Shlomit Argov (ID: 066656265)

Shlomit.argov@gmail.com

MBank Project
John Bryce Java Server 2014

Included files:

1. MBank project
2. MBankTest project
3. MBankAdminConsole project
4. CreateDB project
5. CreateMBankDB project
6. MBankWeb project
7. MBank.app.ear project
8. MBankDBDump.rar (database dump)
9. db-derby-10.10.2.0-bin (Apache Derby Database server application)

Before being able to run the projects you must perform the following steps:

1. From the db-derby-10.10.2.0-bin folder, run the batch file '\db-derby-10.10.2.0-bin\bin\StartNetworkServer.bat'.
   This will start the DB server and assign it the appropriate port.
2. At this point you can choose one of two options:
   a. From the CreateMBankDB project, run the test 'suites.
      CreateCleanDBAndPopulateBasicTableData.java' as a JUnit.
      This will create a new database on the Derby server, called Mbank, in addition to
      creating all Mbank tables, populating the properties table and inserting an administrator
      user into the clients table (this test deletes any previous versions of Mbank database).
   b. Use the provided dump file and load it into the Mbank DB.

How to run the projects:

Mbank core (Mbank project)

Run the project as a java application.

The project contains a Main class with a main method that provides a command line interface (CLI).

The CLI provides access to AdminAction and ClientAction methods so that they can be tested.

Swing UI (MBankAdminConsole project)

Run the project as a java application.

The project contains a MainFrame class with a main that launches the stand-alone UI application.

Use the following credentials in order to login:

Username: system

Password: admin

Web UI (MBankWeb project)

Use Apache JBoss WildFly 8.x server as the web application server (other web application servers were not tested).

Run the project on the server (right click->Run As-> Run on Server).

Use Firefox (other browsers were not tested) and browse to:

http://localhost:8080/MBankWeb/

Use the following credentials in order to login:

Username: testClient1

Password: pwd

Web Services (WSDL + REST APIs)

Use Apache JBoss WildFly 8.x server as the web application server (other web application servers were not tested).

Run the project on the server (right click->Run As-> Run on Server).

Use Firefox (other browsers were not tested) and browse to:

http://localhost:8080/MBankWeb/

Pre-defined users:

- Username: testClient1
  Password: pwd

- Username: testClient2
  Password: pwd

After performing several withdrawal and deposit actions using each of the above users, use the following URLs in order to query the local DB on JBOSS for the corresponding action logs:

WSDL URL (XML format): http://localhost:8080/MBank.ejb/LogWsdlWsBean

- Use `http://localhost:8080/MBank.ejb/LogWsdlWsBean?WSDL` in order to retrieve the possible methods (`getAllLogs()` and `getLogsByClientId()`) and XML request formats.

REST API URL (JSON format): http://localhost:8080/MBankWeb/rest

Available methods are:
- http://localhost:8080/MBankWeb/rest/getAllLogs
- http://localhost:8080/MBankWeb/rest/getLogsByClientId/{clientId}

Detailed specification of projects:

Phase 1: Building System Core

MBank server

a. MBank project – phase 1 implementation, using Java SE.
b. MBankTest project – unit test code for MBank classes, using JUnit 4 framework (you can run all tests by running mbank.AllTests as a JUnit).

MBank database – helper project

a. CreateDB – project for general creation/deletion/population of database tables.
b. CreateMBankDB – used to create/delete/populate specific tables needed for the MBank project.

Phase 2: Building Administration Desktop Application

Mbank Admin Management Application

MBankAdminConsole project – phase 2 implementation, using Java Swing framework.

Phase 3: Building Client Web Application

Mbank Client Dashboard Web Application

MBankWeb project – phase 3 implementation, using JSP and JSTL and implemented using MVC model

Phase 4: Creating Mbank Asynchronous Log System

Mbank WSDL and REST APIs

MBankWeb + Mbank.ejb projects (wrapped in MBank.app.ear) – phase 4 implementation, using EJB (Session, Entity, Message-Driven) and providing WSDL and REST web-service APIs.