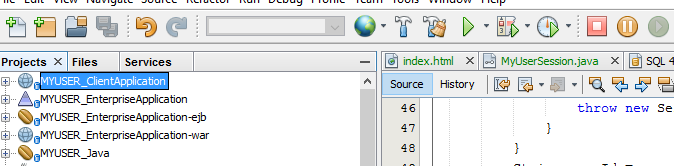
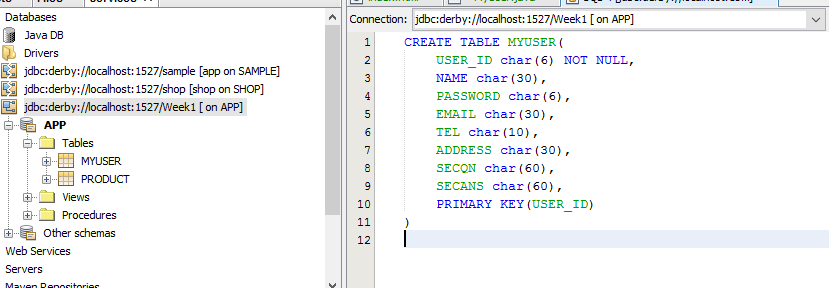
# MYUSER Architecture

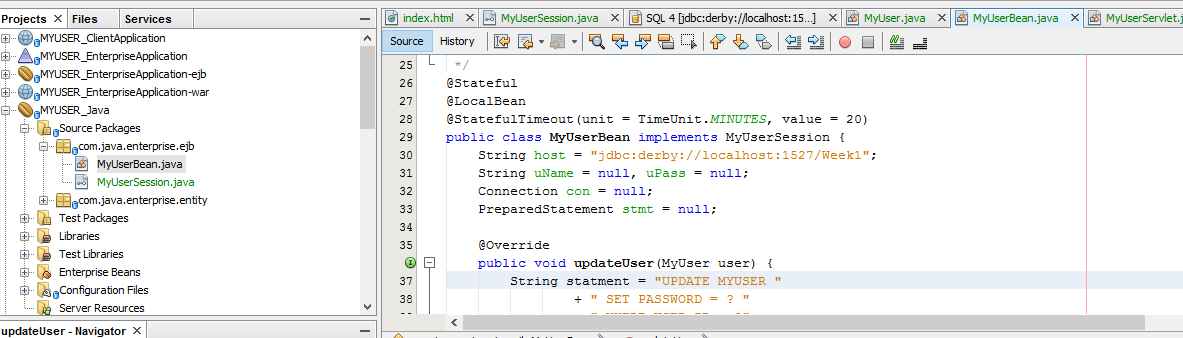
On this task, I have created an application that consists of 3 layers (presentation layer, business layer and data layer). Each tier is responsible for different tasks. I can create a Application with more layer but this is a simple project, so a basic instance of multi layer is enough.



First the data layer. Apache Derby is developed as an open source project under the Apache 2.0 license. Oracle distributes the same binaries under the name Java DB. It is a relational database, it allows me to quickly set up a database for my application. It allows me to integrate and easy to use in my application. Have some alternatives: MySQL, PosgreSQL … but I need more time to Setup, some is not free. So I have chosen Deryby.



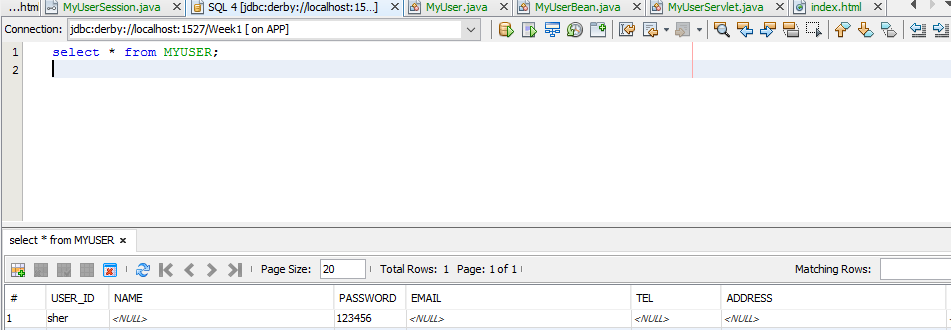
Next to the business layer, I have created two projects. MYUSER\_Java and MYUSER\_EnterpriseApplication-ejb and serves to communicate with the data layer and the presentation layer. MYUSER\_EnterpriseApplication-ejb interacts directly with the data tier. It is responsible for handling the workers business systems such as create, add, edit, delete data on the basis of the application. Every system operation - if you want to communicate with the database - must be conducted through it.

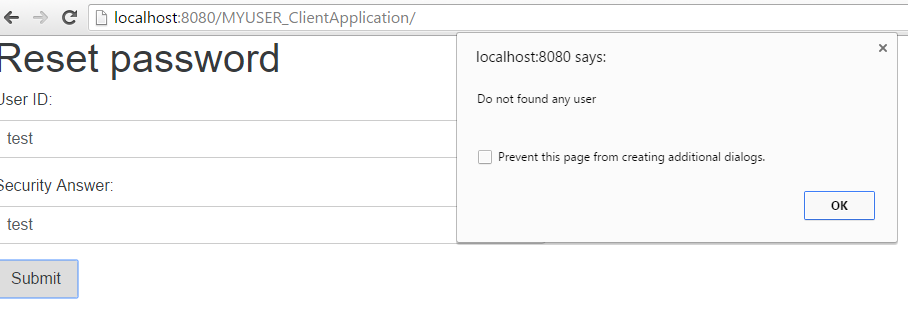


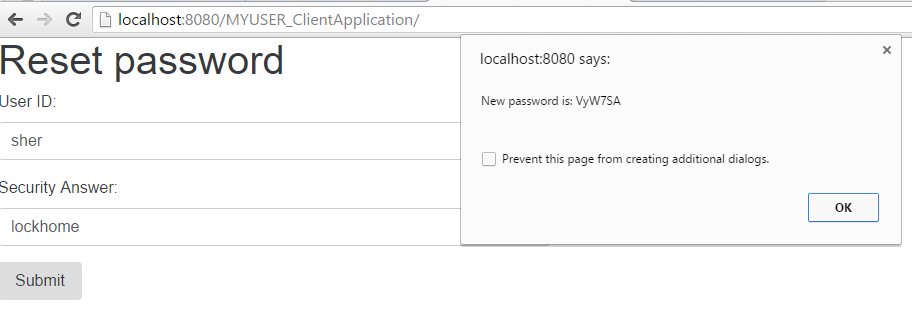
MYUSER\_Java provides interfaces allow communication with MYUSER\_EnterpriseApplication-ejb. It was created with the purpose of providing a unified interface to access the system operations. The idea behind this is to that layer security architecture can support multiple presentation layers e.g. web, mobile, etc. This provides management easier, better abstraction and scalability cupporting large number of simultaneous clients.

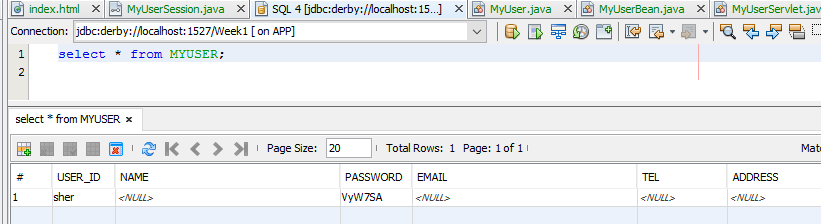
Finally the presentation layer, I created two different projects in order to give users the access methods of system operations. It is not allowed to access MYUSER\_EnterpriseApplication-ejb which only allowed access through MYUSER\_Java. MYUSER\_EnterpriseApplication-war is a web application, which provides a web interface for users. Users can manipulate the operations of the system through a browser. It creates an intuitive view for users. Project runs on the JBoss server.

MYUSER\_ClientApplication is a java application that allows users to manipulate the system operation via the command line. The manipulation is used by MYUSER\_Java offer include tasks such as adding, editing, deleting data. I can create more application with service from Business layer (Mobile, web…), this is a simple project so I chose easy way to present my service.







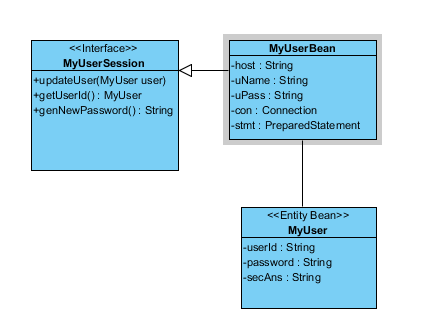


Test case

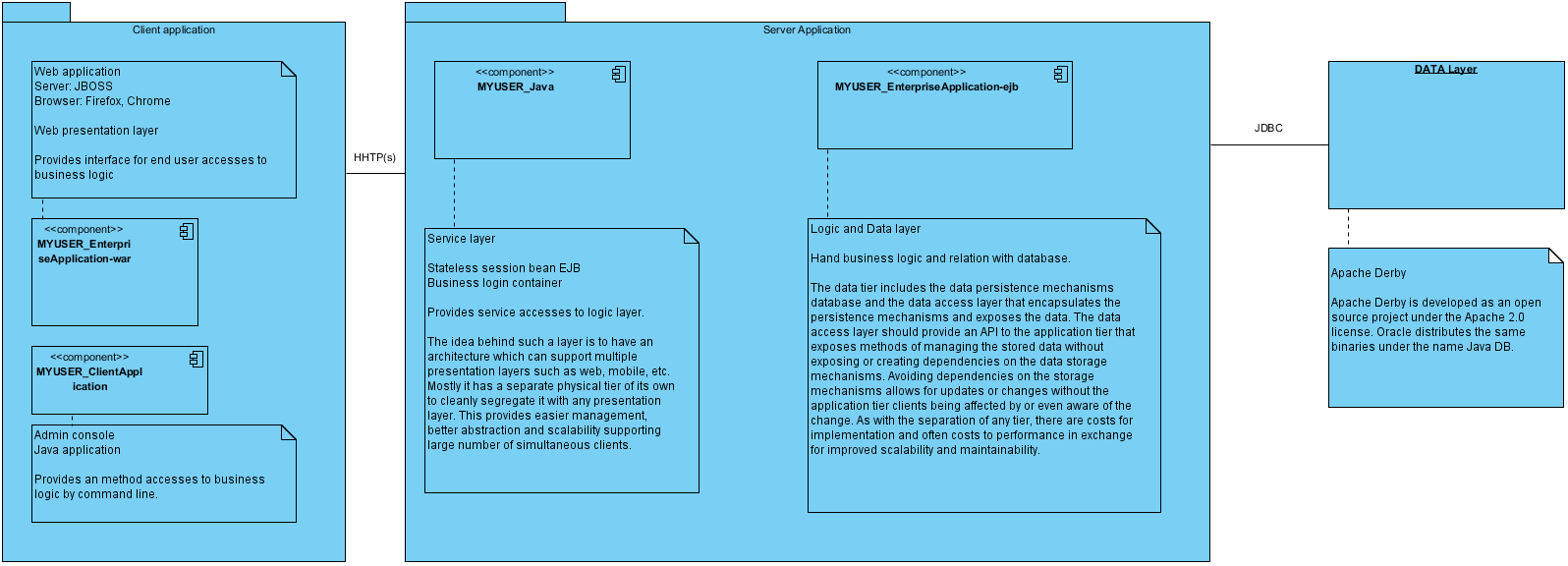
Reset password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Test Steps | Expected Result | Actual Result | P/F |
| 1 | Insert valid User Id and Security answer | Reset password success | Reset password success | P |
| 2 | Insert invalid User Id and Security answer | Reset password fail | Reset password fail | P |
| 3 | Insert valid User Id and invalid Security answer | Reset password fail | Reset password fail | P |
| 4 | Insert invalid User Id and valid Security answer | Reset password fail | Reset password fail | P |

Class diagram



Component Diagram



## Summary

|  |  |
| --- | --- |
| Name | Documentation |
| Image1.png [Client application](#KhllS8qAUPUSXQyB) |  |
| Image1.png [Server Application](#D08VS8qAUPUSXQye) |  |
| Image2.png [MYUSER\_Java](#flGVS8qAUPUSXQy4) | Service layer    Stateless session bean EJB  Business login container    Provides service accesses to logic layer.    The idea behind such a layer is to have an architecture which can support multiple presentation layers such as web, mobile, etc. Mostly it has a separate physical tier of its own to cleanly segregate it with any presentation layer. This provides easier management, better abstraction and scalability supporting large number of simultaneous clients. |
| Image2.png [MYUSER\_EnterpriseApplication-ejb](#d0F1S8qAUPUSXQzf) | Logic and Data layer    Hand business logic and relation with database.    The data tier includes the data persistence mechanisms database and the data access layer that encapsulates the persistence mechanisms and exposes the data. The data access layer should provide an API to the application tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms. Avoiding dependencies on the storage mechanisms allows for updates or changes without the application tier clients being affected by or even aware of the change. As with the separation of any tier, there are costs for implementation and often costs to performance in exchange for improved scalability and maintainability. |
| Image3.png [DATA Layer](#sbQtS8qAUPUSXQ0V) | Apache Derby    Apache Derby is developed as an open source project under the Apache 2.0 license. Oracle distributes the same binaries under the name Java DB. |