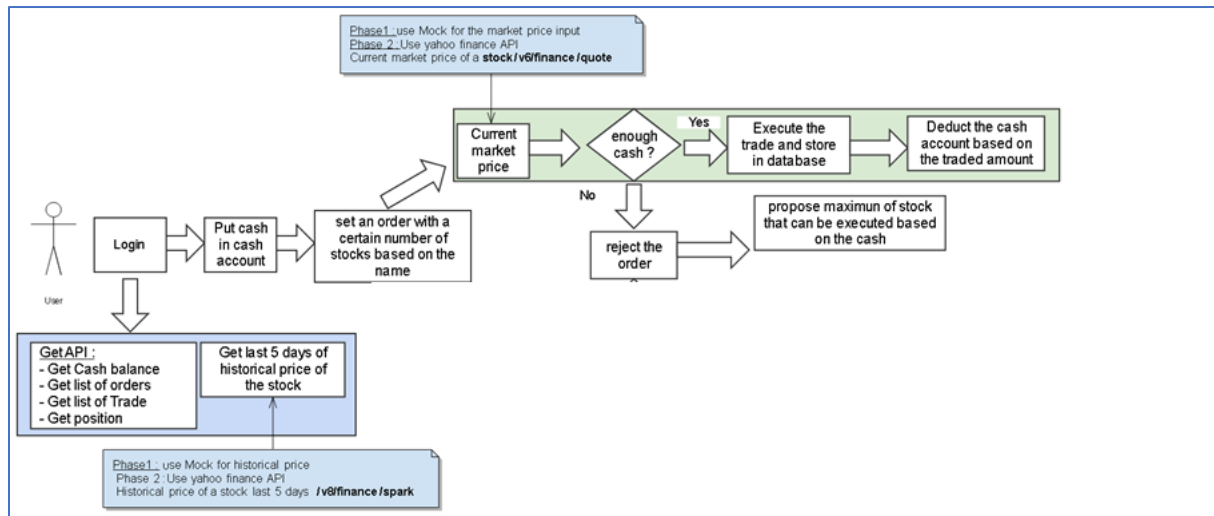


## 1. Trading App Use Case and Flow



## 2. User Stories

#	Title	User Story	Acceptance Criteria
1	TRADING_LOGIN	As a trading user, I should be able to login in application using account username and password	Enter a username and password. If found, the profile is updated with the latest login date time. The system performs a lookup of access and the user can proceed post login
2	TRADING_PROFILE_DE TAILS	As a system, whenever the login is successful fetch user additional profile details and application information	For the valid logged in user, system should be able to fetch <ul style="list-style-type: none"> <li>- Get Cash Balance</li> <li>- Get List of Orders</li> <li>- Get List of Trades</li> <li>- Get Position</li> <li>- Get market historical price of a stock (refer user story 7)</li> </ul>
3	TRADING_ADDCASH	As a trading user, I should have provision to add more cash in cash account to continue trading	On successful login, allow user to add additional cash at any point if balance is insufficient for existing trade , or user wants to add Cash for upcoming trade

4	SEARCH_STOCK	As a trading user, I should be able to search for a specific stock.	System should fetch the stocks based on the stock name provided
5	SET_ORDER	As a trading user, I should be able to set the order by specifying the numbers.	System should accept number of stocks that user intends to buy. This should only be decimal.
6	TRADING_VALIDATE_CASH	As a system, validate cash before executing the order	Validate the quantity of stocks, current market price of stock ( refer user story 7) and respective cash amount.
7	TRADING_REJECT_ORDER	As a system, reject order for insufficient cash amount	If the cash amount is insufficient reject system should reject the order and not allow the Trading user to proceed further During rejection scenario, positively suggest user on the quantity of stocks that can be ordered for the available price
8	TRADING_BOOK_ORDER	As a trading user, I should be able to set an order based on number of stocks	Validate the quantity of stocks, current market price of stock ( refer user story 7) and respective cash amount. If the cash amount is sufficient proceed for the order in system. Post successful order, deduct the amount based on the traded amount from the account
9	TRADING_FINANCE_API_INTEGRATION	As a system, Integrate with the API to get the current price of the stock and historic price of the stock.  Technically this can be done in 2 ways: Option 1: 1. Develop Mock API's which can return current stock price	<ul style="list-style-type: none"> <li>- Current market price of a stock</li> <li>- Historical price of a stock last 5 days</li> </ul>

		<p>2. Develop Mock API's which can return last 5 days of stock</p> <p>Option 2: Integrate with the below APIs: <a href="https://www.yahoofinanceapi.com/v6/finance/quote">https://www.yahoofinanceapi.com/v6/finance/quote</a> for real time Stock Data information</p> <p>or get historical prices of the stock for last 5 days <a href="https://www.yahoofinanceapi.com/v8/finance/spark">/v8/finance/spark</a></p>	
--	--	---	--

### 3. Guidelines

#### **Mandatory**

1. Use scala code
2. Code available in Github/ Bitbucket
3. Model in database cash accounts, order, Trade and Position (Sum of trades of the same stock)
4. Proper validation of the fields (mandatory and format validations)
5. Unit test cases for completed functionalities
6. Javadoc and Swagger documentation. Output to be shown via swagger

**Nice to have:**

Sonar report