Specification Document for The Project

“FlyAway Ticketing Portal”

Phase3: PG FSD Become a Backend Expert

Simplilearn / CalTech

MARCH 2022 COHORT

Prepared By

Bakau Onafuwa

July 8, 2022

Simplilearn \ CalTech \ Bakau Onafuwa

Name	Bakau Onafuwa
Email	Bakau.onafuwa@softgineer.com
GitHub Repository	git@github.com:homozapien/flyaway.git https://github.com/homozapien/flyaway.git
Project Management	Agile
Agile Methodologies	SCRUM

Simplilearn \ CalTech \ Bakau Onafuwa

Objective

The main objective of this assessment project is to develop a web-based application to that avails customers the opportunity to search for flights, book such flights, and make payments for such flights using a dummy payment service. For the prospective customer to be able to achieve this, an administrative account will provide the necessary backend data entities to provide the necessary domain models for the customer interactions.
Notably, this web application was developed to fulfil the core business requirements with additional features (e.g. reporting dashboard, embedded tomcat etc) to make the usage more intuitive and realistic.
Core development concepts like exception handling, code reviews, code refactoring, versioning, and SCRUM framework are equally showcased.

Assumptions

1. It is assumed that the reviewer of the source codes, the compiler, the tester and/or runner of this application is conversant with the Java Technologies and possess some level of programing background principally Servlets, JSPs, and JSTL.

2. This application was developed in Eclipse IDE and based on JDK 17 (Temurin); it is assumed that the same environment will be available for the review and testing of this application.

```
openjdk version "17.0.2" 2022-01-18
OpenJDK Runtime Environment Temurin-17.0.2+8 (build 17.0.2+8)
OpenJDK 64-Bit Server VM Temurin-17.0.2+8 (build 17.0.2+8, mixed mode, sharing)
```

3. It is assumed that the code reviewer and the tester of this application are familiar with Git and can clone the Github repository for this project.

4. The development of this application never focused on the security and boundary contexts of the underlying OS and the development database. A simple “localhost” connection to the underlying database will suffix (refer to the hibernate.cfg.xml).

Simplilearn \ CalTech \ Bakau Onafuwa

5. A Mysql RDBMS will be available in the test environment and database named: flyaway is created *ab initio* as shown thus,

```
baknaf@baknaf1 ~/WorkShop/DEV/eclipse/current/workspace/flyaway master • mysql -V
mysql  Ver 8.0.29 for macos10.14 on x86_64 (Homebrew)
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| flyaway            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| university_batch   |
+--------------------+
6 rows in set (0.05 sec)
```

6. A maven infrastructure is required to be able to re-build / re-package the application as shown thus,

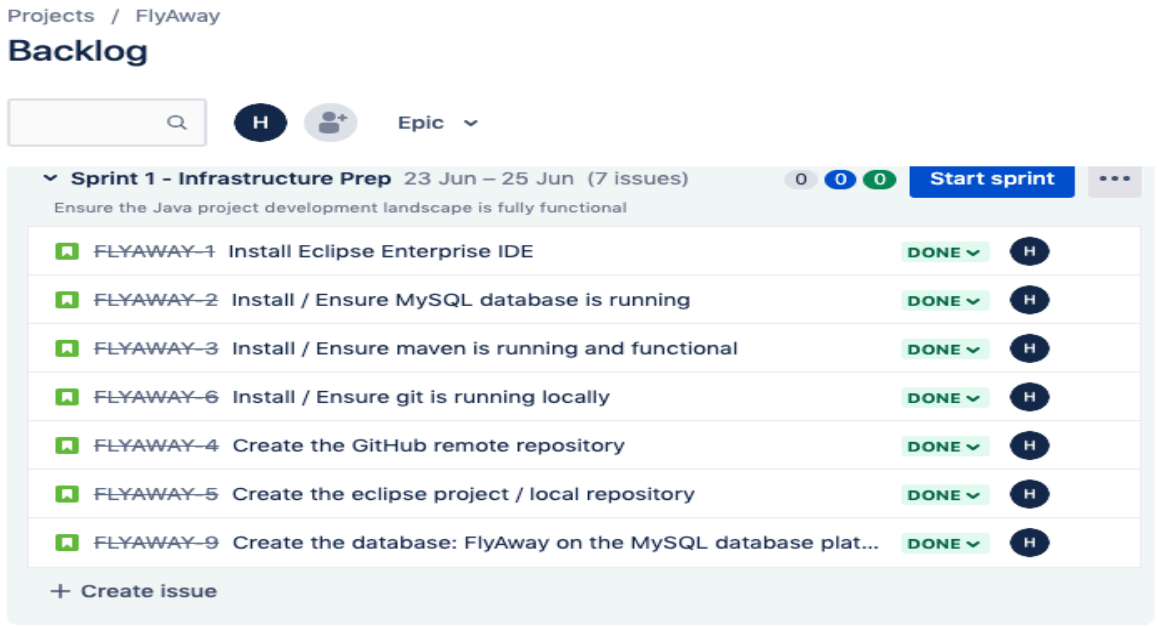
```
baknaf@baknaf1 ~/WorkShop/DEV/eclipse/current/workspace/flyaway master • mvn -v
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: /usr/local/Cellar/maven/3.8.1/libexec
Java version: 17.0.2, vendor: Eclipse Adoptium, runtime: /Library/Java/JavaVirtualMachines/temurin-17.jdk/Contents/Home
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.14.6", arch: "x86_64", family: "mac"
baknaf@baknaf1 ~/WorkShop/DEV/eclipse/current/workspace/flyaway master •
```

Simplilearn \ CalTech \ Bakau Onafuwa

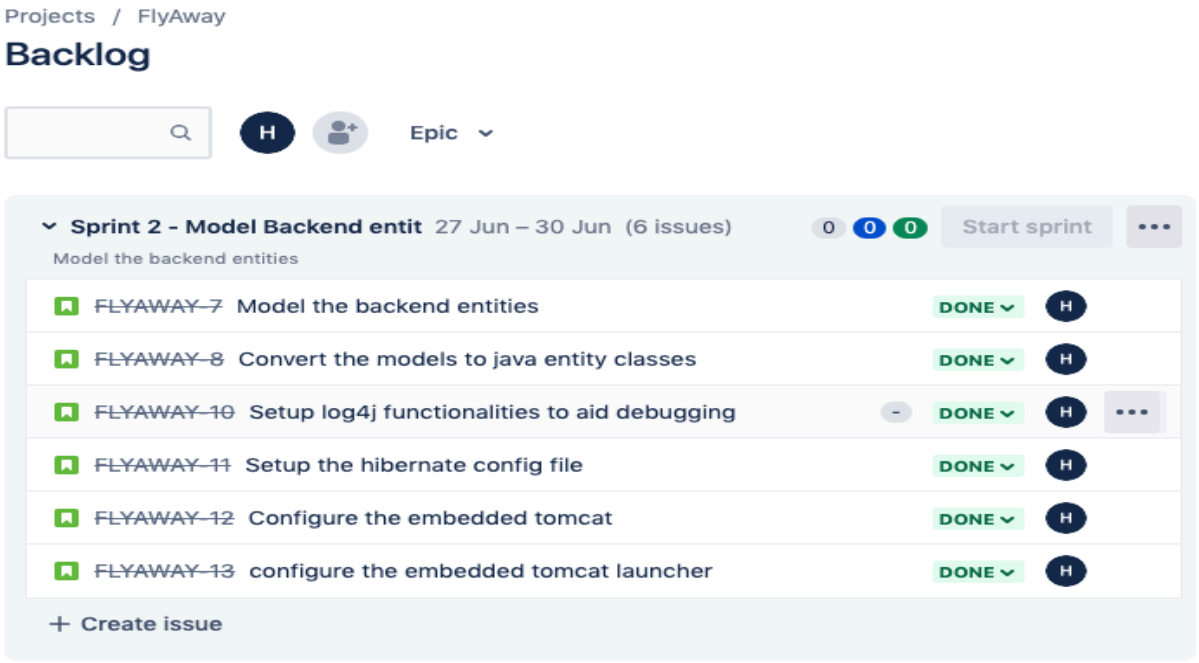
Project Management

This application was developed within a period of 3 weeks. I had employed the SCRUM framework for an iterative development. The user stories in the Product Backlog are high-level (without task break down for the individual story) and were implemented in four iterations as shown below.

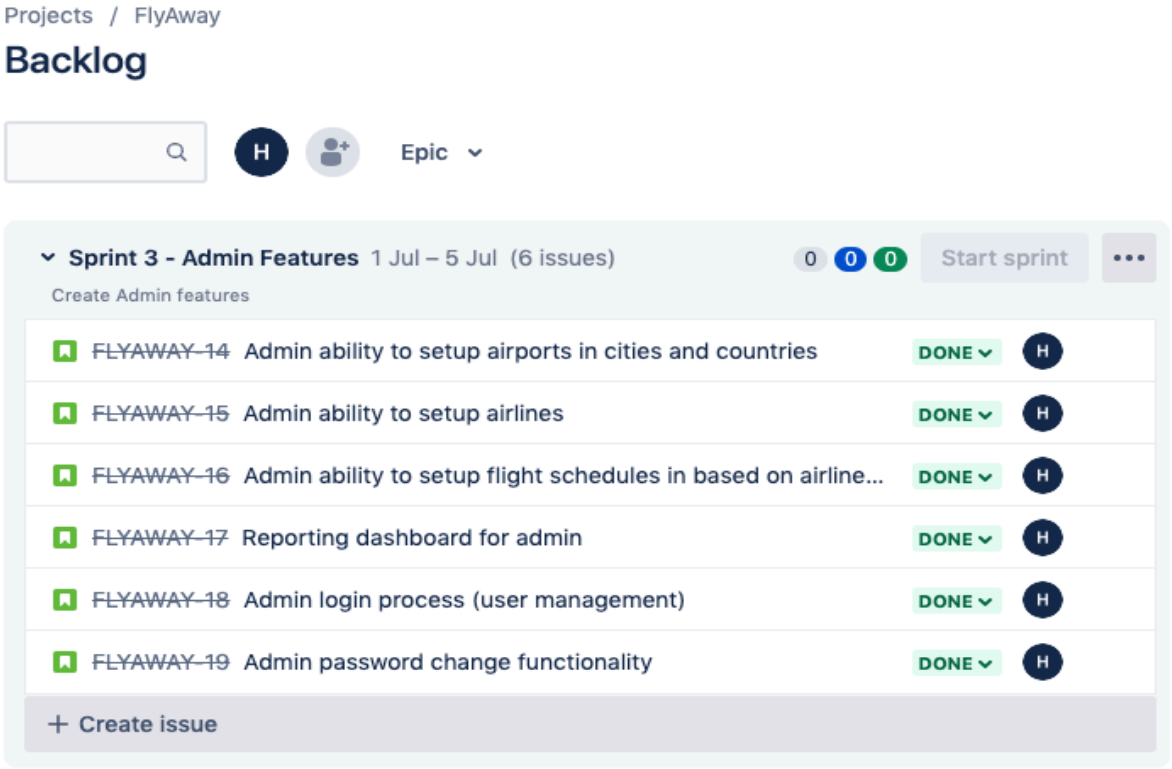
Sprint 1 Goal: Get the environment and infrastructure ready



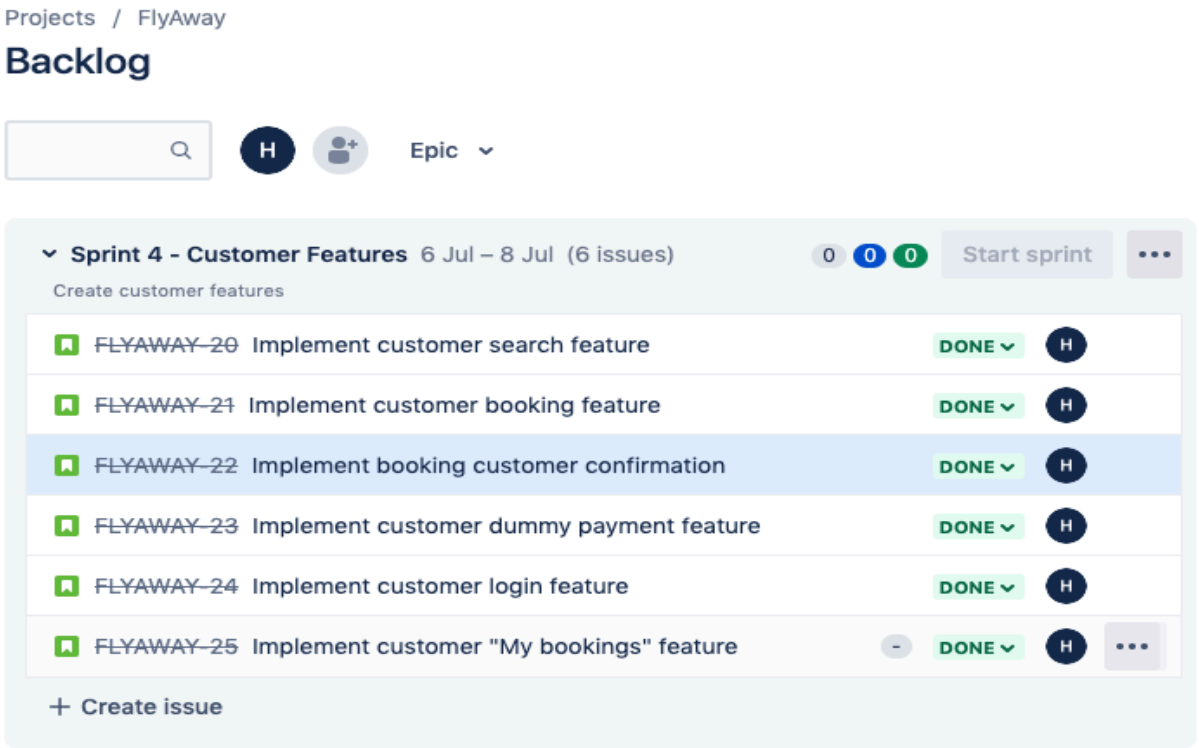
Sprint 2 Goal: Build Application Framework / Model Entities



Sprint 3 Goal: Admin Features

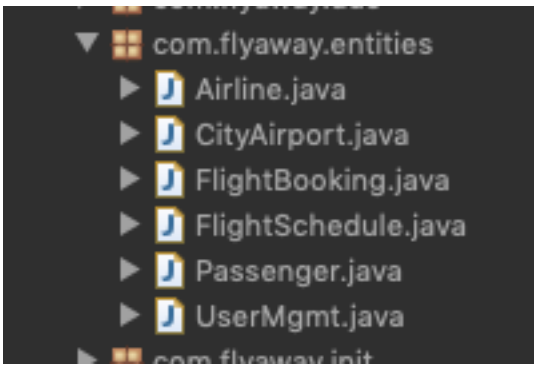


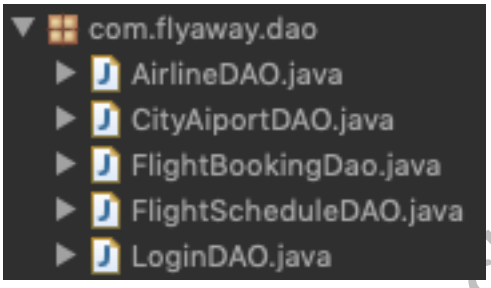
Sprint 4: Customer Feature



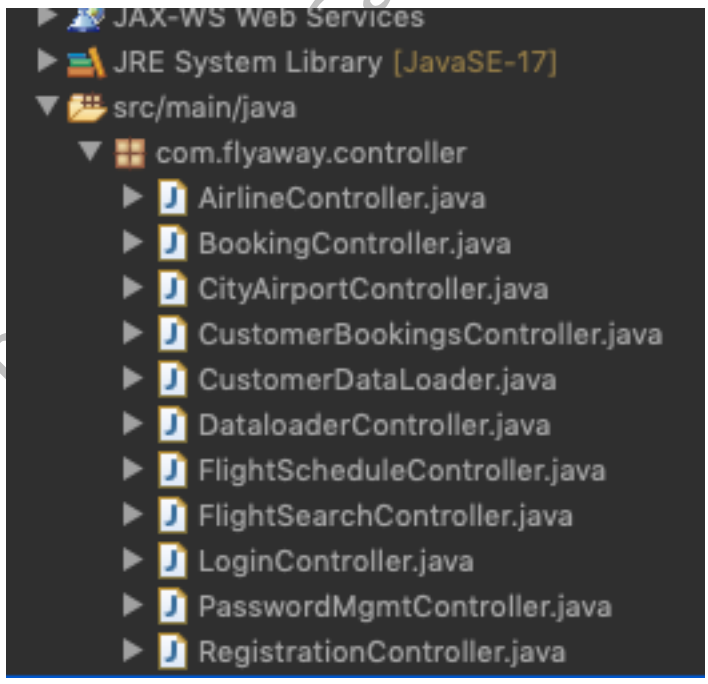
Project Structure

1. Backend: The backend models comprises of Hibernate entities, DAOs, and MySQL database for storage.

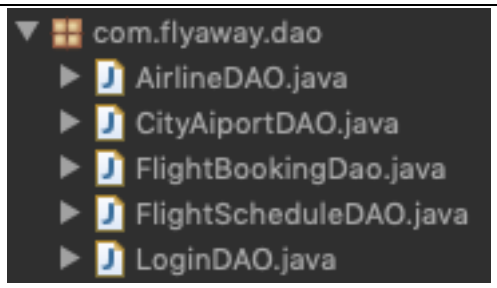




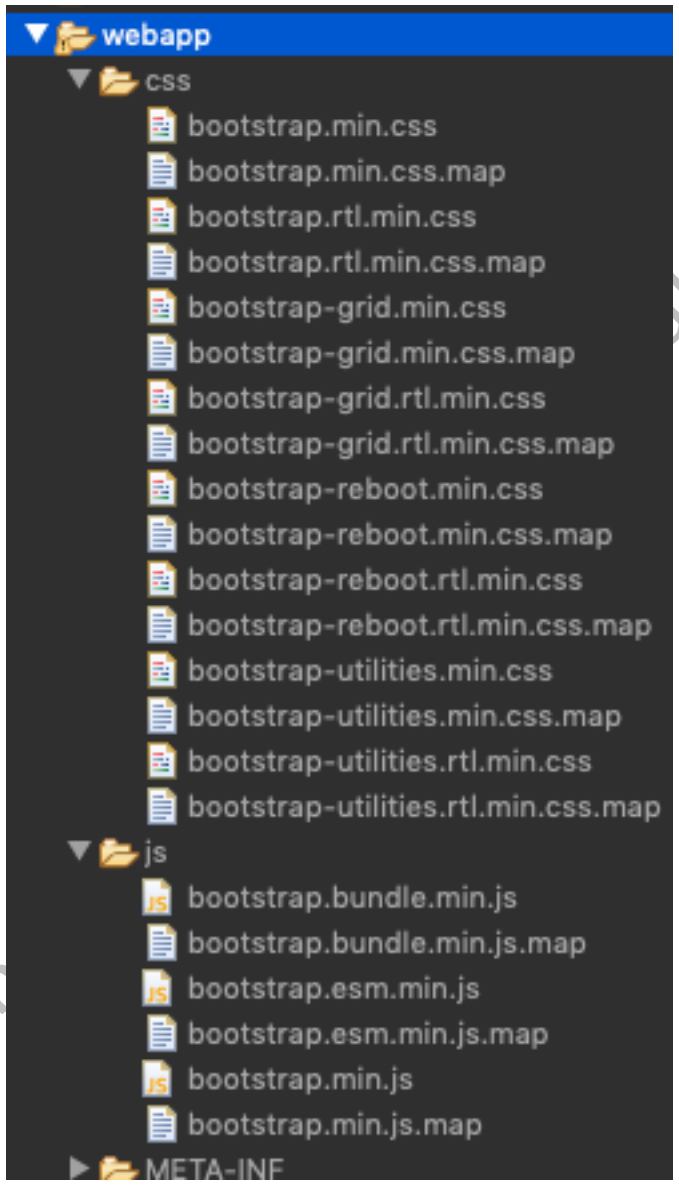
2. Business Logic: The logic is implemented using the Java Servlet (controllers), Service Layers, and POJOs



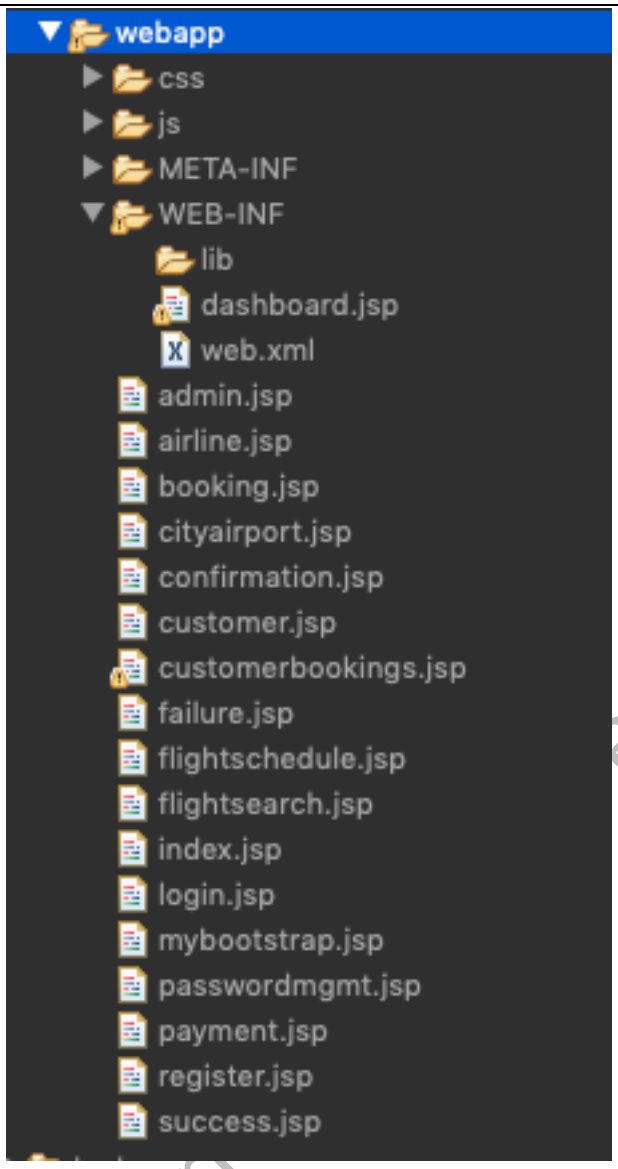
Simplilearn \ CalTech \ Bakau Onafuwa



3. Front end: The views are implemented with JSP with some bootstrap inclusion.



Simplilearn Bakau Onafuwa



Simplilearn Bakau Onafuwa

Execution Flow

This is a web-based application and there is some degree of free typing in some of the UI controls however, these view inputs are validated against the relationships amongst the backing entity beans.

Two types of user roles are allowed during authentication viz:

- Admin: to perform the setup and admin functionalities.
- Customer: use the application based off the setup by the admin.

To use the application, the admin must first login to setup the domain data using the following credential:

- Username: admin@flyaway.com
- Password: flyaway
- UserType: Admin

These values should be inserted automatically by hibernate during bootstrapping processes (otherwise, insert manually from the command prompt using the following insert statement:

```
public class UserMgmt implements Serializable
{
    private static final long serialVersionUID = 3756680645803624405L;
    @Id
    @ColumnDefault("'admin@flyaway.com'")
    private String emailId;
    @ColumnDefault("'flyaway'")
    private String password;
    @ColumnDefault("'Admin'")
    private String typeOfUser;

    public UserMgmt() {
        super();
    }
```

Mysql prompt > Use flyway;

Insert into usermgmt (“emailid”, “password”, “typeofuser”)
Values (‘admin@flyway.com’, “flyaway”, “Admin”)

Note: You may change the admin password with the application

Simplilearn \ Bakau Onafuwa

The admin has the responsibility to create the airport, create airlines, and schedule flights.
Note: From the view, airport could only be created in the following countries: USA, India, Nigeria, France, and UAE.

cityairport.jsp

```
<div class="row mb-3">

    <label for="airportcountry" class="col-sm-2 col-form-label">Airport
        Country</label>

    <div class="col-sm-10">
        <select class="form-select form-select-sm"
            name="airportcountry" aria-label=".form-select-sm example" required>
            <option selected>Open this select menu</option>
            <option value="USA">USA</option>
            <option value="Nigeria">Nigeria</option>
            <option value="India">India</option>
            <option value="France">France</option>
            <option value="UAE">UAE</option>
        </select>
    </div>
</div>
```

Note: The entity relationships ensure that the input values are validated based based on the Foreign Key relationships / joinColumns.

The customer may use the application by searching and booking flights accordingly.

Simplilearn \ CalTech \ Bak

Setup for re-build / re-packaging

Compiling / building the application requires maven to be available on the local machine.

From the project folder, run the following maven goals:

```
[INFO] org/apache/tomcat/util/threads/ThreadPoolExecutor$Worker.class already added, skipping
[INFO] org/apache/tomcat/util/threads/ThreadPoolExecutor.class already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  15.669 s
[INFO] Finished at: 2022-07-08T11:00:46+01:00
[INFO] ------------------------------------------------------------------------
 baknaf@baknaf1 ~/WorkShop/DEV/eclipse/current/workspace/flyaway master • mvn clean compile package
```

During the first-run, the launcher will require the tomcat port for the embedded tomcat server for a successful start as shown here:

Project folder: > java -jar ./target/flyaway-jar-with-dependencies.jar 9191

```
baknaf@baknaf1:~/WorkShop/DEV/eclipse/current/workspace/flyaway
Compressing objects: 100% (17/17), done.
Writing objects: 100% (29/29), 238.56 KiB | 393.00 KiB/s, done.
Total 29 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 8 local objects.
To github.com:homozapien/flyaway.git
   0f880b7..97f53f0  master -> master
 baknaf@baknaf1 ~/WorkShop/DEV/eclipse/current/workspace/flyaway master clear
 baknaf@baknaf1 ~/WorkShop/DEV/eclipse/current/workspace/flyaway master • ? java -jar ./target/flyaway-jar-with-dependencies.jar 9191
```

where 9191 is your desired port on which to run the embedded tomcat. If no port is specified (from the command line), a default of 8080 would be used.

Database:

The hibernate config file is defined as thus:

confirmation.jsp | UserMgmt.java | hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
                                        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
  <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
  <property name="hibernate.connection.password"/>
  <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/flyaway?serverTimezone=America/Chicago</property>
  <property name="hibernate.connection.username">root</property>
  <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>
  <property name="hibernate.hbm2ddl.auto">update</property>
  <property name="hibernate.show_sql">true</property>
  <property name="hibernate.format_sql"> true </property>
  <mapping class="com.flyaway.entities.CityAirport"/>
  <mapping class="com.flyaway.entities.Airline"/>
  <mapping class="com.flyaway.entities.FlightSchedule"/>
  <mapping class="com.flyaway.entities.FlightBooking"/>
  <mapping class="com.flyaway.entities.Passenger"/>
  <mapping class="com.flyaway.entities.UserMgmt"/>
 </session-factory>
</hibernate-configuration>
```

The database name to be created in mysql is: flyaway

Simplilearn | Bakau Onafuwa | CalTech

All mapped entities (i.e. tables) will be created in this database.

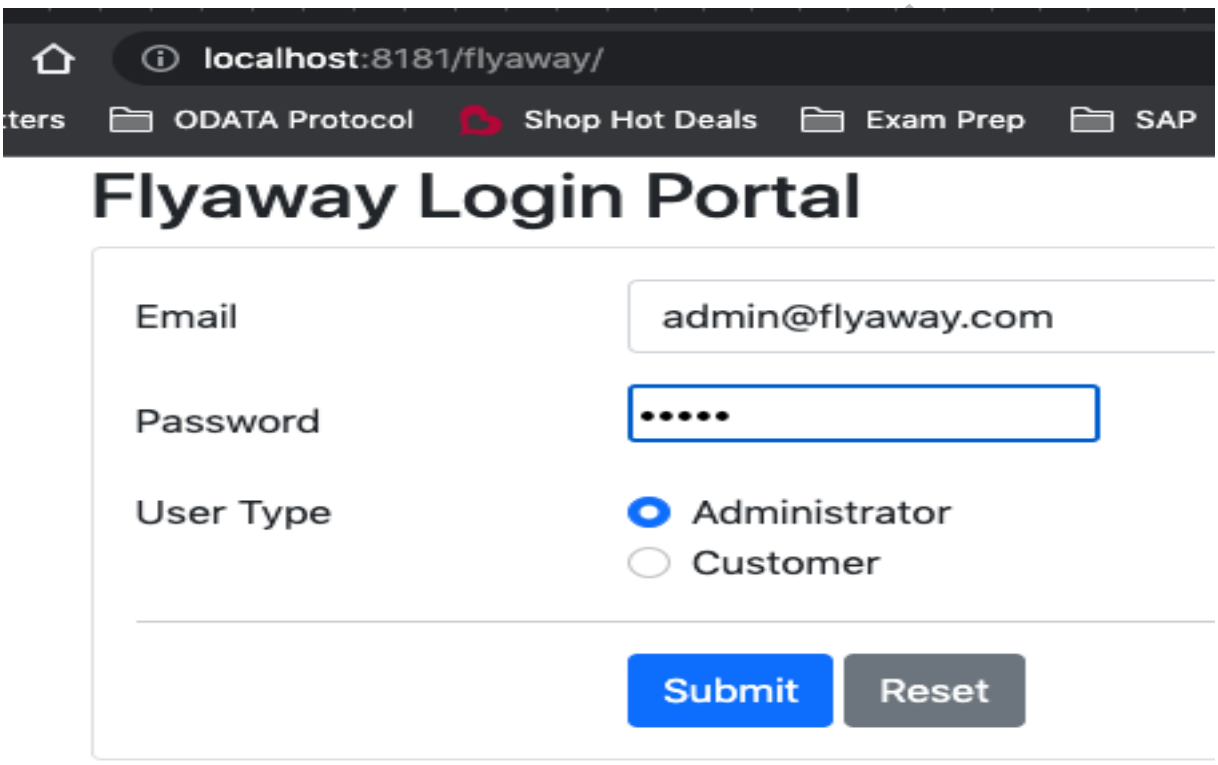
Data Structure & Algorithms

No special algorithms were used.

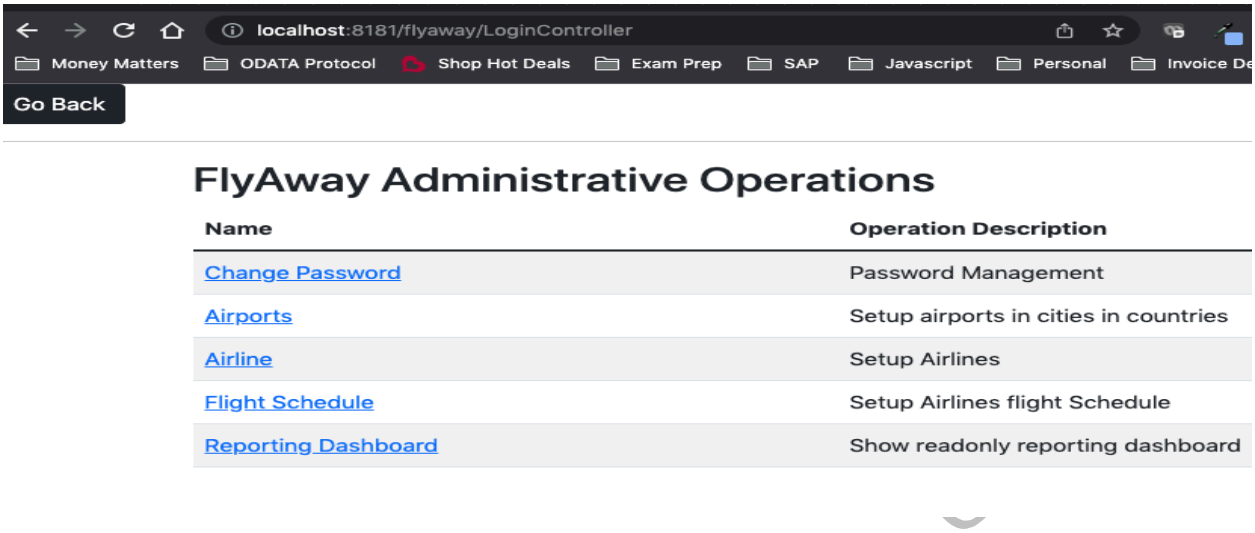
Unit Testing

To successfully compile and run this program, minimum of JDK 1.8 must be running in the test environment.

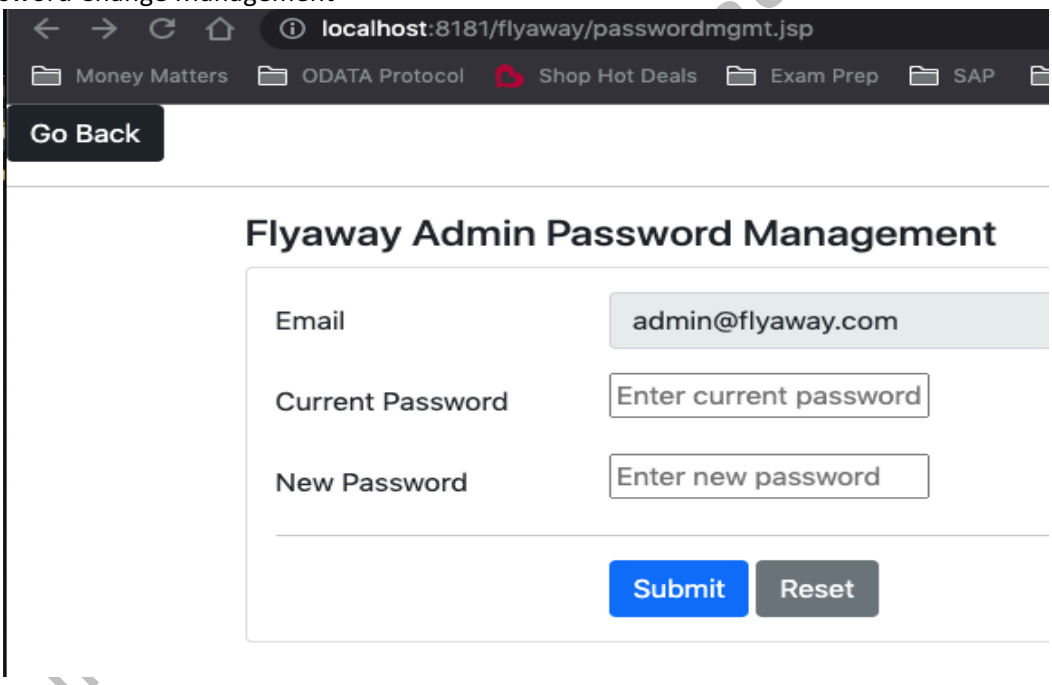
1. Login Page



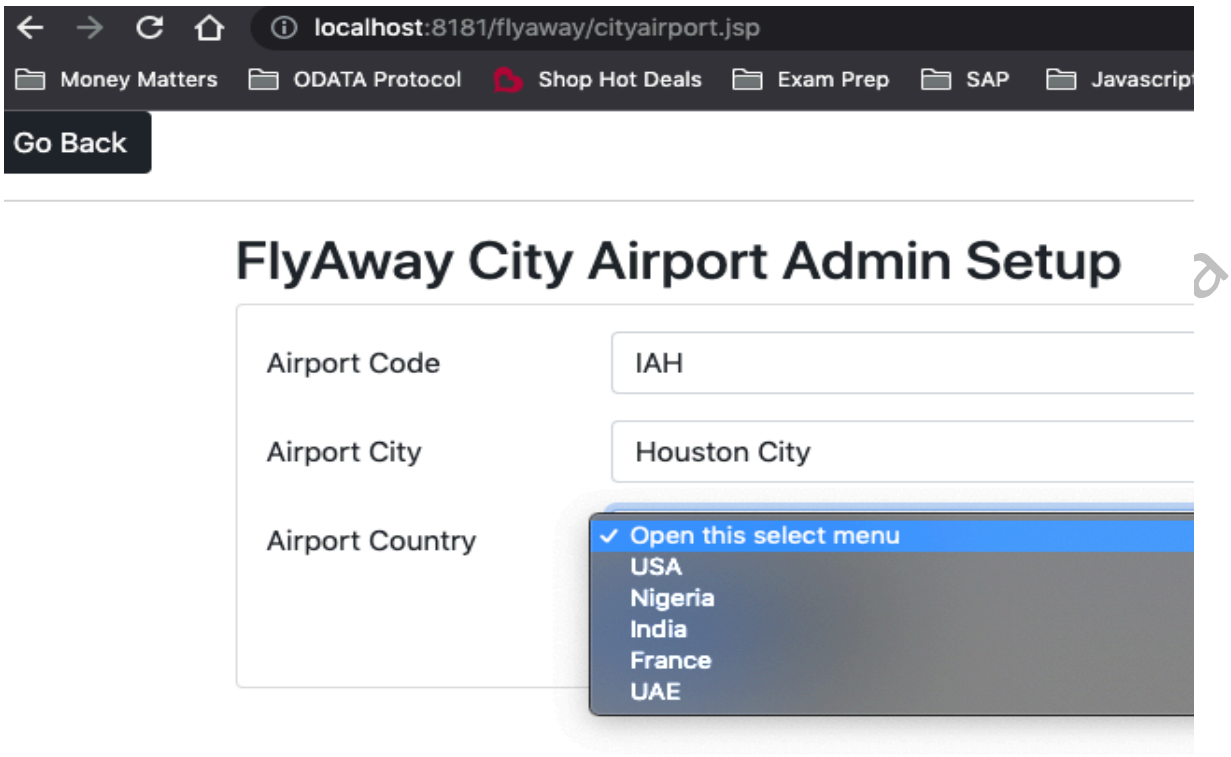
2. Admin Operation



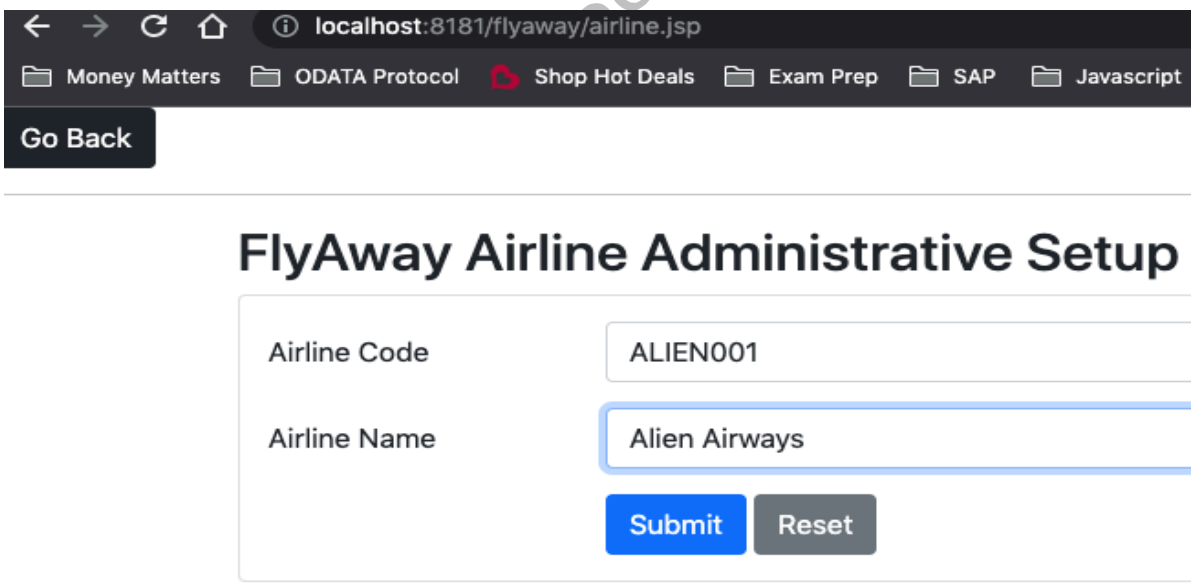
3. Password Change Management



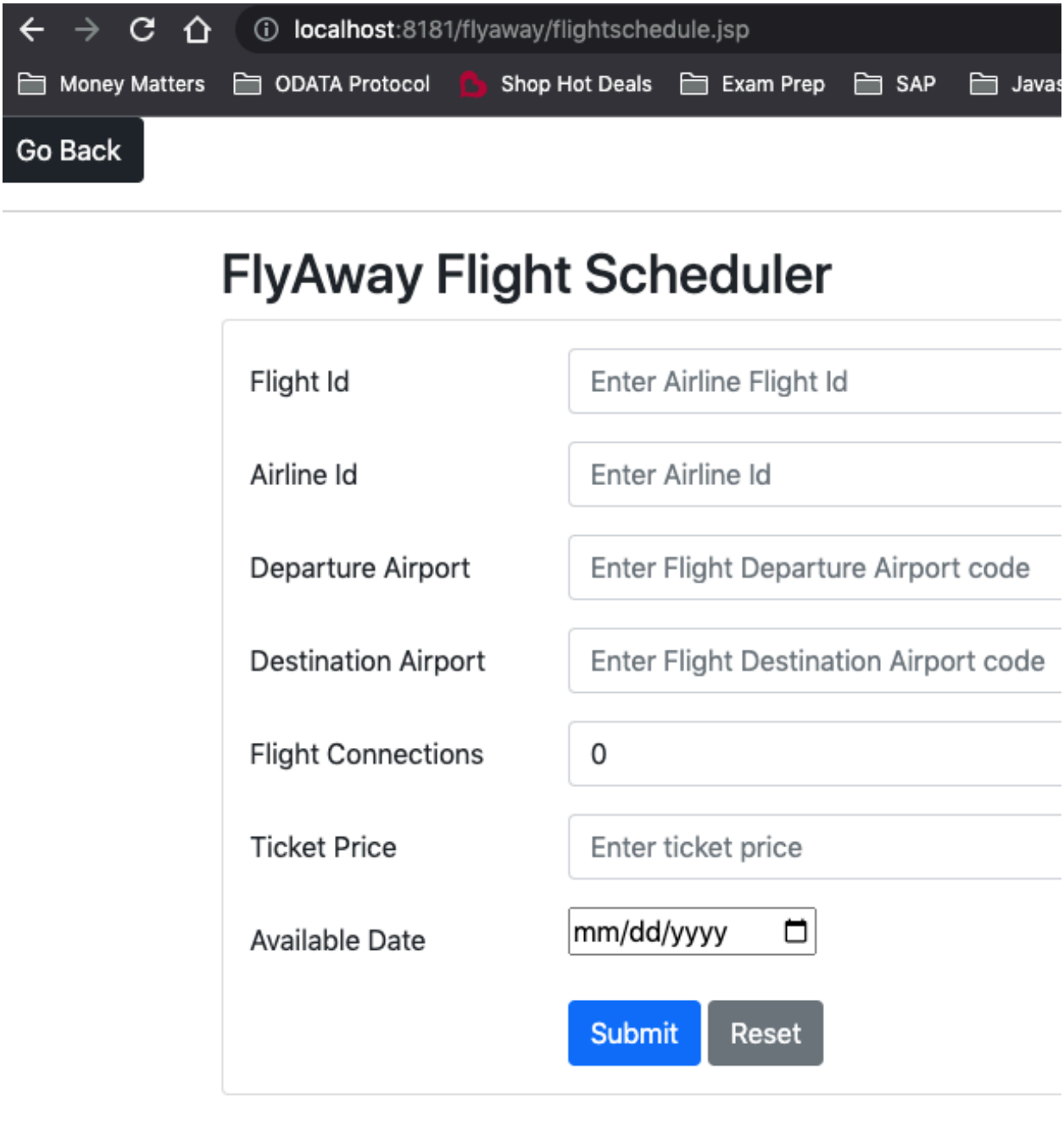
4. Airport Setup



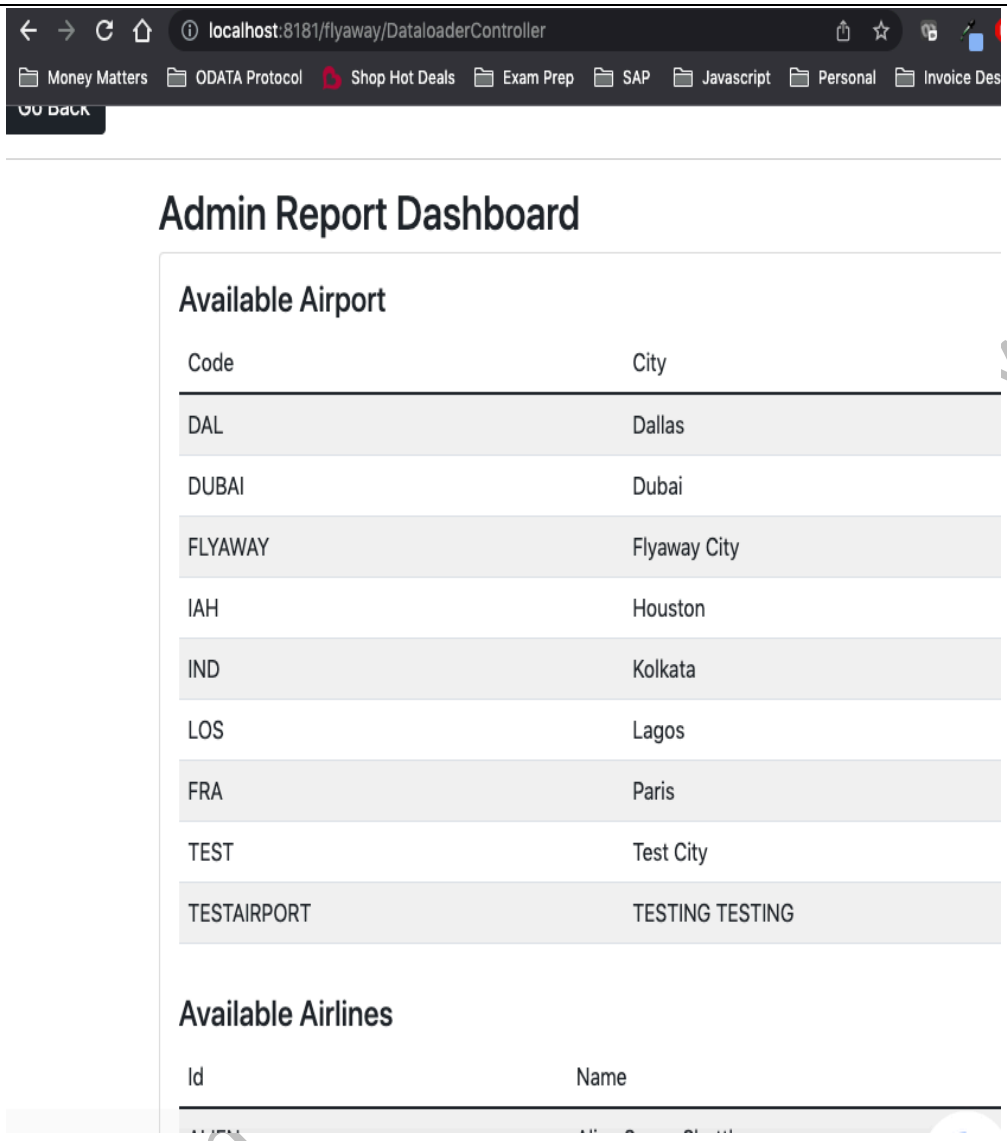
5. Airline Admin Setup



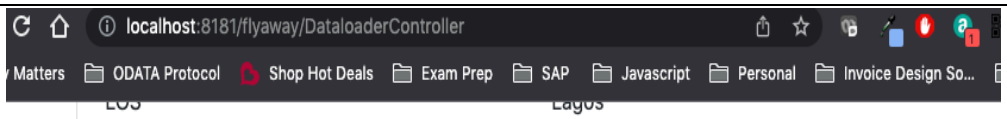
6. Admin Flight Scheduler



7. Admin Report Dashboard



Simplilearn



LOS	Lagos
FRA	Paris
TEST	Test City
TESTAIRPORT	TESTING TESTING

Available Airlines

Id	Name
ALIEN	Alien Space Shuttle
FAST	Very Fast Airline
RISKY	Risky Airways
TEST	Test World Airways
UNIQUE	Unique World Airways

Available Schedules

Flight Id	Airline Id	Departure Airport	Destination Airport	Available Date
flight001	RISKY	DAL	IAH	2022-07-14
flight002	UNIQUE	IAH	LOS	2022-07-14
flight003	RISKY	IAH	LOS	2022-07-14

Simplilearn

flight004	FAST	IAH	LOS	2022-07-14
flight005	RISKY	LOS	FRA	2022-07-21
flight006	ALIEN	DAL	IAH	2022-07-14
flight007	TEST	LOS	IAH	2022-07-14
flight008	TEST	IAH	LOS	2022-07-14

All FlyAway Bookings

Booking Reference	Flight Id	Passenger#	Booked By
EDOFSUHv8o	flight003	1	khlaeed@khameel.com

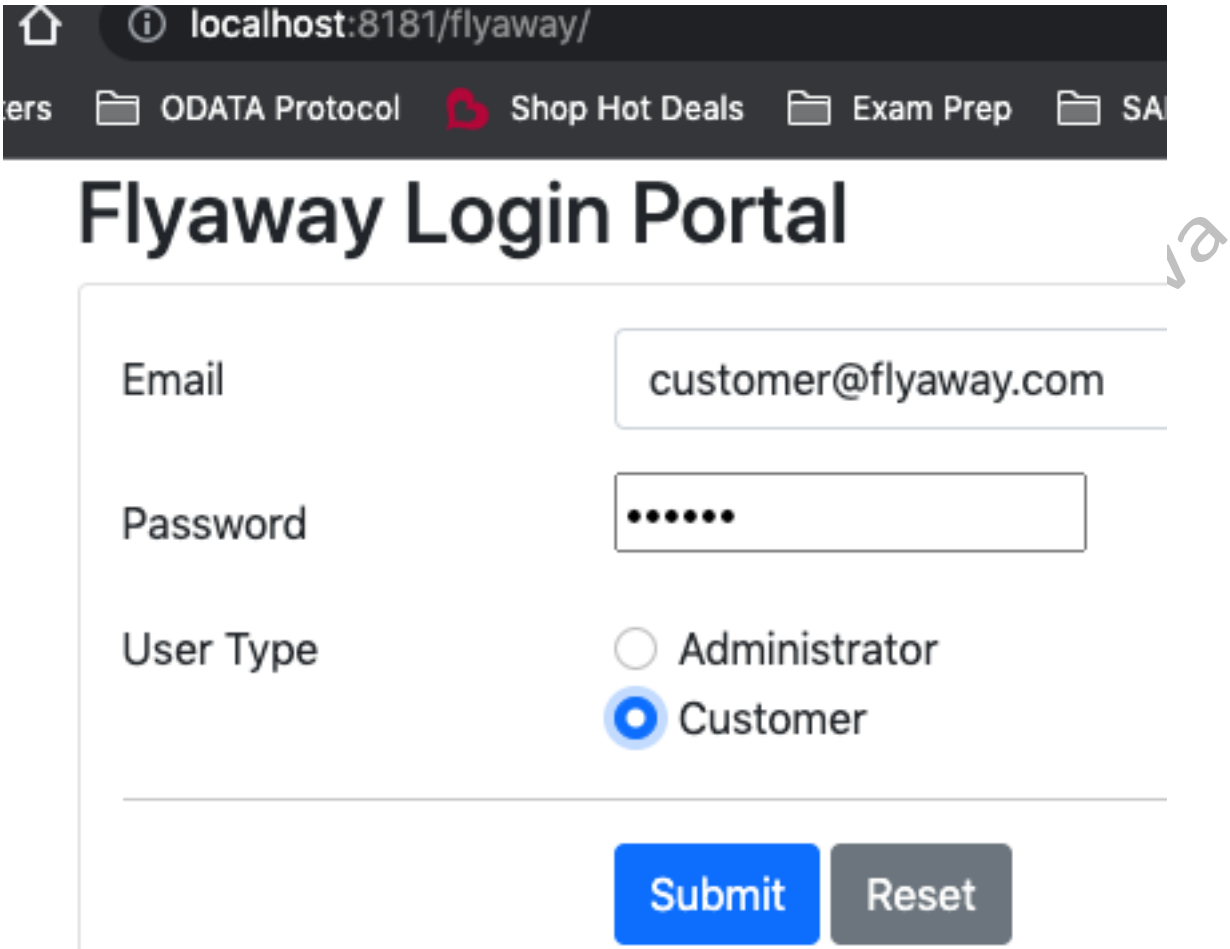
All Booked Passenger Details

Booking Id	Passenger Id	Firstname	Lastname
EDOFSUHv8o	KK001	Khaleed	Khameel

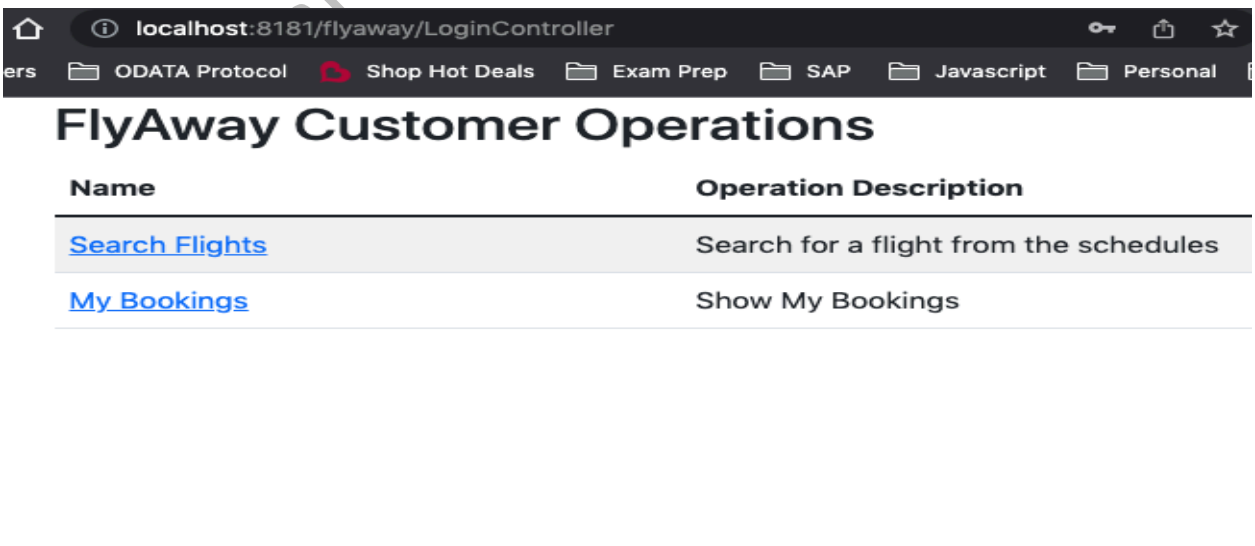
Admin Home

Simplilearn \ CalTe

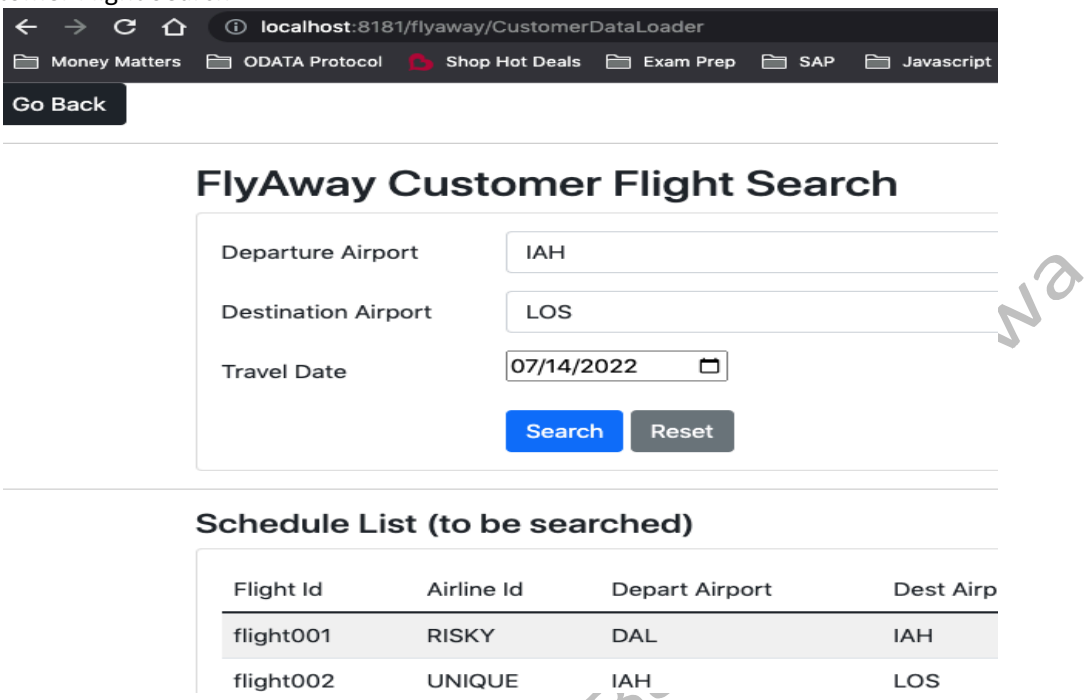
8. FlyAway Customer Login



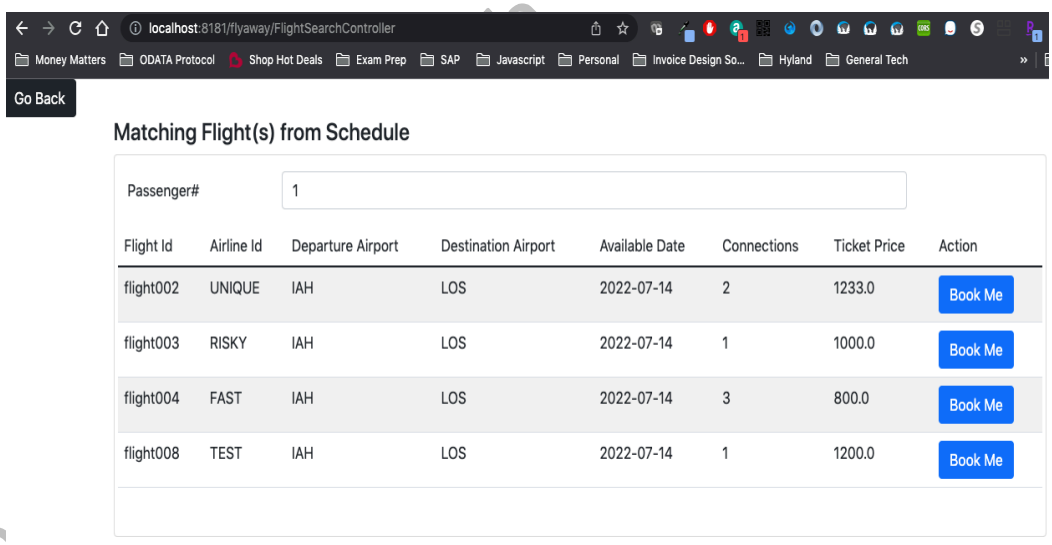
9. FlyAway Customer Operation



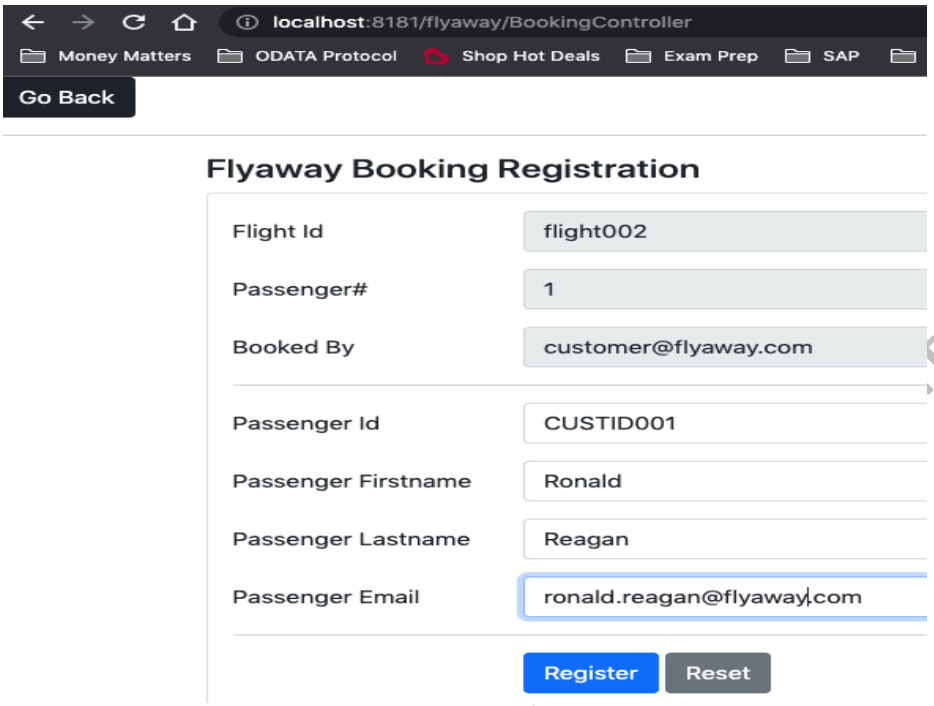
10. Customer Flight Search



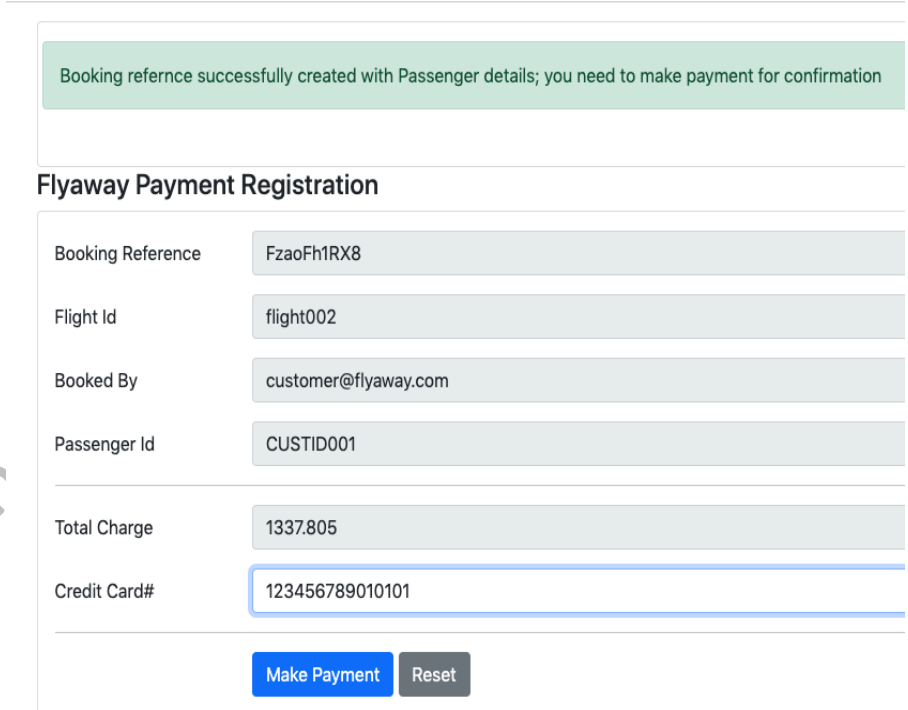
11. Flight Search Result



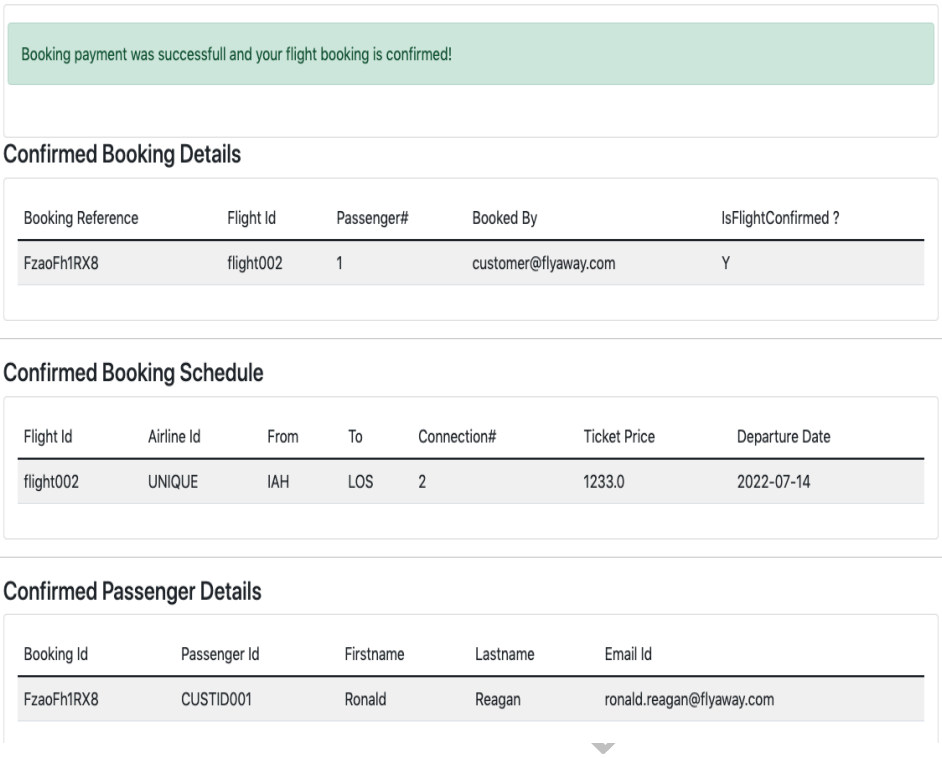
12. Customer Flight Booking Registration



13. Booking Reference / Payment



14. Payment / Booking confirmation



Booking payment was successfull and your flight booking is confirmed!

Confirmed Booking Details

Booking Reference	Flight Id	Passenger#	Booked By	IsFlightConfirmed ?
FzaoFh1RX8	flight002	1	customer@flyaway.com	Y

Confirmed Booking Schedule

Flight Id	Airline Id	From	To	Connection#	Ticket Price	Departure Date
flight002	UNIQUE	IAH	LOS	2	1233.0	2022-07-14

Confirmed Passenger Details

Booking Id	Passenger Id	Firstname	Lastname	Email Id
FzaoFh1RX8	CUSTID001	Ronald	Reagan	ronald.reagan@flyaway.com

Limitations

The application will only allow 1 passenger to be booked at a time against a booking Id (though the relationship between a booking entity and passengers is 1 to many).

Future enhancement is planned to refactor the UI to allow for multiple passengers' profile to applied to a booking Id.

Simplilearn \ CalTech \ Bakau Onafuwa