**Amdocs On-Boarding**

**Training Project**

**On**

**Trading System**

****

**Compiled By:**

**Group 1-**

**Anshu(200693)**

**Manik Chauhan(200697)**

**Pradeep Singh(200707)**

**Prakhar Mishra(200854)**

**Prathmesh(200855)**

**Rachit Raj Singh(200765)**

**Raj Kumar Gupta(200825)**

**Ravi Prakash(200814)**

**Sakshi Kumari(200780)**

**1. Introduction**

**1.1. Purpose of the Document**

This document's goal is to give a general overview of the Core Java-based Desktop Trading System project. The requirements, architecture, functionality, user interface, and testing processes are all described.

**1.2. Project Overview**

Users can carry out a variety of trading tasks online using the Desktop Trading System, a desktop-based program that enables them to manage their trading accounts, create profiles, create buy/sell orders, buy and sell trade stocks, check trade activity, and more. The technology seeks to give traders a safe and convenient interface.

**1.3. Scope**

The following functionalities are included in the project's scope for the desktop trading system:

- User login and registration.

- Management of profiles (account creation, update, and deletion).

- Place a buy/sell order

- See the order book

- Trade equities

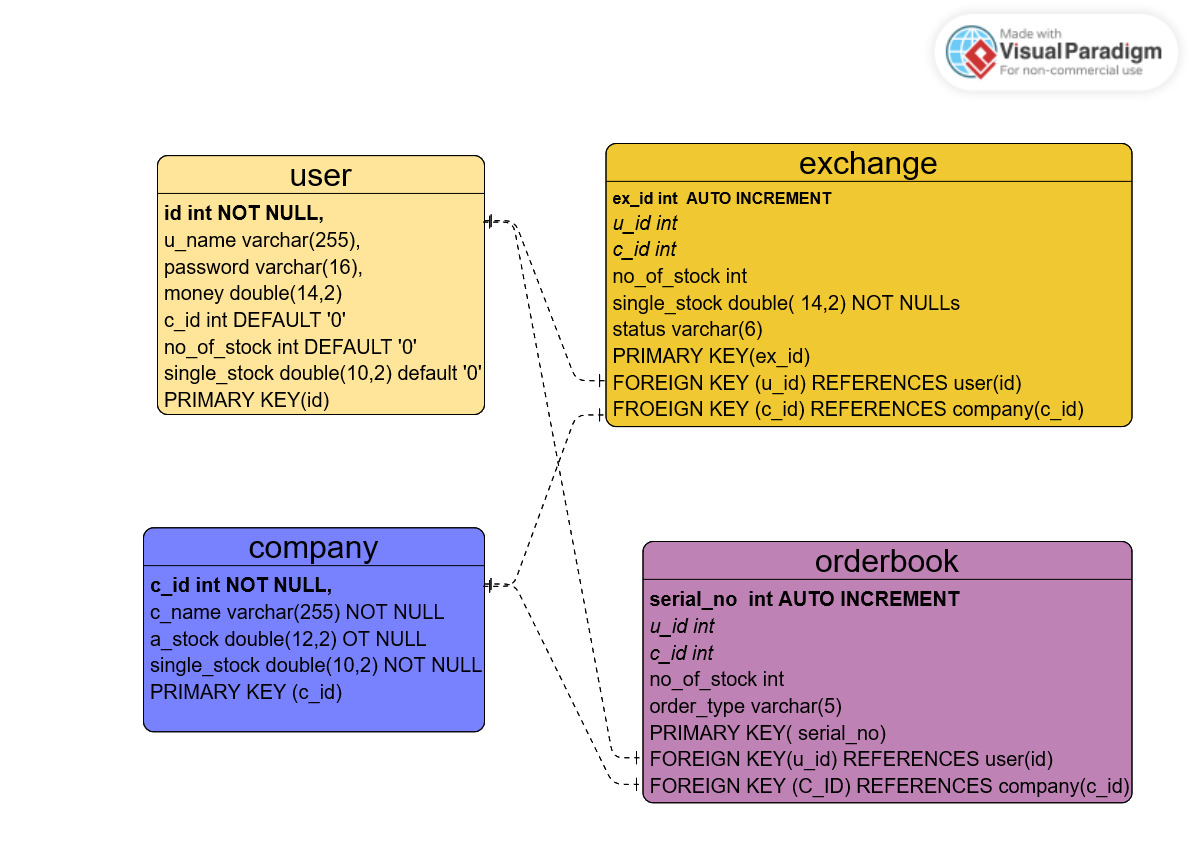
- Examine a user's trading activity

**2. Functionalities**

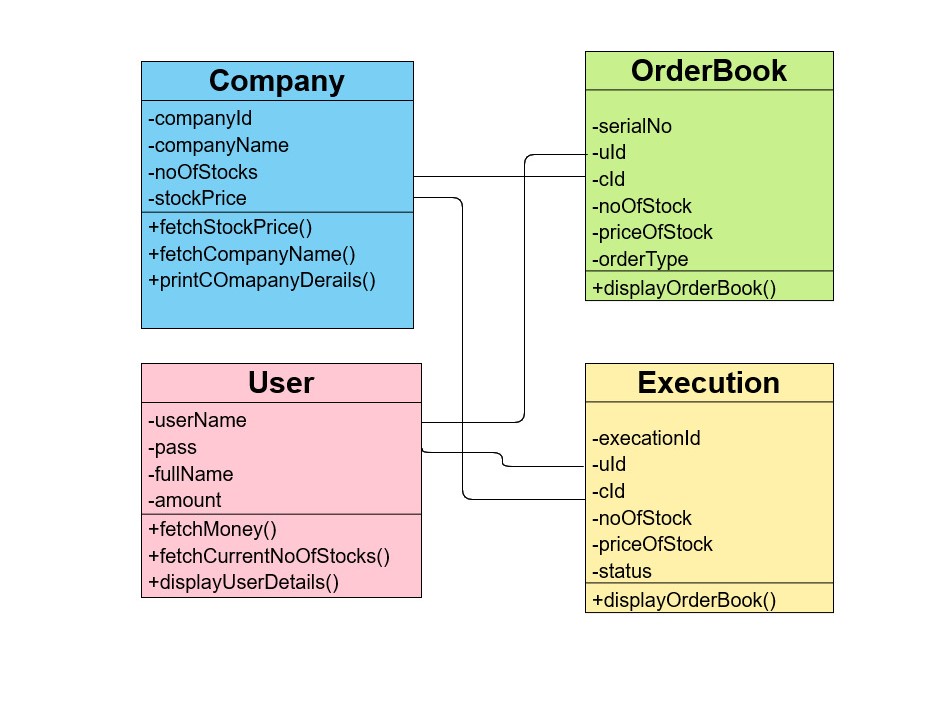
* **User Registration**- Users are able to create new profiles by providing necessary details.
* **User Login-** Registered users are able to log in securely to access their accounts.User are prompted to enter password for the CLI session
* **Trading-** Users are able to create a buy or sell order on an order book. The user can pre-own the stocks to be traded, the user can also own some credit to execute trades (buys)
* **View the order book -** This shows the current order book with the buy and sell orders which can be executed by users.
* **View Activity-**The users can view their trading activity.

**3. Architecture**

**3.1 ER Diagram:**

****

**3.2 Class Diagram:**

****

**4. Working**

**4.1 Exchange(Used for Selling & Buying):**

1. If someone wishes to sell or buy, we can add or change their user information, company information, stock number, price, and status (for instance, "sell" or "buy").
2. Let's say someone wants to buy, but there are already some people selling depending on the sell/buy choice. Then, since the exchange has already occurred, we transfer and update both users' data in the user table, remove them from the exchange table, and also add the transaction to the orderbook table.

**4.2 Update for Transaction:**

1. To update seller info in user table

update user as u join exchange as ex ON u.id=ex.u\_id set u.money=u.money+(ex.no\_of\_stock)\*(ex.single\_stock),u.c\_id=0,u.no\_of\_stock=0,u.single\_stock=0 where u.id=ex.u\_id AND ex.status='sell';

B. To update buyer info in user table.

update user as u join exchange as ex ON u.id=ex.u\_id set u.money=u.money-(ex.no\_of\_stock)\*(ex.single\_stock),u.c\_id=ex.c\_id,u.no\_of\_stock=ex.no\_of\_stock,u.single\_stock=ex.single\_stock where u.id=ex.u\_id AND ex.status='buy'

C. Update order book table.

insert into orderbook(u\_id,c\_id,no\_of\_stock,single\_stock,order\_type) select u\_id,c\_id,no\_of\_stock,single\_stock,status from exchange;

D. Remove both seller and buyer of the above transaction from the exchange table.

delete from exchange where u\_id=s\_id;

delete from exchange where u\_id=b\_id;

Note: s\_id-->seller id

b\_id -->buyer id

here s\_id is seller id which we have already stored

**4.3 Before Transaction:**

**User Table-**

| **id** | **name** | **password** | **money** | **c\_id** | **no\_of\_stock** | **single\_stock** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Ravi** | **ravi** | **999999999.99** | **1** | **500** | **200.00** |
| **2** | **Prakhar** | **prakhar** | **999999999.99** | **0** | **0** | **0.00** |
| **3** | **Raj** | **raj** | **87343.00** | **0** | **0** | **0.00** |
| **4** | **Anshu** | **anshu** | **48332.0** | **0** | **0** | **0.00** |
| **5** | **Sakshi** | **sakshi** | **96933** | **0** | **0** | **0.00** |
| **6** | **Rachit** | **rachit** | **78433.00** | **0** | **0** | **0.00** |
| **7** | **Manik** | **manik** | **84730.00** | **0** | **0** | **0.00** |
| **8** | **Prathmesh** | **prathmesh** | **84732.00** | **0** | **0** | **0.00** |
| **9** | **Pradeep** | **pradeep** | **78359.00** | **0** | **0** | **0.00** |

**Exchange Table-**

| **ex\_id** | **u\_id** | **c\_id** | **no\_of\_stock** | **single\_stock** | **status** |
| --- | --- | --- | --- | --- | --- |
| **1** | **1** | **1** | **500** | **200.00** | **sell** |
| **2** | **2** | **1** | **500** | **200.00** | **buy** |

In the above situation user 1 wants to sell company 1 stock and user 2 wants to buy company 1 stock.

**Order Book-**

**The order book stores the previously performed transactions of the users.Initially this table is empty.**

**4.4 After Transaction:**

**User Table-**

| **id** | **name** | **password** | **money** | **c\_id** | **no\_of\_stock** | **single\_stock** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Ravi** | **ravi** | **1000099999.99** | **0** | **0** | **0.0** |
| **2** | **Prakhar** | **prakhar** | **999999999.99** | **0** | **0** | **0.00** |
| **3** | **Raj** | **raj** | **87343.00** | **0** | **0** | **0.00** |
| **4** | **Anshu** | **anshu** | **48332.0** | **0** | **0** | **0.00** |
| **5** | **Sakshi** | **sakshi** | **96933** | **0** | **0** | **0.00** |
| **6** | **Rachit** | **rachit** | **78433.00** | **0** | **0** | **0.00** |
| **7** | **Manik** | **manik** | **84730.00** | **0** | **0** | **0.00** |
| **8** | **Prathmesh** | **prathmesh** | **84732.00** | **0** | **0** | **0.00** |
| **9** | **Pradeep** | **pradeep** | **78359.00** | **0** | **0** | **0.00** |

**Order Book-**

| **ex\_id** | **u\_id** | **c\_id** | **no\_of\_stock** | **single\_stock** | **order\_type** |
| --- | --- | --- | --- | --- | --- |
| **1** | **1** | **1** | **500** | **200.00** | **sell** |
| **2** | **2** | **1** | **500** | **200.00** | **buy** |

**Exchange Table:**

**After the transaction exchange table is empty.Only one transaction is supported at time.**

**4. User Interface**

The application has an option for user registration.Users can create new profiles by entering the required information.It asks for the name,password and the initial amount.For an existing user there is a login option wherein the user is asked to enter name and password for authentication purpose.

**5. Technologies Used**

The code has been developed on the IDE IntelliJ IDEA with the help of Maven for Java,CLI and MySQL connectivity.

The database has been built on MySQL.

**6. Conclusion**

This was a mini replica of a trading system that the team developed using core java and MySQL.The project is basically about the connectivity of java code to a database source i.e.JDBC and the taking of an input for the application by connecting the code to Command Line Interface(CLI).The project was a very good exercise for a quick review of the java and database topics covered during the college days.

Apart from developing the technical skills the project played a great role in developing the team skills in all the members of our group.It taught us how people with different strengths and weaknesses can channelize their efforts for the attainment of a common goal and handle difference of opinion between different members of the group.It developed our interpersonal communication skills as well.All in all the project was a highly fruitful exercise for strengthening and polishing our hard as well as soft skills.