Final Project Readme

Project Summary

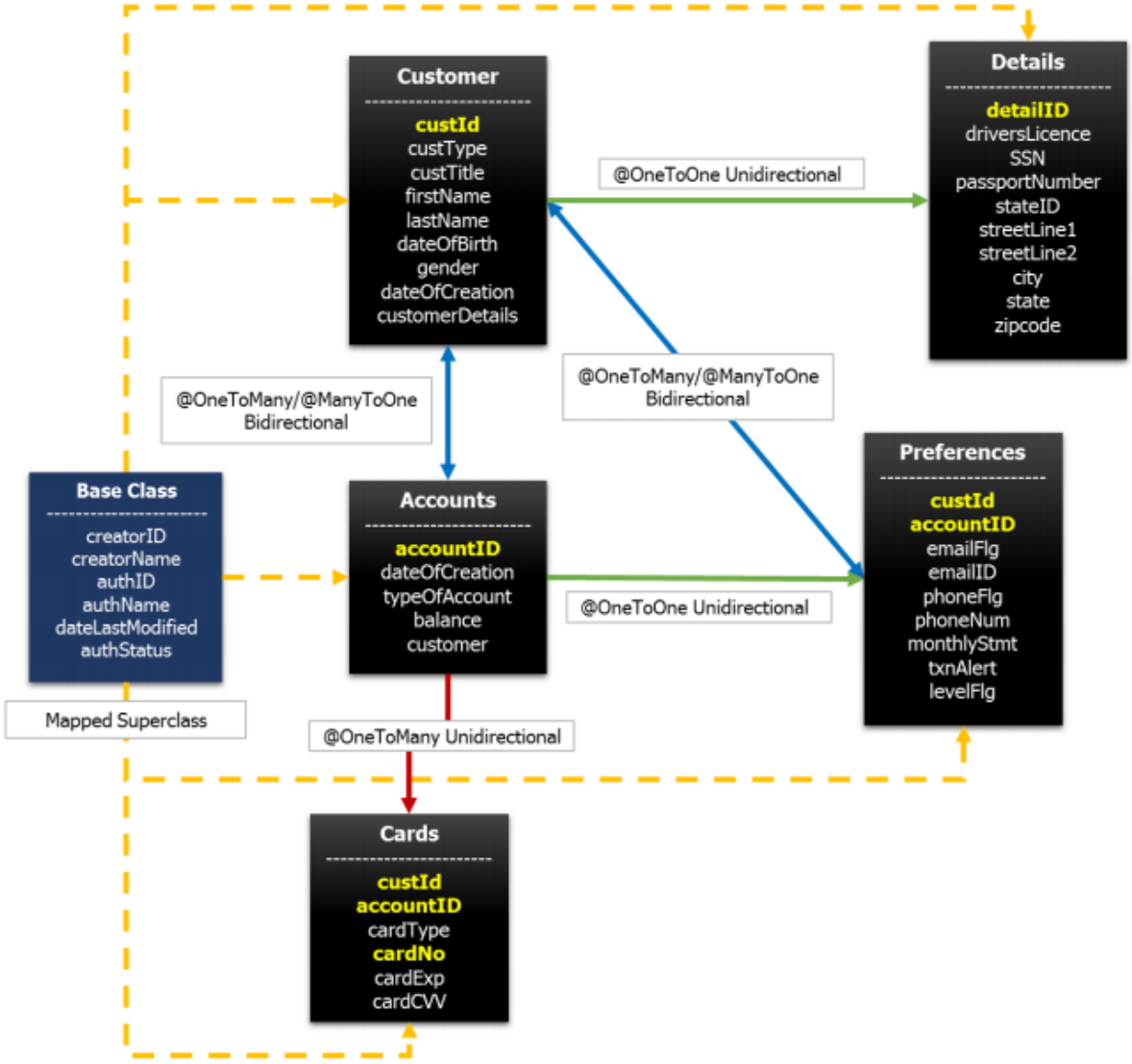
Financial Data Directory – a banking solution application. This basis of this application is to manage customers and their accounts in a bank. It will allow the bank personnel to create new customers and accounts, set preferences for the customer and maintain credit/debit card details of the customer. 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.

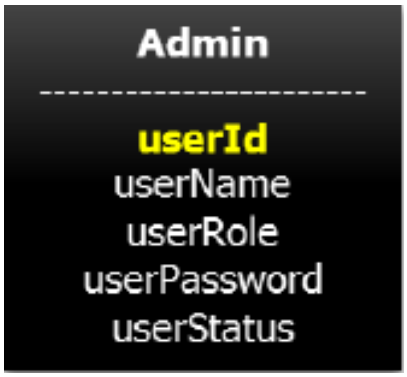
The application revolves around the following entities.
1. **Customer** - Main entity; includes customer type, unique customer ID, name, DOB and date of creation.
2. **Details** - Additional details of the customer like SSN, Passport number, address, etc. There is a one to one relationship of a customer 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"

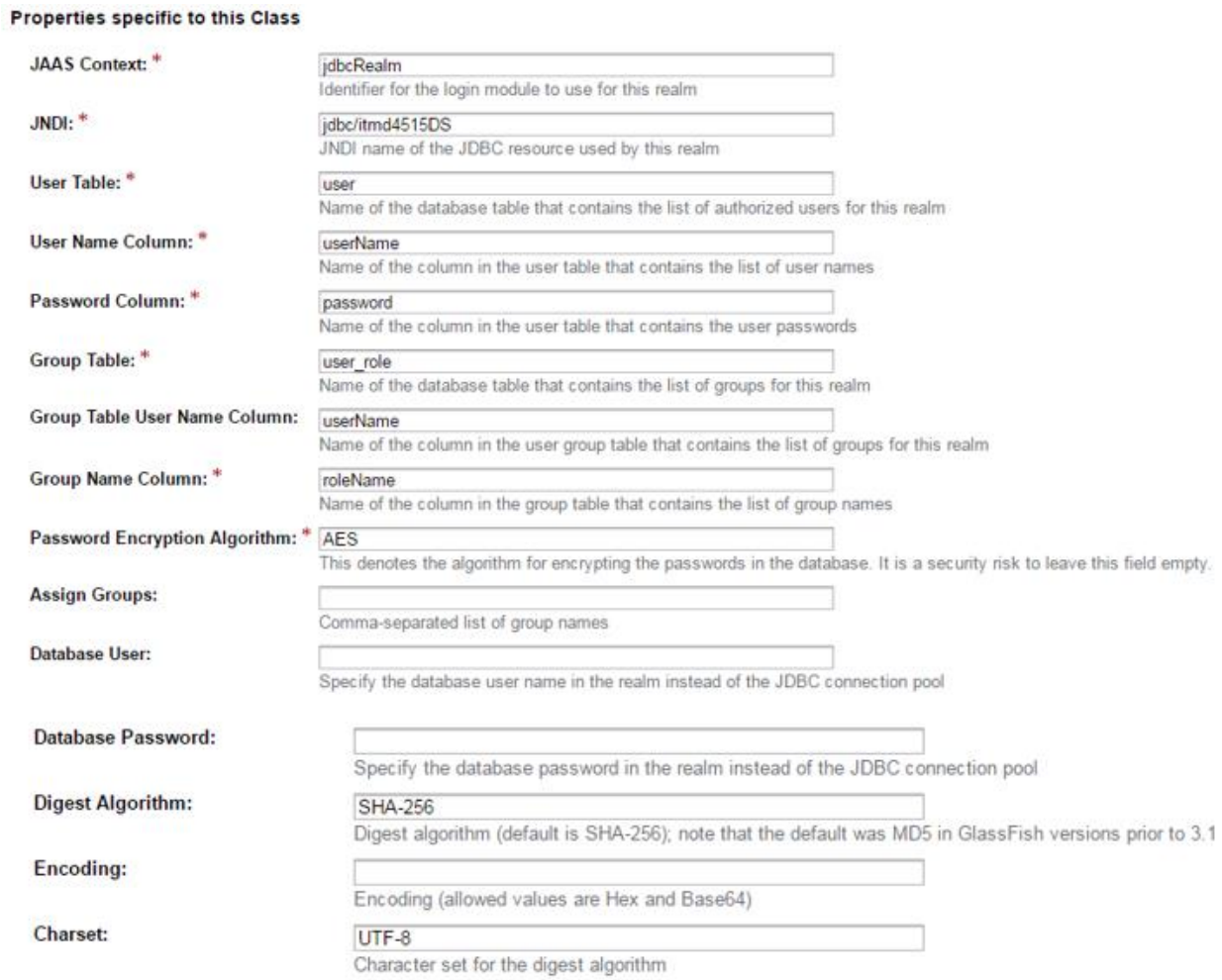
Requirements (Installation, Compile, Runtime, etc)

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.



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

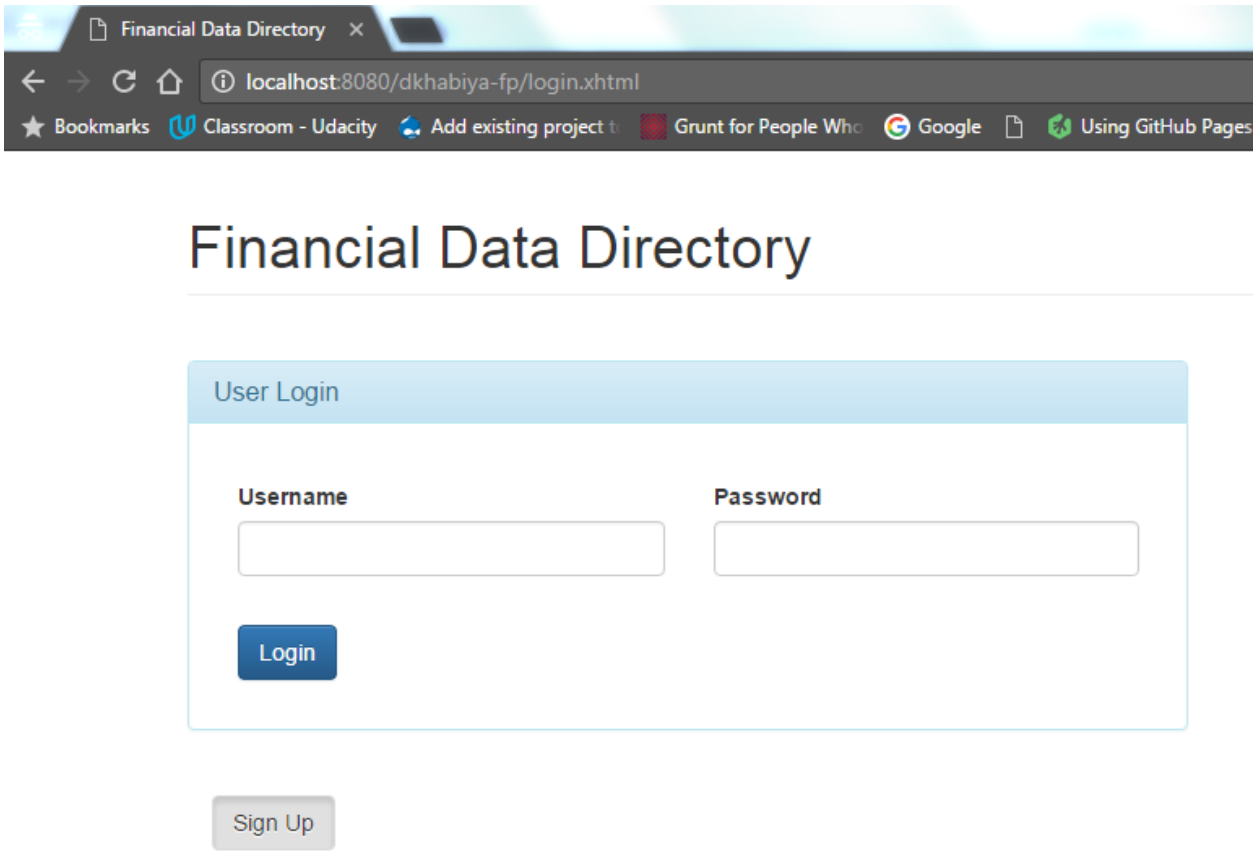
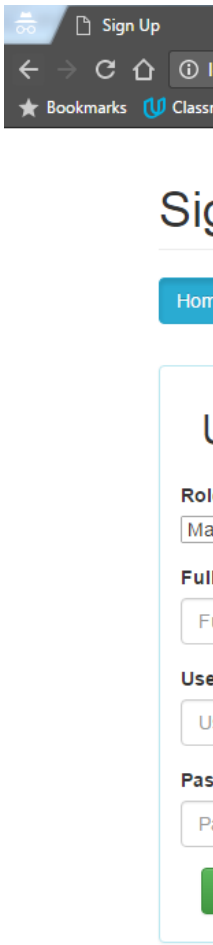
Username	Password
maker1	maker1
maker2	maker2
maker3	maker3
maker4	maker4
checker1	checker1
checker2	checker2
checker3	checker3
checker4	checker4

Sufficient data has been provided to test the application.

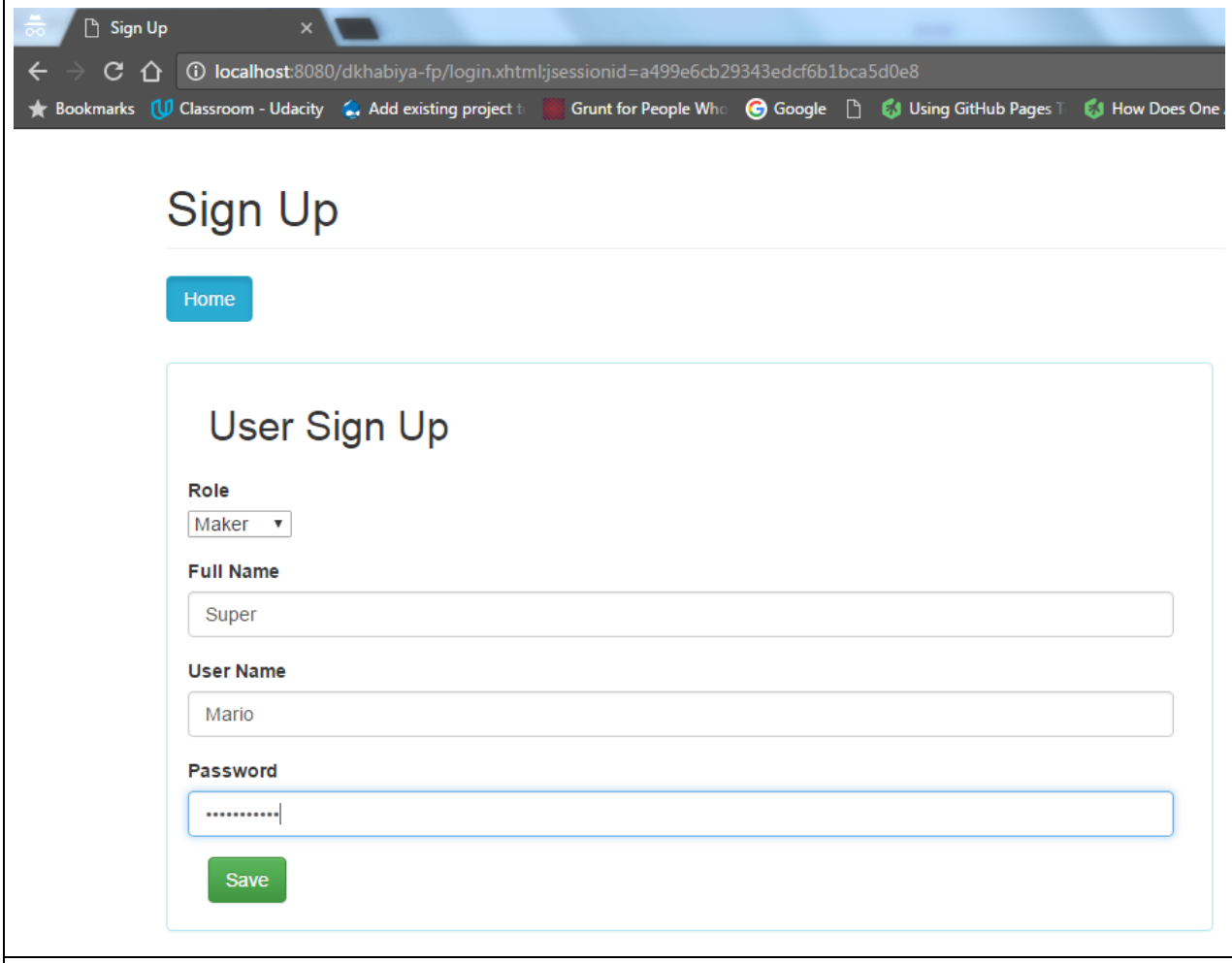
Screen Captures

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

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

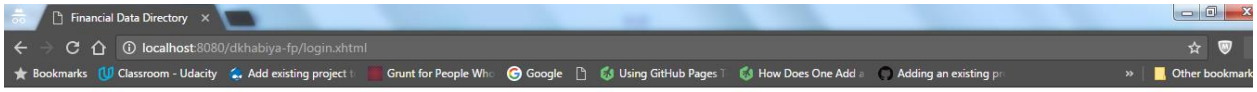
Landing Page	
	

Maker Creation	



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

Maker Landing Page



Financial Data Directory

Welcome Super MAKER

Customer Data

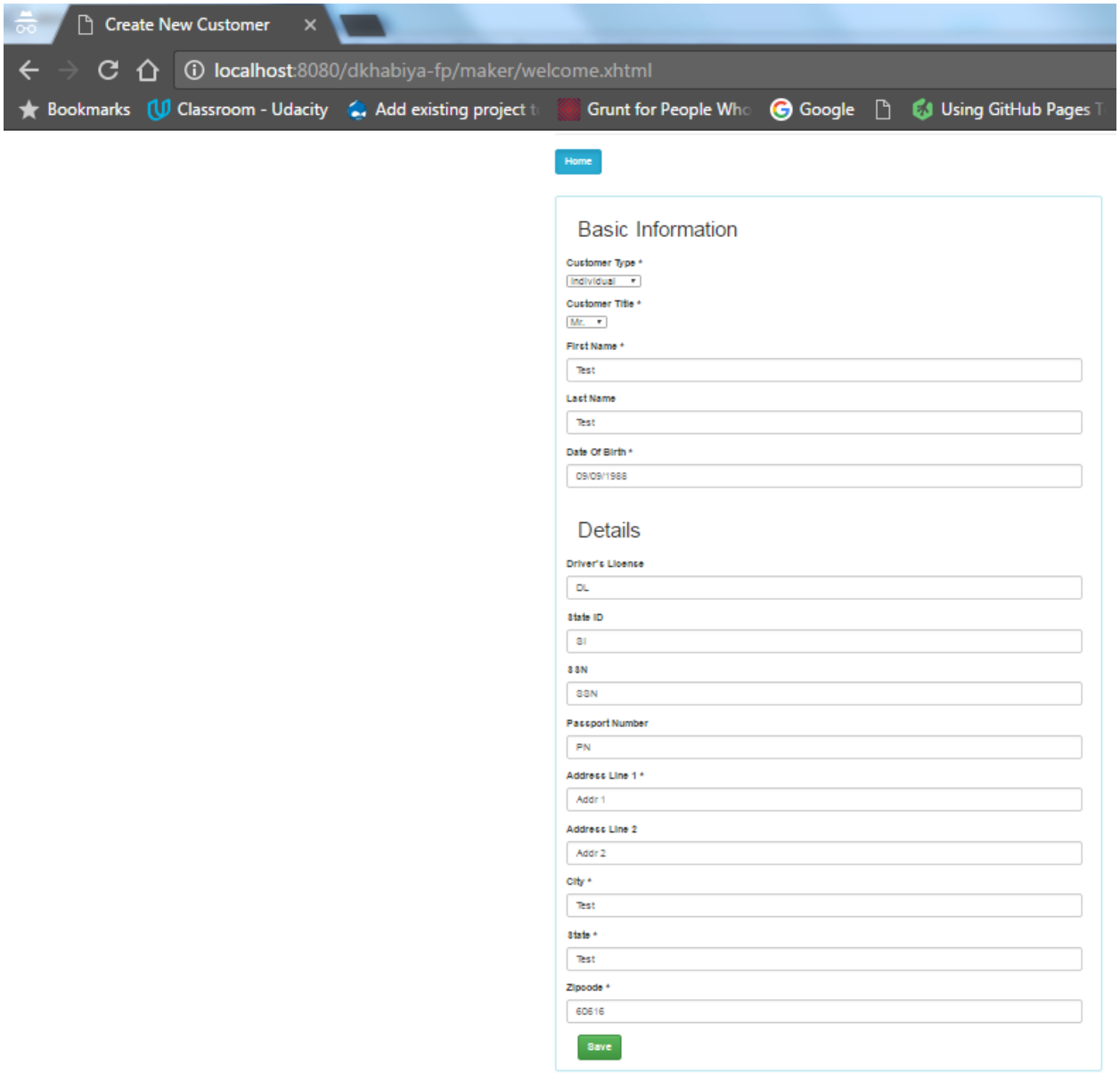
Customer ID	Name	Date Of Birth

*Click on customer ID to view individual customer details.

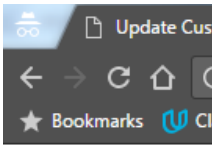
Add New Customer

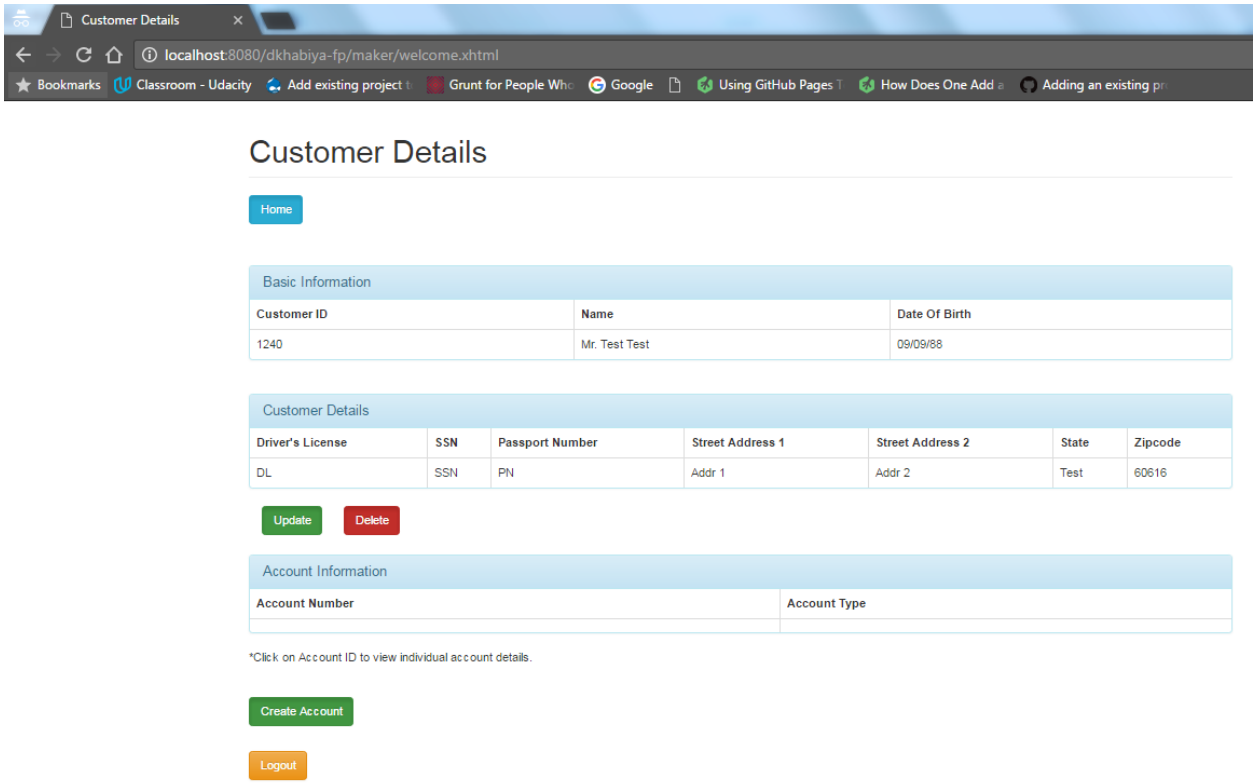
Logout

Enter Information - Maker

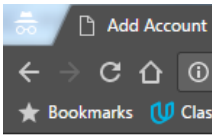


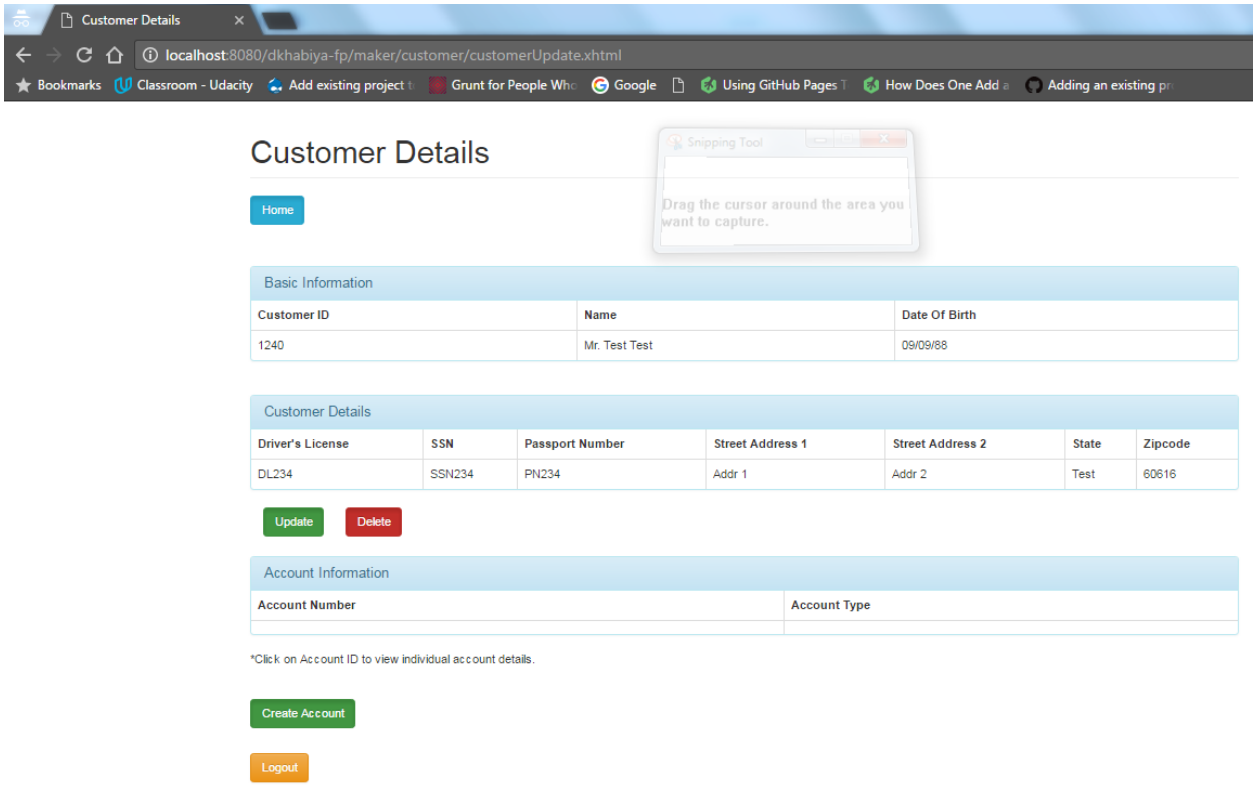
Customer View - Maker



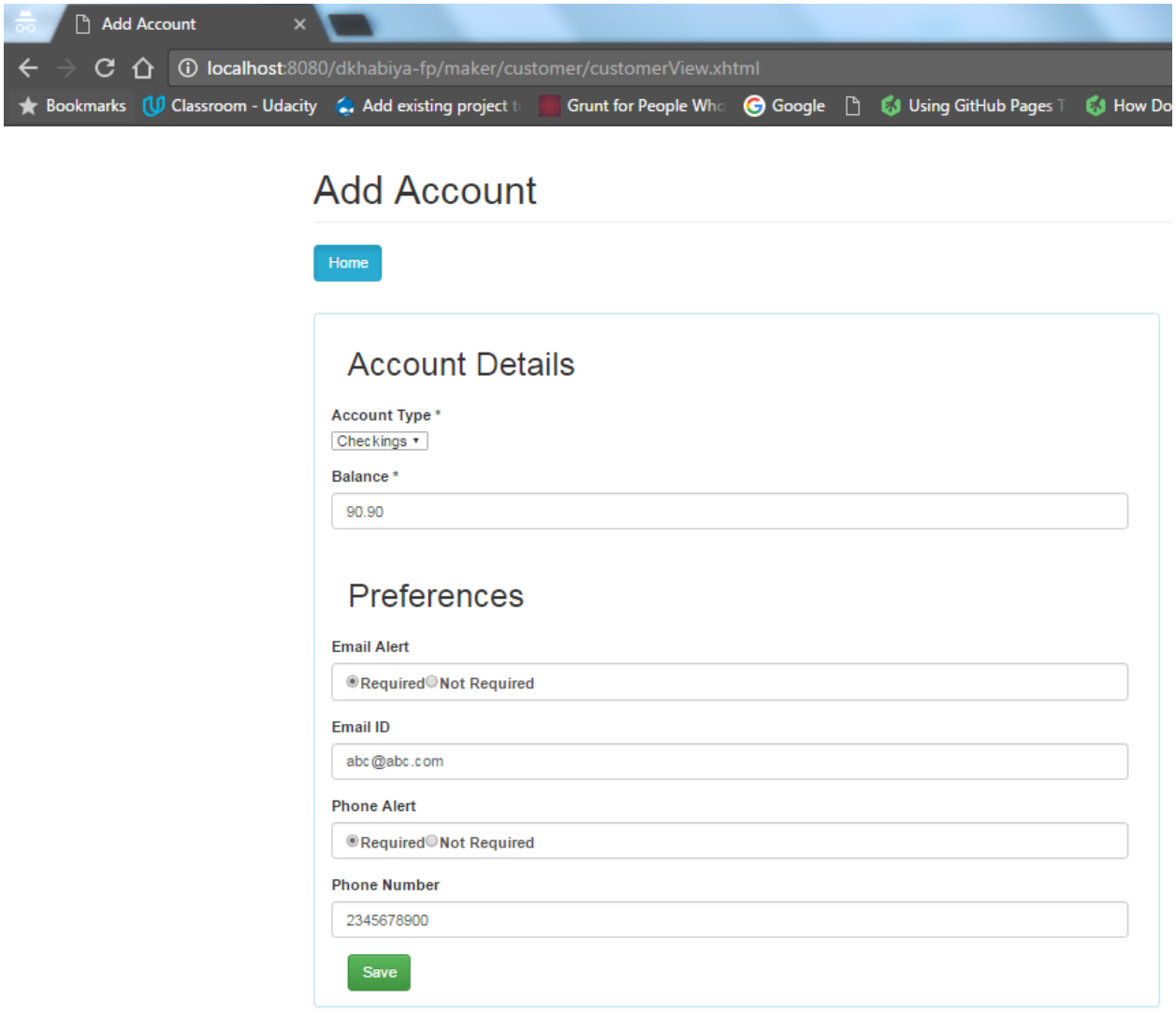


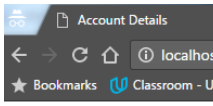
Customer Updated - Maker



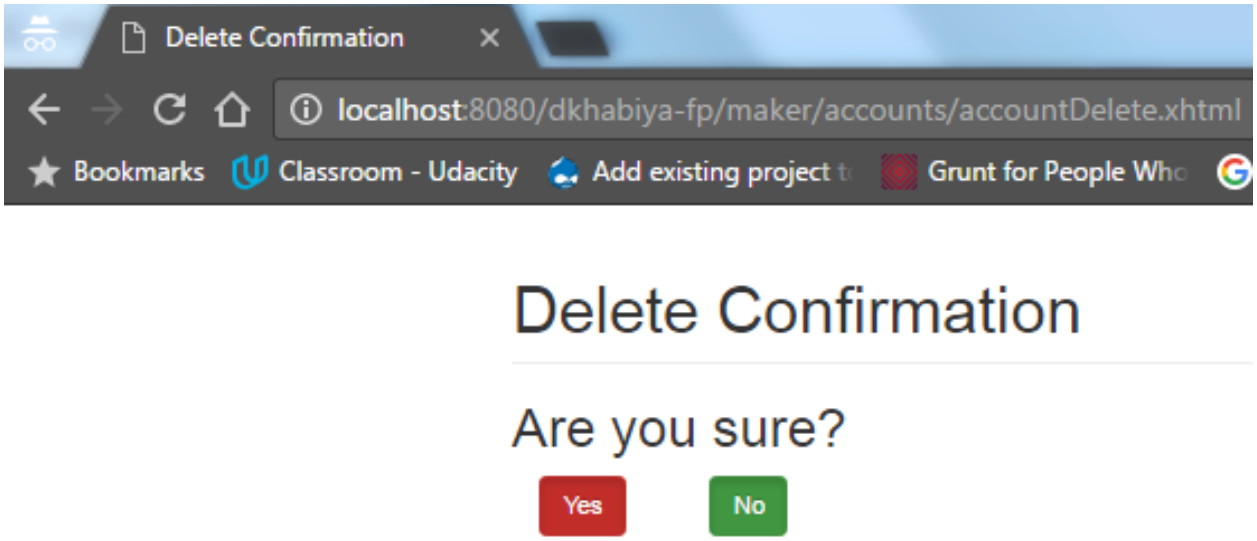


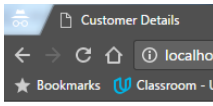
Enter Information - Maker



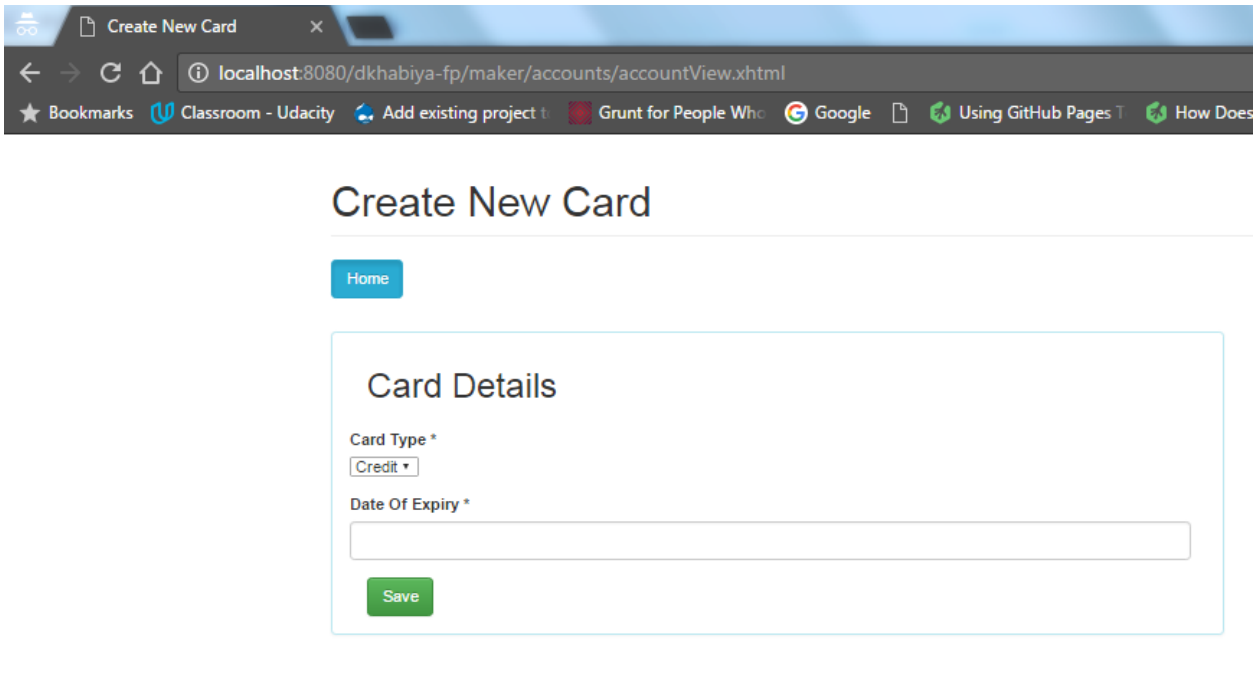


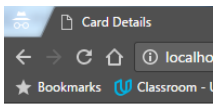
Account Delete - Maker



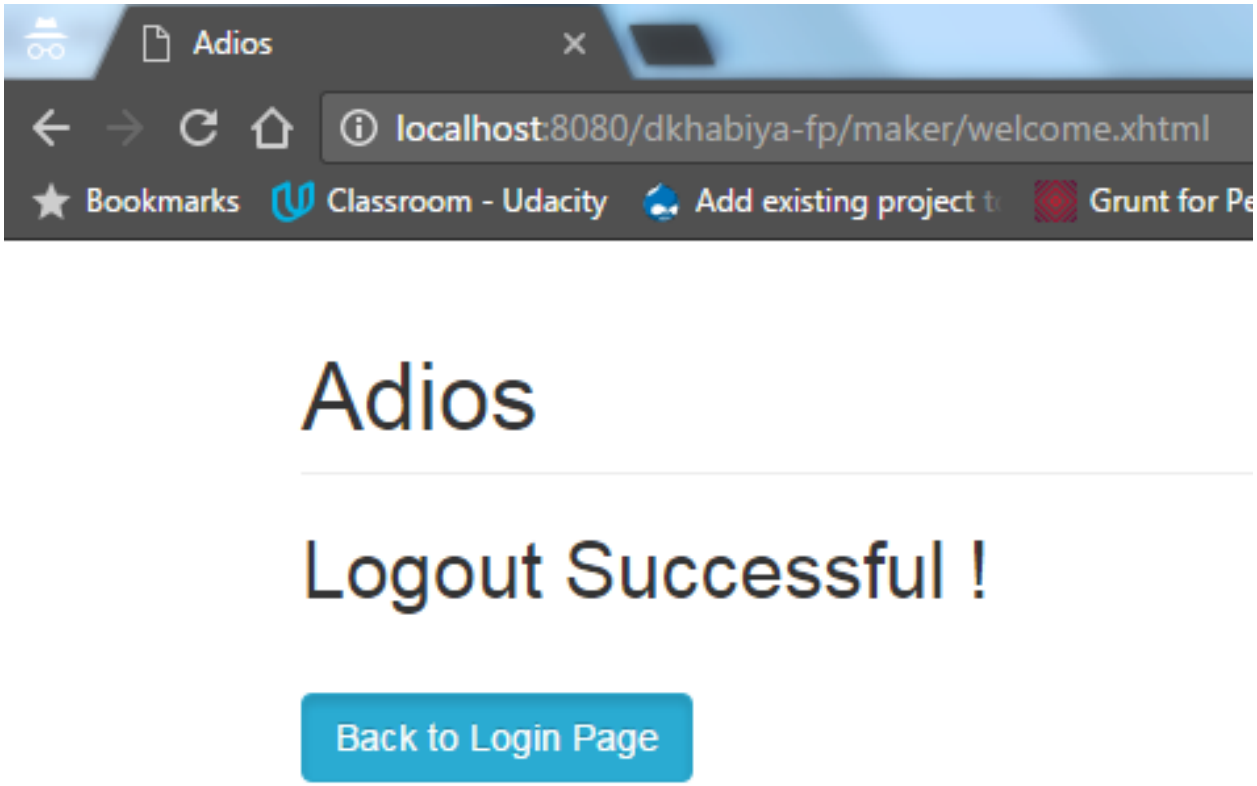


Create New Card - Maker

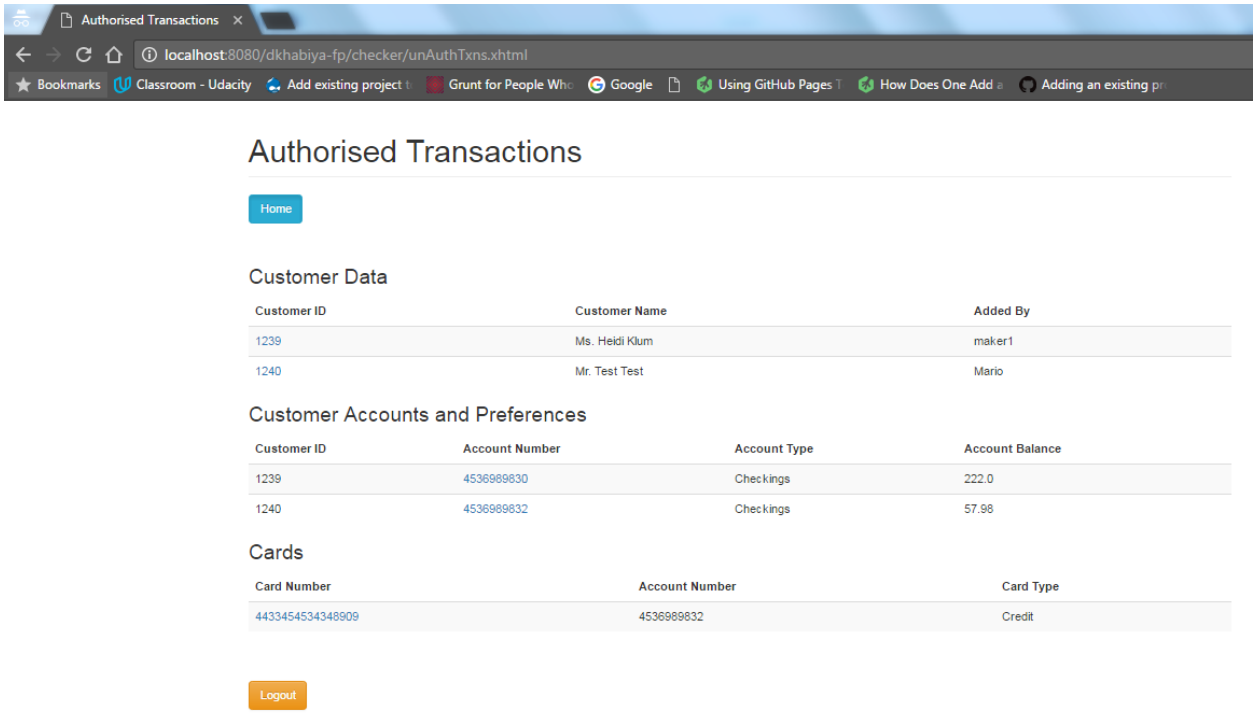


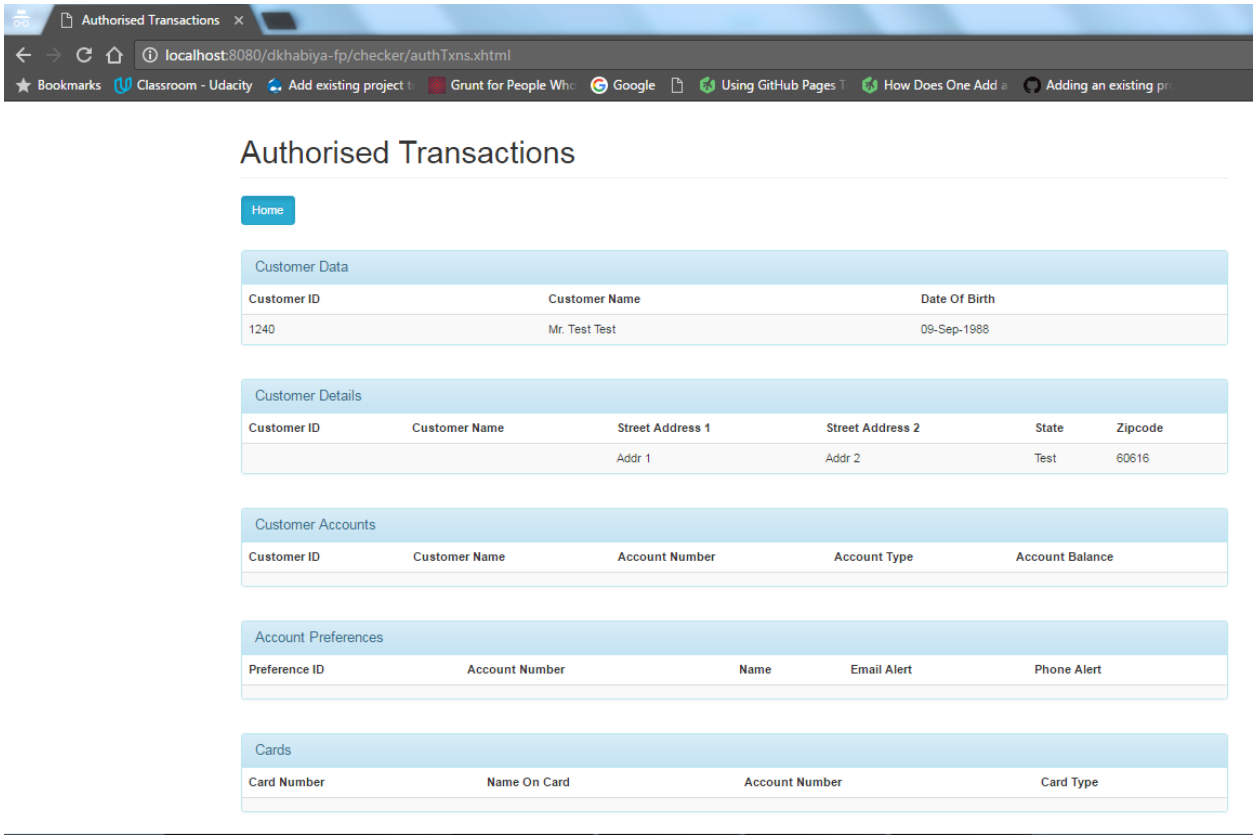
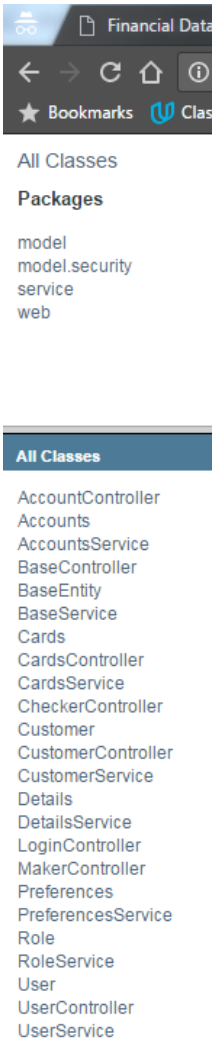


Logout - Maker



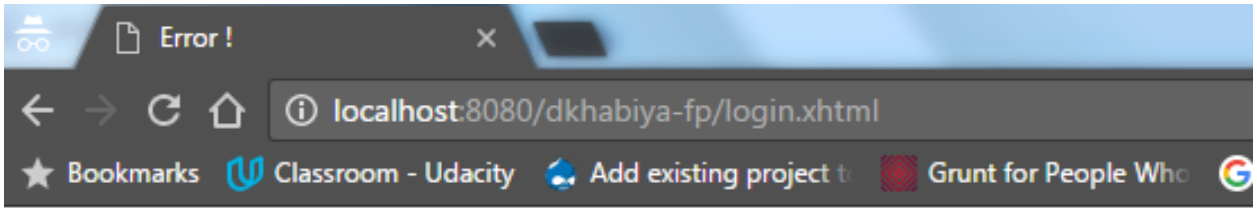
Unauthorized Transactions - Checker



Authorized Transaction	
	

Erroneous Input

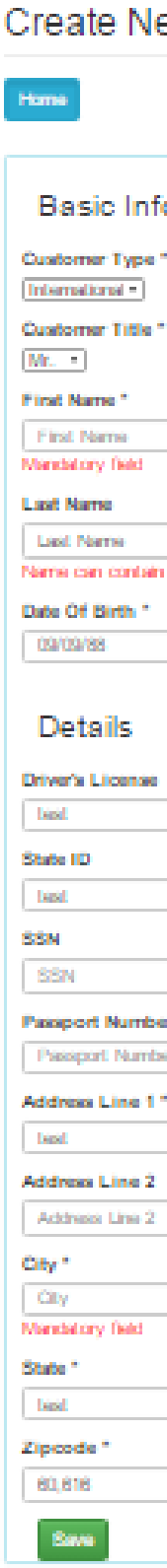
Invalid Login	



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.