

Process MeNtOR 3.0

Cryptocoin Trading System Requirements Model

Version:	1.5
Print Date:	February 16th, 2022
Release Date:	
Release State:	Core
Approval State:	Draft
Approved by:	
Prepared by:	Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Tamer Ali Mohamed
Reviewed by:	
Path Name:	
File Name:	
Document No:	

Document Change Control

Version	Date	Authors	Summary of Changes
1.1	January 31st, 2022	Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	Document creation/initial write and rough draft domain model with brainstormed ideas. Updated group meetings logs.
1.2	February 2nd, 2022	Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	Edited and finalized domain model. Partially inserted introduction. Added gantt chart. Updated group meeting logs.
1.3	February 7th, 2022	Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	Inserted initial sequence, non-functional requirements, product backlog and finalized domain model class definitions. Updated group meeting logs.
1.4	February 13th, 2022	Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	Inserted finalized sequence and activities diagrams. Updated project backlog and group meeting logs.
1.5	February 14th, 2022	Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	Edited the whole project and checked in order to submit. Updated group meeting logs

Document Sign-Off

Name (Position)	Initials	Date
Ian Guenther Green	IGG	January 31st, 2022
Vicky Jiang	VJ	January 31st, 2022
Hala Elewa	HE	January 31st, 2022
Ali Mohamed	AM	January 31st, 2022
Ian Guenther Green	IGG	February 2nd, 2022
Vicky Jiang	VJ	February 2nd, 2022
Hala Elewa	HE	February 2nd, 2022
Ali Mohamed	AM	February 2nd, 2022
Ian Guenther Green	IGG	February 7th, 2022
Vicky Jiang	VJ	February 7th, 2022
Hala Elewa	HE	February 7th, 2022
Ali Mohamed	AM	February 7th, 2022

Ian Guenther Green	IGG	February 13th, 2022
Vicky Jiang	VJ	February 13th, 2022
Hala Elewa	HE	February 13th, 2022
Ali Mohamed	AM	February 13th, 2022
Ian Guenther Green	IGG	February 15th, 2022
Vicky Jiang	VJ	February 15th, 2022
Hala Elewa	HE	February 15th, 2022
Ali Mohamed	AM	February 15th, 2022

Contents

Introduction	6
Purpose	6
Overview	6
References	6
Domain Model	7
Domain Model Class Diagram	7
Domain Model Class Definitions	7
User	7
User Database	8
Login Interface	8
User Interface (UI)	8
Trading Client Action Viewer	9
Cryptocurrency	9
CoinGecko Cryptocurrency Repository	9
Trading Broker	10
Trading Strategy	10
Trade Result	10
Trade Action Log	10
Sequence and Activity Diagrams	11
Sequencing Diagrams	11
Sequence Diagram for Use-Case-1	11
Sequence Diagram for Use-Case-2	12
Sequence Diagram for Use-Case-3	13
Sequence Diagram for Use-Case-4	13
Activity Diagrams	14
Activity Diagram for Use-Case-1	14
Activity Diagram for Use-Case-2	15
Activity Diagram for Use-Case-3	16
Activity Diagram for Use-Case-4	17
Non-Functional Requirements Specification	18
Overview	18
Activities Plan	20

Gantt Chart	20
Project Backlog	21
Group Meeting Logs	22
Test Driven Development	24

1 Introduction

1.1 Purpose

This document details the requirements of a software system that allows users to trade various types of cryptocurrencies based on set trading strategies where the results are displayed onto the screen. This is done to make brokers jobs easier and to provide strategies that they choose to fulfill their needs. The coins entered for a trading broker will be retrieved using the CoinGecko database where they will be traded based on the strategy entered and outputted onto the screen in the form of a table and histogram. A domain model is used to showcase the entire system, with sequence and activity diagrams addressing how each use case will work. The document will also address the non-functional requirements of the system as well as initial test cases for the software.

1.2 Overview

The goal of the project is to implement a sample system that allows the user to choose the best strategy to buy and sell crypto coins with convenience. To enter the program, the user will need to verify their identity by logging in with a username-password combination.

The program requires the user to add and delete trading brokers. When a trading broker is added, information, such as their name, preferred cryptocurrencies, and strategy, will be associated with that particular broker. No two brokers can have the same name.

The program can perform trades based on the chosen strategy of each broker. The system will retrieve the data of the cryptocurrency by implementing CoinGecko's published API. Therefore, prices will be updated in real time and the program can compute the result of the trades.

The system will store every log of the trading activity of each user and display it in a histogram and a table showing the name, strategy, coin traded and other information to help the system with providing the best environment for each user.

The non-functional requirements will be hard disk space of 8GB to install Windows. The user will have everything they need in order to finish what they want to do in either cases, buy or sell or even add and delete some actions.

1.3 References

CG-API: <https://www.coingecko.com/api/documentations/v3>

CG-TERMS: <https://www.coingecko.com/en/glossary>

Eclipse: <http://www.eclipse.org/downloads/index.php>

Maven: <https://maven.apache.org/download.cgi>

RESTAPI: <https://www.w3schools.in/restful-web-services/intro/>

2 Domain Model

2.1 Domain Model Class Diagram

The domain model class diagram for the Cryptocurrency Trading System appears below:

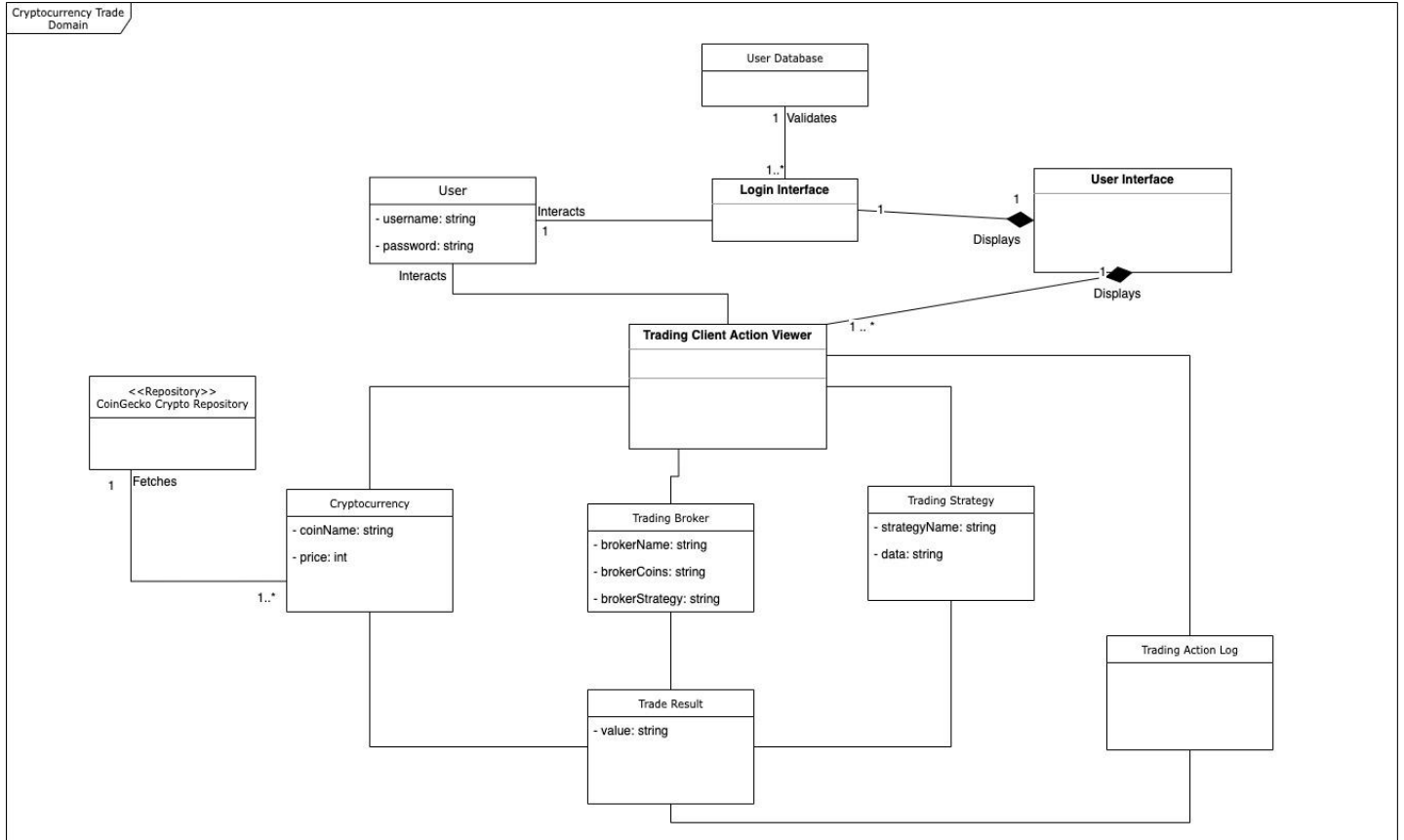


Fig. 2.1 Domain Model Class Diagram

2.2 Domain Model Class Definitions

2.2.1 User

Description	This object is the human user that will be using the program. The user will be able to log into the system and select criteria, such as the currency, trading client, and strategy, to implement a trade. It does through interacting with the User Interface.
Attributes	username: string password: string
Responsibilities	The user must provide their credentials, username and password, to be verified and enter the trading program. It is responsible for selecting the required information and performing the trades by interacting with the user interface.
Business Rules	

2.2.2 User Database

Description	This database, in the form of a simple text file, will store the username and password combinations to verify the identity of the user. This ensures that valid users are allowed to access the trading program.
Attributes	
Responsibilities	Stores data related to the users including the username and password. If the username-password combination is correct, then the main UI of the application will be displayed.
Business Rules	

2.2.3 Login Interface

Description	This interface will allow the user to enter their username and password. Once the information is verified by the user database, it will lead the user to enter the main UI.
Attributes	
Responsibilities	It is responsible for accepting the information that is provided by the user from the keyboard and transferring that data to compare with the user database. Once the username and password combination is verified, it will lead the user to the main UI. If the combination is not verified, it will notify the user with an invalid login window and allow the user to try again.
Business Rules	

2.2.4 User Interface (UI)

Description	The user interface object is the display that the user interacts with so that the user can initiate instructions for the program to proceed with the system's tasks. The UI displays the data that the user is seeking after and also produces the results.
Attributes	
Responsibilities	The UI has three main responsibilities. Firstly, it needs to allow the user to send in instructions to the system such as inputting the username/password combination in the Login interface. Second it allows users to select the trade information in the Trading Client Action Viewer interface. Lastly, it's responsible for displaying the trading results data from the system for the user in the form of a table and a histogram.
Business Rules	

2.2.5 Trading Client Action Viewer

Description	The Trading Client Action Viewer interface serves by showing data and also displaying the results to the Main UI. It is notified by the trading log whether there is new data to be displayed. It is instructed by the end user to perform an instruction.
Attributes	
Responsibilities	This interface allows the user to select criteria and indicate to the system to perform an instruction. It also allows the user to visualize the history of trades that have and have not occurred.
Business Rules	

2.2.6 Cryptocurrency

Description	The cryptocurrency class represents the various cryptocurrency values that will be retrieved from the CoinGecko Repository. Each instance of the object is responsible for two main attributes about each coin: the coin name and the price of the coin.
Attributes	coinName: string price: int
Responsibilities	The object is responsible for providing information on the cryptocoin such as the name and the price. The user will be able to select the various options of cryptocurrencies and the trade result object will use the information from the cryptocoin to conduct the analysis of whether or not the trade is performed.
Business Rules	

2.2.7 CoinGecko Cryptocurrency Repository

Description	The CoinGecko repository is the data repository offered by CoinGecko and stores the values related to each cryptocoin. The system accesses this data using the published REST API.
Attributes	
Responsibilities	It stores data related to the cryptocoin such as the current price and the name. It only accesses the data for the cryptocurrency that is required and asked for by the system.
Business Rules	

2.2.8 Trading Broker

Description	This object represents the trading broker information that will be used to make trades. Each instance of this object is responsible for holding key attributes about each trading broker.
Attributes	brokerName: string
Responsibilities	A trading broker object is responsible for providing information on a trading broker that is associated with a specific name. This object is passed through the Trade Result object to see whether a trade can be made.
Business Rules	

2.2.9 Trading Strategy

Description	This object represents the trading strategy that will have information that will be used to make trades. Each instance of this object is responsible for holding key attributes about each trading strategy.
Attributes	strategyName: string
Responsibilities	A trading strategy object is responsible for providing information on the trading strategy criteria that is associated with a specific name. This object is passed through the Trade Result object to see whether a trade can be made.
Business Rules	

2.2.10 Trade Result

Description	This object represents the result of the trade that is produced.
Attributes	result: string
Responsibilities	It is responsible for tracking the result of the trade and being added to the trade action log and then used to display the results in the UI.
Business Rules	

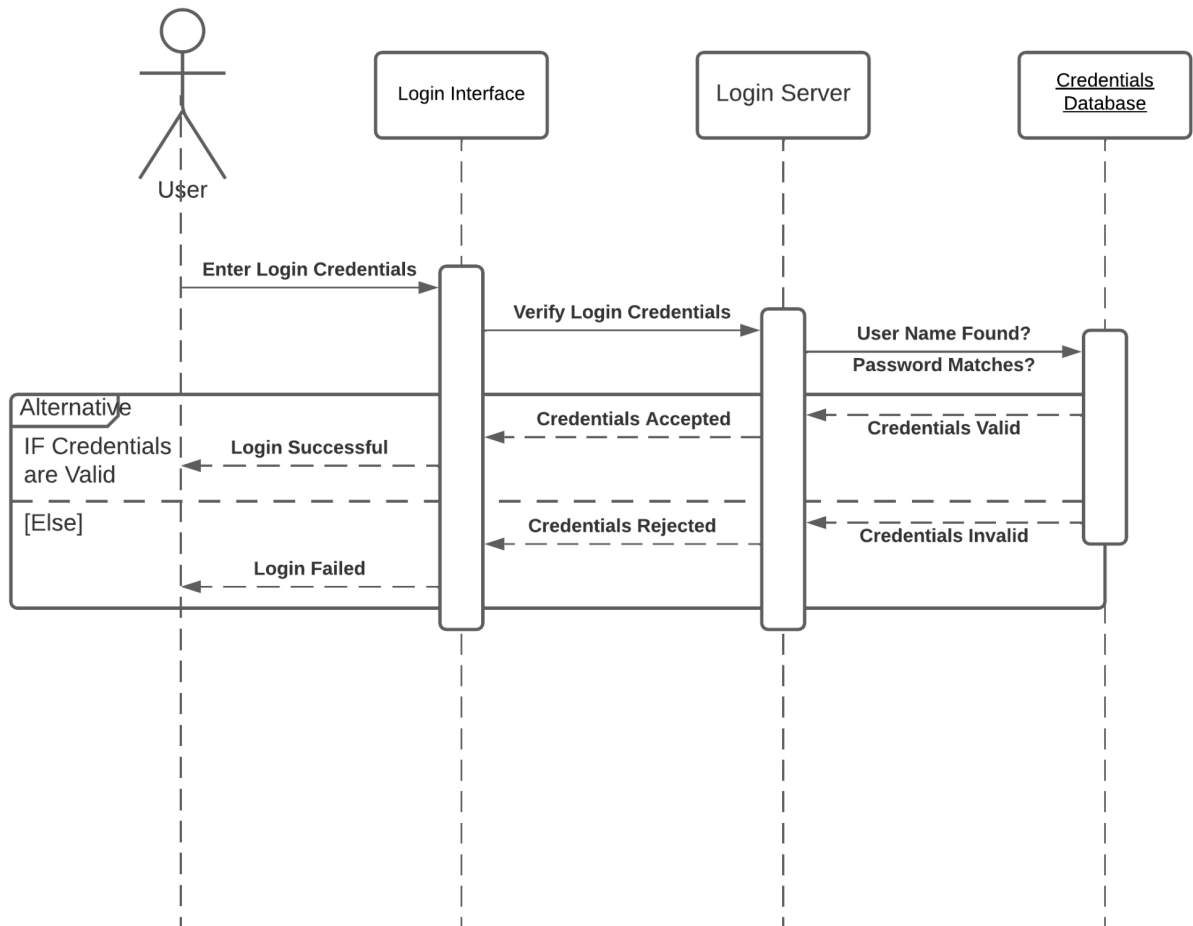
2.2.11 Trade Action Log

Description	The trade action log is an accumulation of all the trade results so far. It is used by the processor to gather the data and display the results in a graph and table in the Trading Client Action Viewer.
Attributes	
Responsibilities	This object is responsible for displaying the visual and graphical information in the UI. It is notified by the UI server to create the view based on the data and returns the table and graphs back to the end user.
Business Rules	

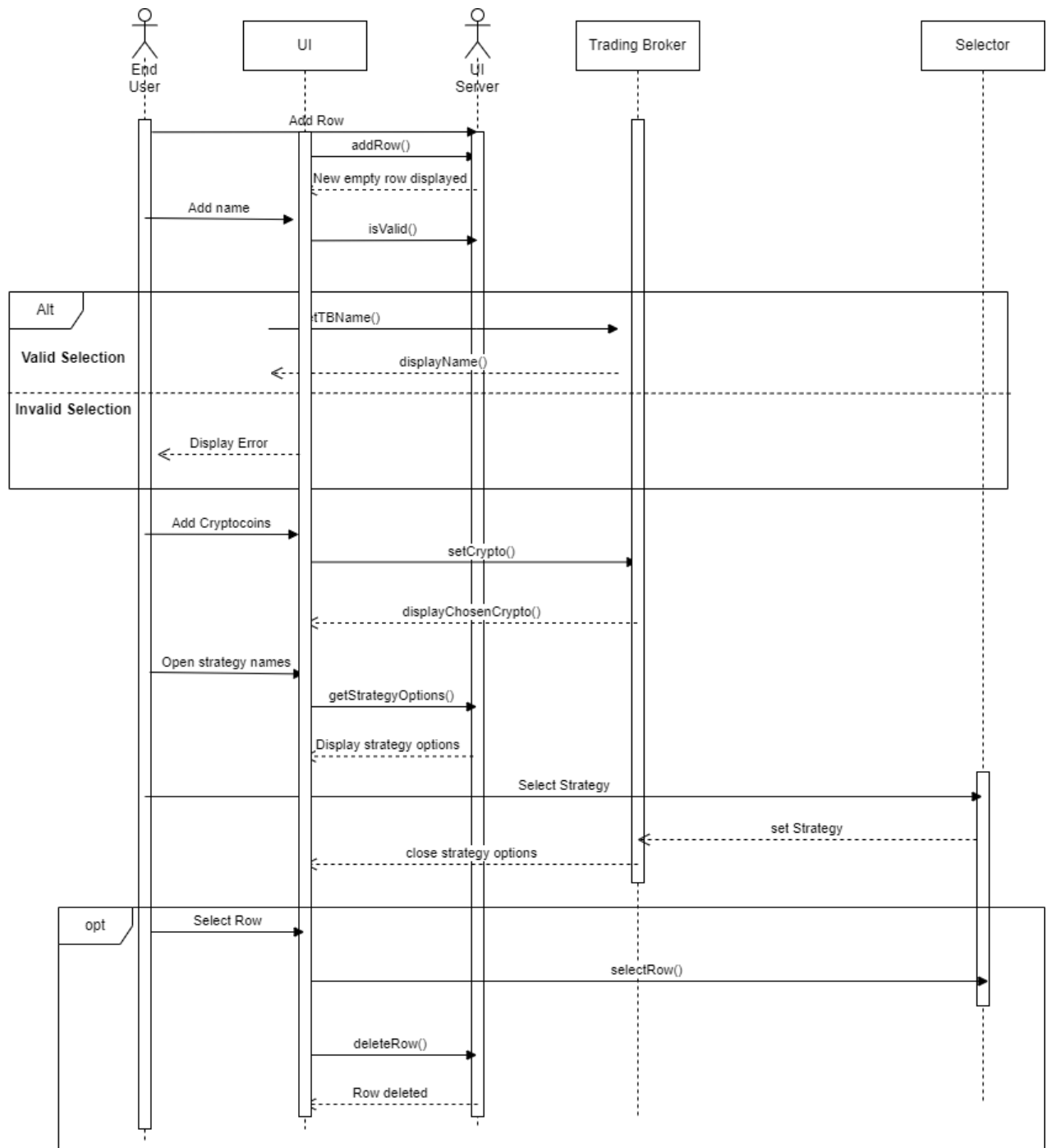
3 Sequence and Activity Diagrams

3.1 Sequencing Diagrams

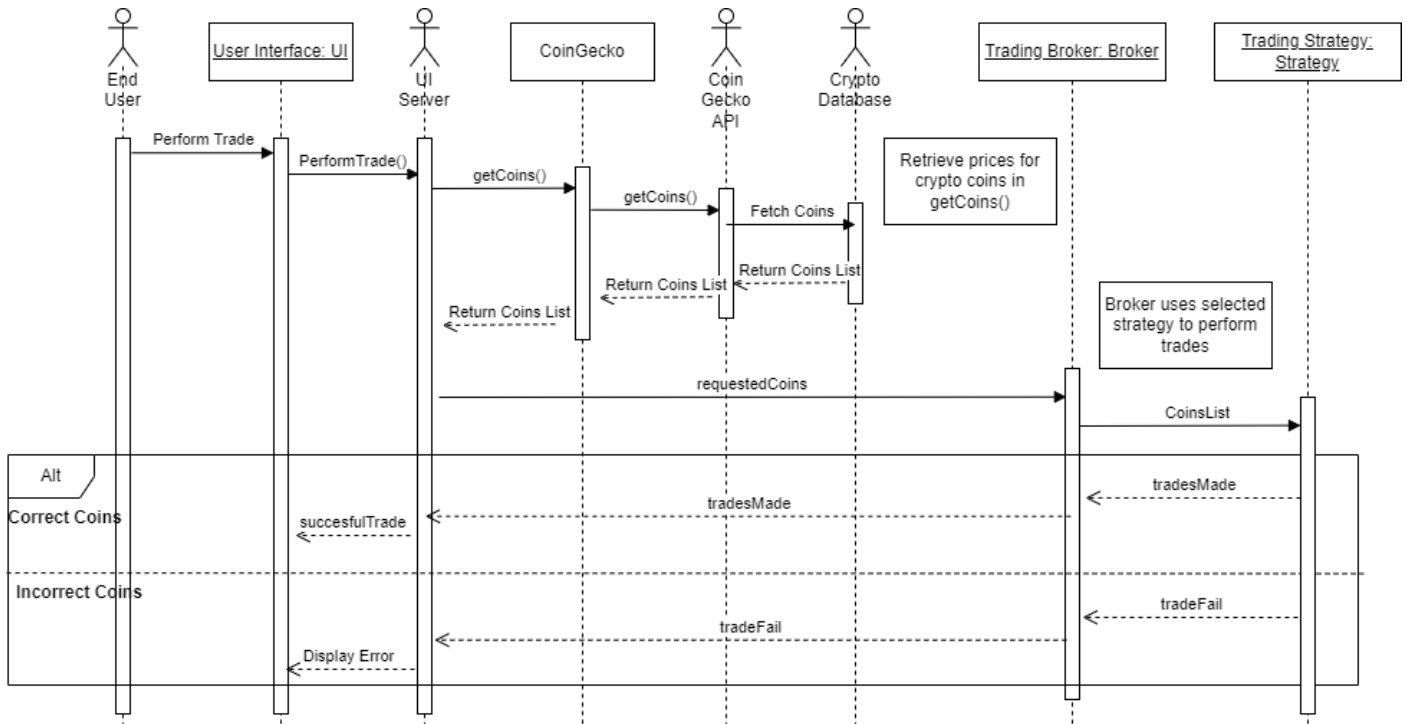
3.1.1 Sequence Diagram for Use-Case-1



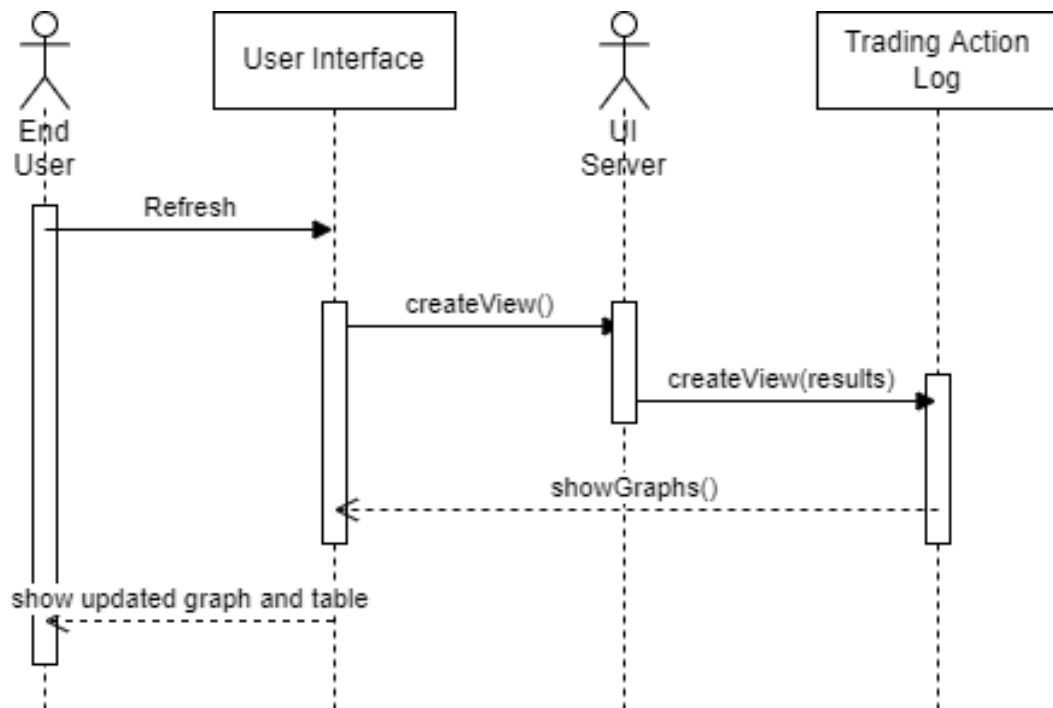
3.1.2 Sequence Diagram for Use-Case-2



3.1.3 Sequence Diagram for Use-Case-3

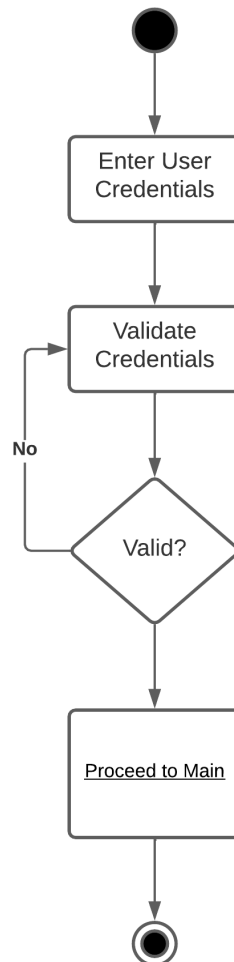


3.1.4 Sequence Diagram for Use-Case-4

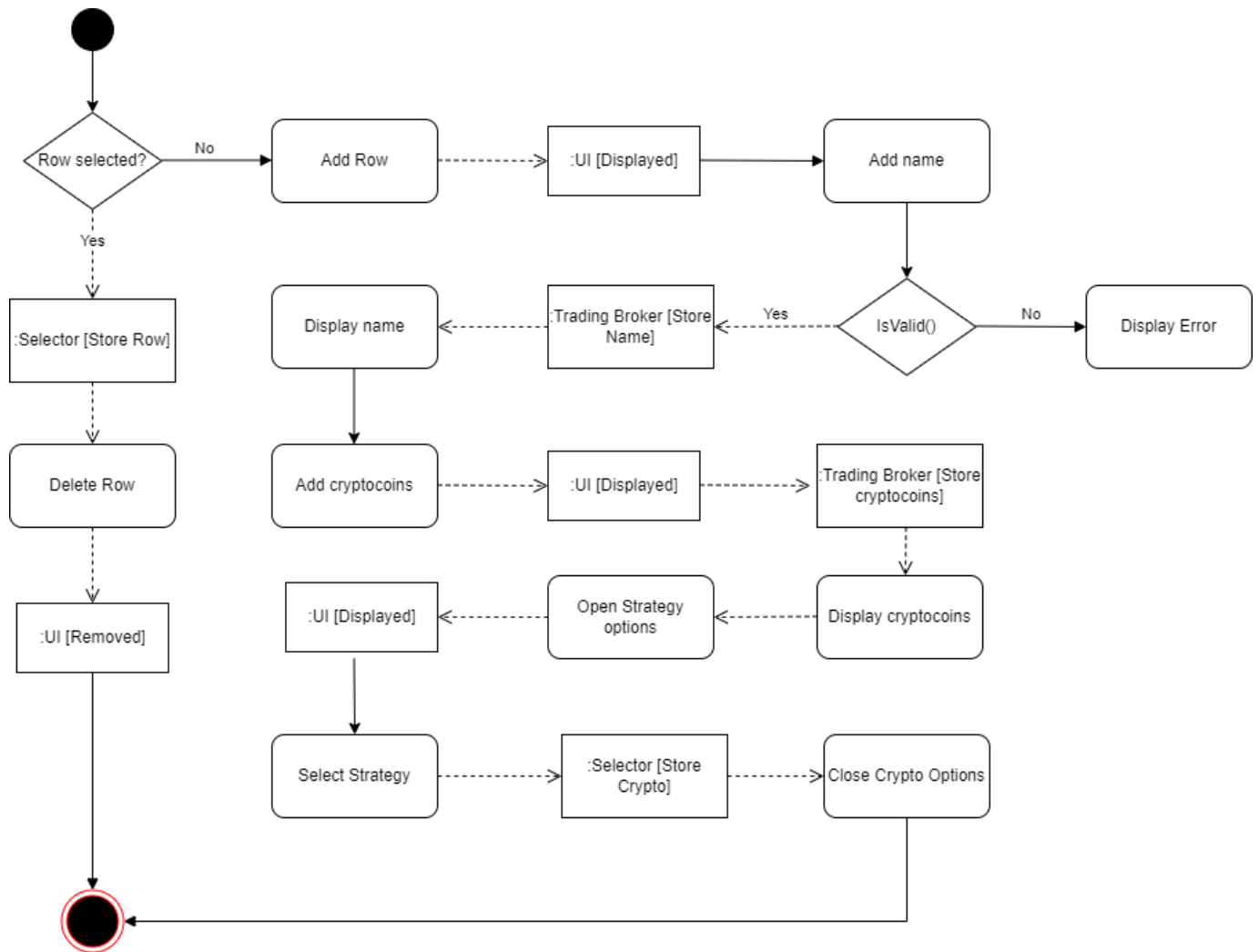


3.2 Activity Diagrams

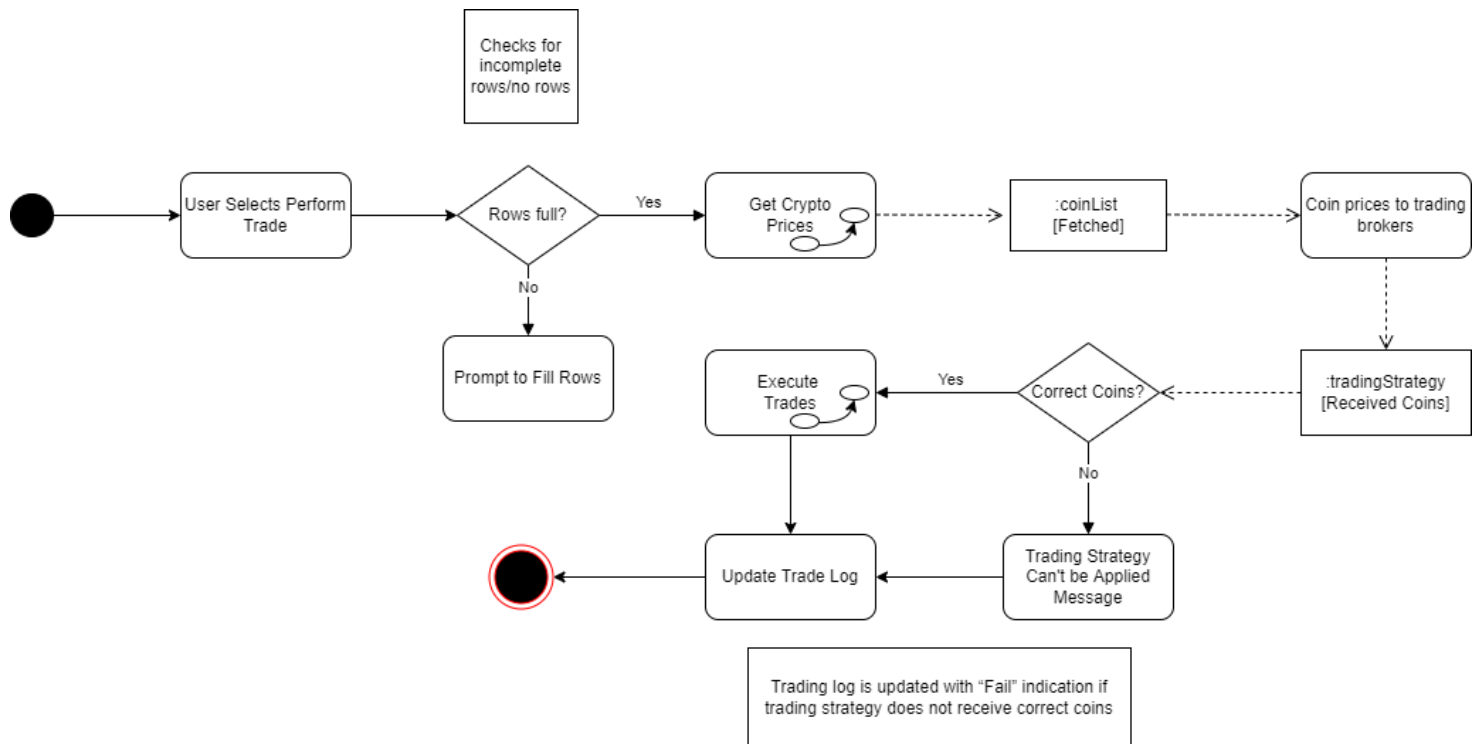
3.2.1 Activity Diagram for Use-Case-1



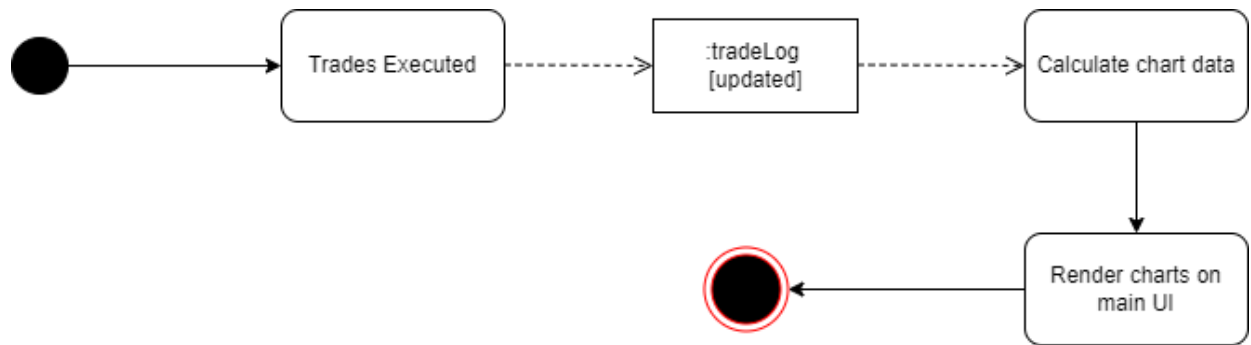
3.2.2 Activity Diagram for Use-Case-2



3.2.3 Activity Diagram for Use-Case-3



3.2.4 Activity Diagram for Use-Case-4



4 Non-Functional Requirements Specification

4.1 Overview

The non-functional requirements of the system comprises utilities, environments and other specifications that are necessary for the smooth operation of the system as a whole. This includes interfaces, development environment, capacity specifications, network and operational parameters.

4.2 Enabling Technologies

4.2.1 Target Development Environment

The system should be developed in a Windows environment and using Java. Eclipse 3.1 will be the Integrated Development Environment (IDE) used for coding purposes.

4.2.2 System Interfaces

In order to be reliable RAID 1 mirroring will be used on the server. This ensures that if one of the disks fails the other disk functions as a single hard drive until the faulty one is replaced. For system installation and activity logging, the disk size used will be 30 GB. Most of that space would be occupied by client information such as User Credentials, Trading Strategy, and Cryptocoins.

4.3 Capacity Planning

Every group member will be using BitBucket as a repository for source code. Each group member will regularly push and pull source code to the BitBucket repository to ensure everything is up to date. BitBucket will provide ample storage for the team to use.

4.4 Network

The computer should have a stable broadband internet connection of 2mb/s download and 1mb/s upload or more. Furthermore, the network should possess efficient packet switching protocols such as MultiProtocol Layer Switching (MPLS) that will transmit the signals without skips or delays.

4.5 Workstations

The minimum system requirements and configurations for the computers used for the development, deployment and execution of the system are:

- A hard disk space of 8GB to install Windows
- A Java Virtual Machine.
- A processor speed of 1 GHz and memory of 128 MB is sufficient.
- A display setting of 1280 x 720 resolution and a 16-bit color palette should be used.
- The computer should have a stable broadband internet connection of 2mb/s download and 1mb/s upload or more.

4.6 Operational Parameters

4.6.1 Usability

The program and system should be usable by any individual who is capable and has the knowledge of using a computer and Java programming language. The UI will be inclusively designed so all the buttons and prompts are clear to the user which will help the user go through the next steps.

4.6.2 Reliability

Each version of the program will be saved externally which will allow the system to recover based on which edition is required.

5 Activities Plan

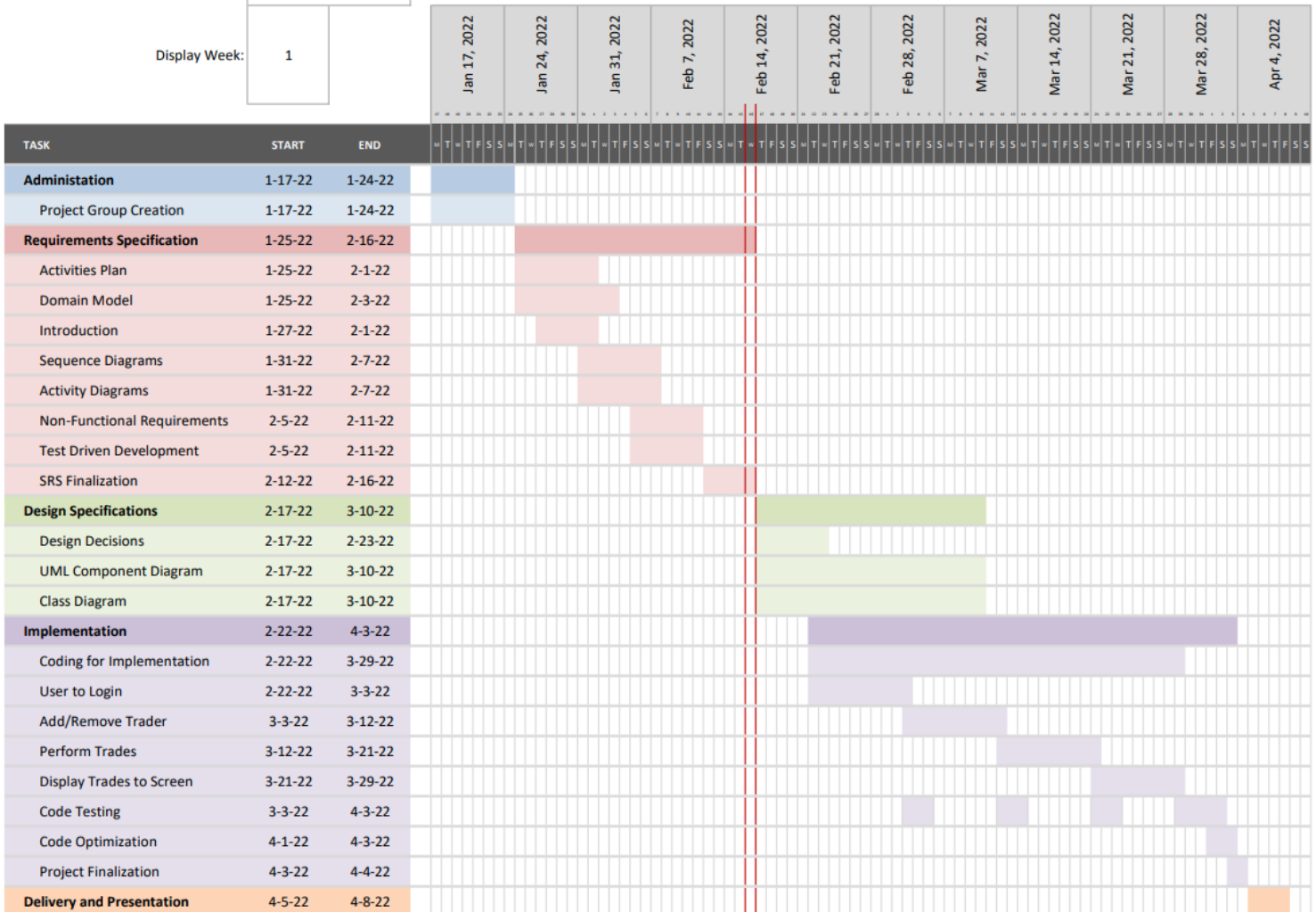
5.1 Gantt Chart

Cryptocurrency Trader

[Group 24]

Start: Mon, 1-17-2022
Today: Wed, 2-16-2022

Display Week: 1



5.2 Project Backlog

Backlog Item	Time (days) Estimate
Activities Plan	7
Domain Model	9
Introduction	5
Sequence Diagrams	7
Activity Diagrams	7
Non-Functional Requirements	6
Test Driven Development	7
Design Decisions	6
UML Component Diagram	21
Class Diagram	21
Creating Trading Strategies	5
Coding for Implementation	35
Allowing User to Login to System	9
Allowing User to Add/Remove a Trading Broker	9
Allow User to Perform Trades	9
Display Trades Made to Screen	8
Code Testing	14
Code Optimization	3
Project Finalization	2

5.3 Group Meeting Logs

Present Group Members	Meeting Date	Issues Discussed / Resolved
Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	January 28th, 2022	<p>First group meeting (1 hour)</p> <p>Introductions/getting to know each other/discussing strengths</p> <p>Sorting out initial logistics</p> <ul style="list-style-type: none"> • Creating shared folder on Google drive • Communication channels • How often to meet • Timeline for when to finish each section by <p>Went through project description and deliverable 1</p> <p>Discussed questions drawn from project description</p> <p>Decided order to complete tasks</p> <ol style="list-style-type: none"> 1. Domain model first as a group 2. Sequence and activity diagrams 3. Remaining sections individually
Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	January 31st, 2022	<p>Group Meeting 2 (1 hour)</p> <p>Group research and study for how to make a domain model for a software system</p> <p>Brainstorming ideas for domain model</p>
Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	February 2nd, 2022	<p>Group Meeting 3 (1 hour)</p> <p>Work session on domain model</p> <ul style="list-style-type: none"> • Vicky decided to meet with course TA to ensure our domain model was on the right track and ask clarification questions <p>General discussion of progress made on project and how we will handle time before due date</p> <p>Divided sequence and activity diagrams by use case</p> <ol style="list-style-type: none"> 1. Use case 1 - Ali 2. Use case 2 - Vicky 3. Use case 3 - Ian 4. Use case 4 - Hala

Hala Elewa, Ian Guenther Green, Vicky Jiang	February 7th, 2022	<p>Group Meeting 4 (1 hour)</p> <p>Discussion on progress of sequence and activity diagrams</p> <p>Vicky discussed information that TA provided regarding domain model</p> <p>Divided remaining tasks</p> <ol style="list-style-type: none"> 1. Introduction - Hala 2. Non-Functional Requirements - Ali 3. Activities plan, Product Backlog and Sprint Backlog - Ian 4. Test Driven Development - Vicky
Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	February 12th, 2022	<p>Group Meeting 5 (1.5 hours) - discussion and work session</p> <p>Domain Model</p> <ul style="list-style-type: none"> • Vicky and Hala had worked on last parts of domain model prior to this meeting • Full group performed final edits • Vicky finished class descriptions <p>Discussed where we are at with our respective sections</p> <ul style="list-style-type: none"> • Ali finished most of NFR - had some questions regarding system requirements etc • Hala finished introduction and rough draft of sequence diagram • Ian has rough draft of sequence diagram and edits for domain model • Vicky did domain model descriptions
Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	February 13th, 2022	<p>Group Meeting 6 (30 minutes) - brief section updates</p> <p>Update on where we are at with sections</p> <p>New section assignments</p> <ul style="list-style-type: none"> • Use case 4 sequence diagram - Vicky • Use case 4 activity diagram - Ian • Test driven development - Hala
Hala Elewa, Ian Guenther Green, Vicky Jiang, Ali Mohamed	February 14th, 2022	<p>Group Meeting 7 (45 minutes)</p> <p>Group editing of entire document for submission</p>

6 Test Driven Development

Test ID	UC1.1
Category	<i>username and password stored on file or DB</i>
Requirements Coverage	<i>UC1-Successful-User-Login</i>
Initial Condition	The username and password are correct and working
Procedure	<ol style="list-style-type: none"> 1. The user selects login 2. The user provides a user name 3. The user provides a password 4. The user logs-in into the system and is presented with the main UI window
Expected Outcome	The login got closed and the UI got displayed
Notes	alphanumeric usernames and passwords with no other special character

Test ID	UC1.2
Category	<i>username and/or password entered not correct</i>
Requirements Coverage	<i>UC1-Unsuccessful-User-Login</i>
Initial Condition	The username and password are not correct and not working
Procedure	<ol style="list-style-type: none"> 1. Try to login again 2. The user provides a user name 3. The user provides a password 4. If the password or the username are not correct then try again
Expected Outcome	The login is still open
Notes	alphanumeric usernames and passwords with no other special character, unlimited trials.

Test ID	UC2.1
Category	Adding and removing a trading broker
Requirements Coverage	<i>UC2, successful-broker name, stores the chosen coin and the strategy.</i>
Initial Condition	the name of the broker is working and the chosen coin and strategy stored

Procedure	<ol style="list-style-type: none"> 1. The user selects the name. 2. The user selects coins. 3. The user selects strategy. 4. The user clicks on to perform trading. 5. Users can delete trading brokers by pressing on the remove row.
Expected Outcome	Information about the selected coin and strategy.
Notes	If the broker name already exists, the system will deny adding a row that has the exact broker name.

Test ID	UC2.2
Category	<i>Trading broker name is already exists</i>
Requirements Coverage	<i>UC1-Unsuccessful-trader-name</i>
Initial Condition	If the trader name is not working because it is already stored
Procedure	<ol style="list-style-type: none"> 1. The user selects the name. 2. The user selects coins. 3. The user selects strategy. 4. The user clicks on to perform trading. 5. Users can delete trading brokers by pressing on the remove row.
Expected Outcome	a message displayed and broker is not added
Notes	

Test ID	UC3.1
Category	<i>Performing Trading</i>
Requirements Coverage	<i>UC3, check the prices for each coin and compare it with a strategy to notify the right client with the pre-selected coins.</i>
Initial Condition	The broker should have selected the coins and the strategy.
Procedure	<ol style="list-style-type: none"> 1. The server UI triggers the computation. 2. Get the prices for the selected coins. 3. Notify the correct client.
Expected Outcome	updated information for each broker with the strategy that suits them. Trades made.
Notes	The broker should get only notifications about their chosen coins.

Test ID	UC3.2
Category	<i>Performing Trading</i>
Requirements Coverage	<i>UC3, incorrect trader</i>
Initial Condition	The broker should have selected the coins and the strategy.
Procedure	<ol style="list-style-type: none"> <i>1. The server UI triggers the computation.</i> <i>2. Get the prices for the selected coins.</i> <i>3. notify the wrong client</i>
Expected Outcome	No trades made, display error
Notes	The broker should get only notifications about their chosen coins.

Test ID	UC4.1
Category	displaying the trading action for all trading clients
Requirements Coverage	<i>Display the name, strategy, coins, action, quantity, unit-price, time-stamp in a histogram and a table.</i>
Initial Condition	column will be increased by 1 if client has more trades
Procedure	<ol style="list-style-type: none"> <i>1. get the name, strategy, coins, action, quantity, unit-price, time-stamp.</i> <i>2. display them in a histogram.</i> <i>3. display how many times each strategy has been used by traders in a table.</i> <i>4. if necessary, add a new column.</i>
Expected Outcome	A histogram and a table displayed in the main UI that has all of these information.
Notes	The table will consist of the most used strategy.