Requirements Specifications

P3: Autonomous Trading Bot

Student ID	Name
23100197	Ahmed Tahir Shekhani
23100011	Suleman Mahmood
23100198	Ali Asghar
23100176	Syed Talal Hasan
23110345	Muhammad Ammar Ibrahim

Table of Contents

1. Introduction

A web application with an autonomous trading bot instance that will trade to generate profitable returns on stocks. The bot will be trained on the PSX data. The bot's decision will be based on the concepts of game theory, mathematical models, financial techniques, and especially artificial intelligence. The primary web app will allow the user to provide the bot's configuration, which includes, target return, risk appetite, and duration of the instance.

With recent advancements in deep learning frameworks and access to faster gpus, training complex models that can predict on time series data has opened new avenues to explore stock market trading. We plan on using models that have a memory component in them, such as LSTM (Long Short Term Memory) to make predictions and trades on the stock market.

The overall objective for the application would be to achieve the return target provided by the analyst while configuring the bot and minimize loss according to the risk factor provided. The potential users of this application would be trade analysts or managers who will use the bot to run its instances according to their requirements.

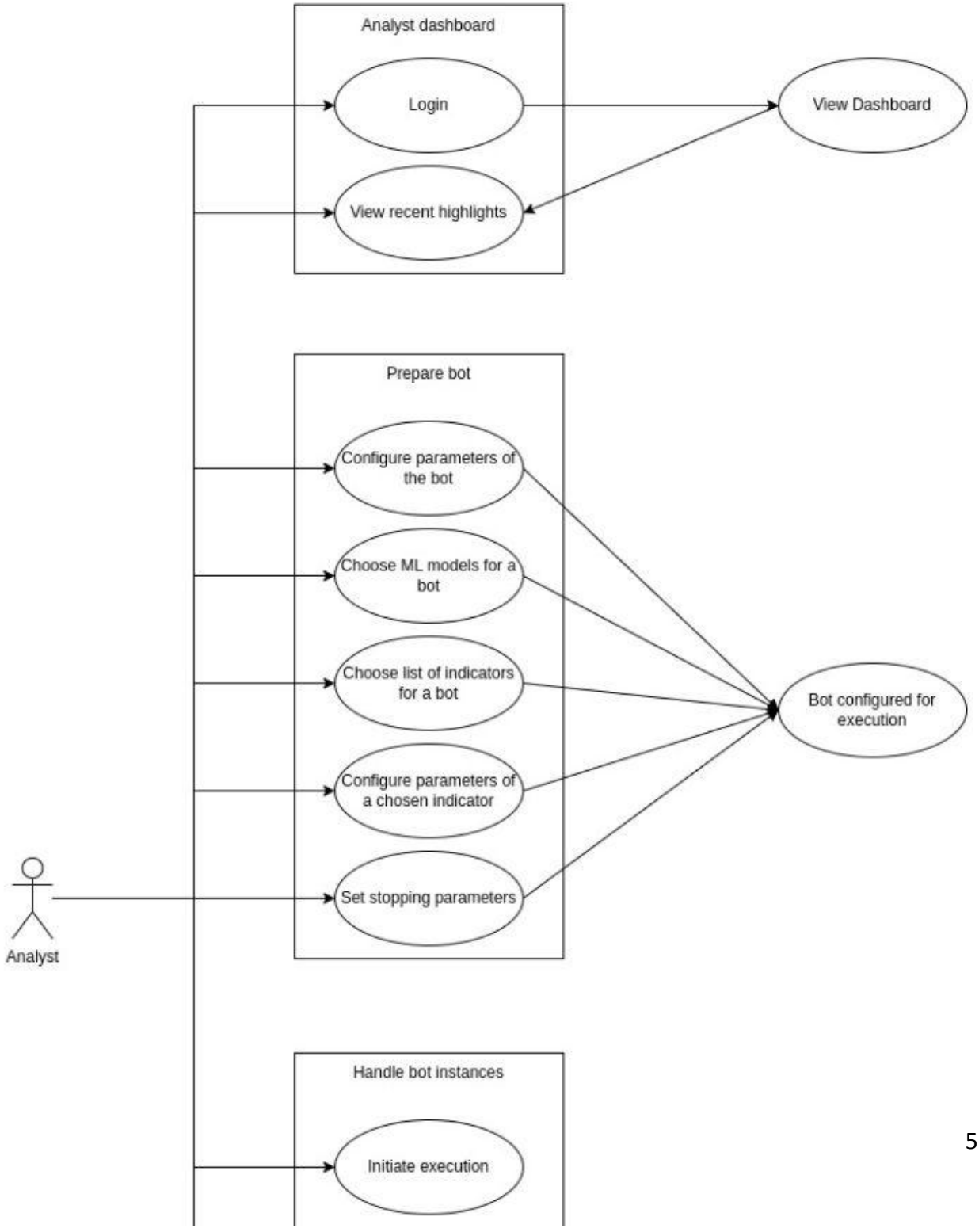
Technical details:
The project's tech stack would be Next.js for frontend web application, Flask for backend server, and PostgreSQL for our persistent storage. The application would follow three-tier architecture with a repository pattern for the persistent layer, models, and command layer for modifying the state.

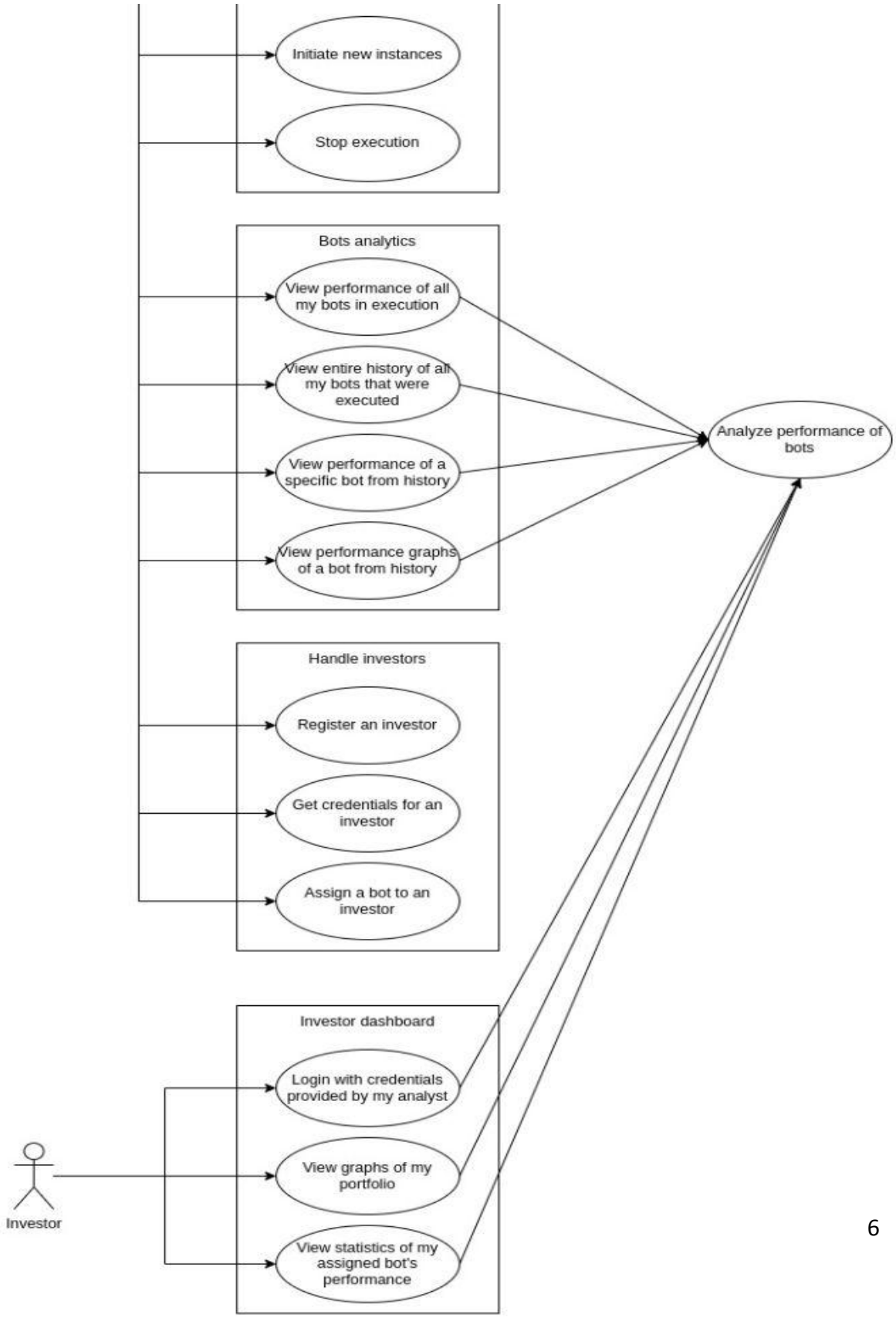
2. System Actors

Actor Name	Description
Analysts	Analysts will configure and start the execution of the bot, analyze its statistics and assign investors to their bots.
Investors	Investors will use the bot for investment purposes and can view a summary report of the performance of the bot for the a specific range of time period (monthly, quarterly, half-yearly, yearly, or other combinations of days or months)

3. Use Cases

3.1 Use Case Diagrams




Initiate new instances
Stop execution
Bots analytics
View performance of all my bots in execution
View entire history of all my bots that were executed
View performance of a specific bot from history
View performance graphs of a bot from history
Analyze performance of bots
Handle investors
Register an investor
Get credentials for an investor
Assign a bot to an investor
Investor dashboard
Login with credentials provided by my analyst
View graphs of my portfolio
View statistics of my assigned bot's performance
Investor

3.2 Description of Use Cases

3.2.1 Login with credentials

Identifier	UC-001
Purpose	The analyst is verified to be a authorized user
Pre-conditions	User enters the username and password
Post-conditions	User is redirected to the home page of the trading platform

Step #	Typical Course of Action
1.	User opens the trading website
2.	Website displays a login prompt to the user
3.	User enters username and password
4.	Username is matched in the database of authorized users
5.	If the username is present, check if the corresponding password is entered correctly
6.	If the password is correct, send a message to redirect to the home page
Step #	**Alternate Courses of Action**
	-
Step #	**Exception Paths**
1.	In step 4: If username doesn't match, display a username error message In step 5: If password doesn't match display a failed login message

3.2.2 View Recent Highlights

Identifier	UC-002
Purpose	Analyst want to see recent predictions made by the model
Pre-conditions	User is logged in with analyst credentials
Post-conditions	A highlight page displays recent activity highlights from the database

Step #	Typical Course of Action
1	Analyst clicks on the view highlights tab
2	A request is sent to the database to fetch recent activity of the bot
3.	Each entry of the activity is sorted based on time
4.	The activity is displayed on the display page
5.	Use case ends

Step #	Alternate Courses of Action
	-

Step #	Exception Paths
1.	In step 2: If there are no activities in the database a display message telling the user of no recent activity is shown

3.2.3 Configure Model Parameters to make Predictions

Identifier	UC-003
Purpose	To provide analyst granularity to run the model
Pre-conditions	i) User is logged in using analyst credentials ii) User has access to stocks to make predictions on
Post-conditions	Model makes predictions on the defined parameters by the user

Step #	Typical Course of Action
1.	Analyst navigates to the model definition page
2.	User defines the parameters of maximum drawdown, balance and list of stocks to the run the model on
3.	A request is sent to the database to check if each of stock in the list of stocks defined is part of the model database
4.	If the stocks model is part of the model directory in the database run prediction for the model
5.	Combine results from each model to form a prediction vector
6.	Show the user the best result in terms of user defined parameters from the vector
7.	Save model prediction in user highlights
8.	Use case ends

Step #	Alternate Courses of Action
	-

Step #	Exception Paths
1	In step 4 if the model execution is terminated before the complete run of the model, the results will not be displayed to the user or stored in the database

3.2.4 Define Stopping parameters of the bot

Identifier	UC-004
Purpose	Automatically exit a trade if the trade requirements are met
Pre-conditions	User has started a valid trade
Post-conditions	Trade is ended and the total return from the trade is added to the account of the analyst

Step #	Typical Course of Action
1.	Before running the model user defines stopping conditions; duration and target percentage return
2.	A cron job checks for the state of model
3.	If one of the preset user termination condition is met, the trade is exited
4.	Result of the bot is stored in user highlights
5.	Trade profit or losses are reflected in the analysts account
6.	Use case end

Step #	Alternate Courses of Action
1.	From step 2: If user terminates a trade manually, trade is exitted with the results stored in the database and the trade profits and loss reflected in the accounts
Step #	**Exception Paths**
1.	If a user tries to exit a trade in non trading times, send an exception.

3.2.5 User initiates a bot for execution

Identifier	UC-005
Purpose	Enable the analyst to enter trades
Pre-conditions	User is logged in with analyst credentials. Time for execution is during trading times of the stock market
Post-conditions	A bot state is run and linked to a chronejob

Step #	**Typical Course of Action**
1.	User defines the trade parameters before running the bot.
2.	Based on the parameters, models from the model directories are run and send their predictions to the bot
3.	The bot chooses the best prediction instance and initiates a trade on the best predicted stock
4.	A cron job is initialized to exit trades based on user defined exit conditions
5.	A running model state is linked to the the cron job
6.	Instance of the model is stored on the analysts running bot directory
7.	The use case ends.
Step #	**Alternate Courses of Action**
	-
Step #	**Exception Paths**
1.	In Step 5: If the analyst tries to enter a trade in non trading times, the model initialization will fail and display an error message

3.2.6 Forcefully terminate the execution of the bot

Identifier	UC-006
Purpose	To exit a trade forcefully, before parametric termination condition is met
Pre-conditions	A bot has entered a trade that needs to be exited
Post-conditions	The bots execution is terminated and the associated loss or profits are added to the analysts account

Step #	Typical Course of Action
1.	User sends a command to exit an instance of the bot execution
2.	System checks if the bot is executing
3.	If the bot is running, it is exited from the trade
4.	Store profits or losses in the analysts account
5.	Save the model history in user highlights
6.	The use case ends

Step #	Alternate Courses of Action
1	-

Step #	Exception Paths
1.	In Step 1: If user sends a command to exit in non trading time, it should send an error message

3.2.7 Initiate multiple bot instances

Identifier	UC-007
Purpose	Enable analyst to make multiple trades based on different parameters
Pre-conditions	User is logged in with analyst credentials. Time for execution is during trading times of the stock market
Post-conditions	A bot state is run and linked to a chronejob

Step #	Typical Course of Action
1.	User defines the trade parameters before running the bot.
2.	Based on the parameters, models from the model directories are run and send their predictions to the bot
3.	The bot chooses the best prediction instance and initiates a trade on the best predicted stock
4.	A cron job is initialized to exit trades based on user defined exit conditions
5.	A running model state is linked to the the cron job
6.	The new model instance is added to the list of running bots with a special bot id
8.	The use case ends.
Step #	**Alternate Courses of Action**
	-
Step #	**Exception Paths**
1.	In Step 5: If the analyst tries to enter a trade in non trading times, the model initialization will fail and display an error message

3.2.8 Analyst registers a new investor

Identifier	UC-008
Purpose	The analyst registers an investor who can view their bot's performance.
Pre-conditions	User is logged in using analyst credentials
Post-conditions	The investor is registered, and his details added to the database.

Step #	Typical Course of Action
1.	The analyst is logged in to the website.
2.	The analyst goes to the 'Register Investor' page.
3.	The website asks the analyst for new investor details.
4.	The analyst enters the name, phone number, email address, residential address, and NTN number of the investor.
5.	The email address, phone number and NTN number are checked if they are valid.
6.	If details are valid, the database is checked if the email address is already there.
7.	If the email address is not in the database, the analyst is shown a prompt to confirm the details about the investor.
8.	If the analyst confirms, the user is registered, and details are added to the database.
9.	Use case ends.

Step #	Alternate Courses of Action
10.	

Step #	**Exception Paths**
1.	In step 5, if any of the details is invalid, the analyst is prompted about the error and be told to enter again.

	2. In step 6, if the email address is already in the database, the analyst is prompted that email already exists. 3. In step 7, if the analyst doesn't confirm, he/she is redirected to the enter details page.

3.2.9 Analyst gets credentials for a registered investor

Identifier	UC-009
Purpose	The analyst gets login credentials after registering an investor.
Pre-conditions	UC-008 use case is completed successfully.
Post-conditions	The analyst gets credentials for the investor, and the credentials are added to the database.

Step #	Typical Course of Action
1.	The analyst goes to the 'Get credentials' option on the website.
2.	The website asks the analyst to enter investor details (name and email address).
3.	The analyst enters the name and email address of the newly registered investor.
4.	The name and email address are checked from the database.
5.	If valid, the website provides a username and password for the investor, and adds it to the database.
6.	The use case ends.

Step #	Alternate Courses of Action

Step #	Exception Paths
1.	In step 5, if the name and email address are not present in the database, the website prompts the analyst regarding the error, and execution proceeds to step 2.

3.2.10 Assign bots to investor

Identifier	UC-010
Purpose	Analyst assigns one or more bots to an investor.
Pre-conditions	Use case 9 is completed successfully.
Post-conditions	One or more bots are assigned to a specific investor.

Step #	Typical Course of Action
1.	Analyst goes to the 'Assign bots to the investor' page on the website.
2.	Website asks the user for investor details.
3.	The entered details are checked in the database to check if they are valid.
4.	If valid, the analyst is required to enter the number of bots.
5.	The analyst enters the number of bots.
6.	The input is checked if the input is a valid number.
7.	If valid, the number of bots input is assigned to the investor.
8.	The use case ends.

Step #	Alternate Courses of Action

Step #	Exception Paths
1.	In step 3, if the details are not valid, the user is prompted with an error and execution proceeds to step 2. 2. In step 6, if it is not a valid number, the user is prompted with an error and execution proceeds to step 4.

3.2.11 View performance of all bots

Identifier	UC-011
Purpose	Analyst can view the performance of all his/her bots that are currently executing.
Pre-conditions	The Analyst is logged in and authenticated
Post-conditions	Analyst is displayed performance of bots currently running.

Step #	Typical Course of Action
1.	Analyst clicks on the view performance tab.
2.	A request is sent to the database to fetch performance(profit%) of all bots currently executing initiated by the analyst.
3.	The result is displayed to the analyst on the website.
4.	Use case ends.

Step #	Alternate Courses of Action
1.	
Step #	**Exception Paths**
1.	In step 2, if there are no bots currently running initiated by the analyst, a display message is shown saying “No current bots”.

3.2.12 View history of all bots for analyst

Identifier	UC-012
Purpose	The analyst can view the entire history of all bots run by him/her.
Pre-conditions	The Analyst is logged in and authenticated
Post-conditions	Analyst is displayed the history of bots run by him/her.

Step #	Typical Course of Action
1.	Analyst clicks on the view history tab.
2.	A request is sent to the database to fetch history of all bot instances initiated by the analyst.
3.	The result is displayed to the analyst on the website.
4.	Use case ends.

Step #	Alternate Courses of Action
1.	
Step #	**Exception Paths**
1.	In step 2, if there is no history of bots initiated by the analyst, a display message is shown saying “No history”.

3.2.13 View entire performance of a bot

Identifier	UC-013
Purpose	The analyst can view the entire performance of a bot.
Pre-conditions	The Analyst is logged in and authenticated
Post-conditions	Entire performance of the bot is displayed to the analyst.

Step #	Typical Course of Action
1.	Analyst clicks on the 'View entire performance of a bot' tab.
2.	Analyst selects the bot that is required.
3.	A request is sent to the database to fetch the complete trade history of the bot selected by the analyst.
4.	The result is displayed to the analyst on the website.
5.	Use case ends.

Step #	Alternate Courses of Action
1.	

Step #	Exception Paths
1.	In step 3, if the bot selected by the analyst is still currently running, the user is prompted with an error and execution proceeds to step 2.

3.2.14 View multiple graphs of a bot

Identifier	UC-014
Purpose	The analyst can view the multiple graphs of a bot.
Pre-conditions	The Analyst is logged in and authenticated
Post-conditions	Selected graphs of the bot are displayed to the analyst.

Step #	Typical Course of Action
2.	Analyst clicks on the 'View graphs of a bot' tab.
3.	Analyst selects the bot that is required.
4.	The analyst is asked to select graphs required from multiple choices
5.	Analyst selects the graphs that are required.
6.	A request is sent to the database to fetch the complete trade history of the bot selected by the analyst.
7.	The graphs are displayed to the analyst on the website.
8.	Use case ends.

Step #	Alternate Courses of Action
9.	
Step #	**Exception Paths**
10.	In step 2, if there is no history of bots initiated by the analyst, a display message is shown saying "No history for selected bot".

3.2.15 As an analyst, I want to choose which ML models to use for a bot.

Identifier	UC-015
Purpose	Analyst can choose which ML models to use for a specific bot
Pre-conditions	1. The Analyst is logged in and authenticated 2. The bot for which the ML model is to be chosen hasn't started it's execution yet
Post-conditions	An ML model is selected for the chosen bot

Step #	**Typical Course of Action**
1.	The user opens the configuration of the bot.
2.	The user chooses an option to view a list of available ML models
3.	The user selects one ML model
4.	The ML is then selected for that particular bot
Step #	**Alternate Courses of Action**
Step #	**Exception Paths**
1.	In step 2: There are no available ML models so display an empty list In step 3: If the ML model isn't compatible with the chosen bot configuration then display an error message

3.2.16 As an analyst, I want to choose a list of indicators to use for a bot.

Identifier	UC-016
Purpose	Analyst can choose a list of indicators for a specific bot
Pre-conditions	1. The Analyst is logged in and authenticated 2. The bot for which the list of indicators is to be chosen hasn't started it's execution yet
Post-conditions	A list of indicators is selected for the chosen bot

Step #	**Typical Course of Action**
1.	The user opens the configuration of the bot.
2.	A list of available indicators is shown
3.	The user selects a particular indicator
4.	A form is displayed with the configuration parameters for the indicator
5.	The user further enters configuration parameters for the selected indicator
6.	The user confirms the indicator's configuration
7.	The user repeats from Step 2
Step #	**Alternate Courses of Action**
Step #	**Exception Paths**
1.	In step 2: There are no indicators so an empty list is shown In step 4: There are no configuration parameters for the chosen indicator so the indicator is selected without choosing any parameters for it

3.2.17 As an analyst, I want to configure the parameters of my chosen indicators for the bot.

Identifier	UC-017
Purpose	Analysts can change the configuration parameters of a specific bot
Pre-conditions	1. The Analyst is logged in and authenticated 2. The bot for which the list of indicators is to be configured hasn't started it's execution yet
Post-conditions	Configuration parameters of a specific bot are changed

Step #	Typical Course of Action
1.	The user opens the configuration of the bot.
2.	A list of chosen indicators is shown
3.	The user selects a particular indicator
4.	A form is displayed with the configuration parameters for the indicator with the previous values already inserted in the fields
5.	The user further changes the configuration parameters for the selected indicator for the selected parameters
6.	The user confirms the indicator's configuration
7.	The user repeats from Step 2 for different indicators

Step #	Alternate Courses of Action

Step #	Exception Paths
1.	In step 2: There are no chosen indicators for the bot so an empty list is displayed and instead a list of available indicators is shown to add them

3.2.18 As an investor, I want to be able to login to my dashboard with the credentials provided by my analyst.

Identifier	UC-018
Purpose	The investor successfully logins in and is authenticated
Pre-conditions	
Post-conditions	Investor is redirected to his dashboard

Step #	Typical Course of Action
1.	User opens the trading website
2.	Website displays a login prompt to the user
3.	User enters username and password
4.	Username is matched in the database of authorized users
5.	If the username is present, check if the corresponding password is entered correctly
6.	If the password is correct, redirect the user to the dashboard

Step #	Alternate Courses of Action

Step #	Exception Paths
1.	In step 4: If username doesn't match, display a username error message In step 5: If password doesn't match display a failed login message

3.2.19 As an investor, I want to be able to see various graphs for my portfolio.

Identifier	UC-019
Purpose	Investor can visually analyze the performance of his portfolio by checking the graphs
Pre-conditions	1. Investor must be logged in and authenticated 2. Investor must have at least invested some amount
Post-conditions	

Step #	Typical Course of Action
1.	Investor visits their dashboard
2.	A simple line graph of balance over time is displayed on the dashboard
3.	A select is displayed on top of the graph
4.	Investor clicks on the select to display a list of available graphs
5.	Investor selects a graph
6.	The bottom line graph changes to the newly selected graph
7.	Investor can then analyze his portfolio using the displayed graph

Step #	Alternate Courses of Action

Step #	Exception Paths
1.	In step 4: There are no available graphs so an empty list is displayed

3.2.20 As an investor, I want to be able to see statistics of my bot's performance.

Identifier	UC-020
Purpose	Investor can analyze his portfolio by viewing the various statistics
Pre-conditions	1. Investor must be logged in and authenticated 2. Investor must have at least invested some amount
Post-conditions	

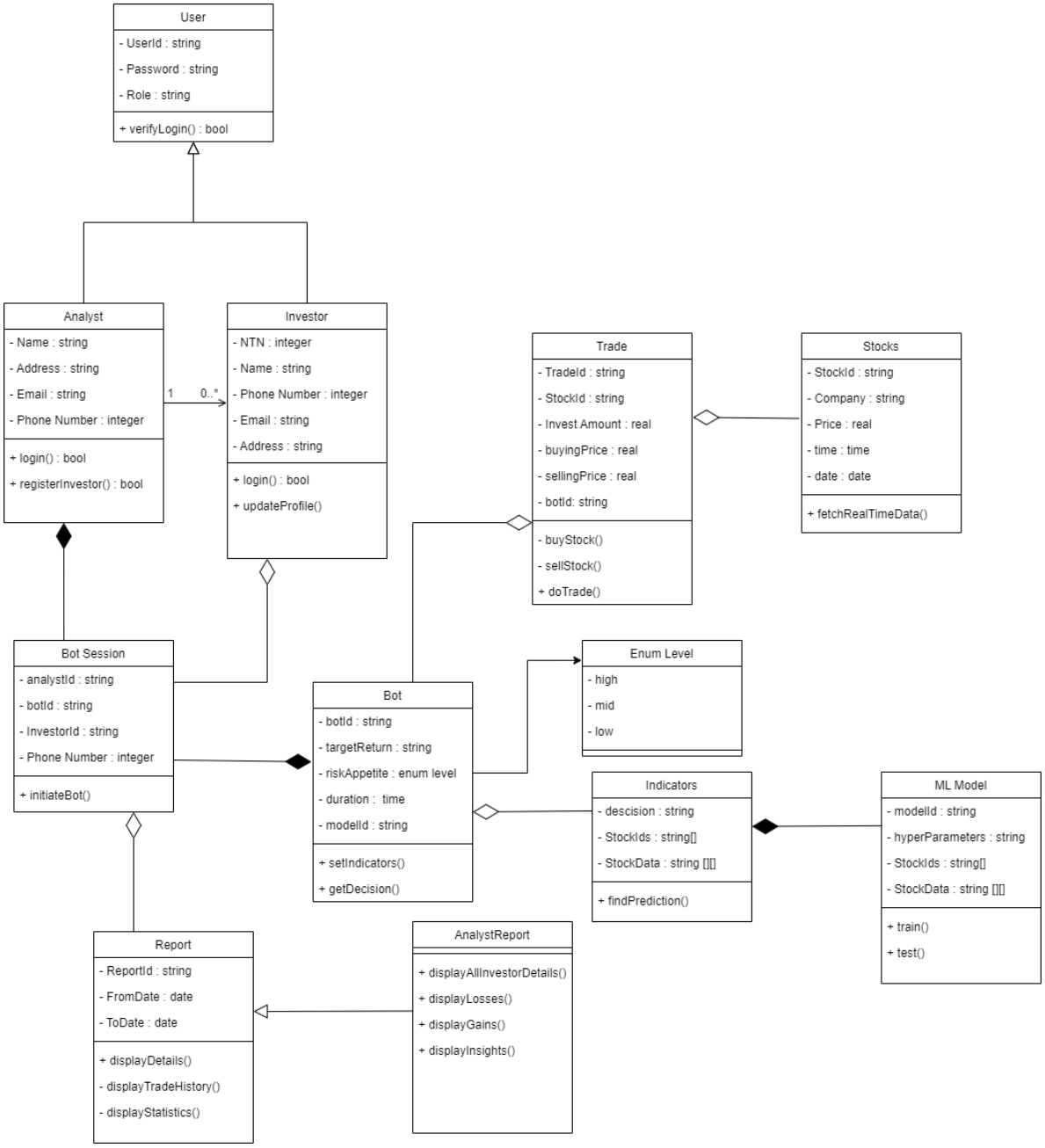
Step #	Typical Course of Action
1.	Investor visits their dashboard
2.	A table of various statistics(balance, drawdown, total trades, successful trades, un-successful trades) is displayed on the dashboard

Step #	Alternate Courses of Action

Step #	Exception Paths

4. Class Diagram

4.1 Diagram



4.2 Description

User: This is a parent class to keep the stakeholder's id and password. The child classes include analysts and investors with parent features and their separate functions of login. Investors can update their profile which includes the variables of the objects. Analysts are associated with investors where one analyst can register multiple investors.

Bot: It will help to create separate instances of bots with various combinations of configuration. It will use the indicator instances with the configuration to make decisions.

BotSession: It will keep track of bot sessions currently running, which are a composition of bot and analyst objects. They will be destroyed after the session is completed.

Trade: It will keep track of trades and store the stock ids, buying and selling prices etc. It will be responsible for managing trade according to the decisions taken by the bot.

Stocks: Every instance of stocks will represent a single stock with their relevant information for trading. These data will be used by the Trade class to make sales and purchases.

Report: It will be responsible for displaying the statistics, trade history, and comparisons of previous trades with the latest ones. It will help keep track of generated reports.

Indicators: These will create the indicators with specific values for each bots according to their configuration parameters. Machine Learning Model Class will be used in it to assist the decision

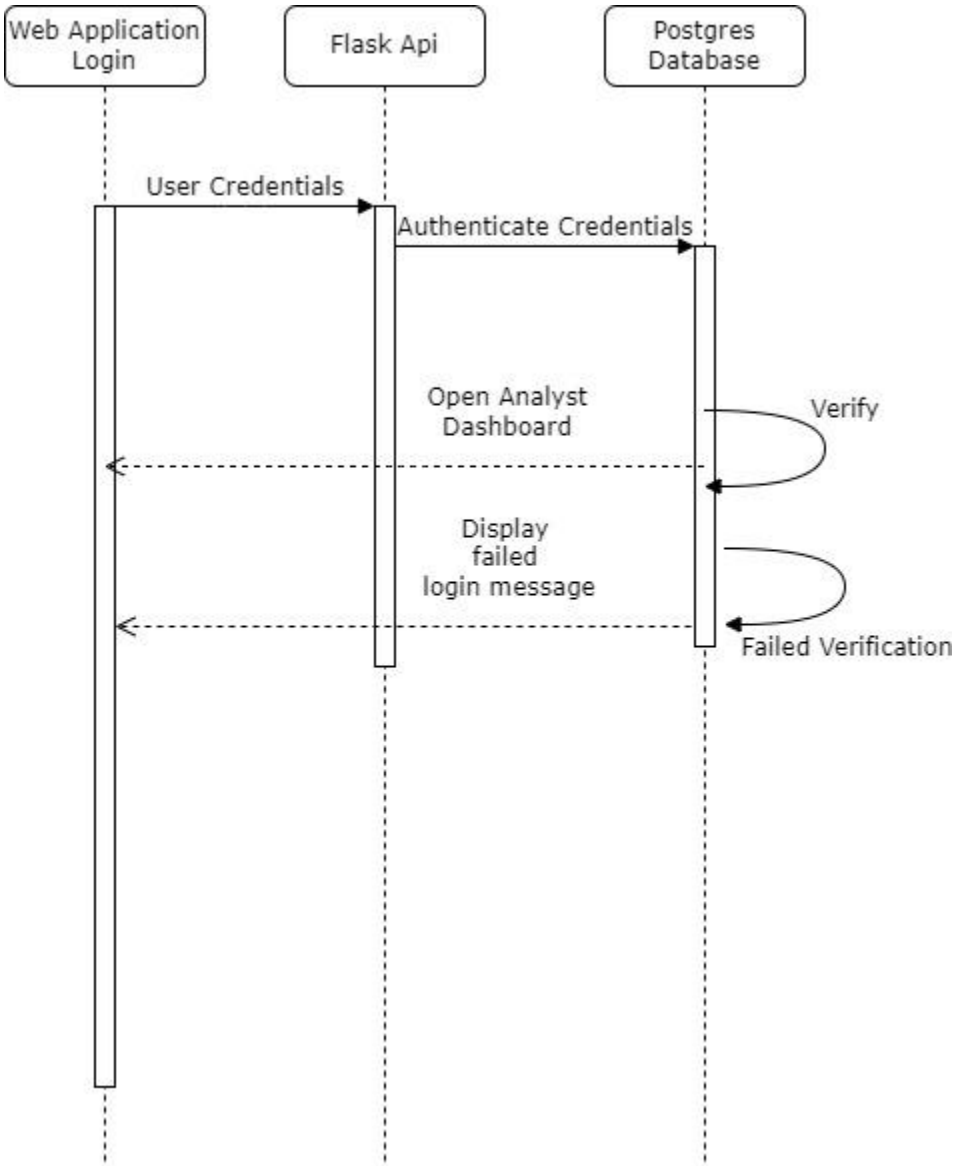
The following indicators will be used:

- On-balance volume (OBV)
- Accumulation/distribution line
- Average directional index
- Aroon oscillator
- Moving average convergence divergence (MACD)
- Relative strength index (RSI)
- Stochastic oscillator

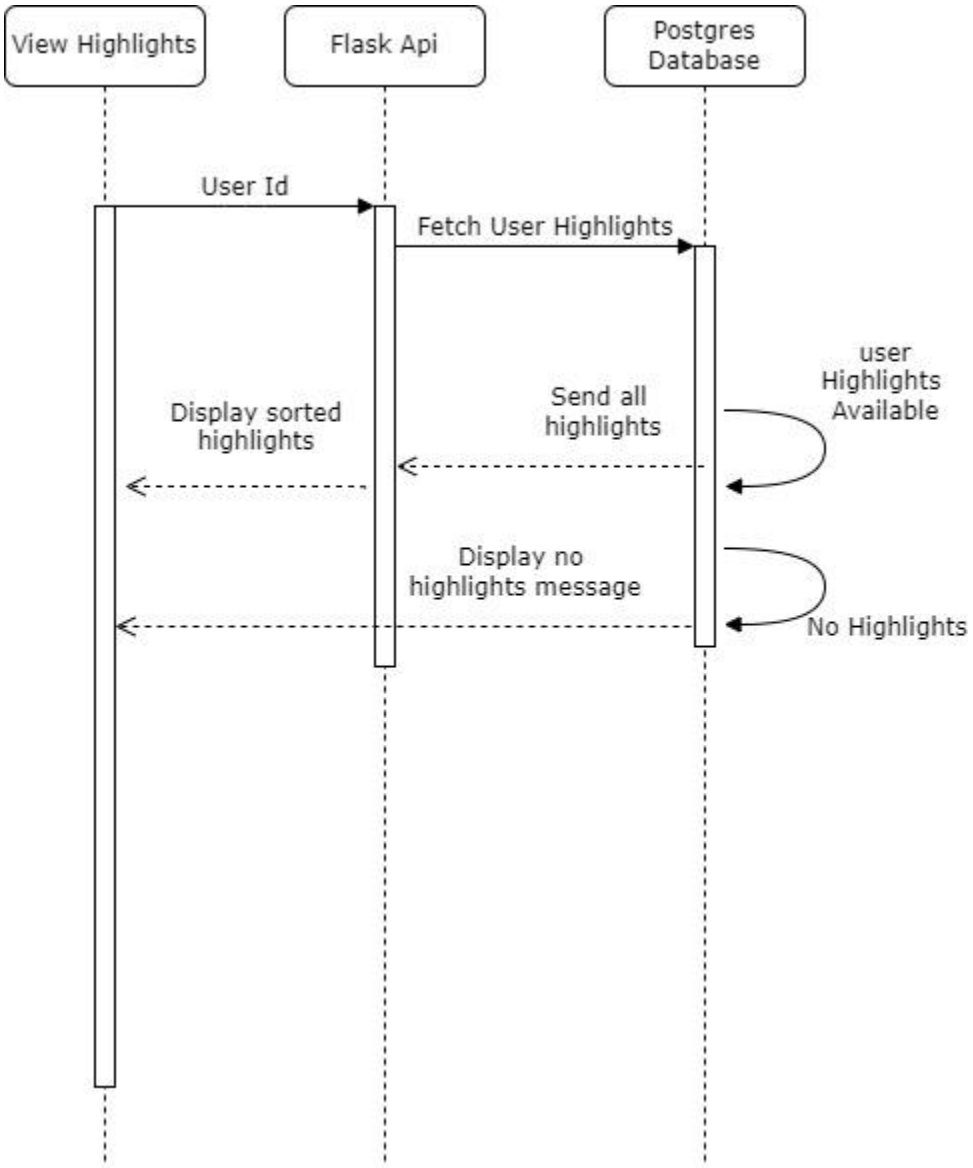
All the class names, attributes and their relations are displayed on the above class diagram.

5. Sequence Diagrams

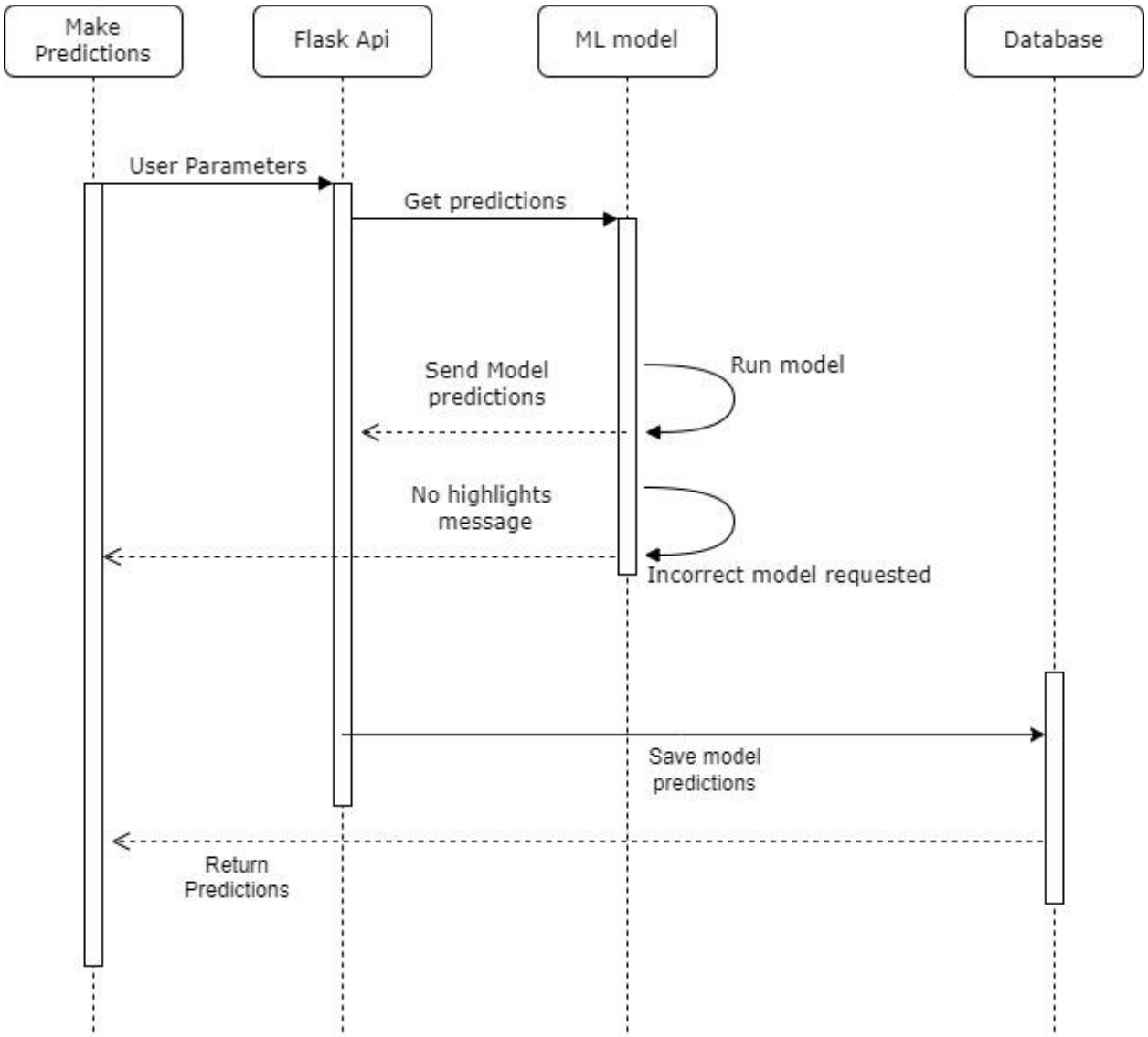
5.1: Login with credentials



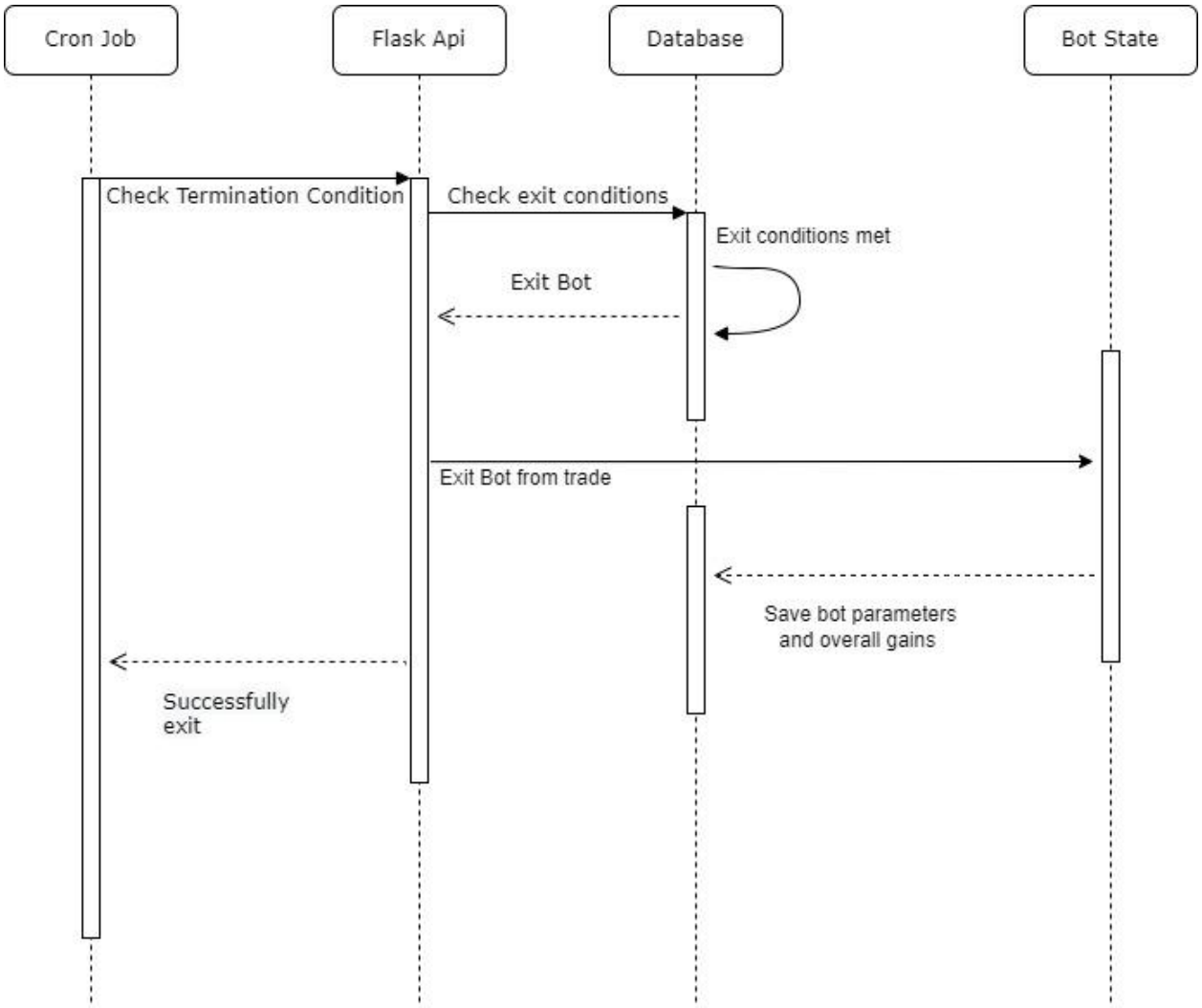
5.2: View Recent Highlights



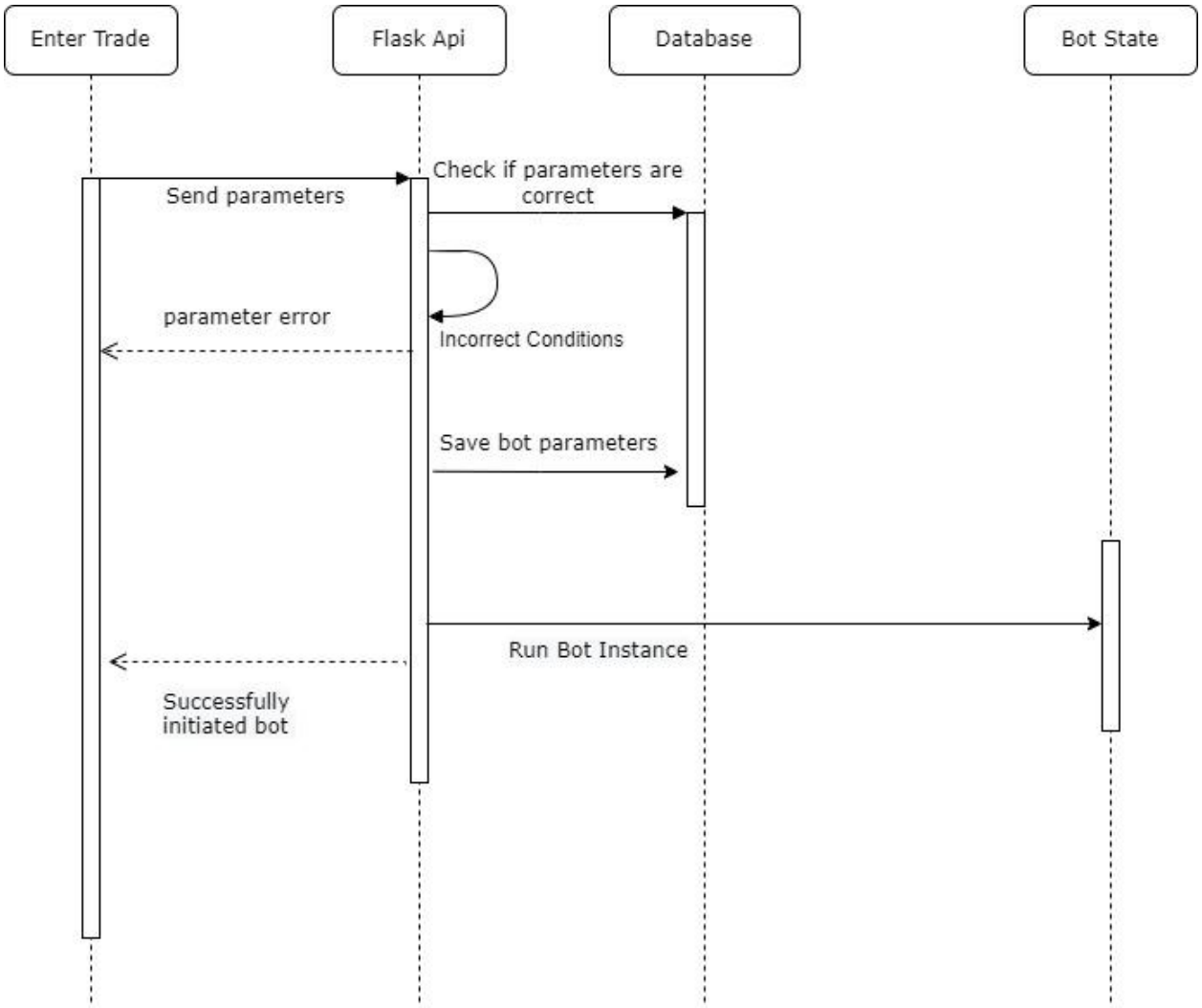
5.3: Configure Model Parameters to make Predictions



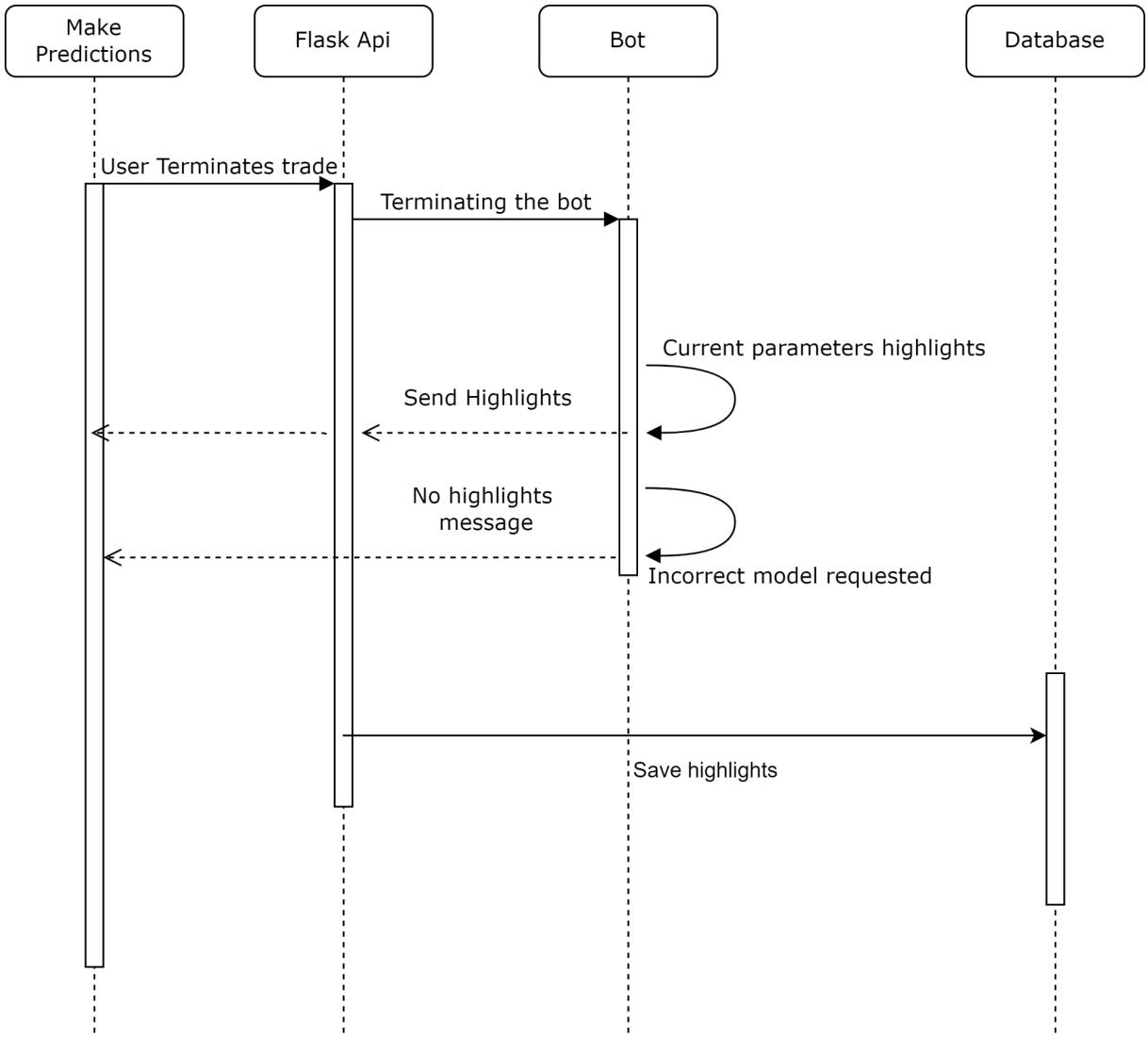
5.4: Define Stopping parameters of the bot



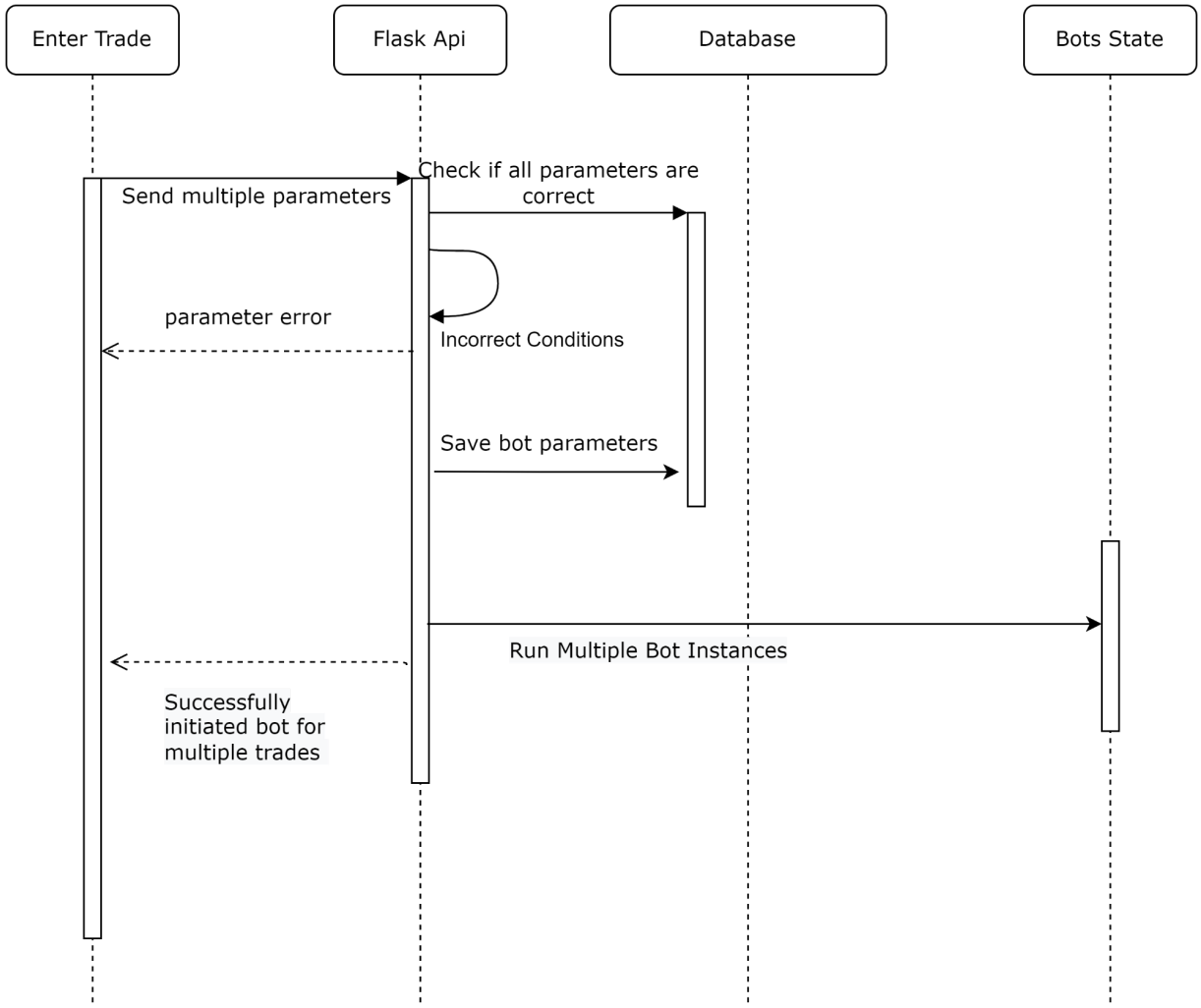
5.5: User initiates a bot for execution



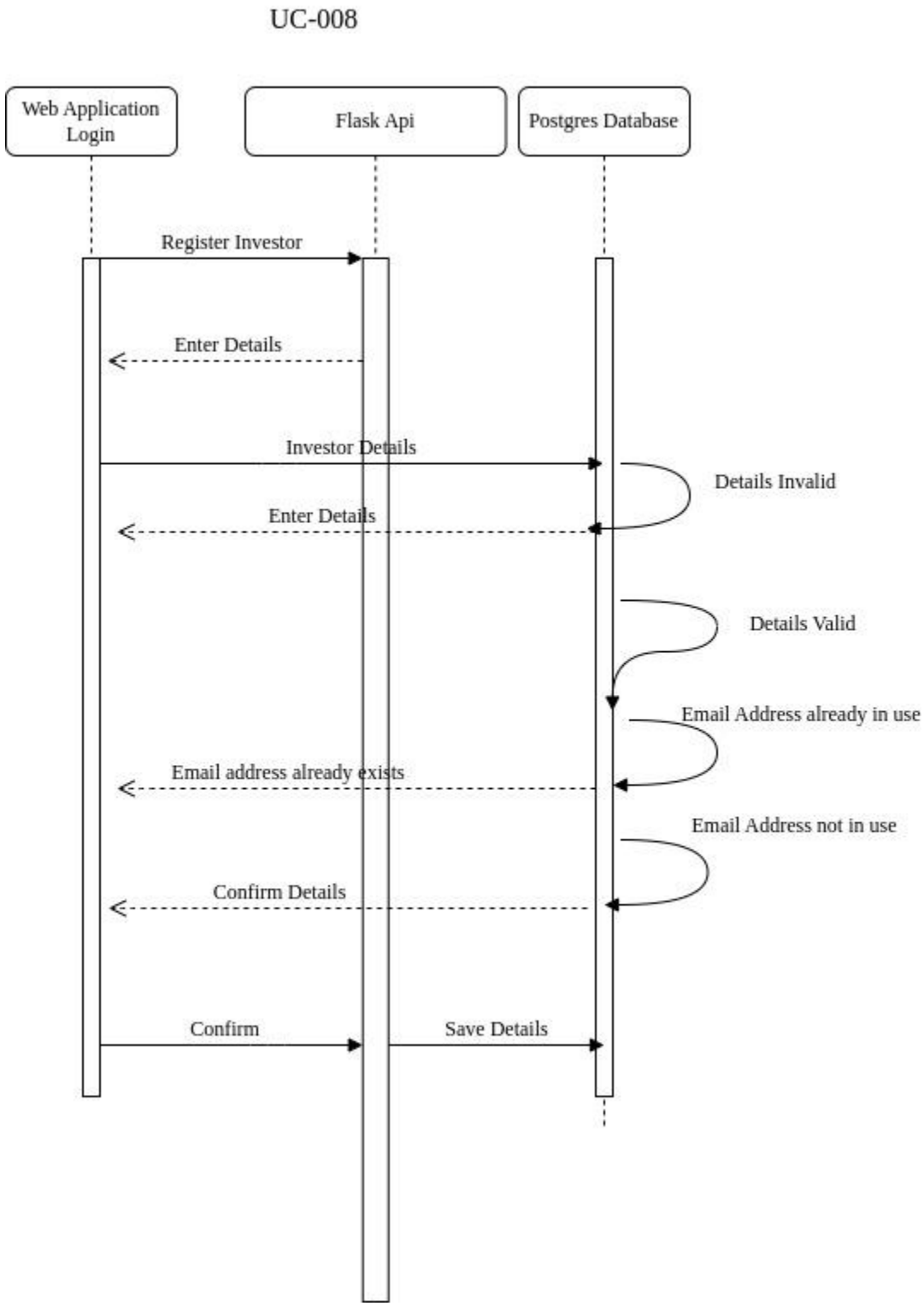
5.6 Forcefully terminate the execution of the bot



5.7 Initiate multiple bot instances

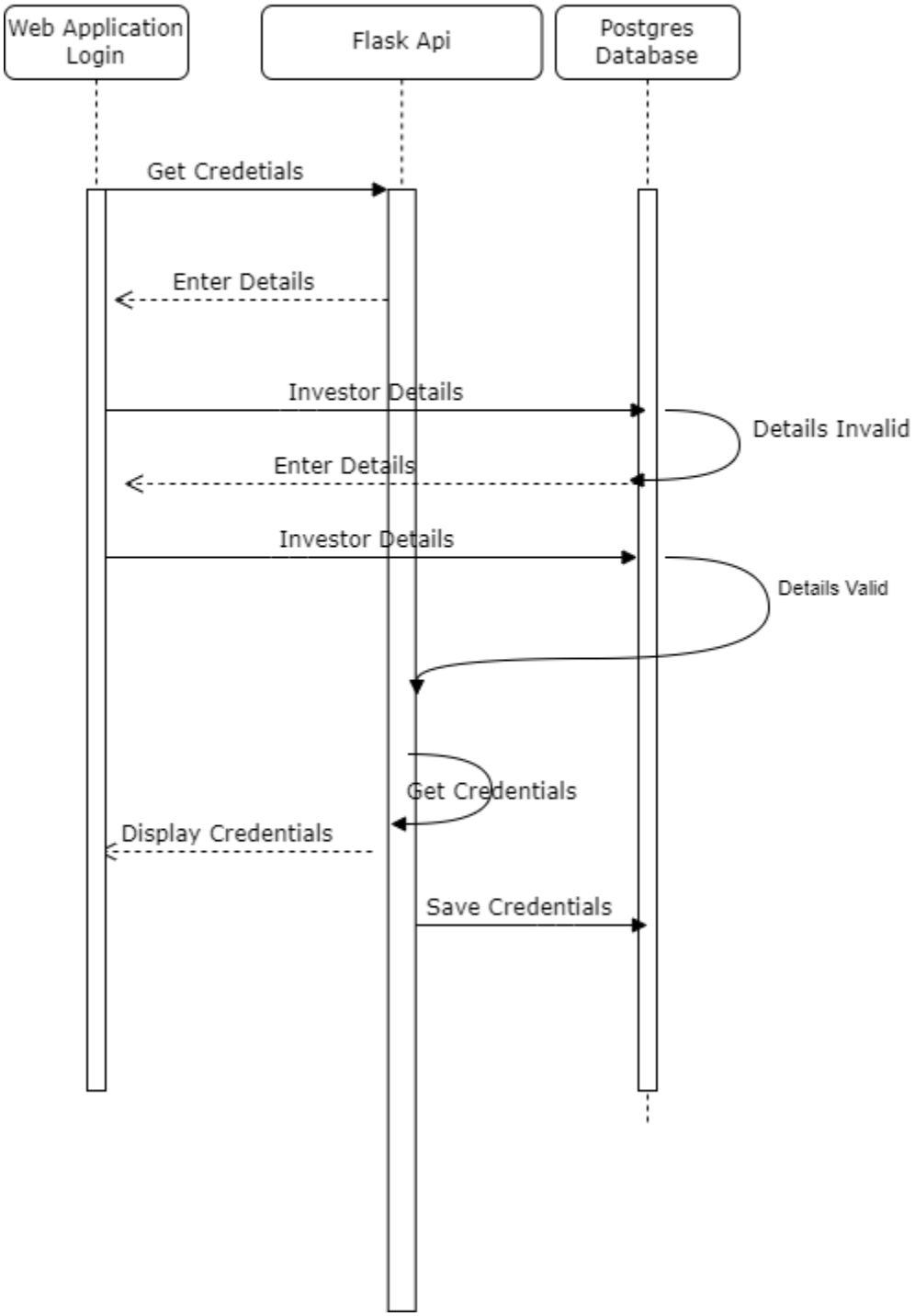


5.8 Analyst registers a new investor



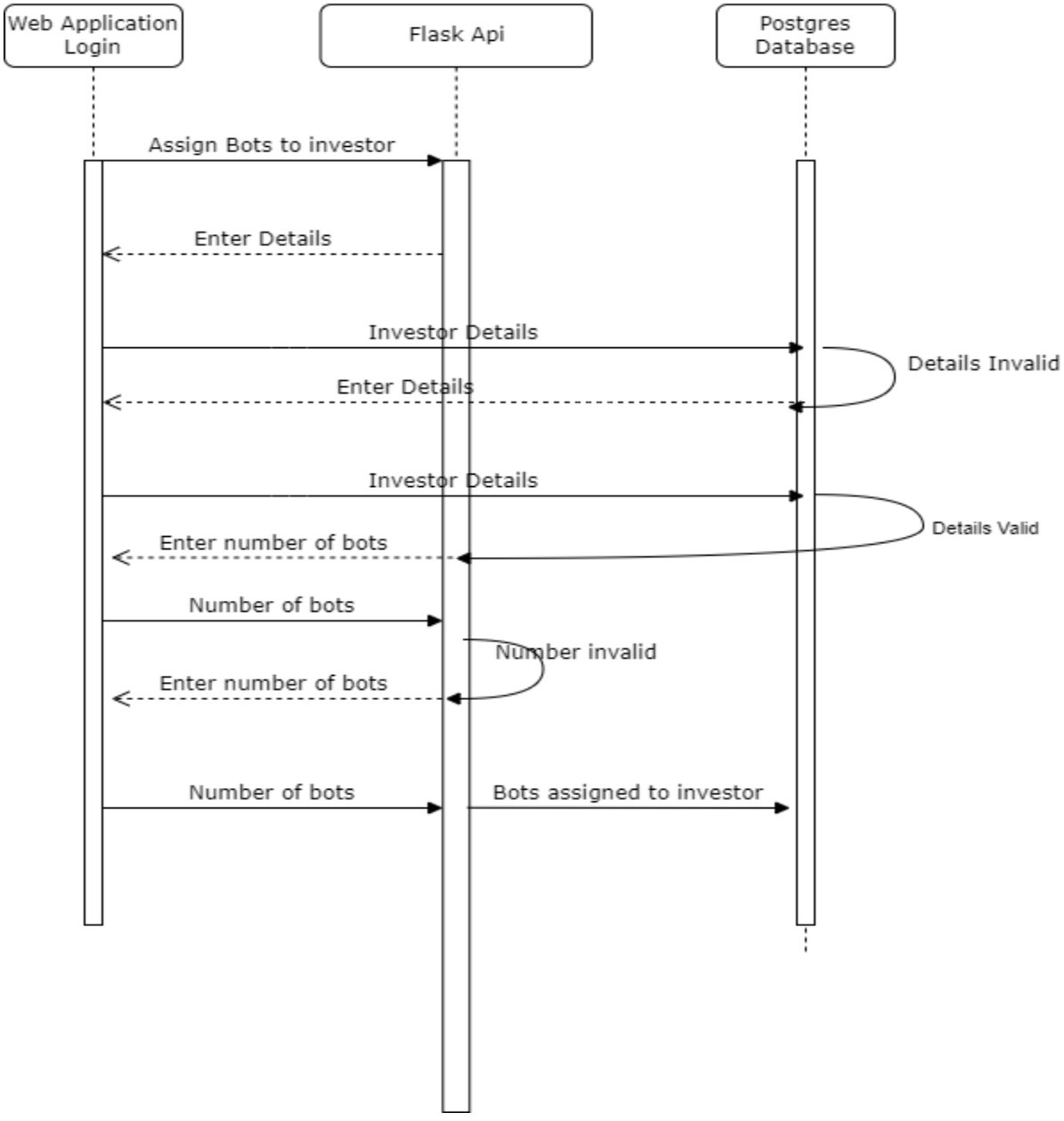
5.9 Analyst gets credentials for a registered investor

UC-009



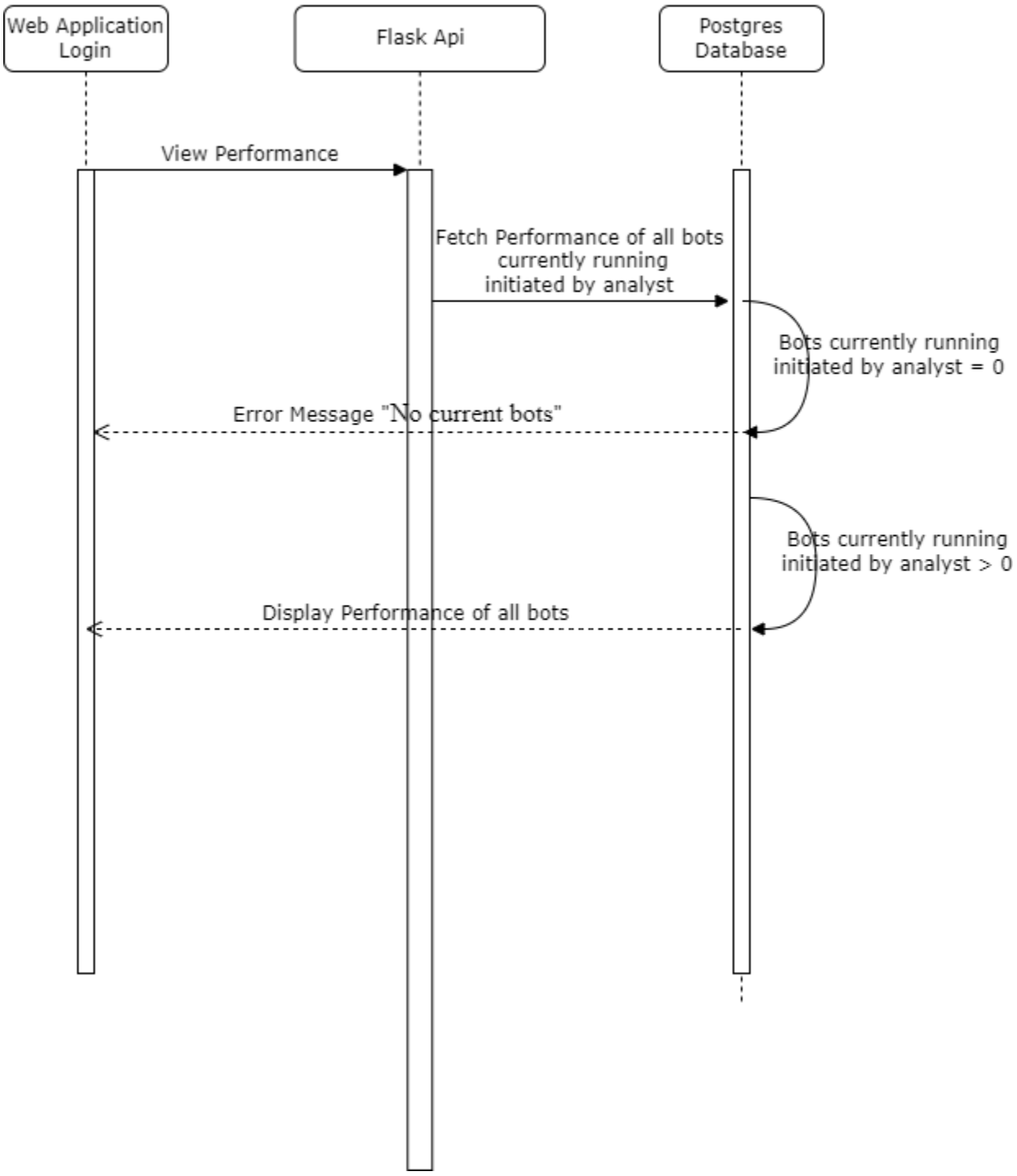
5.10 Assign bots to investor

UC-010



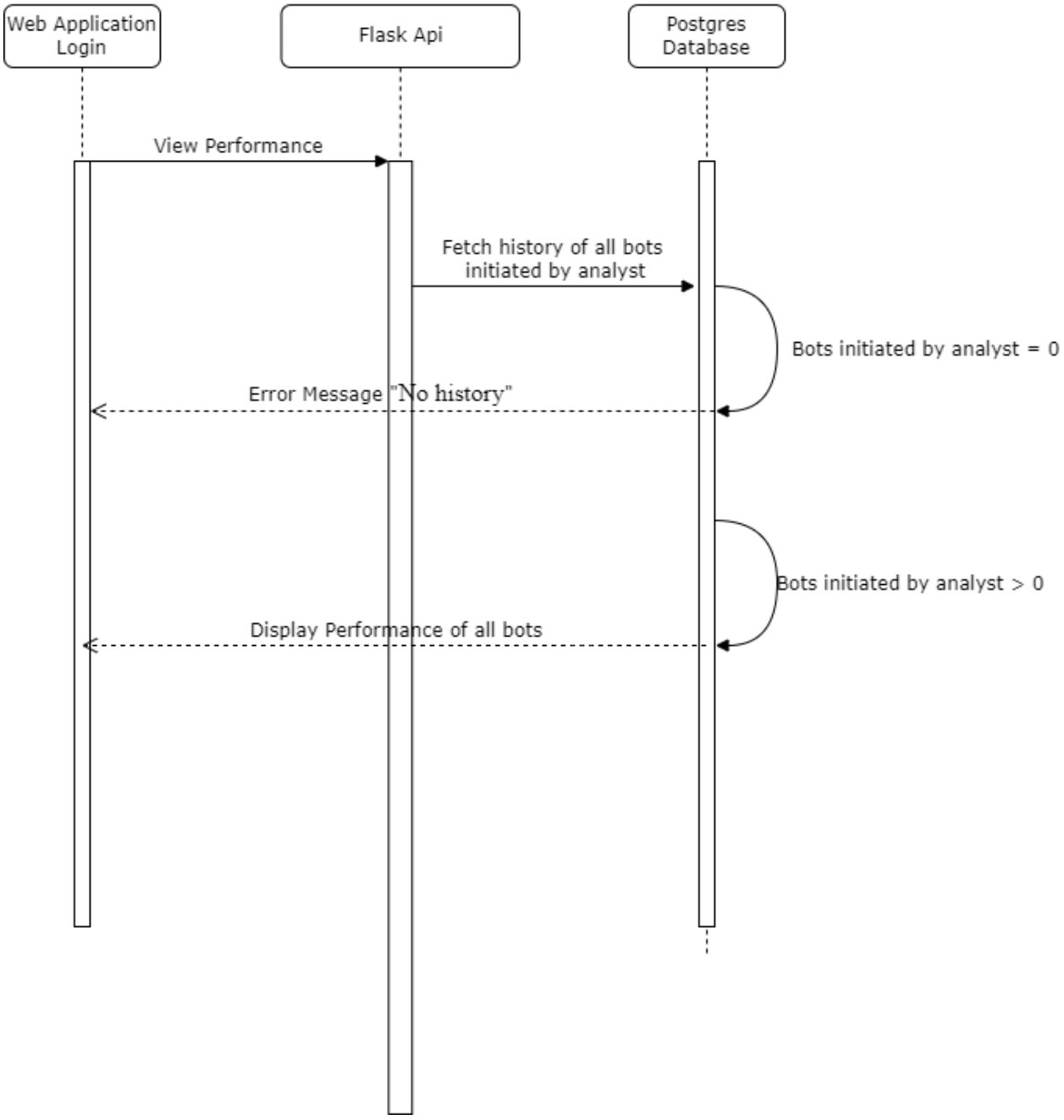
5.11 View performance of all bots

UC-011



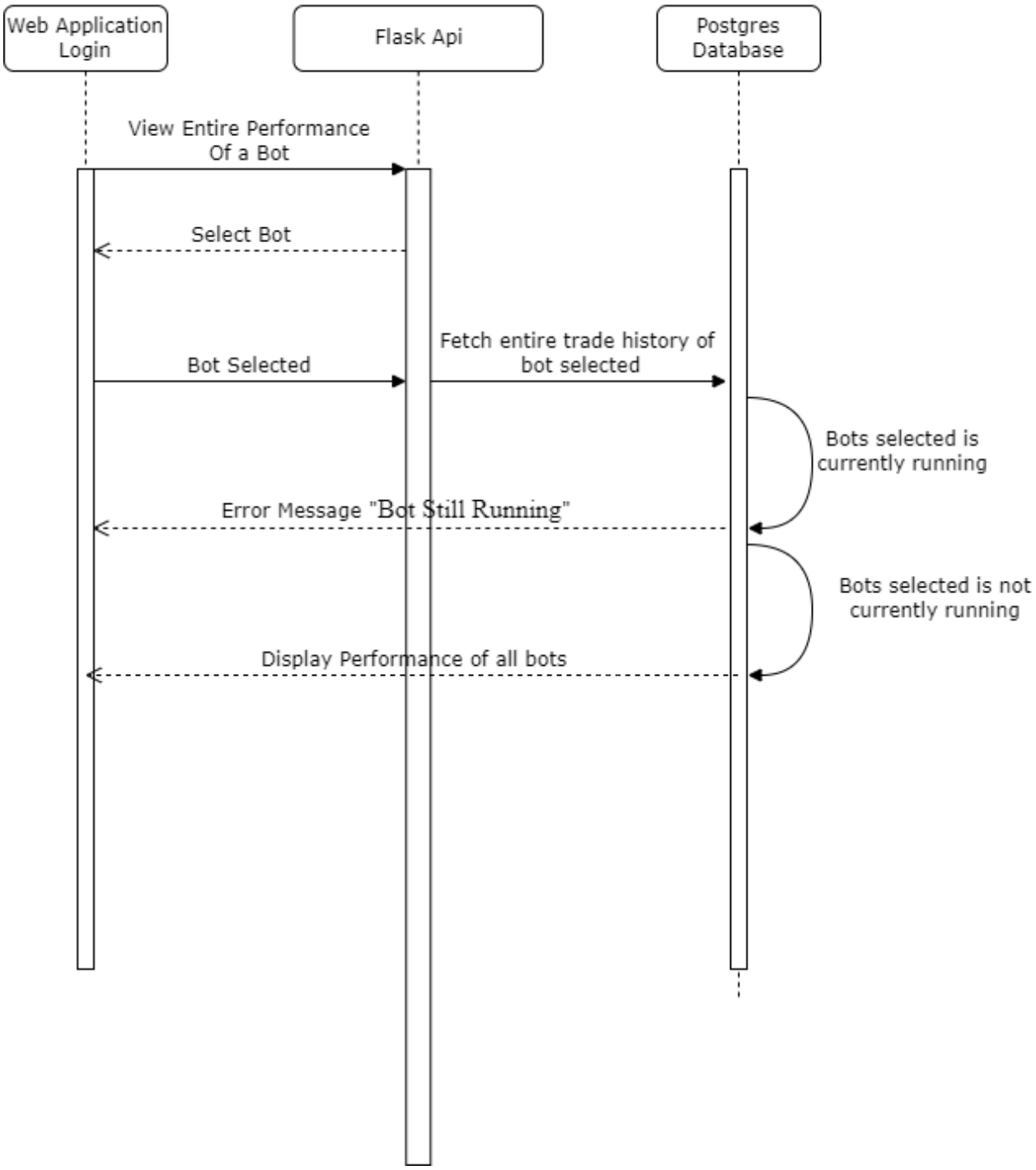
5.12 The analyst can view the entire history of all bots run by him/her.

UC-012



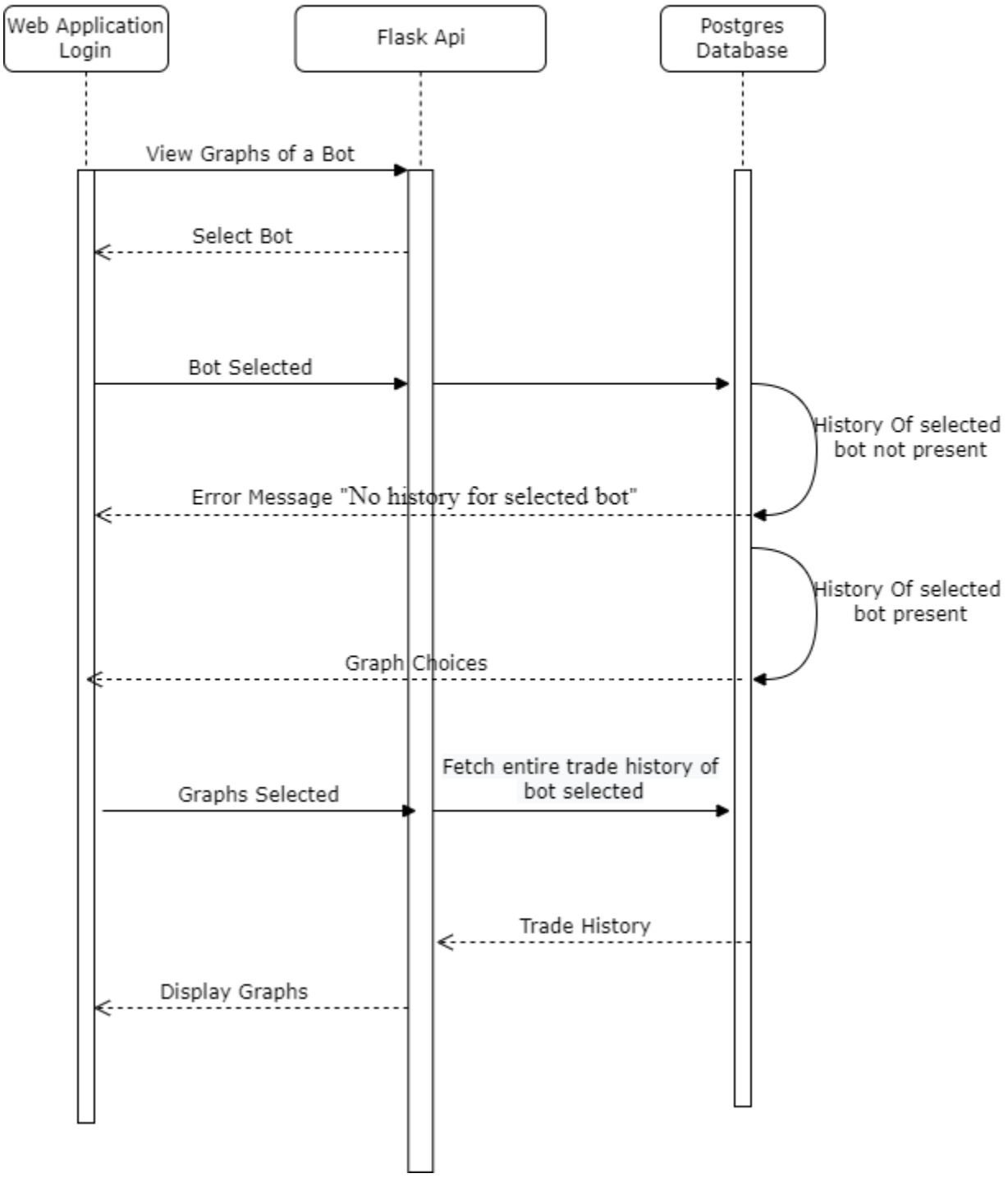
5.13 The analyst can view the entire performance of a bot.

UC-013

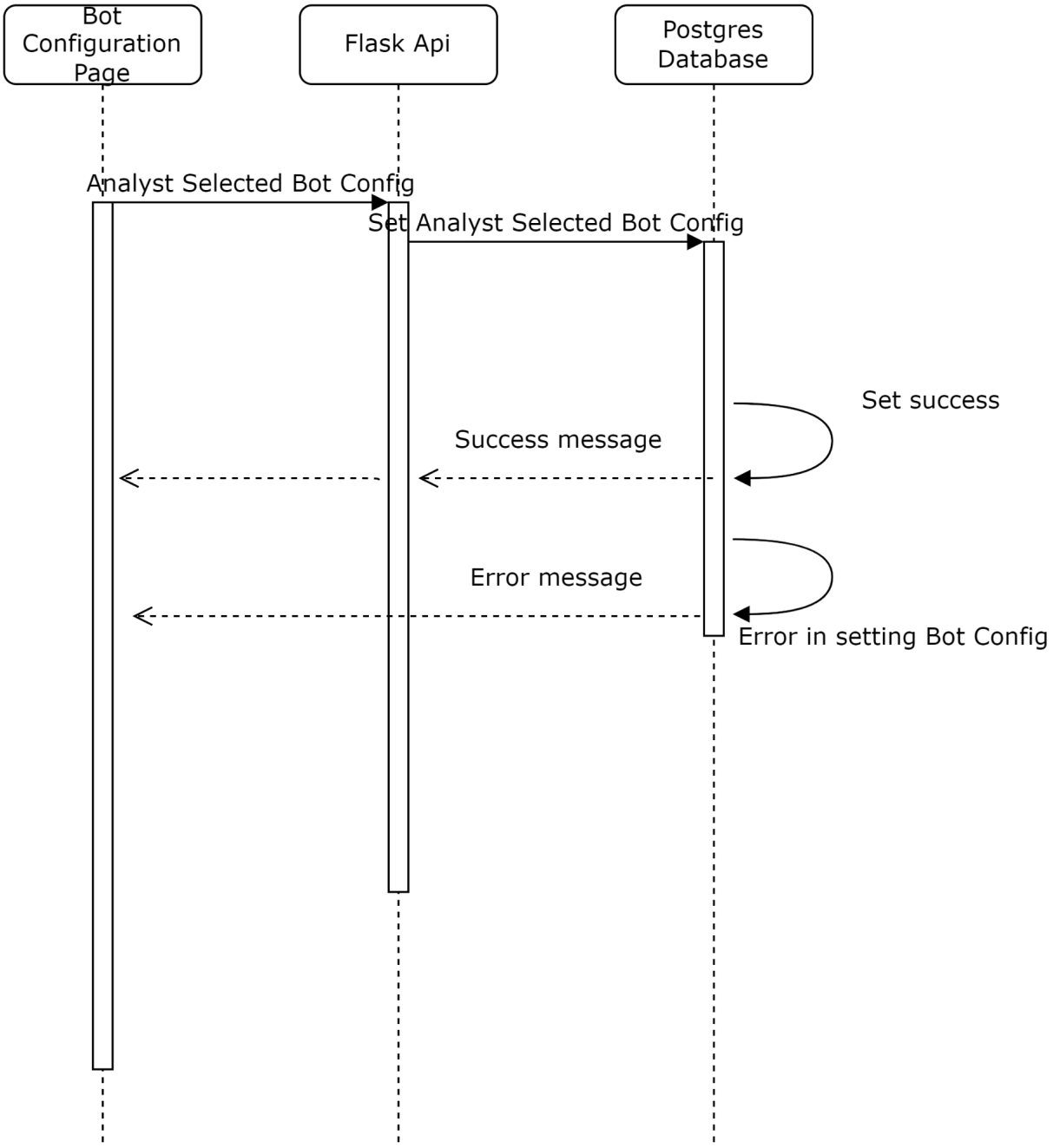


5.14 The analyst can view the multiple graphs of a bot.

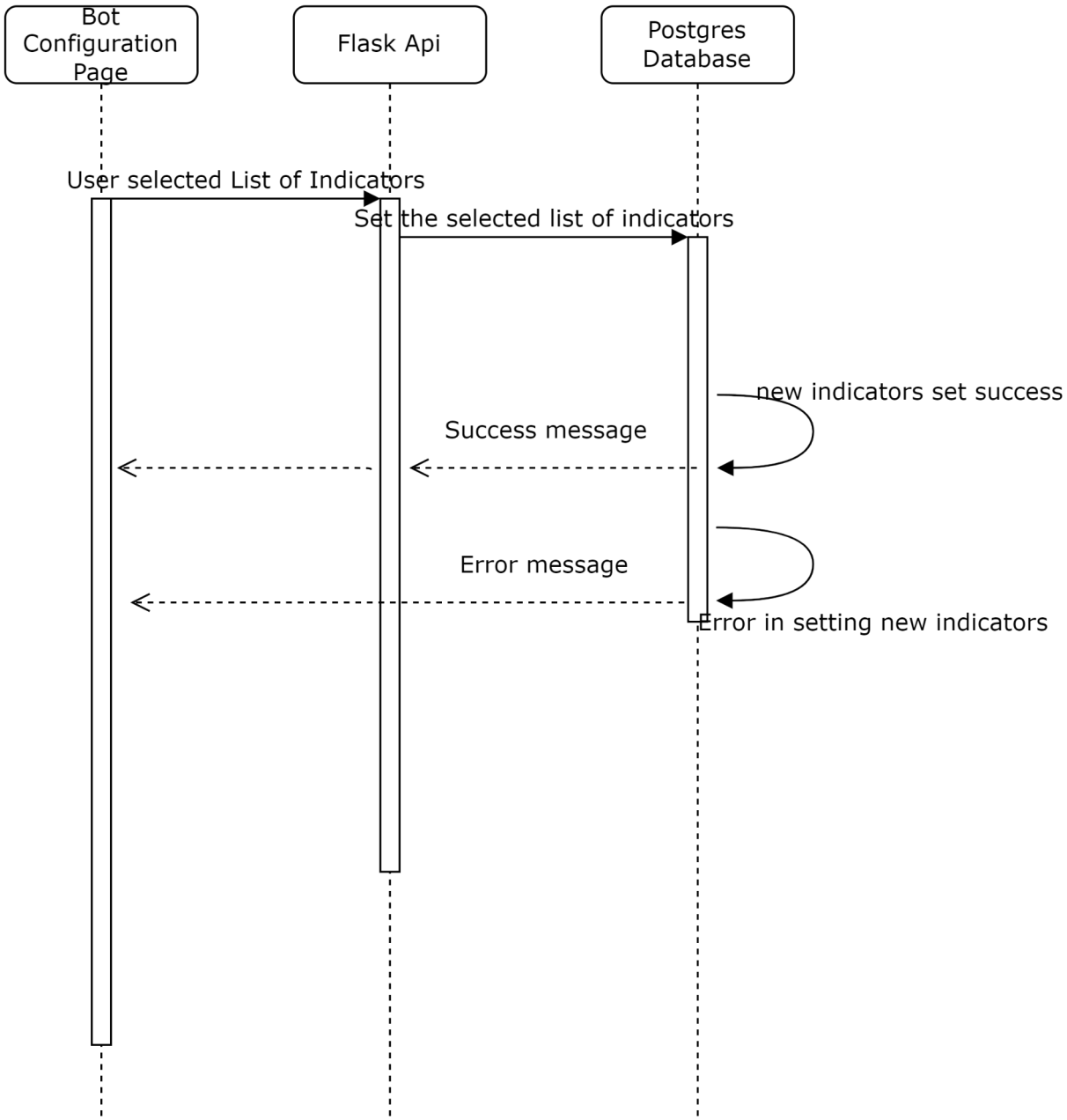
UC-014



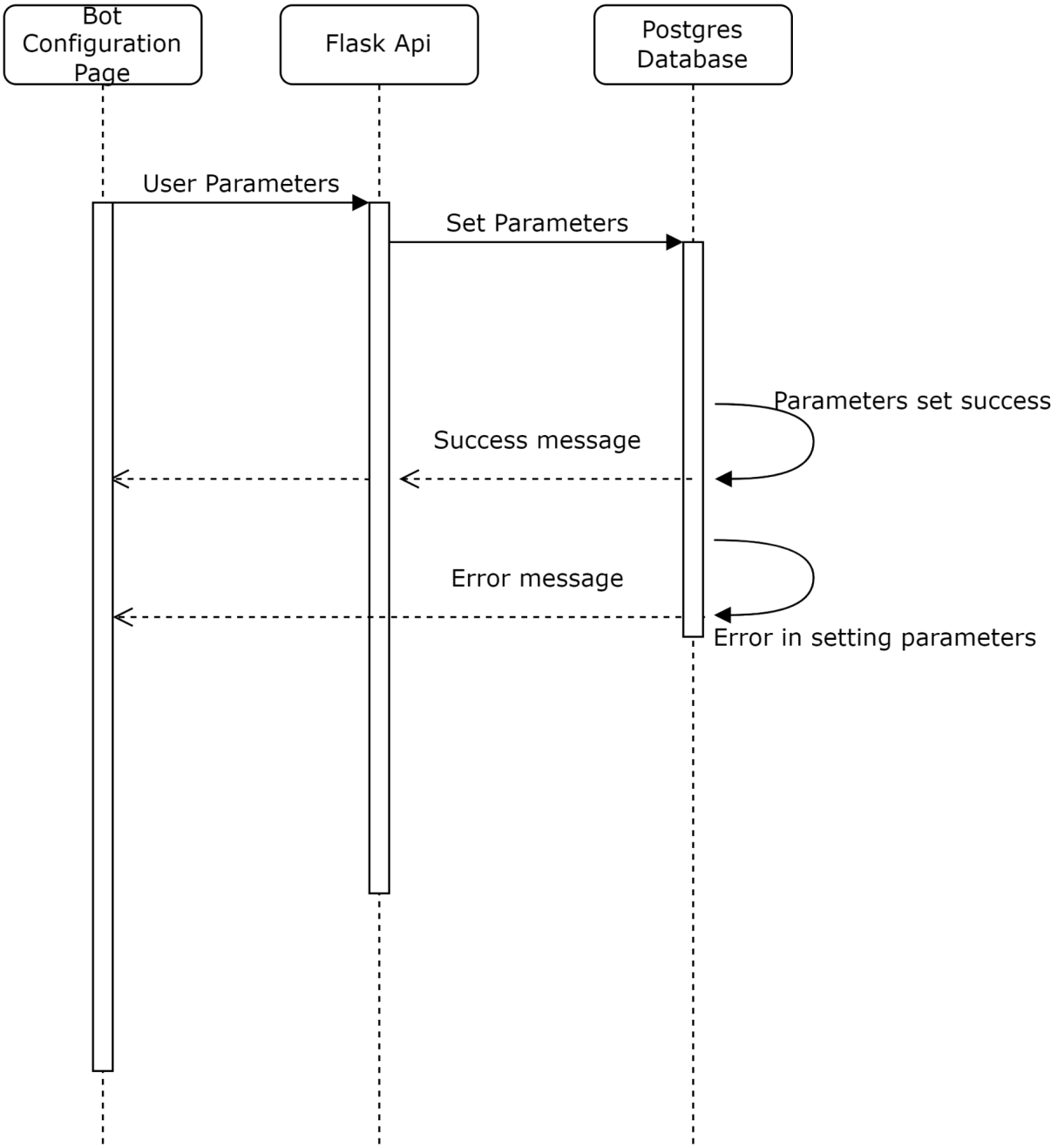
5.15 As an analyst, I want to choose which ML models to use for a bot.



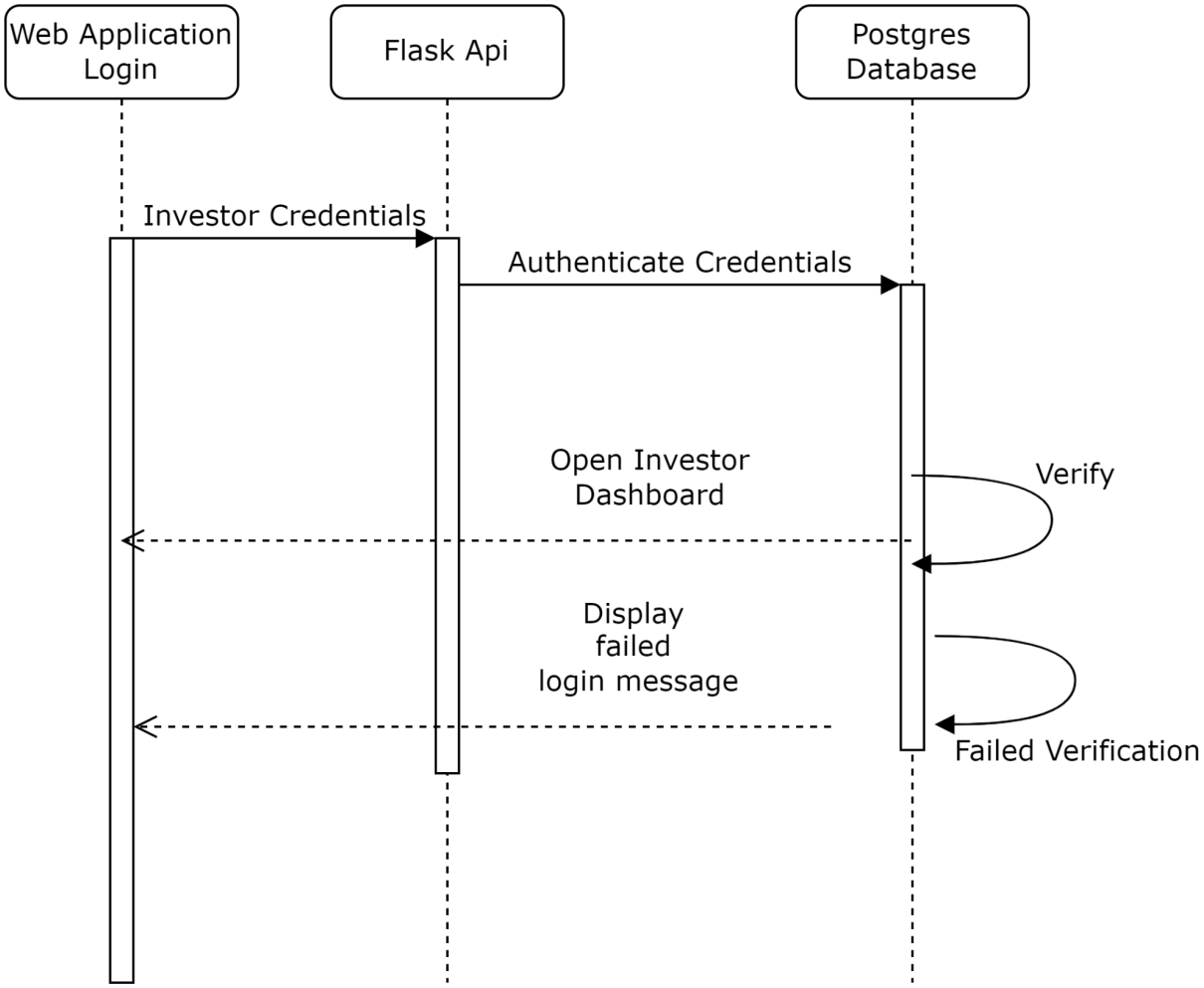
5.16 As an analyst, I want to choose a list of indicators to use for a bot.



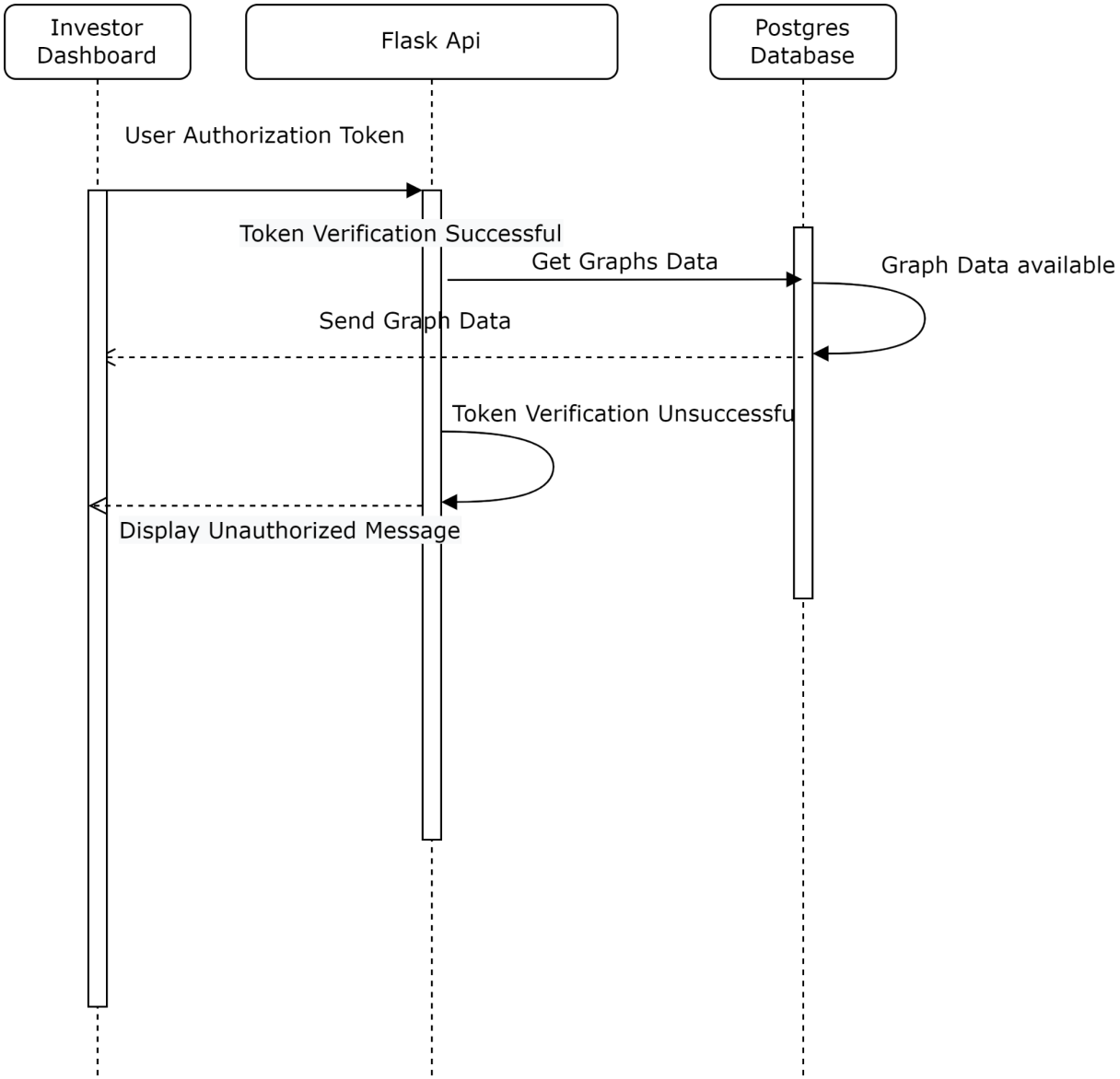
5.17 As an analyst, I want to configure the parameters of my chosen indicators for the bot.



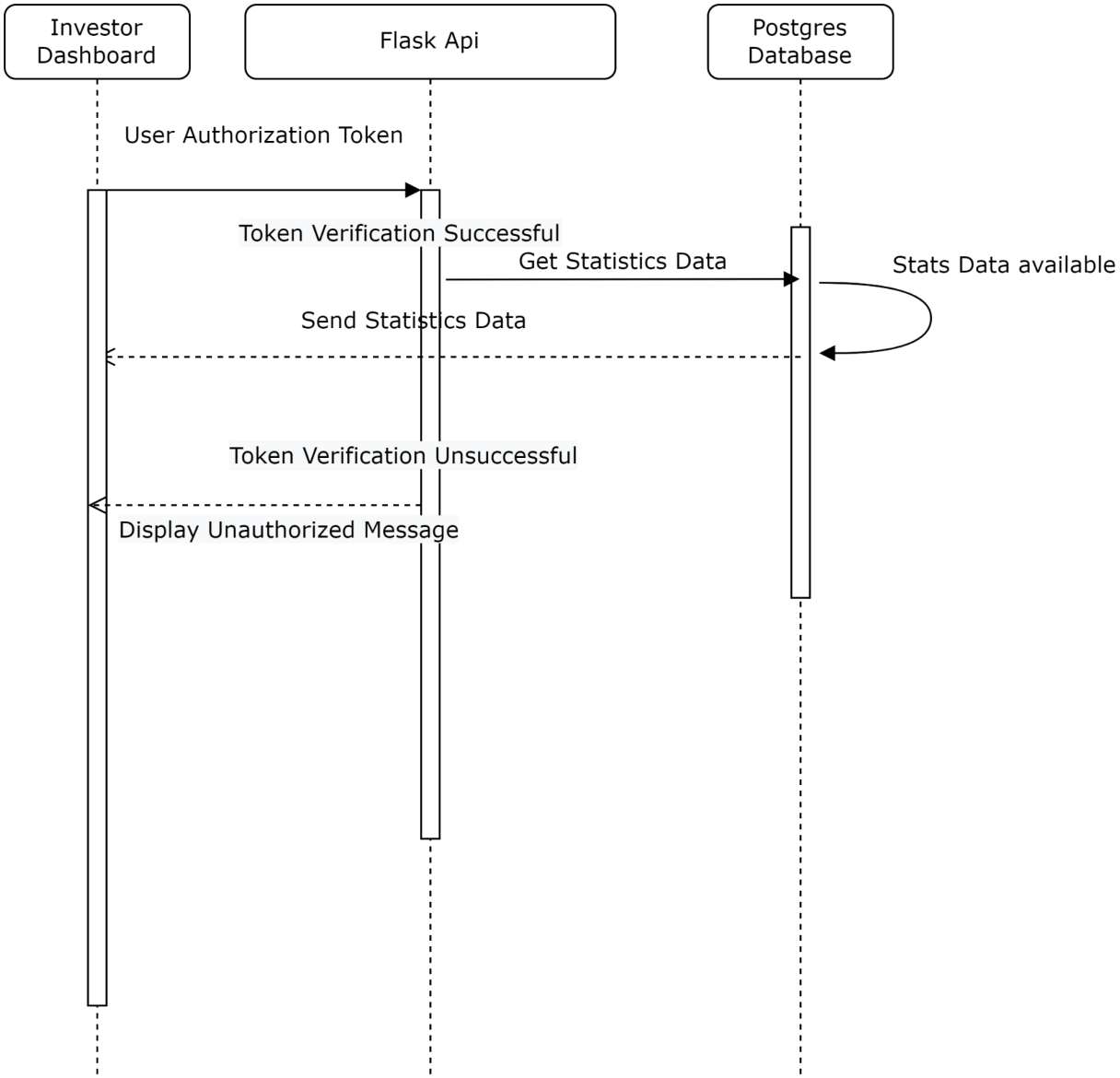
5.18 As an investor, I want to be able to login to my dashboard with the credentials provided by my analyst.



5.19 As an investor, I want to be able to see various graphs for my portfolio.



5.20 As an investor, I want to be able to see statistics of my bot's performance.



6. State Diagrams

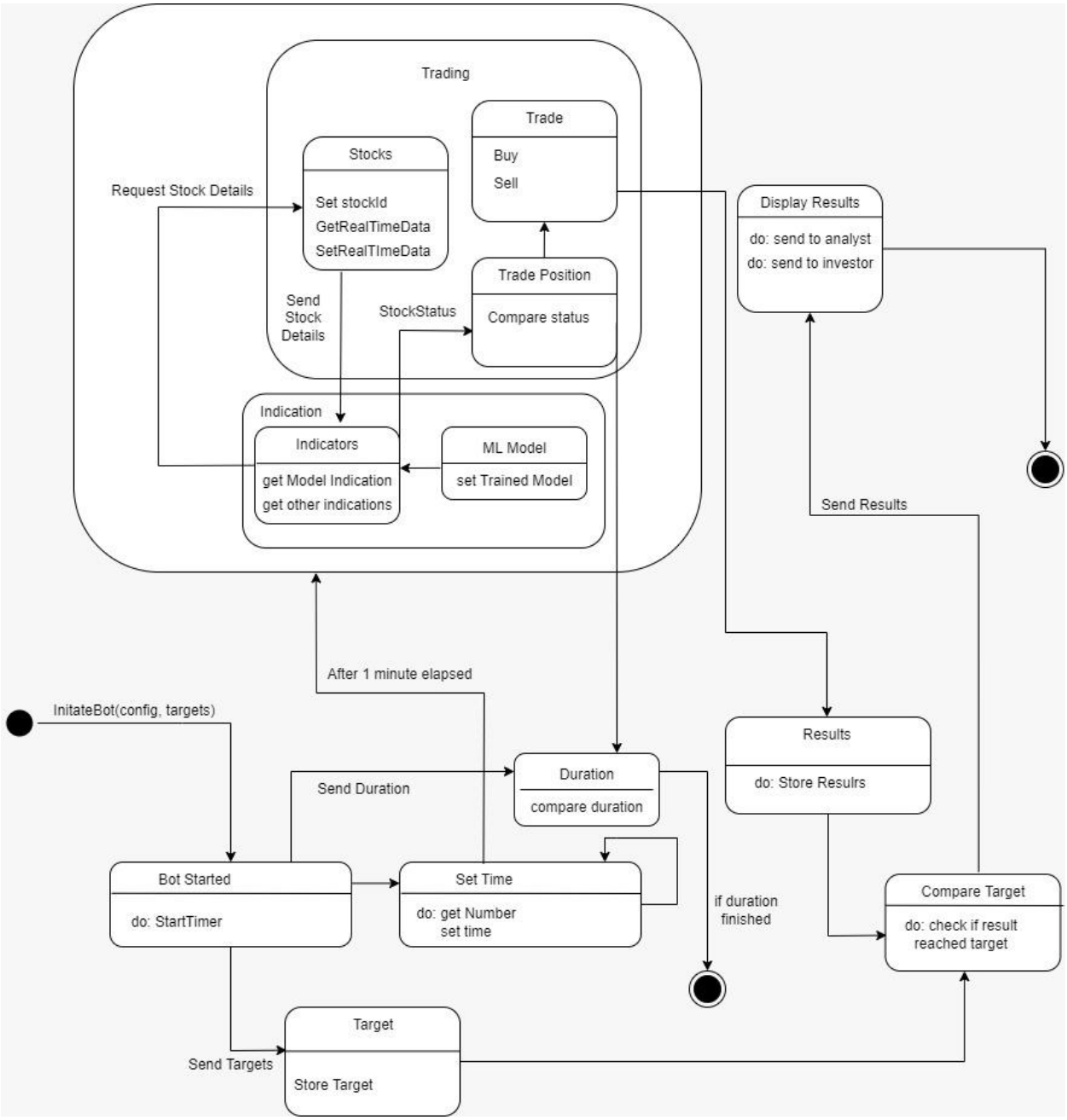
6.1 Diagram details

The following diagram shows the overall system's state where our main focus is on bot making the trades. The bot is initiated, which will start the timer. After every 1 the session will check for the values of stocks. The trading instance will check if it has to sell the stock and it fulfills the target return value provided in the bot config then it will sell and close the profit. It will use the indicators to get the buying on the selling point of stock and make the decisions accordingly.

The bot will stop if the target return is reached, or duration is completed. It will store the summary so that analysts can keep track of the bot results and understand the pitfalls to manipulate the configuration for the next investment.

When making a trade, the system is responsible to store trade data that can be visible to investors and analysts. The report will be displayed according to the duration for which the bot was running and made successful trades. Analysts and investors can generate reports of the trades of a specific range of time period could be in days or months depending on their demands.

6.2 Diagram


Trading
Trade
Buy
Sell
Stocks
Set stockId
GetRealTimeData
SetRealTImeData
Request Stock Details
Display Results
do: send to analyst
do: send to investor
Trade Position
Compare status
Send
Stock
Details
StockStatus
Indication
Indicators
get Model Indication
get other indications
ML Model
set Trained Model
Send Results
After 1 minute elapsed
InitateBot(config, targets)
Results
do: Store Resulrs
Duration
compare duration
Send Duration
Bot Started
do: StartTimer
Set Time
do: get Number
set time
if duration
finished
Compare Target
do: check if result
reached target
Target
Send Targets
Store Target

7. Non-functional Requirements / Quality Attributes

Sr#	Requirements
1	The system should complete execution of any request within 1 minute.
2	The system should entirely and accurately respond to requests at least 90% of the time it is invoked.
3	The time to get current stock prices from pakistan stock exchange should not be more than 5 seconds
4	The system should be available from Monday to Friday 9 am to 5 pm.
5	The bot should ping the trade instance every 1 minute to get the decision
6	The personal data of the investors and analysts will be stored in encrypted form
7	The trade profits, invested amounts, buying price, and selling will be in encrypted form as they are sensitive
8	Passwords should be salted and hashed before storage.
9	Multiple running bots will still ensure the individual performance of decision-making and request handling is catered within 60 seconds.
10	There must be at least 5 unique stock's pairs where the bots can trade

8. Who Did What?

Name of the Team Member	Tasks done
Syed Talal Hasan	Use cases, Sequence Diagrams, Introduction
Suleman Mahmood	Use case diagram and use cases
Ahmed Tahir Shekhani	Class Diagram with details, state diagram with detail, and nonfunctional requirements
Ali Asghar	Use cases and Sequence Diagrams
Ammar Ibraheem	Use cases and Sequence Diagrams

9. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

Section Title	Reviewer Name(s)
Class Diagram and Description	Syed Talal Hasan
Class diagram, State diagram	Suleman Mahmood
Use Case Diagram, Use cases	Ahmed Tahir Shekhani
Introduction	Ali Asghar
Non-functional requirements	Ammar Ibraheem