Web Apps Report

Candidate Number: 184521

# Design

The design of the web application adheres to the three-tier architectural model being the web layer, application layer and database layer with each layer being implemented for this application. For example, the web layer is responsible for the presentation and communication of the application and is implemented with the .XHTML files for the index, login, admin, user, and error pages. Each of these files contains html code to display to the user the application as well as allow the user to interact with the application. The next layer, the application layer, handles all the logic of the application allowing it to function as intended and fulfil its purpose. In this application this is handled by the EJB’s and JSF’s of the program, with them being written in java, with two main classes being considered being deposits and users. The final tier is the data layer which is the layer that connects to the web application’s database to control all of the data that must be stored for the application to run. This is done via the entity classes of which there are three, System Users, System Deposits, and System Groups. On the creation of one of these classes the entity is added to the database via the JDBC connection with the Payara server.

The model view controller (MVC) software pattern is similar in nature to the three-tier architectural model where some tiers share an equivalent in the MVC. For example, the Model part of the MVC encapsulates core data as well as its functionality. Therefore, the model is the combination of the data and the application layer of the three-tier model. Therefore, in this application the model is made up of the entities, JSFs, the EJBs and the databases. The view of the MVC represents the .XHTML files of the web application or the web layer of the three-tier model. Finally, the Controller of the MVC is represented by the Payara server itself as this is responsible for processing the users inputs and making requests to the model to produce new views. This is the same definition as the controller of the MVC

# Concurrent Users

System would use a single-process, single-threaded server therefore an infinite loop would be executed where checks are constantly carried out to see if data is returned from inputs and outputs. Therefore, if data is being returned the other users shall wait in a queue but if no data is being returned then the next connection in the queue is given access to the database. Due to connections being refused once no transfer of data between the database and the connection takes place the queue is moved through quickly with each connection lasting milli seconds. This therefore allows multiple concurrent users to access the data of the database. As for functionality, every time a new connection is made a new instance of the application shall be made on a different IP allowing multiple users to use the application at once.

# Security

The main method of securing the program is the utilisation of JDBC realms which require credentials to be entered which are then checked against the user database to sign a user in. If the users details are correct the user is signed if not the user is sent to the sign in error page and asked to retry. When a user tries to access a page the realm looks at the users associated user group to see if the group has access to the page or not. For this application users and admins can access the deposit page but only admins can access the admin page, all other pages don’t need you to be signed in to access them such as the index, registration, and error pages.

This security method is better than not having one at all as it means only authenticated users can access the right pages and only Admins can view all the data available and add more admins. On top of this this method is better than a basic JDBC realm as encryption is used on the user and admins passwords so the entered password must be encrypted in the same fashion before they can be checked to be the same. The encryption method used here is hash SHA-256 and means that if someone is to access the database outside of the application, they are unable to see what the users password actually is as it shows many, seemingly random, characters.

However, this method is not perfect as this encryption method is the default one provided by the realm if when passwords are chosen to be encrypted and therefore is not the most secure. This is especially true as the database does not require username or password to connect to it and therefore could easily be accessed. This does mean other confidential information such as users names and balances are open to anyone who can connect to the database. Therefore, a better method would be to encrypt all of the details in the database instead of only the user’s passwords. Annotations have also been added on restricted functions such as the ability to register admins so only admins can call them.

# Extension

A new instance of the web application could be hosted on different servers whenever a new user connects with a queue for accessing the database when requests are made. Therefore, if one server were to fail only that user would be inconvenienced and also this users browser would simply be able to try other IP addresses associated with the application to reconnect the user to a different instance. Therefore, the exact place the user was at in terms of accessing or completing transactions wouldn’t be the same, but the user would be able to begin again from the index without much interruption. An example of how this could be done would be with High Availability Deployment Architecture (HADA) which would also allow for maintenance to be performed on one server without having to undeploy the whole application. The way HADA operates is by instead of having a single server, multiple are clustered together that monitor each other for faults. This architecture still has a single primary server but with multiple backups that can also host the application at the same time. If one of the servers does have a fault the backup servers can step in to cover it whilst it is brought back online. Therefore, the service would never be lost as the users would onnect to either the primary server or the backup server depending on which servers are operating and the capacity the servers are currently running at.