

Final Project Readme

Project Summary

Financial Data Directory – a banking solution application. This basis of this application is to manage customers and their accounts in a bank. It will allow the bank personnel to create new customers and accounts, set preferences for the customer and maintain credit/debit card details of the customer. Every customer will have a unique customer ID which will be used to form relationships with the other entities in the application.

Design

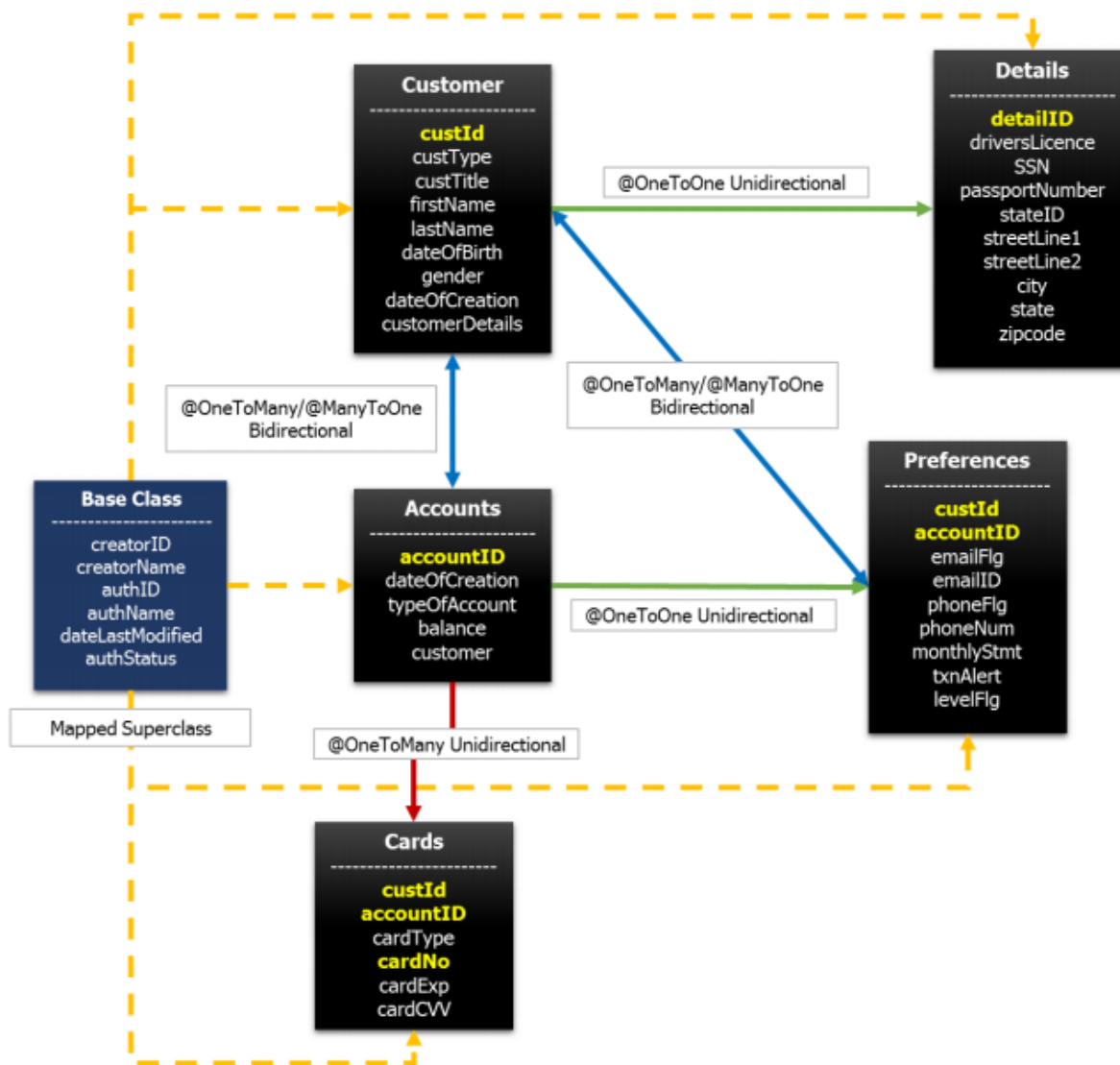
There are two users in this application.

1. **Makers** - These are the users that are responsible for adding customers, their details, preferences, and create accounts/cards. The transactions done by a teller will be sent for authorization. A teller can perform add, update and delete operations.
2. **Checkers** – They responsible for overseeing transactions made by the teller. A transaction will be in the unauthorized state until a checker, authorizes the transaction.

The application revolves around the following entities.

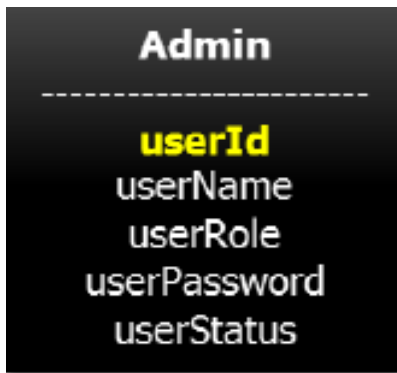
1. **Customer** - Main entity; includes customer type, unique customer ID, name, DOB and date of creation.
2. **Details** - Additional details of the customer like SSN, Passport number, address, etc. There is a one to one relationship of a customer and his details. The two entities are linked by a unique customer ID.
3. **Accounts** - A customer can have multiple accounts for example a savings account, checking account, deposit account, etc. A unique account ID is maintained for every account that is created. A one to many relationship exists between the customer and the account. The Customer entity and Account entities are linked using customer ID.
4. **Preferences** – This entity will include the customer and account level preferences. Preferences include monthly statement delivery formats, transaction alert methods, customer/account change alerts, etc. This entity support the many to many relationship with the customer and account entities.
5. **Cards** – Each customer can have a credit and debit card attached to their account. A unique card number will be provided for each card the customer has.
6. **Admin** – This entity consists of users that add, modify and delete customers, accounts, etc. It contains two types of users – makers and checkers. The users with the maker role can initiate the add/modify/delete transactions. These transactions can then be authorized by the checker.
7. **BaseEntity** – This a normal POJO class that includes all the necessary transaction details like date created, maker id, checker id, transaction status etc. This entity is inherited by all other entities.

Below is the diagrammatic explanation of the entities and their relationships with each other.



The attributes marked in yellow are the primary/foreign keys. Base class is the superclass that is inherited by all entities.

The Admin Entity holds the details of the users that can perform CRUD operations in the database. It holds information like the userID, name, password etc.



All the above entities have been implemented as a part of my final project. CRUD operations are available for each of the entities through the application.

Development Insights

Through this project I was able to understand several concepts of developing a web based application. Realized the ease of using components and templates, which can be useful while building applications that need to use the same template for different operations. Tried out different bean states to see which suits my application best and was able to see what difference a single word can make in the application. Using JSF was a delight; the options provided made it easy to create and manipulate the front end. JSF also has a great way of connecting to the beans, which in turn connect to the services that help persist data. Layering of the application is achieved.

Note: Code was based on "Instructor Examples"

Requirements (Installation, Compile, Runtime, etc)

The project is a Maven Web Application with the dependencies included in pom.xml. Below are the security constraints configured in Glassfish.

In the Glassfish Server admin console, under "**Configurations --> server-config --> Security --> Realms.**

Create a realm with the following name - "**itmd4515Realm**" and select "**com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm**" class.

The realm details are as below.

Properties specific to this Class

JAAS Context: *	<input type="text" value="jdbcRealm"/> Identifier for the login module to use for this realm
JNDI: *	<input type="text" value="jdbc/itmd4515DS"/> JNDI name of the JDBC resource used by this realm
User Table: *	<input type="text" value="user"/> Name of the database table that contains the list of authorized users for this realm
User Name Column: *	<input type="text" value="userName"/> Name of the column in the user table that contains the list of user names
Password Column: *	<input type="text" value="password"/> Name of the column in the user table that contains the user passwords
Group Table: *	<input type="text" value="user_role"/> Name of the database table that contains the list of groups for this realm
Group Table User Name Column:	<input type="text" value="userName"/> Name of the column in the user group table that contains the list of groups for this realm
Group Name Column: *	<input type="text" value="roleName"/> Name of the column in the group table that contains the list of group names
Password Encryption Algorithm: *	<input type="text" value="AES"/> This denotes the algorithm for encrypting the passwords in the database. It is a security risk to leave this field empty.
Assign Groups:	<input type="text"/> Comma-separated list of group names
Database User:	<input type="text"/> Specify the database user name in the realm instead of the JDBC connection pool
Database Password:	<input type="text"/> Specify the database password in the realm instead of the JDBC connection pool
Digest Algorithm:	<input type="text" value="SHA-256"/> Digest algorithm (default is SHA-256); note that the default was MD5 in GlassFish versions prior to 3.1
Encoding:	<input type="text"/> Encoding (allowed values are Hex and Base64)
Charset:	<input type="text" value="UTF-8"/> Character set for the digest algorithm

The following are the username and password combinations that can be used.

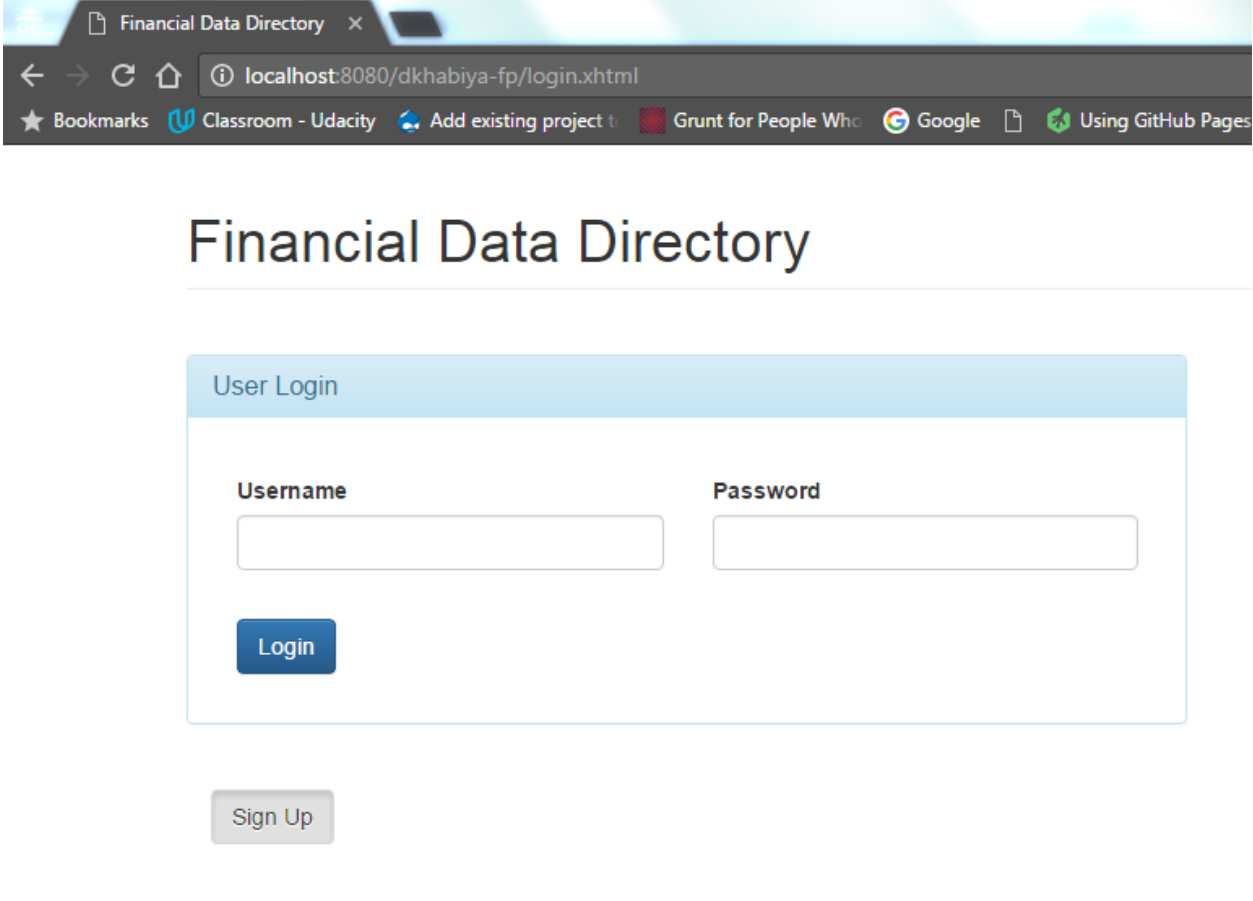
Username	Password
maker1	maker1
maker2	maker2
maker3	maker3
maker4	maker4
checker1	checker1
checker2	checker2
checker3	checker3
checker4	checker4

Sufficient data has been provided to test the application.

Screen Captures

Application URL - <http://localhost:8080/dkhabiya-fp/customer>

JavaDocs URL - <http://localhost:8080/dkhabiya-fp/apidocs/index.html>

Landing Page	
	
Maker Creation	

Sign Up

Home

User Sign Up

Role

Maker

Full Name

Super

User Name

Mario

Password

.....

Save

	NAME	roleName	USERNAME	PASSWORD
▶	Koji Kondo	CHECKER	koji	a665a45920422f9d417e4867efdc4fb8a04a1f3f...
	Super	MAKER	Mario	a665a45920422f9d417e4867efdc4fb8a04a1f3f...

Maker Landing Page

Financial Data Directory

Welcome Super MAKER

Customer Data

Customer ID	Name	Date Of Birth

*Click on customer ID to view individual customer details.

Add New Customer

Logout

Enter Information - Maker

Create New Customer

localhost:8080/dkhabiya-fp/maker/welcome.xhtml

BookmarksClassroom - UdacityAdd existing project tGrunt for People WhoGoogleUsing GitHub Pages T

Home

Basic Information

Customer Type *

Individual

Customer Title *

Mr.

First Name *

Test

Last Name

Test

Date Of Birth *

09/09/1998

Details

Driver's License

DL

State ID

SI

SSN

SSN

Passport Number

PN

Address Line 1 *

Addr 1

Address Line 2

Addr 2

City *

Test

State *

Test

Zipcode *

60616

Save

Financial Data Directory

localhost:8080

BookmarksClassroom - Udacity

Financial Data Directory

Welcome

Customer Data

Customer ID

1240

*Click on customer ID to view details

Add New Customer

Logout

Customer View - Maker

Customer Details

Home

Basic Information

Customer ID	Name	Date Of Birth
1240	Mr. Test Test	09/09/88

Customer Details

Driver's License	SSN	Passport Number	Street Address 1	Street Address 2	State	Zipcode
DL	SSN	PN	Addr 1	Addr 2	Test	60616

Update

Delete

Account Information

Account Number	Account Type

*Click on Account ID to view individual account details.

Create Account

Logout

Update Customer

←

→

↺

🏠

🔍

★ Bookmarks

🔗 Classroom - Udacity

➕ Add existing project

🔴 Grunt for People Who

🔍 Google

📄 Using GitHub Pages

🔍 How Does One Add

🔗 Adding an existing pr

Customer Updated - Maker

Customer Details

localhost:8080/dkhabiya-fp/maker/customer/customerUpdate.xhtml

Bookmarks Classroom - Udacity Add existing project Grunt for People Who Google Using GitHub Pages How Does One Add Adding an existing pr

Customer Details

Home

Shipping Tool

Drag the cursor around the area you want to capture.

Basic Information

Customer ID	Name	Date Of Birth
1240	Mr. Test Test	09/09/88

Customer Details

Driver's License	SSN	Passport Number	Street Address 1	Street Address 2	State	Zipcode
DL234	SSN234	PN234	Addr 1	Addr 2	Test	60616

Update

Delete

Account Information

Account Number	Account Type

*Click on Account ID to view individual account details.

Create Account

Logout

Enter Information - Maker

Add Account

← → ↺ 🏠 ⓘ

★ Bookmarks 🌐 Clas

Add Account

localhost:8080/dkhabiya-fp/maker/customer/customerView.xhtml

BookmarksClassroom - UdacityAdd existing project tGrunt for People WhcGoogleUsing GitHub Pages THow Do

Add Account

Home

Account Details

Account Type *
Checkings

Balance *
90.90

Preferences

Email Alert
RequiredNot Required

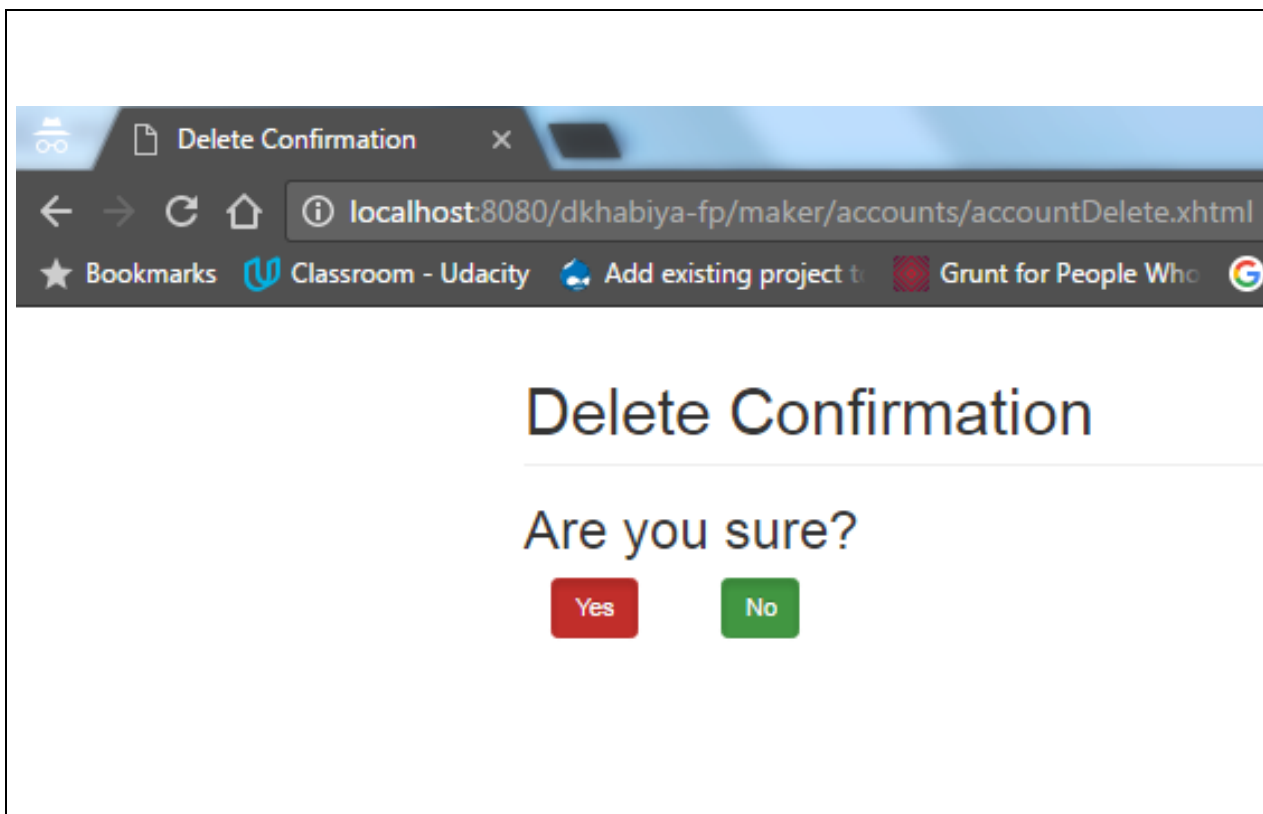
Email ID
abc@abc.c om

Phone Alert
RequiredNot Required

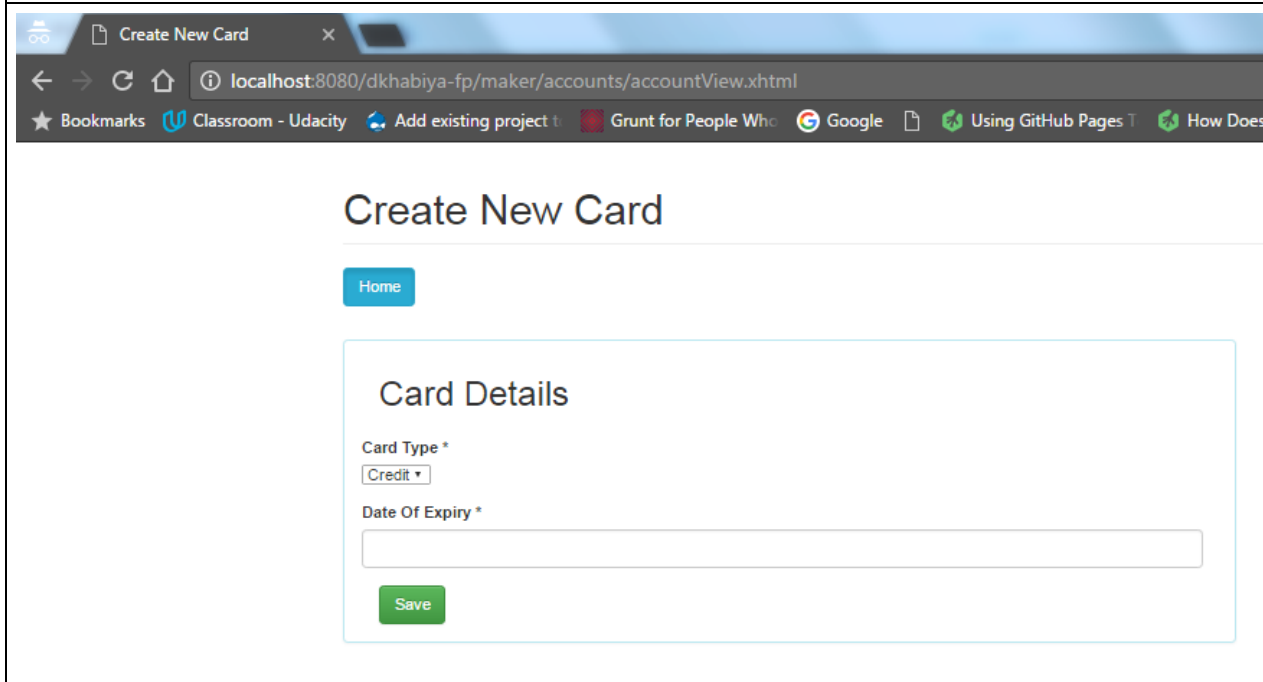
Phone Number
2345678900

Save

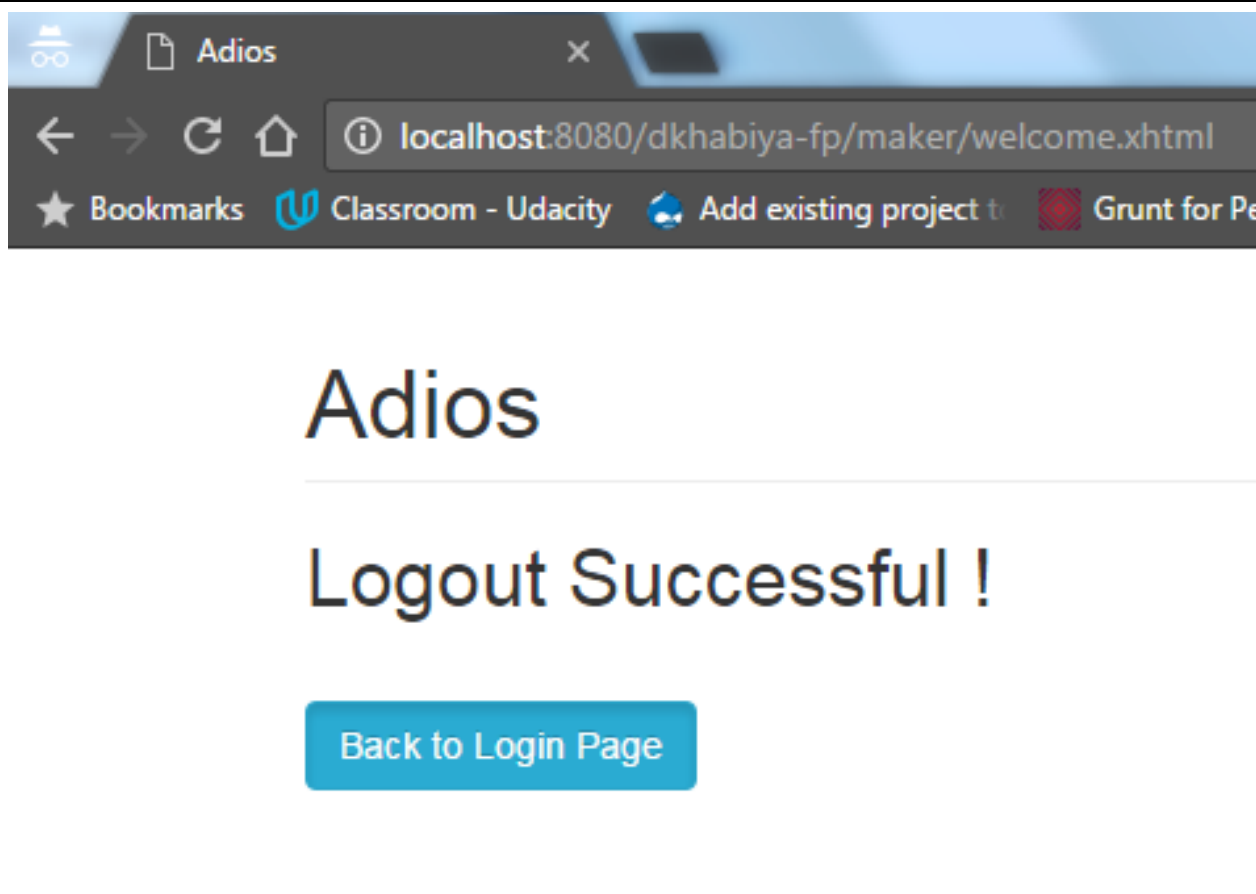
Account Delete - Maker



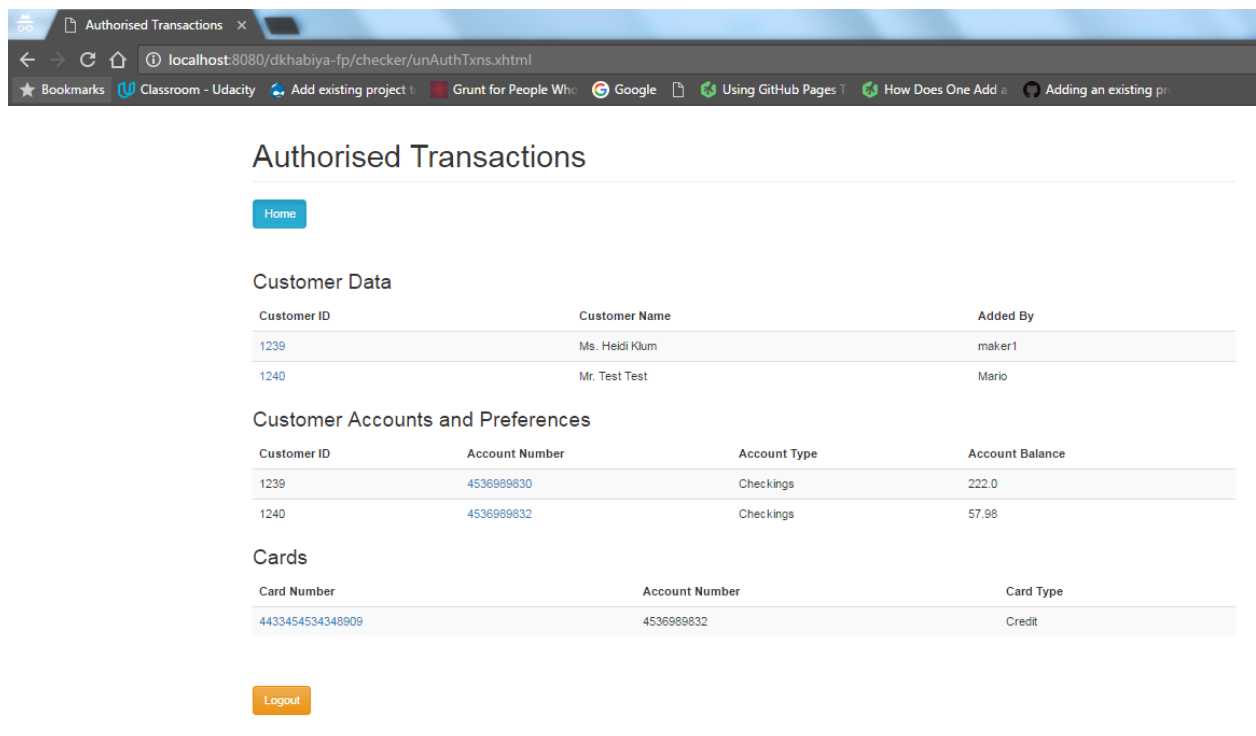
Create New Card - Maker



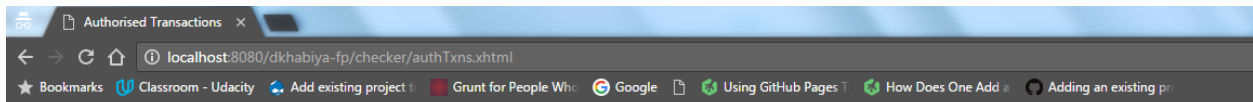
Logout - Maker



Unauthorized Transactions - Checker



Authorized Transaction



Authorised Transactions

Home

Customer Data

Customer ID	Customer Name	Date Of Birth
1240	Mr. Test Test	09-Sep-1988

Customer Details

Customer ID	Customer Name	Street Address 1	Street Address 2	State	Zipcode
		Addr 1	Addr 2	Test	60616

Customer Accounts

Customer ID	Customer Name	Account Number	Account Type	Account Balance

Account Preferences

Preference ID	Account Number	Name	Email Alert	Phone Alert

Cards

Card Number	Name On Card	Account Number	Card Type

Financial Data

← → ↺ ↻ ⓘ

★ Bookmarks ↗ Class

All Classes

Packages

model
model.security
service
web

All Classes

AccountController
Accounts
AccountsService
BaseController
BaseEntity
BaseService
Cards
CardsController
CardsService
CheckerController
Customer
CustomerController
CustomerService
Details
DetailsService
LoginController
MakerController
Preferences
PreferencesService
Role
RoleService
User
UserController
UserService

Erroneous Input

Invalid Login

Create New

Home

Basic Info

Customer Type *

International

Customer Title *

Mr.

First Name *

First Name

Mandatory field

Last Name

Last Name

Name can contain

Date Of Birth *

09/09/99

Details

Driver's License

Issd

State ID

Issd

SSN

SSN

Passport Number

Passport Number

Address Line 1 *

Issd

Address Line 2

Address Line 2

City *

City

Mandatory field

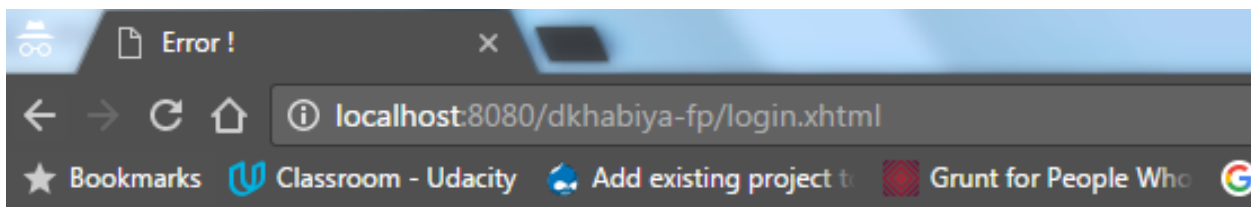
State *

Issd

Zipcode *

80,818

Save



Error !

Invalid user name or password.

[Back to Login Page](#)

Expected Results

An application that supports user creation, login/logout and creation/manipulation of data. The expected results are:

1. User creation with different roles
2. User authentication
3. Security implementation
4. Data viewing
5. Data addition
6. Data manipulation

Issues Faced

While completing the project the following issues were faced.

1. Was unable to maintain flow of data between pages because the controllers were "Stateless". Changed them to "SessionScoped" and made it work.
2. Was having difficulty in saving user information while creating/updating entities.