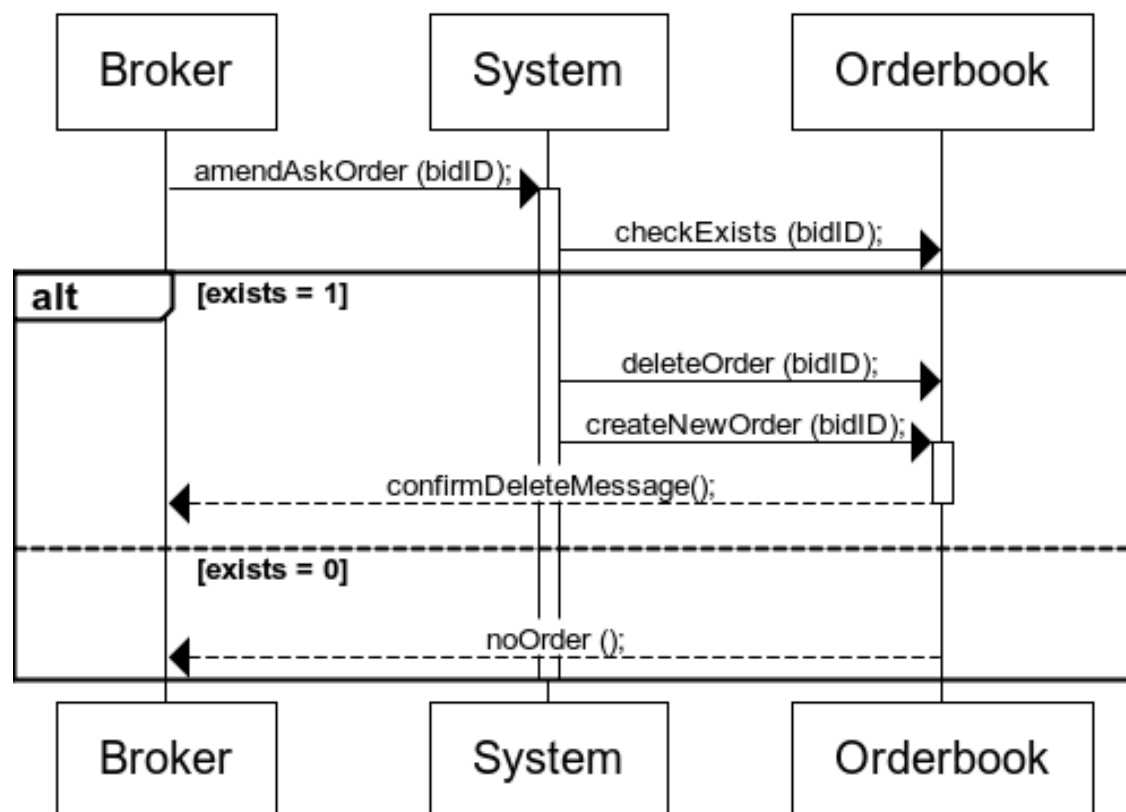


Figure 6: Use Case 4: amendAskOrder

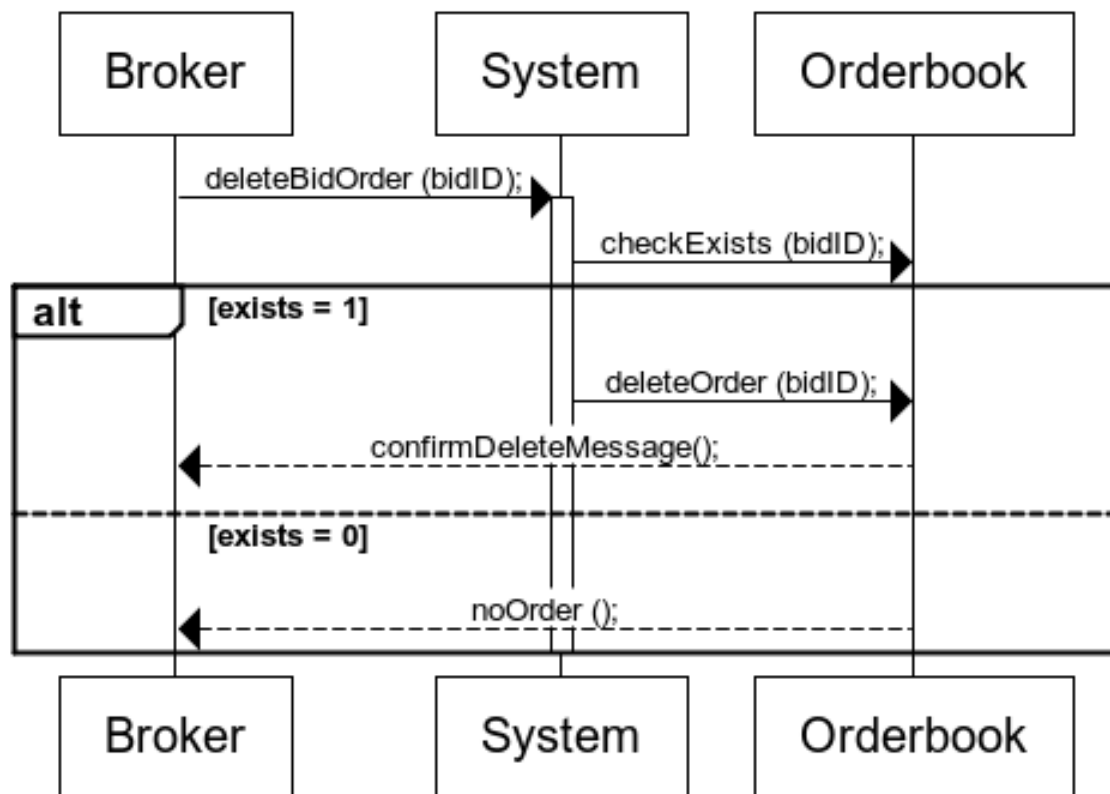


Figure 7: Use Case 5: deleteBidOrder

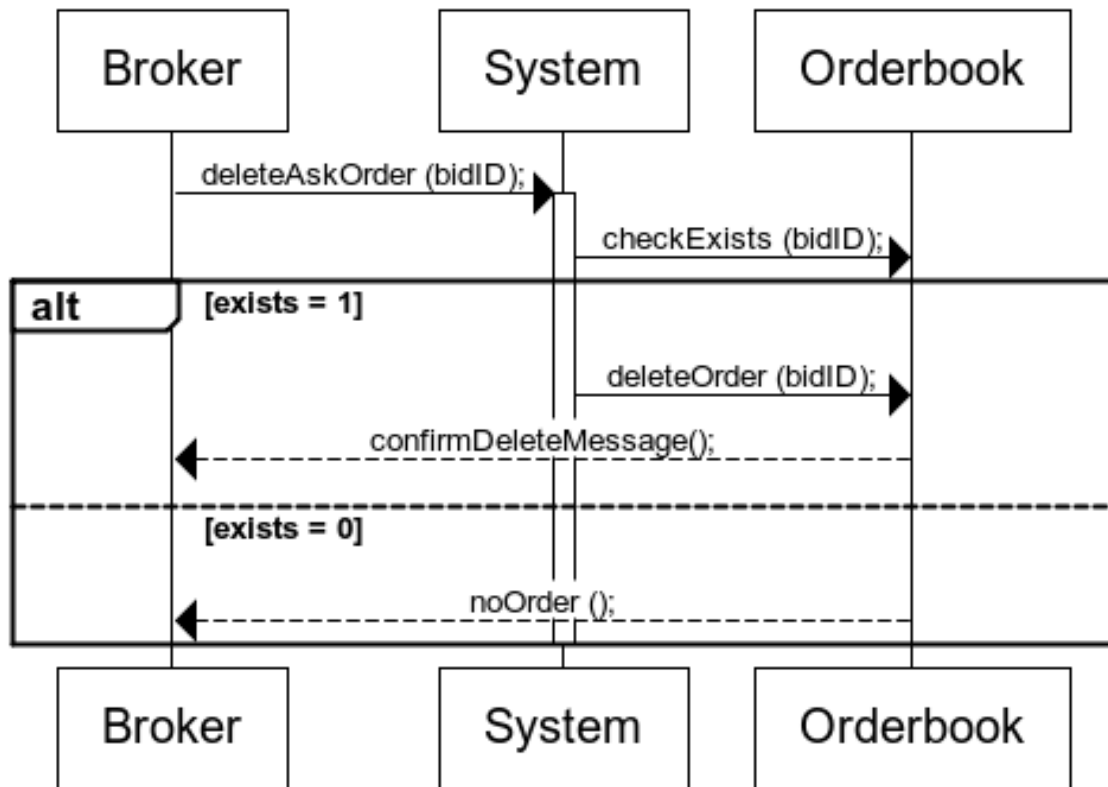


Figure 8: Use Case 6: deleteAskOrder

## 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