FNB Connect Assessment Solution

==================================

REST API design and implementation with RESTeasy and Spring Framework

====================================================================

The application uses a multi-layered architecture, based on the “Law of Demeter (LoD) or principle of least knowledge”

The first layer is the REST support implemented with RESTeasy, has the role of a facade and delegates the logic to the business layer.

The business layer is where the logic happens.

The data access layer is where the communcation with the pesistence storage takes place. I used in-memory database "HyperSQL" running in oracle mode. Spring embbeded database was used to create HSQL in-memory database.

The system generate token on successful login. It then uses this token to authenticate subsequent calls to the application.

Validation also included to make sure user capture a number and that the amount captured is greater than or equal to the amount due.

REST Client side implemented with HTML and jQuery framework.

Running the project.

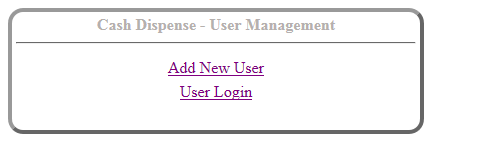
===================

The project was developed using Eclipse IDE. I included all the jars required in the lib folder (could have used ant/maven but had problems configuring build from my network). Apache Tomcat v7.0.69 was used as web server and this was configured to run inside inside Eclipse. Could have configured ant build to deploy war file into Tomcat given time but

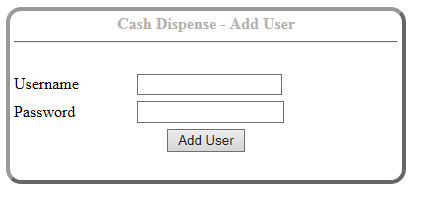
running project under Eclipse yield desired results.

After staring Tomcat under Eclipse the url for accessing home page: http://localhost:8080/CashDispenseSystem/

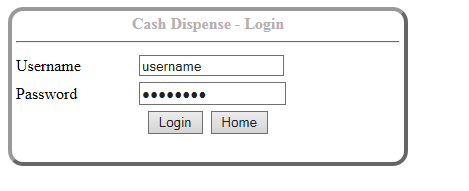
The screen below will be presented.



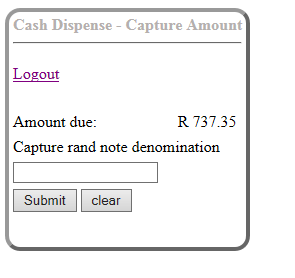
From the screen above you need to add user “Add New User” to the application first before you can login. The screen for add new user is as below.



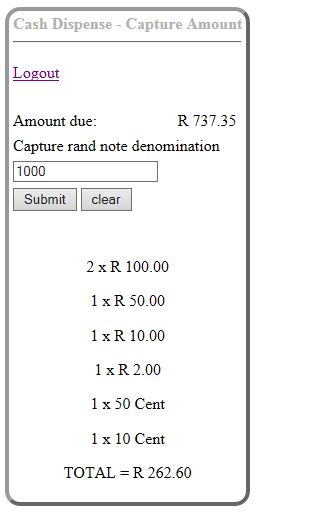
After successfully adding of new user then you can login. Login screen will appear as below.



After successfully login the system randomly generates amount due for you. You then need to capture rand note in the text field below.



e.g. Capture 1000 and submit. The system with then subtract amount due from captured note and gives back change in denomination as below.



Home button, logout and clear were added to add more user usability to the system

Given more time.

===============

> The exception mapper fro REST API could have been handled better.

> Token refresh could have done better

> css style sheet could have been applied.

> properly handing of session expiring.