UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
Faculty of Engineering, Built Environment & IT
University of Pretoria

TradeSim

Software Requirements Specification

AiPi

September 2022

Rachel Hamilton, u20453478

A. Introduction:

This documentation has all the necessary documentation from the AiPi team.

Team members:

* Michael Viljoen

* Rachel Hamilton

* Regan Shen

* Sean Nkosi

C. Project:

TradeSim

1 Introduction

1.1. Purpose of product:

The purpose of the following documentation is to provide the software requirements (functional and non functional), acceptance criteria, constraints, architectural requirements, diagrams and an overview of the TradeSim system. Our role is to build a TradeSim system that will be responsible for containing all information for a user to create and customize their own mini-ETF's by setting rules for their ETF. The user will be able to track how well their ETF is doing, and use this to decide where they want to invest their real money in the future.

1.2 Scope of product:

TradeSim users will be able to sign up and sign into their account and have access to their profile where they are able to create and customize ETF's by setting their own rules. The goal is to create a system that will allow a user to create an account, and create their own rule-based mini-ETF's - they can then track their ETF's and compare them in time. The result will show the user how much capital they'll need to invest in the various stocks to make their own "mini-ETF" competitive against the established ETF on the market. The user is able to download these rules to a file, so they can use them for future reference when investing with their real money. By using the rules

the user has created, they'll be able to enhance the performance of their own mini ETF. Users are able to change their mini-ETF rules, to enhance the performance of their mini-ETF over time.

1.3 Acronyms, Abbreviations and Definitions:

- FR – Functional Requirements
- NFR – Non Functional Requirements
- API - Application Programming Interface.
- UI - User Interface
- C – Constraint
- ETF – Exchange-traded fund

1.4. Overview:

The document follows the following scheme:

- An Overall description
- Specific Requirements
- Acceptance Criteria
- Architectural requirements and design patterns
- Diagrams

2 Overall Description

2.1 User Stories:

• The TradeSim user creates an account

• The TradeSim user logs in to their account

• The TradeSim user clicks on create ETF and creates their ETF according to name and amount they want to invest.

• The TradeSim user sets and applies rules to their mini-ETF.

• The TradeSim user can then go the home page and select on their mini-ETF for viewing

• The TradeSim user selects a date in history to visualise their mini-ETF at that point in time.

• The TradeSim user can make multiple mini-ETF's with different rules.

• The TradeSim user visualises selected mini-ETF's and compares them at a date in time.

• The TradeSim user edits a selected mini-ETF's by changing their rules to improve the performance of their mini-ETF.

• The TradeSim user selects the mini-ETF's to share and can download the rules as a file.

• The TradeSim user imports a mini-ETF to the TradeSim system to have as their own ETF.

2.2 User Characteristics:

The TradeSim system is intended for the use of: Users who want to invest in an ETF through the stock exchange. Users intend on creating their own mini-ETF's and track its performance over time to make real-life decisions with their money in the future.

2.3 Assumption and Dependencies:

- Assumptions:
  - The user is someone who wants to invest in an ETF.
  - The user has an internet connection.

  - The user is using a PC that has a UI.

- Dependencies:
  - Time management - Our group is dependent on time management to be able to complete our project.
  - Lack of knowledge to create a solution for a requirement - Lack of knowledge to implement a feature has a big effect on our team completing this project.

3 Specific Requirements

3.1 Functional Requirements:

- FR.1. The TradeSim system must allow a user to register a TradeSim account.
- FR.2. The TradeSim system must allow a user to login to their registered account on TradeSim.
- FR.3. The TradeSim system must allow a user to create their mini-ETF by name and amount they want to invest into it.
- FR.4. A user should be able to define the rules against which the system should trade in their mini-ETF. Our team's rules consist of the following:
  - FR.4.1. The system must allow the user to request certain companies by name or ticker.
    - FR.4.1.1. The system must allow the user to set a percentage in their specified company they want to invest in.
  - FR.4.2. The system must allow the user to request a percentage in a specific sector.
    - FR.3.2.1. The system must allow the user to set a percentage in their specified sector they want to invest in.

- FR.4.3. The system must allow the user to request percentage in a specific industry.
    - FR.3.3.1. The system must allow the user to select a specific percentage in their specified industry they want to invest in.
- FR.4.4. The system must allow the user to request a minimum amount of companies they want to invest in.
- FR.4.5. The system must allow the user to set a time period in which it must reconsider its stocks.
- FR.4.6. The system must allow the user to define a percentage that a stock can drop before the ETF sells the stock automatically.
- FR.4.7. The system must allow the user to set the market cap min and max values.
- FR.4.8. The system must allow the user to set the earnings min and max value.
- FR.4.9. The system must allow the user to set an amount to invest.
- FR.4.10. The system must allow the user to reject specific companies by name or ticker.
- FR.4.11. The system must allow the user to reject specific sectors by name or ticker.
- FR.4.12. The system must allow the user to reject specific industries by name or ticker.
- FR.4.13. The system must allow the user to invest in companies based in specific countries.
    - FR.1.13.1. The system must allow the user to set a percentage based on a country of a specific company.
- FR.4.14. The system must allow the user to reject companies based in specific countries.
- FR.4.15. The system must allow the user to set a minimum and maximum price for shares.
- FR.4.16. The system must allow the user to set a balance period and/or balance threshold percentage).

- FR.5. The system must apply the rules against historic market data.
- FR.6. The system must compare the performance of the mini-ETF against other indexes in the market.
- FR.7. The system must allow the user to visualise the performance of their mini-ETF's.
- FR.8. The system must allow the user to edit their mini-ETF's. (change the rules)
- FR.9. The system must allow the user to be able to save their rules and also test these different rules against each other.
- FR.10. The system must allow the user to be able delete their mini-ETF.
- FR.11. The system must allow the user to be able to reset their mini-ETF.
- FR.10. The TradeSim system must allow a user to logout of their signed in account on TradeSim.

3.2 External Interface Requirements:

3.2.1. User Interfaces

- The TradeSim system can only be accessed by a PC with a UI.

3.2.2. Hardware Interfaces

- Connection to the internet.

3.2.3. Software Interfaces

- Web browser.

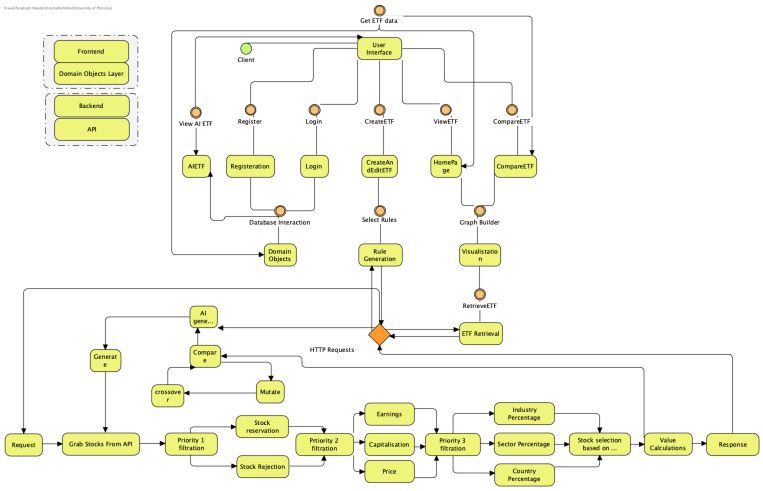
3.3 Performance Requirements:

All the features of the TradeSim system must function as expected just as any trading system on a web app should.

3.4 Design Constraints:

The TradeSim system is dependent on the design requests from the project owners, and therefore we cannot go forward without communication with the project owners informing us about the design requirements needed.

4 Architectural Design



Architectural Design Strategy

- We used generating test cases as our architectural design strategy. This shows all test cases the user can perform in our TradeSim system, from registration to creating their mini-ETF's. Each component's output is based on a test case, and this takes the user to the next component. This is shown in the diagram above.

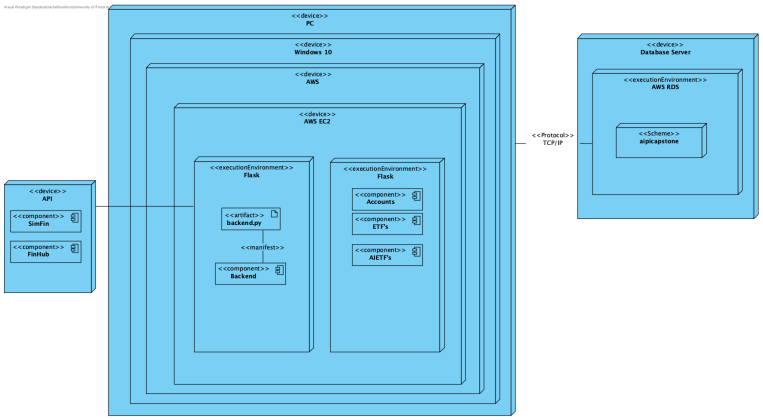
Architectural Styles

- Pipe and filters
  - The TradeSim system architectural design consists of a chain of

elements, it is placed in order of process - so that the output of each element is the input of the next. The backend (bottom half of diagram) is the pipe and filter style.

- Component Based
  - All TradeSim system processes are in separate components so that all of the data and functions inside each component are related. Components are modular. The frontend (top half of the diagram) uses a component based style.

5 Deployment Diagram



3.5 Quality Requirements:

1. Data integrity - The TradeSim user must have access to certain information such as their rules created, that must remain private. This information cannot be leaked and should only be accessed by the TradeSim user.

2. Security - Certain information on the users TradeSim account that needs to remain private, for the user's eyes only, can not be accessed by anyone else unless they login to their TradeSim account.

3. Performance - All the features of the TradeSim system must function as expected according to the information and requirements given by the project owners.

4. Availability - The TradeSim system must be accessible to the TradeSim users at all needed times.

5. Usability - TradeSim must be mobile friendly. The TradeSim system must provide all people of all technological literacy to be able easily utilise the platform. The user interface of the platform must be simple and understandable to ensure users have a good experience when using the platform. The system will provide users with guidance or descriptions on how to utilise the functionality of the system.

6. Reliability - Users must be able to utilise the TradeSim system's functionality consistently and reliably. The number of users utilising the system should not negatively impact the system's functionality.

7. Maintainability - Users must be able to delete their ETF's and reset their ETF's. ETF's must be maintained and their information must be updated at each point in time. When rules are updated or information is changed, the system must be maintained.

3.6 Architectural Constraints:

C1. The system must be maintained and managed by the AiPi team.

C2. All system implementations and documentation must be done by the AiPi team.

C3. The TradeSim system design must be the design the AiPi team follows throughout the project.

C4. The frontend engineers must use a python framework to complete this project.

C5. The backend engineers must use Python to complete this project.

C6. The API Engineer of AiPi must use Symfin or Finhub to complete this project.

C7. The Data Engineer of AiPi must use SQL to complete this project.

3.7 Architectural Requirements:

3.7.1 Flexibility:

- The TradeSim system must function on a device that has a UI.
- The TradeSim system must function using all web browsers.

3.7.2 Maintainability:

- There must be constant communication between developers of the AiPi team and the project owners in order to identify and fix errors to improve the quality of the feature.
- Clear documentation of requirements must be provided to ensure maintainability.

3.7.3 Security

- Certain information on the TradeSim user's account that needs to remain private, for the user's eyes only, can not be accessed by anyone else unless they login to the company representatives profile.

3.7.4 Availability:

- The TradeSim user's account must be available and easily accessed by any device, desktop or mobile with a UI.
- The TradeSim user's account must be available at all times.

3.7.5 Reliability:

- TradeSim is reliable on internet connection.
- The TradeSim system must ensure a stable experience to the user.

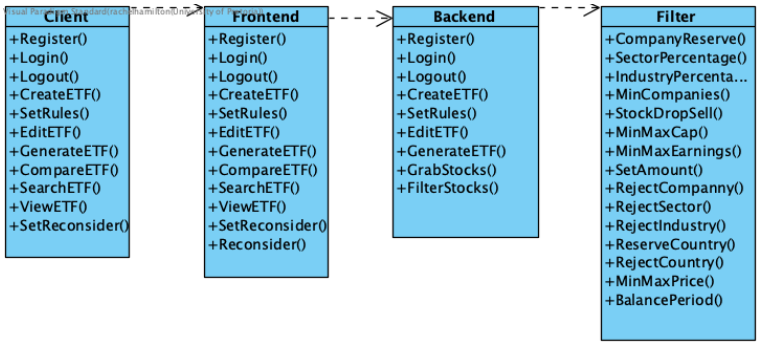
3.6.6 Usability:

- TradeSim must be mobile friendly.

6 Attributes

- Availability
  - The TradeSim system must be accessible to the TradeSim users at all needed times.
- Security
  - Private information belonging to the TradeSim user cannot be leaked, and must remain private.

7 Class Diagram



8 Acceptance criteria

4.1 All information that the TradeSim user gives to the system must be in the database and displayed on the user interface without hiccups. (e.g. rules).

4.3 The system must allow the user to be able to import and set rules for their mini-ETF.

4.4 The system must allow the user to login to their account.

9 Use-Case Diagram

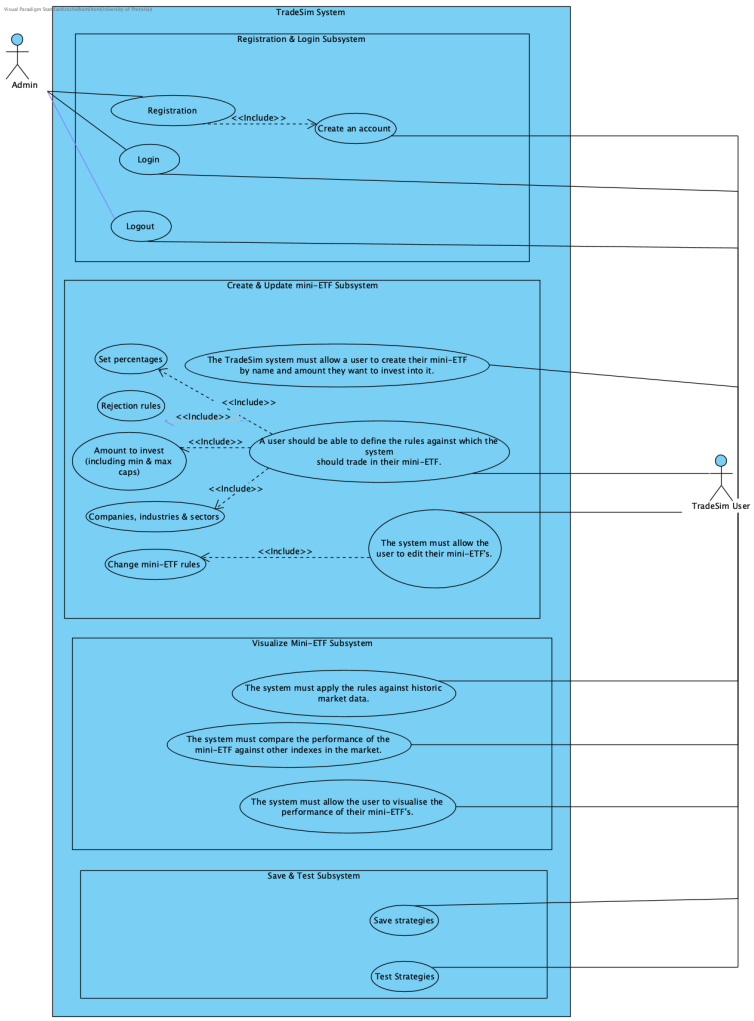


Figure 1 is a use-case diagram representing the TradeSim system.

10 Service Contracts

Use Case ID	UC1
Use Case Name	Create account
Priority	High
Actors	TradeSim user
Precondition	The user is on the registration page.
Reason	The user wants to create an account with TradeSim
Main Scenario	1: The User inputs their details (Full Name, Email Address, Phone Number, Password, etc.) 2: The User presses the 'Register' button. 3: The system will validate the users' details. 4.1: If the users' details are invalid, prompt the user to try again. 4.2: If the users' details are valid, add them to the database and notify the user that the account has been successfully created. 5: The verification request will be added to the admins' 'verification requests'.
Alternate Scenario	None
Postcondition	The user will be redirected to the front page of the website, now logged into their account.

Use Case ID	UC2
Use Case Name	Login to TradeSim
Priority	High
Actors	TradeSim user
Precondition	The user must have an existing account with TradeSim
Reason	The user wants to login into their respective account.
Main Scenario	1: The user presses the 'Login' button. 2: The user enters their details. 3: The user presses the 'Submit' button. 4: The system validates the user's details, using the database. 5.1: The details are valid, the system notifies the user they have successfully logged in. 5.2: The details are incorrect, the system prompts the user to try again.
Alternate Scenario	None
Postcondition	The user must be now logged into their account and be redirected to their respective home page.

Use Case ID	UC3
Use Case Name	Create ETF
Priority	High
Actors	Tradesim user
Precondition	The user must be on their TradeSim account
Reason	The user wants to create their mini-ETF by name and amount they want to invest in.
Main Scenario	1: The user selects the create ETF button in the nav-bar 2: The system takes the user to a new page 3: The user types in the name of their mini-ETF and the amount they want to invest in 2 separate textboxes.
Alternate Scenario	None
Postcondition	The user can set the rules for their mini-ETF

Use Case ID	UC4
Use Case Name	Define the rules against which the system should trade.
Priority	High
Actors	Tradesim user
Precondition	The user must have created their mini-ETF by name and amount they want to invest in
Reason	The user wants to define rules against the system should trade.
Main Scenario	1: The user selects the rule from a dropdown menu which they want to apply to their mini-ETF. 2: The user can click a plus icon at the bottom of the page to add more rules 3: The user confirms their decision by clicking a submit button, and the rules are applied to their mini-ETF.
Alternate Scenario	None
Postcondition	The user can examine their mini-ETF on the home page.

Use Case ID	UC5
Use Case Name	Apply rules against historic mock data.
Priority	High
Actors	Tradesim user
Precondition	The user must be on their TradeSim account and have chosen their rules.
Reason	The user wants to apply rules against historic mock data.
Main Scenario	1: Once the user has created their ETF, the user will be able to see the performance of the ETF from previous times. 2: The user clicks on their ETF on the home page and they are taken to a new page 3. The user selects a date in which they want to see their ETF 4. A graph with data is shown about their ETF at that chosen date.
Alternate Scenario	None
Postcondition	The user can track the performance of their ETF over time.

Use Case ID	UC6
Use Case Name	Mini-ETF performance should be measured against other indexes.
Priority	High
Actors	Tradesim user
Precondition	The user must be on their TradeSim account and have created multiple ETF's
Reason	The system compares the performance of the mini-ETF's to other ETF's.
Main Scenario	1. The user selects on 2 of their ETF's under the compare ETF's tab on the nav bar. 2. This takes the user to a new page 3. On the new page, the user selects a date where they want to compare their ETF's 4. 2 graphs with corresponding data about both ETF's are shown according to their selected date 5. The user is able to compare both ETF's at that date in time.
Alternate Scenario	None
Postcondition	The user can make decisions and/or changes to their ETF's according to the data they have viewed.

Use Case ID	UC7
Use Case Name	Visualise the performance of the mini-ETF's
Priority	High
Actors	Tradesim user
Precondition	The user must have created a mini-ETF and rules for their mini-ETF.
Reason	The user wants to visualise the performance of the mini-ETF's to see how successful their ETF is doing using their rules.
Main Scenario	1: This will be displayed in the form of a graph. 2: Performance of the ETF will be visualised on the home page.
Alternate Scenario	None
Postcondition	The user can save or test their strategy.

Use Case ID	UC8
Use Case Name	Edit ETF
Priority	High
Actors	Tradesim user
Precondition	The user must have created a mini-ETF and rules for their mini-ETF.
Reason	The user wants to edit the rules of their ETF to improve performance of the mini-ETF.
Main Scenario	1. The user selects Edit ETF in the nav-bar on the home page. 2. The user then selects the ETF they want to edit. 3. The user then changes the rules of their ETF and clicks on the apply changes button 4. Changes are then saved and updated to their mini-ETF.
Alternate Scenario	None
Postcondition	The user can view their ETF on the home page.

Use Case ID	UC9
Use Case Name	Save Strategies
Priority	High
Actors	Tradesim user
Precondition	The user must be on their TradeSim account and have chosen their rules.
Reason	The user wants to apply rules against historic mock data.
Main Scenario	1: Once the user has created their rules, their rules are saved on a database. 2: The strategies are saved automatically
Alternate Scenario	None
Postcondition	The rule is used on their mini-ETF

Use Case ID	UC10
Use Case Name	Log out of TradeSim
Priority	High
Actors	Tradesim user
Precondition	The user must be currently logged into their account.
Reason	The user wants to log out of their account.
Main Scenario	1: The user presses the 'Logout' button. 2: The system prompts the user to confirm their decision. 3.1: The user confirms their decision, and the user is logged out of their account. 3.2: The user denies their decision and is returned to the previous page.
Alternate Scenario	None
Postcondition	The user is redirected to the login page.

11 Traceability matrix

Registration & login Subsystem				
		Requirements		
		FR1	FR2	FR8
Use Case	UC1	x		
	UC2		x	
	UC10			x

Create & update Mini-ETF Subsystem																		
		Requirements																
		FR3	FR4.1	FR4.2	FR4.3	FR4.4	FR4.5	FR4.6	FR4.7	FR4.8	FR4.9	FR4.10	FR4.11	FR4.12	FR4.13	FR4.14	FR4.15	FR 4.16
Use Case	UC3	x																
	UC4		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	UC8		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Compare Mini-ETF Subsystem			
		Requirements	
		FR5	FR6
Use Case	UC5	x	
	UC6		x
	UC7		x

Save Strategies Subsystem		
		Requirements
		FR7
Use Case	UC9	x

Traceability matrix of Subsystems					
		Subsystems			
		Registration & login	Create & update ETF	Compare ETF	Save Strategies
Use Case	UC1	x			
	UC2	x			
	UC3		x		
	UC4		x		
	UC5			x	
	UC6			x	
	UC7			x	
	UC8		x		
	UC9				x
	UC10	x			