Oil Transaction management System

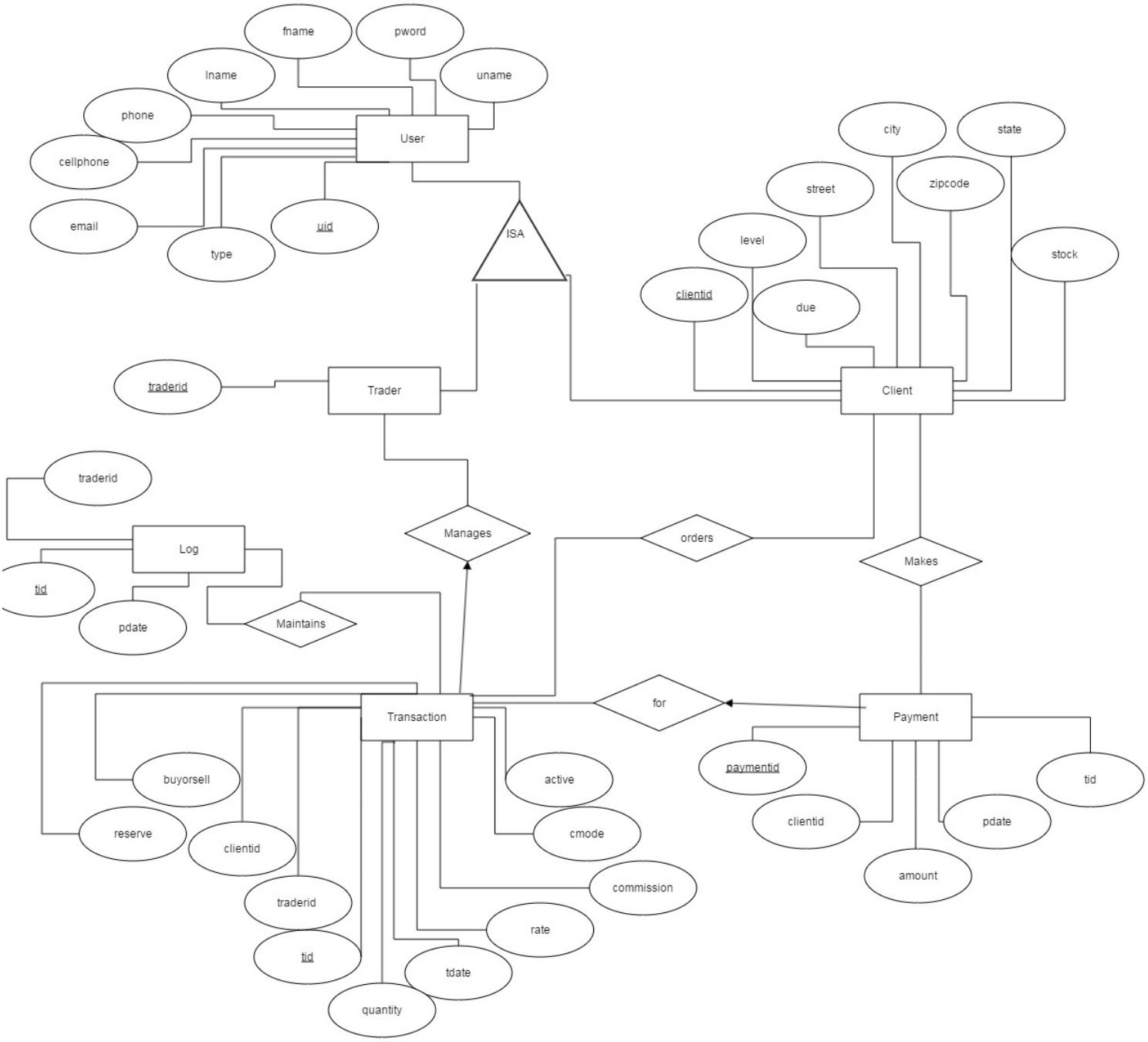
Apoorv Sahay (axs159031)

Nidhi Jain (nxj150930)

Nipun Agarwal (nxa150830)

Sreenivasan Shobana Venkatraman (sxs153630)

ER/EER Diagrams:



Entities:

The entities in the system are

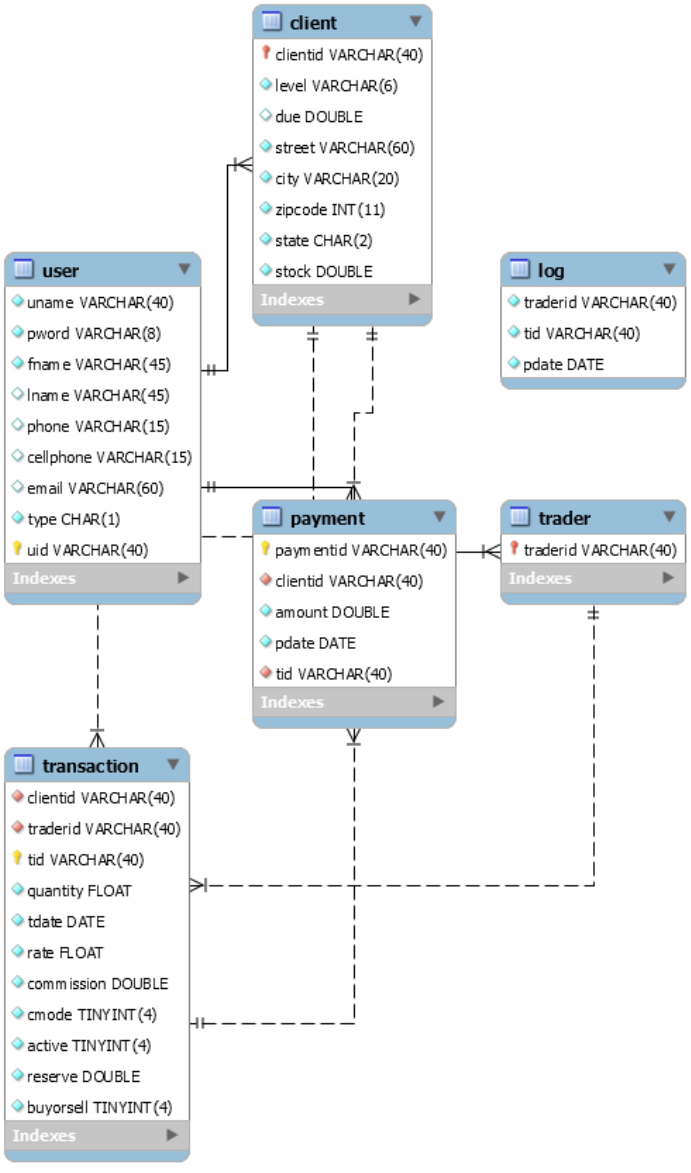
- Client
- Trader

- Payment
- Transaction
- Log

Relationships:

The following are the relationships exhibited among the entities given above:

- ISA relationship: Trader ISA User; Client ISA User
- Client orders a transactions
- Trader manages a Transactions
- Transaction maintains a Log
- Client makes a Payment



Major use cases / user flows:

There are three actors in the system and we explain below the major use cases for each of the actors:

1) Client:

A Client can log into the application and issue his own transactions, view and manage them. The transactions can be buy or sell transactions. He can also make the payments for his previous transactions. Every Client has a “Due amount which is treated like his e-wallet”. Payments are deducted and credited to this Due amount.

2) Trader:

A Trader can issue buy or sell transaction requests on behalf of clients. He can also view the outstanding payments by his clients and can cancel unauthorised payments. Canceling a transaction as a whole will cause the transaction to be nullified and the payment is credited back to the client's due amount.

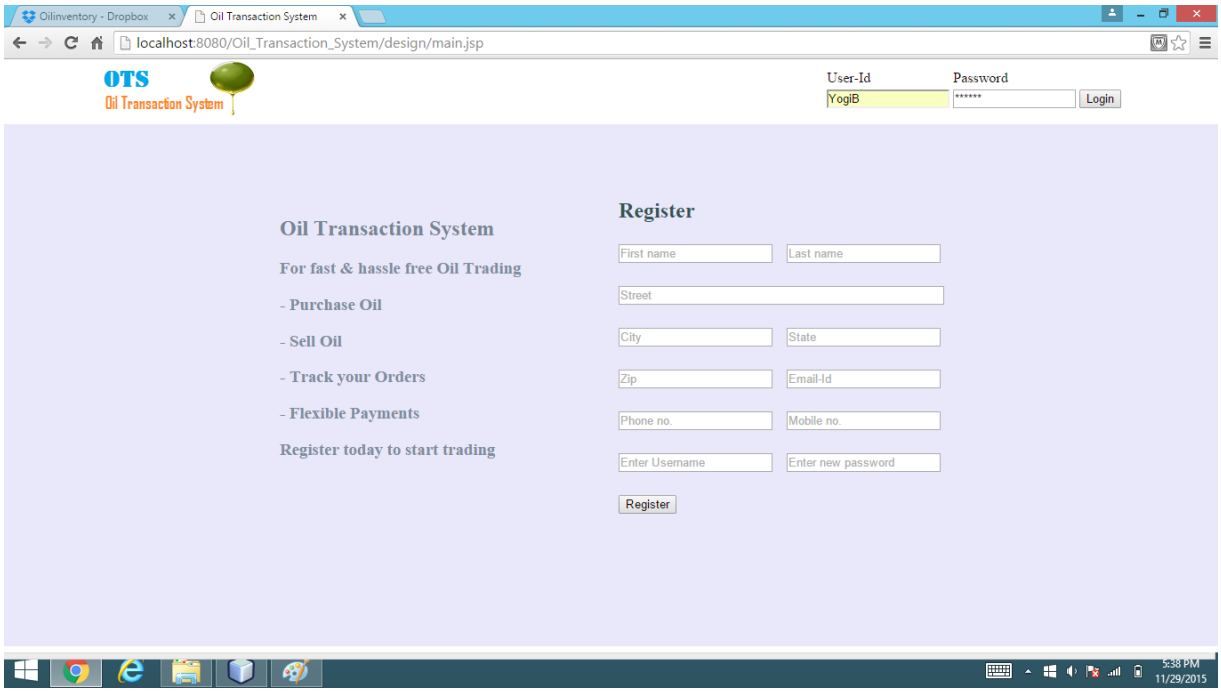
3) Manager:

A manager also plays on the role of the sysadmin. The manager can view the recent transactions in terms of daily, weekly and monthly granularity.

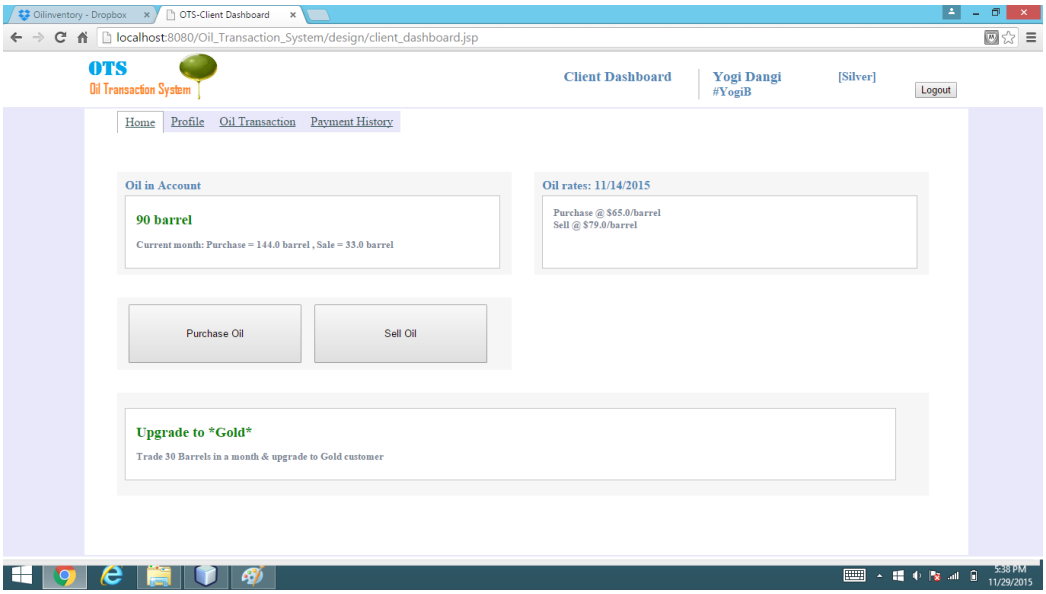
Client:

- Login/Logout

All users can login or logout with their valid account. The application uses cookies and server side sessions to keep track of authenticated users and to prevent unauthorised usage. It redirects all unauthenticated requests to the login page.



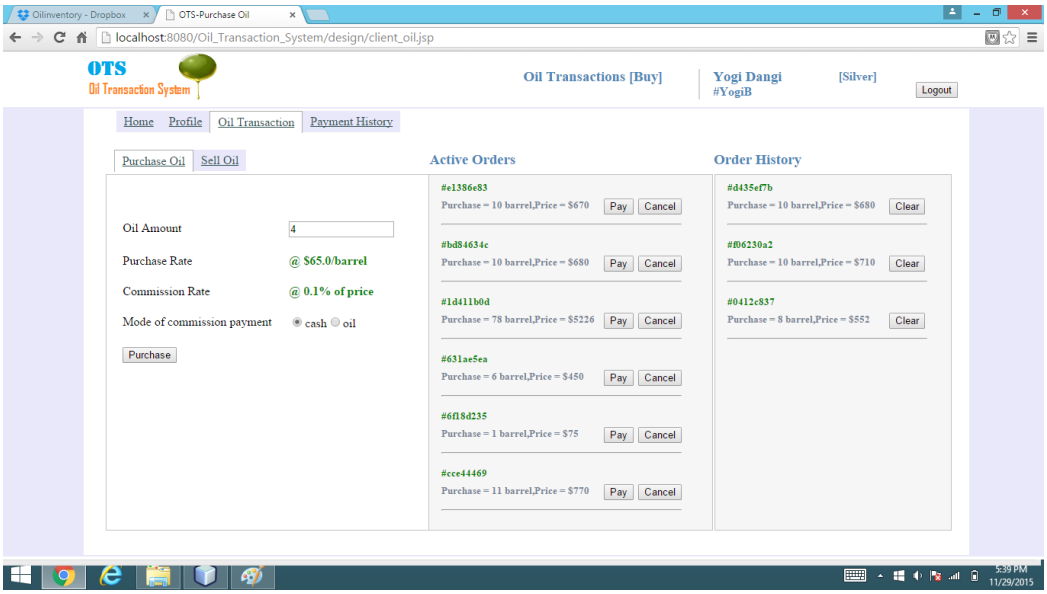
- Register



If the user does not have a previous login already, they can sign up as the required kind of user

- Buying Transaction

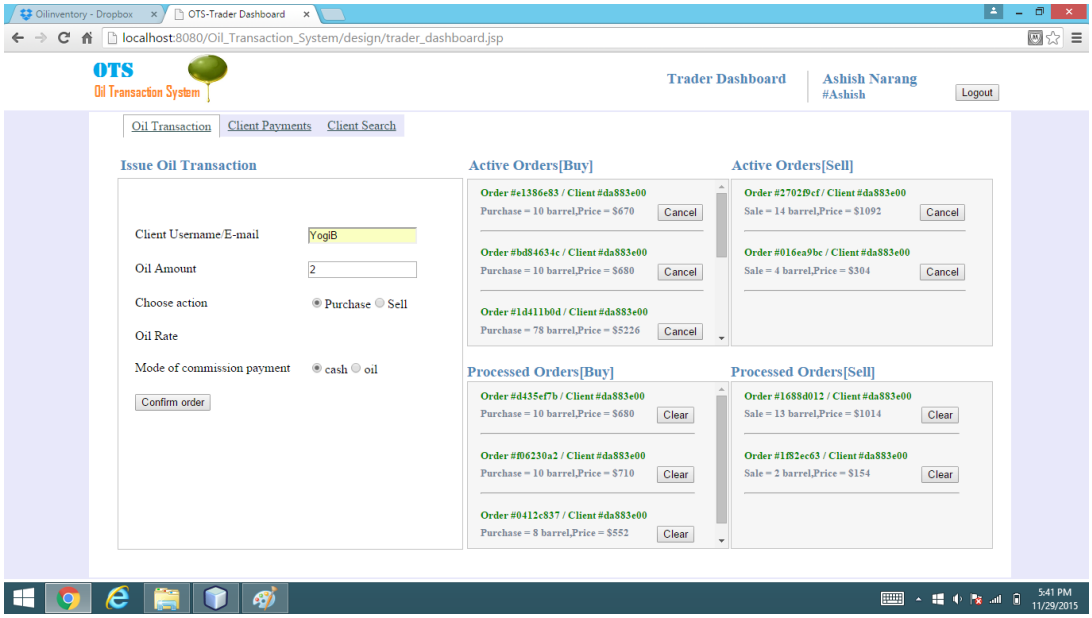
The client can issue a buying transaction of the requisite number of barrels. They can pay in terms of cash or oil. Cash payment is deducted from their Due amount and oil payment is deducted from their oil balance in barrels.



- Selling Transaction

The clients can sell any number of oil barrels that they already have. The proceeds of the sale minus the commission will be credited back to their due amount.

Trader:



- Login/LogoutRegister
- Buying Transaction

The trader can issue a Buy transaction to any of his clients. The workflow is similar to the that of the client buying his oil except for the selection of the client

- Selling Transaction

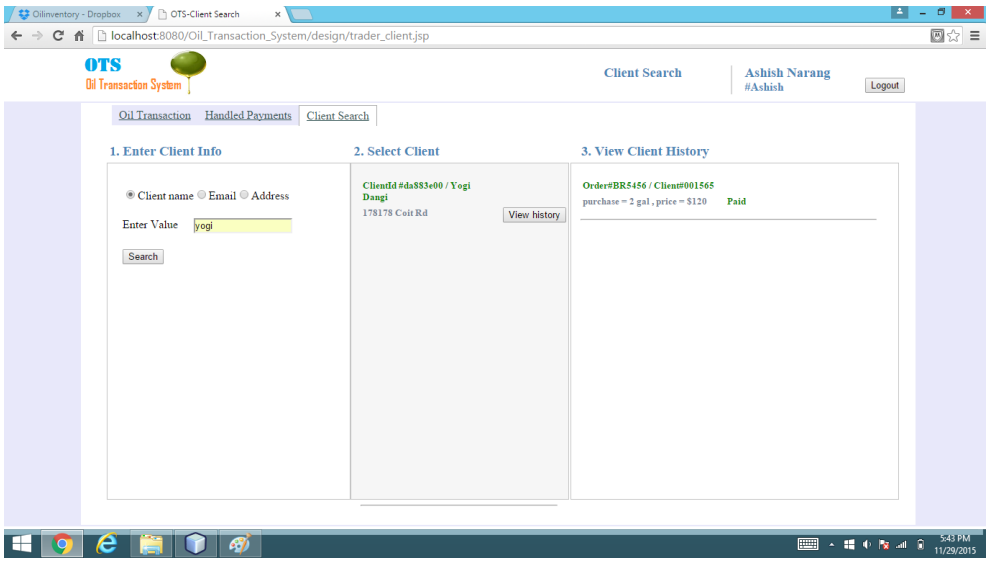
The trader can issue sell transactions on behalf of clients. The workflow is similar to the that of the client buying his oil except for the selection of the client.

- Canceling Transactions of a Client

The trader can choose to cancel a transaction of a client. In such a case any payment if made will be credited back to the due amount of the client

- Canceling Payment of a Client

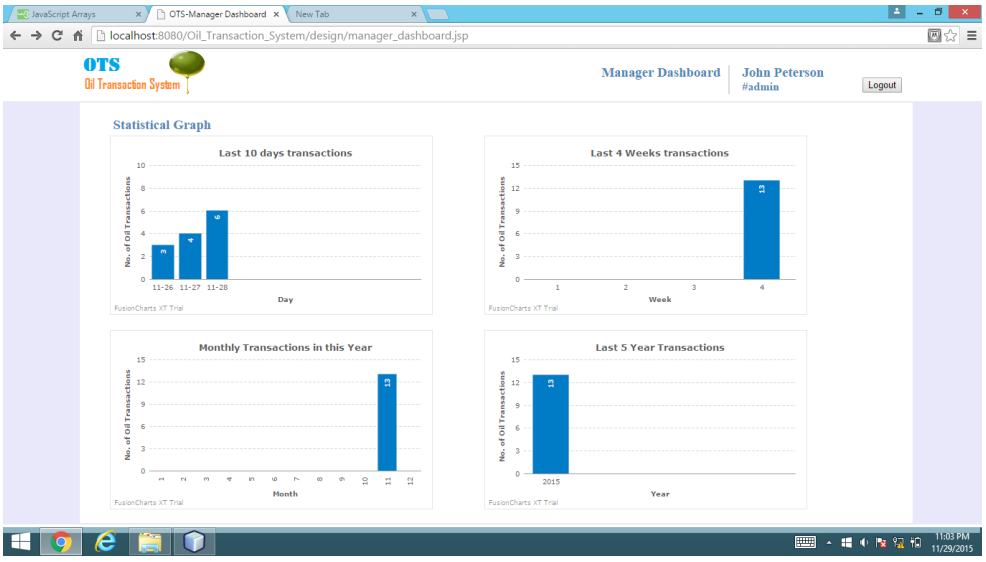
The trader can cancel individual payments made by a client. Just that payment is affected.



Manager:

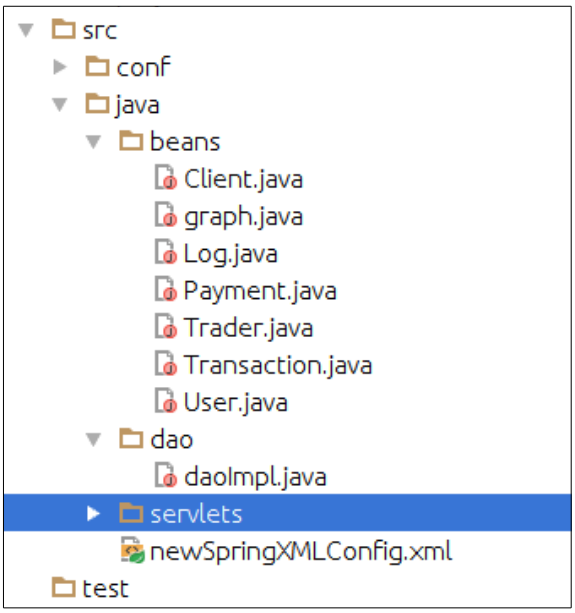
- Login/Logout
- Register
- View transactions of a time period

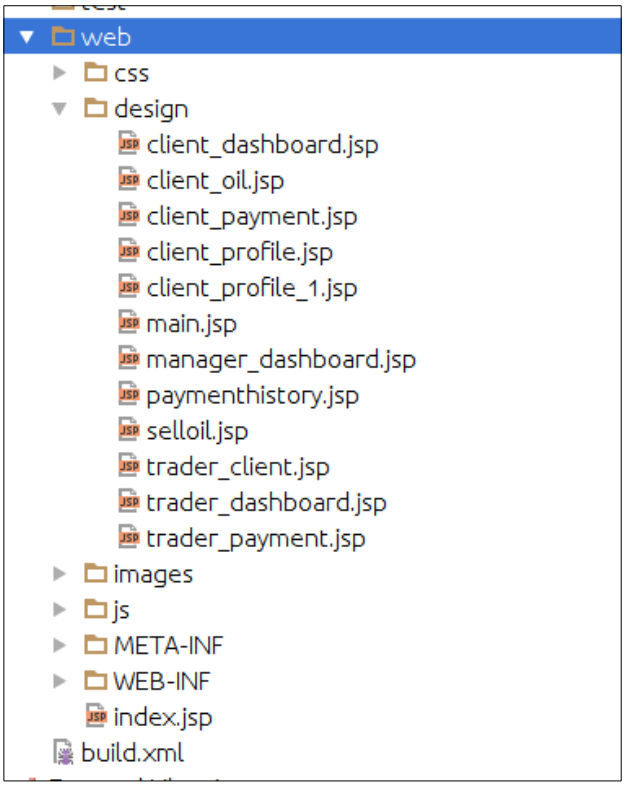
The manager also plays on the role of the admin and he can log in to view the recent transactions in terms of rich charts and graphs. This can be viewed daily, weekly or in monthly granularity.



Software Architecture / Code Overview:

- The whole application follows modular design and best practices.
- There are beans handling the operations in each table and there are Data Access Objects(DAO) that operates on these beans and provides higher level of abstraction to the servlets and JSPs.
- Spring framework is used to manage MVC in the back end.
- Jquery javascript library is used to handle DOM modifications in the front end.
- Fusion Charts library is used to render rich graphical charts in the manager's dashboard.





Security in the application:

All pages in the application are protected using Sessions. Only logged in users are allowed access and there are checks in each and every page. If the credentials dont match, we force them to the login page.

The application is protected against SQL injection attacks by using prepared statements. Hence valid Sql cannot be injected in the input parameters to cause a disastrous behavior in the database.

Concluding remarks:

The Oil Transaction Management System project furnished in this report thus integrates several new

technologies, implements all the provided functionalities and can serve as a scalable solution capable of being hosted online in a server to provide service to Traders, Clients and the manager.