

FLIGHT RESERVATION APP

DATABASE

**CREATE DATABASE FLIGHT
USE FLIGHT**

**CREATE TABLE USER
(ID INT NOT NULL AUTO_INCREMENT,
FIRST_NAME VARCHAR (20),
LAST_NAME VARCHAR (20),
EMAIL VARCHAR (125),
PASSWORD VARCHAR (20),
PRIMARY KEY(ID),
UNIQUE KEY(EMAIL)
)
SELECT * FROM USER**

**CREATE TABLE FLIGHT
(ID INT NOT NULL AUTO_INCREMENT,
FLIGHT_NUMBER VARCHAR (20) NOT NULL,
OPERATING_AIRLINES VARCHAR (20) NOT NULL,
DEPARTURE_CITY VARCHAR (20) NOT NULL,
ARRIVAL_CITY VARCHAR (20) NOT NULL,
DATE_OF_DEPARTURE DATE NOT NULL,
ESTIMATED_DEPARTURE_TIME TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY(ID)
)
SELECT * FROM FLIGHT**

**CREATE TABLE PASSENGER
(ID INT NOT NULL AUTO_INCREMENT,
FIRST_NAME VARCHAR (20),
LAST_NAME VARCHAR (20),
MIDDLE_NAME VARCHAR (20),
EMAIL VARCHAR (125),
PHONE VARCHAR (10),
PRIMARY KEY(ID)
)
SELECT * FROM PASSENGER**

**CREATE TABLE RESERVATION
(ID INT NOT NULL AUTO_INCREMENT,
CHECKED_IN TINYINT (1),
NUMBER_OF_BAGS INT,
PASSENGER_ID INT,
FLIGHT_ID INT,
CREATED TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY(ID),**

```
FOREIGN KEY(PASSENGER_ID) REFERENCES PASSENGER(ID) ON DELETE CASCADE,  
FOREIGN KEY(FLIGHT_ID) REFERENCES FLIGHT(ID)  
)  
SELECT * FROM RESERVATION
```

```
INSERT INTO FLIGHT VALUES (1,'AA1','AMERICAN AIRLINES','AUS','NYC',  
STR_TO_DATE('02-05-2018','%m-%d-%Y'),'2018-02-05 03-14-07')
```

```
INSERT INTO FLIGHT VALUES (2,'AA2','AMERICAN AIRLINES','AUS','NYC',  
STR_TO_DATE('02-05-2018','%m-%d-%Y'),'2018-02-05 05-14-07')
```

```
INSERT INTO FLIGHT VALUES (3,'AA3','AMERICAN AIRLINES','AUS','NYC',  
STR_TO_DATE('02-05-2018','%m-%d-%Y'),'2018-02-05 06-14-07')
```

```
INSERT INTO FLIGHT VALUES (4,'SW1','SOWTH WEST','AUS','NYC',  
STR_TO_DATE('02-05-2018','%m-%d-%Y'),'2018-02-05 07-14-07')
```

```
INSERT INTO FLIGHT VALUES (5,'UA1','UNITED AIRLINES','NYC','DAL',  
STR_TO_DATE('02-05-2018','%m-%d-%Y'),'2018-02-05 10-14-07')
```

```
INSERT INTO FLIGHT VALUES (6,'UA1','UNITED AIRLINES','NYC','DAL',  
STR_TO_DATE('02-05-2018','%m-%d-%Y'),'2018-02-05 10-14-07')
```

```
INSERT INTO FLIGHT VALUES (7,'SW1','SOWTH WEST','AUS','NYC',  
STR_TO_DATE('02-06-2018','%m-%d-%Y'),'2018-02-06 07-14-07')
```

```
INSERT INTO FLIGHT VALUES (8,'SW2','SOUTH WEST','AUS','NYC',  
STR_TO_DATE('02-06-2018','%m-%d-%Y'),'2018-02-06 08-14-07')
```

```
INSERT INTO FLIGHT VALUES (9,'SW3','SOUTH WEST','NYC','DAL',  
STR_TO_DATE('02-06-2018','%m-%d-%Y'),'2018-02-06 10-14-07')
```

```
INSERT INTO FLIGHT VALUES (10,'UA1','UNITED AIRLINES','NYC','DAL',  
STR_TO_DATE('02-06-2018','%m-%d-%Y'),'2018-02-06 10-14-07')
```

POM.XML FILES

ALL DEPENDENCIES

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.4</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.FlightReservationapp</groupId>
  <artifactId>FlightReservationapp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>FlightReservationapp</name>
  <description>A Flight Reservation App</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.apache.tomcat.embed</groupId>
      <artifactId>tomcat-embed-jasper</artifactId>
      <scope>provided</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>
    <dependency>
      <groupId>com.itextpdf</groupId>
```

```

        <artifactId>itextpdf</artifactId>
        <version>5.5.13.2</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-mail</artifactId>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

APPLICATION.PROPERTIES FILE

#DB Connection

spring.datasource.url=jdbc:mysql://localhost:3306/flight

spring.datasource.username=root

spring.datasource.password=lshdwivedi#96@23

This line will automatically create table in database and when you restart

server table will be dropped and created again

spring.jpa.hibernate.ddl-auto=update

#Suffix & Prefix

spring.mvc.view.prefix=/WEB-INF/jsp/

spring.mvc.view.suffix=.jsp

server.servlet.context-path=/flights

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.username=ishaandemo@gmail.com

spring.mail.password=atmatygtvoonfqqb

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.starttls.enable=true

ENTITY LAYERS

ABSTRACT ENTITY

```
package com.FlightReservationapp.entities;

import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.MappedSuperclass;

@MappedSuperclass
public class AbstractEntity {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
}
```

FLIGHT ENTITY

```
package com.FlightReservationapp.entities;

import java.sql.Timestamp;
import java.util.Date;

import javax.persistence.Entity;

@Entity
public class Flight extends AbstractEntity {

    private String flightNumber;
    private String operatingAirlines;
    private String departureCity;
    private String arrivalCity;
    private Date dateOfDeparture;
    private Timestamp estimatedDepartureTime;

    public String getFlightNumber() {
        return flightNumber;
    }

    public void setFlightNumber(String flightNumber) {
        this.flightNumber = flightNumber;
    }

    public String getOperatingAirlines() {
        return operatingAirlines;
    }
}
```

```

    public void setOperatingAirlines(String operatingAirlines) {
        this.operatingAirlines = operatingAirlines;
    }

    public String getDepartureCity() {
        return departureCity;
    }

    public void setDepartureCity(String departureCity) {
        this.departureCity = departureCity;
    }

    public String getArrivalCity() {
        return arrivalCity;
    }

    public void setArrivalCity(String arrivalCity) {
        this.arrivalCity = arrivalCity;
    }

    public Date getDateOfDeparture() {
        return dateOfDeparture;
    }

    public void setDateOfDeparture(Date dateOfDeparture) {
        this.dateOfDeparture = dateOfDeparture;
    }

    public Timestamp getEstimatedDepartureTime() {
        return estimatedDepartureTime;
    }

    public void setEstimatedDepartureTime(Timestamp estimatedDepartureTime) {
        this.estimatedDepartureTime = estimatedDepartureTime;
    }
}

```

PASSENGER ENTITY

```

package com.FlightReservationapp.entities;

```

```

import javax.persistence.Entity;

```

```

@Entity

```

```

public class Passenger extends AbstractEntity {
    private String firstName;
    private String lastName;
    private String middleName;
    private String email;
    private String phone;

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
}

```

```

    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getMiddleName() {
        return middleName;
    }

    public void setMiddleName(String middleName) {
        this.middleName = middleName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }
}

```

RESERVATION ENTITY

```

package com.FlightReservationapp.entities;

```

```

import javax.persistence.Entity;
import javax.persistence.OneToOne;

```

```

@Entity

```

```

public class Reservation extends AbstractEntity {
    private boolean checkedIn;
    private long numberOfBags;
    @OneToOne
    private Passenger passenger;
    @OneToOne
    private Flight flight;

    public boolean isCheckedIn() {
        return checkedIn;
    }

    public void setCheckedIn(boolean checkedIn) {
        this.checkedIn = checkedIn;
    }
}

```

```

    }

    public long getNumberOfBags() {
        return numberOfBags;
    }

    public void setNumberOfBags(long numberOfBags) {
        this.numberOfBags = numberOfBags;
    }

    public Passenger getPassenger() {
        return passenger;
    }

    public void setPassenger(Passenger passenger) {
        this.passenger = passenger;
    }

    public Flight getFlight() {
        return flight;
    }

    public void setFlight(Flight flight) {
        this.flight = flight;
    }
}

```

USER ENTITY

package com.FlightReservationapp.entities;

import javax.persistence.Entity;

@Entity

```

public class User extends AbstractEntity {

    private String firstName;
    private String lastName;
    private String email;
    private String password;

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}

```



```

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}

```

REPOSITORY LAYERS

FLIGHT REPOSITORY

```

package com.FlightReservationapp.repositories;

```

```

import java.util.Date;
import java.util.List;

```

```

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

```

```

import com.FlightReservationapp.entities.Flight;

```

```

public interface FlightRepository extends JpaRepository<Flight, Long> {

```

```

    @Query("from Flight where departureCity=:departureCity and arrivalCity=:arrivalCity and dateOfDeparture=:dateOfDeparture")
    List<Flight> findFlights(@Param("departureCity") String from, @Param("arrivalCity") String to,
        @Param("dateOfDeparture") Date departureDate);
}

```

PASSENGER REPOSITORY

```

package com.FlightReservationapp.repositories;

```

```

import org.springframework.data.jpa.repository.JpaRepository;

```

```

import com.FlightReservationapp.entities.Passenger;

```

```

public interface PassengerRepository extends JpaRepository<Passenger, Long> {
    }

```

RESERVATION REPOSITORY

```
package com.FlightReservationapp.repositories;

import org.springframework.data.jpa.repository.JpaRepository;

import com.FlightReservationapp.entities.Reservation;

public interface ReservationRepository extends JpaRepository<Reservation, Long> {

}
```

USER REPOSITORY

```
package com.FlightReservationapp.repositories;

import org.springframework.data.jpa.repository.JpaRepository;

import com.FlightReservationapp.entities.User;

public interface UserRepository extends JpaRepository<User, Long> {

    User findByEmail(String emailId);

}
```

DTO {DATA TRANSFER OBJECT} LAYER

RESERVATION REQUEST

```
package com.FlightReservationapp.dto;

public class ReservationRequest {

    private long flightId;
    private String firstName;
    private String lastName;
    private String middleName;
    private String email;
    private String phone;
    private String nameOfTheCard;
    private String cardNumber;
    private String cvv;
    private String expiryDate;
    public long getFlightId() {
        return flightId;
    }
    public void setFlightId(long flightId) {
        this.flightId = flightId;
    }

}
```

```
public String getFirstName() {  
    return firstName;  
}  
public void setFirstName(String firstName) {  
    this.firstName = firstName;  
}  
public String getLastName() {  
    return lastName;  
}  
public void setLastName(String lastName) {  
    this.lastName = lastName;  
}  
public String getMiddleName() {  
    return middleName;  
}  
public void setMiddleName(String middleName) {  
    this.middleName = middleName;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}  
public String getPhone() {  
    return phone;  
}  
public void setPhone(String phone) {  
    this.phone = phone;  
}  
public String getNameOfTheCard() {  
    return nameOfTheCard;  
}  
public void setNameOfTheCard(String nameOfTheCard) {  
    this.nameOfTheCard = nameOfTheCard;  
}  
public String getCardNumber() {  
    return cardNumber;  
}  
public void setCardNumber(String cardNumber) {  
    this.cardNumber = cardNumber;  
}  
public String getCvv() {  
    return cvv;  
}  
public void setCvv(String cvv) {  
    this.cvv = cvv;  
}  
public String getExpiryDate() {  
    return expiryDate;  
}  
public void setExpiryDate(String expiryDate) {  
    this.expiryDate = expiryDate;  
}  
}
```

RESERVATION UPDATE REQUEST

package com.FlightReservationapp.dto;

```
public class ReservationUpdateRequest {  
  
    private long id;  
    private boolean checkedIn;  
    private int numberOfBags;  
    public long getId() {  
        return id;  
    }  
    public void setId(long id) {  
        this.id = id;  
    }  
    public boolean isCheckedIn() {  
        return checkedIn;  
    }  
    public void setCheckedIn(boolean checkedIn) {  
        this.checkedIn = checkedIn;  
    }  
    public int getNumberOfBags() {  
        return numberOfBags;  
    }  
    public void setNumberOfBags(int numberOfBags) {  
        this.numberOfBags = numberOfBags;  
    }  
}
```

SERVICES LAYERS

RESERVATION SERVICE(INTERFACE)

package com.FlightReservationapp.services;

import com.FlightReservationapp.dto.ReservationRequest;
import com.FlightReservationapp.entities.Reservation;

```
public interface ReservationService {  
  
    Reservation bookFlight(ReservationRequest request);  
}
```

RESERVATION SERVICE IMPL(CLASS)

package com.FlightReservationapp.services;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.FlightReservationapp.dto.ReservationRequest;
import com.FlightReservationapp.entities.Flight;

```
import com.FlightReservationapp.entities.Passenger;
import com.FlightReservationapp.entities.Reservation;
import com.FlightReservationapp.repositories.FlightRepository;
import com.FlightReservationapp.repositories.PassengerRepository;
import com.FlightReservationapp.repositories.ReservationRepository;
import com.FlightReservationapp.utilities.EmailUtil;
import com.FlightReservationapp.utilities.PDFgenerator;
```

```
@Service
```

```
public class ReservationServiceImpl implements ReservationService {
```

```
    @Autowired
```

```
    private PassengerRepository passengerRepo;
```

```
    @Autowired
```

```
    private FlightRepository flightRepo;
```

```
    @Autowired
```

```
    private ReservationRepository reservationRepo;
```

```
    @Autowired
```

```
    private PDFgenerator pdfGenerator;
```

```
    @Autowired
```

```
    private EmailUtil emailUtil;
```

```
    @Override
```

```
    public Reservation bookFlight(ReservationRequest request) {
```

```
        Passenger passenger = new Passenger();
        passenger.setFirstName(request.getFirstName());
        passenger.setLastName(request.getLastName());
        passenger.setMiddleName(request.getMiddleName());
        passenger.setEmail(request.getEmail());
        passenger.setPhone(request.getPhone());
        passengerRepo.save(passenger);
```

```
        long flightId = request.getFlightId();
        Optional<Flight> findById = flightRepo.findById(flightId);
        Flight flight = findById.get();
```

```
        Reservation reservation = new Reservation();
        reservation.setFlight(flight);
        reservation.setPassenger(passenger);
        reservation.setCheckedIn(false);
        reservation.setNumberOfBags(0);
```

```
        String filePath = "C:\\ishaan\\sts 14\\FlightReservationapp\\tickets\\reservation" +
reservation.getId()
                + ".pdf";
```

```
        reservationRepo.save(reservation);
```

```
        pdfGenerator.generateItinerary(reservation, filePath);
        emailUtil.sendItinerary(passenger.getEmail(), filePath);
        return reservation;    }
```

CONTROLLER LAYERS

FLIGHT CONTROLLER

```
package com.FlightReservationapp.controller;
```

```
import java.util.Date;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.format.annotation.DateTimeFormat;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.FlightReservationapp.entities.Flight;
```

```
import com.FlightReservationapp.repositories.FlightRepository;
```

```
@Controller
```

```
public class FlightController {
```

```
    @Autowired
```

```
    private FlightRepository flightRepo;
```

```
    @RequestMapping("/findFlights")
```

```
    public String findFlights(@RequestParam("from") String from, @RequestParam("to") String to,  
        @RequestParam("departureDate") @DateTimeFormat(pattern = "MM-dd-yyyy") Date
```

```
departureDate, ModelMap model) {
```

```
        List<Flight> findFlights = flightRepo.findFlights(from , to , departureDate);
```

```
        model.addAttribute("findFlights", findFlights);
```

```
        return "displayFlights";
```

```
    }
```

```
    @RequestMapping("/showCompleteReservation")
```

```
    public String showCompleteReservation(@RequestParam("flightId") Long flightId, ModelMap  
model) {
```

```
        Optional<Flight> findById = flightRepo.findById(flightId);
```

```
        Flight flight = findById.get();
```

```
        model.addAttribute("flight", flight);
```

```
        return "showReservation";
```

```
    }
```

```
}
```

PASSENGER CONTROLLER

```
package com.FlightReservationapp.controller;
```

```
import org.springframework.stereotype.Controller;
```

```
@Controller
```

```
public class PassengerController {}
```

RESERVATION CONTROLLER

```
package com.FlightReservationapp.controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import com.FlightReservationapp.dto.ReservationRequest;
import com.FlightReservationapp.entities.Reservation;
import com.FlightReservationapp.services.ReservationService;
```

```
@Controller
```

```
public class ReservationController {
    @Autowired
    private ReservationService reservationService;

    @RequestMapping("/confirmReservation")
    public String confirmReservation(ReservationRequest request, ModelMap model) {
        Reservation reservationId = reservationService.bookFlight(request);
        model.addAttribute("reservationId", reservationId.getId());

        return "confirmReservation";
    }
}
```

RESERVATION REST CONTROLLER

```
package com.FlightReservationapp.controller;
```

```
import java.util.Optional;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
import com.FlightReservationapp.dto.ReservationUpdateRequest;
import com.FlightReservationapp.entities.Reservation;
import com.FlightReservationapp.repositories.ReservationRepository;
```

```
@RestController
```

```
public class RreservationRestController {

    @Autowired
    private ReservationRepository reservationRepo;

    @RequestMapping("/reservation/{id}")
    public Reservation findReservation(@PathVariable("id") Long id) {
        Optional<Reservation> findById = reservationRepo.findById(id);
        Reservation reservation = findById.get();
        return reservation;
    }

    @RequestMapping("/reservation")
    public Reservation updateReservation(@RequestBody ReservationUpdateRequest request) {
        Optional<Reservation> findById = reservationRepo.findById(request.getId());
```

```

        Reservation reservation = findByld.get();
        reservation.setCheckedIn(request.isCheckedIn());
        reservation.setNumberOfBags(request.getNumberOfBags());
        return reservationRepo.save(reservation);
    }
}

```

USER CONTROLLER

```
package com.FlightReservationapp.controller;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

```

```

import com.FlightReservationapp.entities.User;
import com.FlightReservationapp.repositories.UserRepository;

```

```
@Controller
```

```
public class UserController {
```

```
    @Autowired
```

```
    private UserRepository userRepo;
```

```
    @RequestMapping("/showLoginPage")
```

```
    public String showLoginPage() {
        return "login/login";
    }

```

```
    @RequestMapping("/showReg")
```

```
    public String showReg() {
        return "login/showReg";
    }

```

```
    @RequestMapping("/saveReg")
```

```
    public String saveReg(@ModelAttribute("user") User user) {
        userRepo.save(user);
        return "login/login";
    }

```

```
    @RequestMapping("/verifyLogin")
```

```
    public String verifyLogin(@RequestParam("emailId") String emailId, @RequestParam("password")
String password , ModelMap model) {
        User user = userRepo.findByEmail(emailId);
        if(user!=null) {
            if(user.getEmail().equals(emailId) && user.getPassword().equals(password)) {
                return "findFlights";
            }else {
                model.addAttribute("error", "Invalid username/password");
                return "login/login";
            }
        }else {
            model.addAttribute("error", "Invalid username/password");
            return "login/login";
        }
    }
}

```



```
    }  
}  
}
```

UTILITIES LAYERS

EMAIL UTILITY

```
package com.FlightReservationapp.utilities;
```

```
import org.springframework.stereotype.Component;
```

```
import java.io.File;
```

```
import javax.mail.MessagingException;
```

```
import javax.mail.internet.MimeMessage;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.mail.javamail.JavaMailSender;
```

```
import org.springframework.mail.javamail.MimeMessageHelper;
```

```
@Component
```

```
public class EmailUtil {
```

```
    @Autowired
```

```
    private JavaMailSender sender;
```

```
    public void sendItinerary(String toAddress , String filePath) {
```

```
        MimeMessage message = sender.createMimeMessage();
```

```
        try {
```

```
            MimeMessageHelper messageHelper = new MimeMessageHelper(message ,
```

```
            true);
```

```
                messageHelper.setTo(toAddress);
```

```
                messageHelper.setSubject("ITINERARY OF FLIGHT");
```

```
                messageHelper.setText("Please Find The Attached");
```

```
                messageHelper.addAttachment("Itinerary", new File(filePath));
```

```
                sender.send(message);
```

```
        }catch(MessagingException e) {
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```

PDF GENERATOR

```
package com.FlightReservationapp.utilities;
```

```
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
```

```
import org.springframework.stereotype.Component;
```

```
import com.FlightReservationapp.entities.Reservation;
```

```
import com.itextpdf.text.Document;
```

```
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;
```

```
@Component
```

```
public class PDFgenerator {
```

```
    public void generateItinerary(Reservation reservation, String filePath) {
        Document document = new Document();
        try {
            PdfWriter.getInstance(document, new FileOutputStream(filePath));
            document.open();
            document.add(generateTable(reservation));
            document.close();
        } catch (FileNotFoundException | DocumentException e) {
            e.printStackTrace();
        }
    }
}
```

```
    private PdfPTable generateTable(Reservation reservation) {

        PdfPTable table = new PdfPTable(2);
        PdfPCell cell;

        cell = new PdfPCell(new Phrase("FLIGHT ITENERARY"));
        cell.setColspan(2);
        table.addCell(cell);

        cell = new PdfPCell(new Phrase("FLIGHT DETAILS"));
        cell.setColspan(2);
        table.addCell(cell);

        table.addCell("Departure City");
        table.addCell(reservation.getFlight().getDepartureCity());

        table.addCell("Arrival City");
        table.addCell(reservation.getFlight().getArrivalCity());

        table.addCell("Flight Number");
        table.addCell(reservation.getFlight().getFlightNumber());

        table.addCell("Operating Airlines");
        table.addCell(reservation.getFlight().getOperatingAirlines());

        table.addCell("Departure Date");
        table.addCell(reservation.getFlight().getDateOfDeparture().toString());

        table.addCell("Departure Time");
        table.addCell(reservation.getFlight().getEstimatedDepartureTime().toString());

        cell = new PdfPCell(new Phrase("PASSENGER DETAILS"));
        cell.setColspan(2);
        table.addCell(cell);
    }
}
```

```

        table.addCell("First Name");
        table.addCell(reservation.getPassenger().getFirstName());

        table.addCell("Last Name");
        table.addCell(reservation.getPassenger().getLastName());

        table.addCell("Middle Name");
        table.addCell(reservation.getPassenger().getMiddleName());

        table.addCell("Email");
        table.addCell(reservation.getPassenger().getEmail());

        table.addCell("Mobile Number");
        table.addCell(reservation.getPassenger().getPhone());

        return table;
    }
}

```

VIEW LAYERS

UNDER WEBAPP FOLDER CREATE INDEX.HTML PAGE

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>HOME PAGE</title>
</head>
<body>

    <a href="showLoginPage">Click Here To Login</a><br/>

    <a href="showReg">Click Here To Create New Account</a>
</body>
</html>

```

UNDER WEB-INF/JSPS/LOGIN CREATE LOGIN.JSP

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>LOGIN</title>
</head>
<body>
    <h2>LOGIN HERE...</h2>
    <form action="verifyLogin" method="post">
        <pre>
            EMAILID <input type="text" name="emailId"/>
            PASSWORD<input type="text" name="password"/>
            <input type="submit" value="login">

```

```
        </pre>
    </form>
    ${error}
</body>
</html>
```

UNDER WEB-INF/JSPS/LOGIN CREATE SHOWREG.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>NEW REGISTRATION</title>
</head>
<body>
    <h2>CREATE NEW ACCOUNT</h2>
    <form action="saveReg" method = "post">
        <pre>
            FIRST NAME <input type = "text" name = "firstName"/>
            LAST NAME <input type = "text" name = "lastName"/>
            EMAIL <input type = "text" name = "email"/>
            PASSWORD <input type = "text" name = "password"/>
            <input type = "submit" value = "save"/>
        </pre>
    </form>
</body>
</html>
```

UNDER JSPS FOLDER CREATE CONFIRM RESERVATION.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>CONFIRMATION</title>
</head>
<body>
    Your Ticket Is Booked . Your Reservation Id is :${reservationId}
</body>
</html>
```

UNDER JSPS FOLDER CREATE DISPLAY FLIGHTS.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>DISPLAY FLIGHTS</title>
```

```

</head>
<body>
  <h2>FLIGHT SEARCH RESULTS</h2>
  <h3>LIST OF FLIGHTS</h3>
  <table border="5">
    <tr>
    <tr>
      <th>AIRLINES</th>
      <th>DEPARTURE CITY</th>
      <th>ARRIVAL CITY</th>
      <th>DEPARTURE TIME</th>
      <th>SELECT FLIGHT</th>
    </tr>
    <c:forEach var="findFlights" items="${findFlights}">
      <tr>
        <td>${findFlights.operatingAirlines}</td>
        <td>${findFlights.departureCity}</td>
        <td>${findFlights.arrivalCity}</td>
        <td>${findFlights.estimatedDepartureTime}</td>
        <td><a
href="showCompleteReservation?flightId=${findFlights.id}">Select</a></td>
      </tr>
    </c:forEach>
  </table>
</body>
</html>

```

UNDER JSPS FOLDER CREATE DISPLAY FLIGHTS.JSP

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SEARCH FLIGHTS</title>
</head>
<body>
  <h2>FIND FLIGHTS</h2>
  <form action="findFlights" method="post">
    <pre>
      FROM:    <input type="text" name="from"/>
      TO:      <input type="text" name="to"/>
      DEPARTURE DATE:<input type="text" name="departureDate"/>
      <input type="Submit" value="Search"/>
    </pre>
  </form>
</body>
</html>

```

UNDER JSPS FOLDER CREATE SHOW RESERVATION.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>RESERVATION DETAILS</title>
</head>
<body>
    <h2>FLIGHT DETAILS</h2>
    FLIGHT NUMBER      :${flight.flightNumber}      <br/>
    OPERATING AIRLINES  :${flight.operatingAirlines}<br/>
    DEPARTURE CITY      :${flight.departureCity}      <br/>
    ARRIVAL CITY        :${flight.arrivalCity}        <br/>
    DEPARTURE DATE      :${flight.dateOfDeparture}    <br/>

    <h2>ENTER PASSENGER DETAILS</h2>
    <form action="confirmReservation" method="post">
        <pre>
            FIRST NAME <input type = "text" name = "firstName"/>
            LAST NAME  <input type = "text" name = "lastName"/>
            MIDDLE NAME <input type = "text" name = "middleName"/>
            EMAIL       <input type = "text" name = "email"/>
            PHONE       <input type = "text" name = "phone"/>
            <input type = "hidden" name = "flightId" value = "${flight.id}"/>
        </pre>
    <h2>ENTER THE PAYMENT DETAILS</h2>
        NAME ON THE CARD<input type = "text" name = "nameOfTheCard"/>
        CARD NUMBER    <input type = "text" name = "cardNumber"/>
        CVV            <input type = "text" name = "cvv"/>
        EXPIRY DATE    <input type = "text" name = "expiryDate"/>
        <input type = "submit" value = "Complete Reservation"/>
    </pre>
</form>
</body>
</html>
```

CHECKIN MODULE

THIS MODULE IS FOR DOING MICROSERVICES AND FOR CONNECTING THIS MODULE TO FLIGHT RESERVATION MODULE

CONTROLLER LAYER

RESERVATION CONTROLLER

```
package com.CheckIn.controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.CheckIn.integration.ReservationRestfulClient;
import com.CheckIn.integration.dto.Reservation;
import com.CheckIn.integration.dto.ReservationUpdateRequest;
```

```
@Controller
```

```
public class ReservationController {
```

```
    @Autowired
```

```
    private ReservationRestfulClient restClient;
```

```
    @RequestMapping("/startCheckIn")
```

```
    public String startCheckIn() {
        return "startCheckIn";
    }
```

```
    @RequestMapping("/proceedCheckIn")
```

```
    public String proceedCheckIn(@RequestParam("id") Long id, ModelMap model) {
        Reservation reservation = restClient.findReservation(id);
        model.addAttribute("reservation", reservation);
        return "displayResrvation";
    }
```

```
    @RequestMapping("/proceedToCheckIn")
```

```
    public String proceedToCheckIn(@RequestParam("id") Long id,
    @RequestParam("numberOfBags") int numberOfBags) {
```

```
        ReservationUpdateRequest request = new ReservationUpdateRequest();
        request.setId(id);
        request.setNumberOfBags(numberOfBags);
        request.setCheckInStatus(true);
```

```
        restClient.updateReservation(request);
```

```
        return "confirmReservation";
```

```
    }
}
```

INTEGRATION LAYERS

RESERVATION RESTFUL CLIENT(INTERFACE)

```
package com.CheckIn.integration;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import com.CheckIn.integration.dto.Reservation;
```

```
import com.CheckIn.integration.dto.ReservationUpdateRequest;
```

```
public interface ReservationRestfulClient {  
    public Reservation findReservation(Long id);  
    public Reservation updateReservation(@RequestBody ReservationUpdateRequest request);  
}
```

RESERVATION RESTFUL CLIENT IMPL(CLASS)

```
package com.CheckIn.integration;
```

```
import org.springframework.stereotype.Component;
```

```
import org.springframework.web.client.RestTemplate;
```

```
import com.CheckIn.integration.dto.Reservation;
```

```
import com.CheckIn.integration.dto.ReservationUpdateRequest;
```

```
@Component
```

```
public class ReservationRestfulClientImpl implements ReservationRestfulClient {
```

```
    @Override
```

```
    public Reservation findReservation(Long id) {
```

```
        RestTemplate restTemplate = new RestTemplate();
```

```
        Reservation reservation =
```

```
restTemplate.getForObject("http://localhost:8080/flights/reservation/" + id,  
                        Reservation.class);
```

```
        return reservation;
```

```
    }
```

```
    @Override
```

```
    public Reservation updateReservation(ReservationUpdateRequest request) {
```

```
        RestTemplate restTemplate = new RestTemplate();
```

```
        Reservation reservation =
```

```
restTemplate.postForObject("http://localhost:8080/flights/reservation", request,  
                        Reservation.class);
```

```
        return reservation;
```

```
    }
```

```
}
```


INTEGRATION.DTO LAYERS

FLIGHT.JAVA

package com.CheckIn.integration.dto;

import java.sql.Timestamp;

import java.util.Date;

public class Flight {

private long id;

private String flightNumber;

private String operatingAirlines;

private String departureCity;

private String arrivalCity;

private Date dateOfDeparture;

private Timestamp estimatedDepartureTime;

public long getId() {

return id;

 }

public void setId(**long** id) {

this.id = id;

 }

public String getFlightNumber() {

return flightNumber;

 }

public void setFlightNumber(String flightNumber) {

this.flightNumber = flightNumber;

 }

public String getOperatingAirlines() {

return operatingAirlines;

 }

public void setOperatingAirlines(String operatingAirlines) {

this.operatingAirlines = operatingAirlines;

 }

public String getDepartureCity() {

return departureCity;

 }

public void setDepartureCity(String departureCity) {

this.departureCity = departureCity;

 }

public String getArrivalCity() {

return arrivalCity;

 }

public void setArrivalCity(String arrivalCity) {

this.arrivalCity = arrivalCity;

 }

public Date getDateOfDeparture() {

return dateOfDeparture;

 }

public void setDateOfDeparture(Date dateOfDeparture) {

this.dateOfDeparture = dateOfDeparture;

 }

public Timestamp getEstimatedDepartureTime() {

return estimatedDepartureTime;

 }

public void setEstimatedDepartureTime(Timestamp estimatedDepartureTime) {

```
        this.estimatedDepartureTime = estimatedDepartureTime;
    }
}
```

PASSENGER.JAVA

package com.CheckIn.integration.dto;

```
public class Passenger {

    private long id;
    private String firstName;
    private String lastName;
    private String middleName;
    private String email;
    private String phone;
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getMiddleName() {
        return middleName;
    }
    public void setMiddleName(String middleName) {
        this.middleName = middleName;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

RESERVATION.JAVA

package com.CheckIn.integration.dto;

```
public class Reservation {  
  
    private long id;  
    private boolean checkedIn;  
    private int NumberOfBags;  
    private Passenger passenger;  
    private Flight flight;  
    public long getId() {  
        return id;  
    }  
    public void setId(long id) {  
        this.id = id;  
    }  
    public boolean isCheckedIn() {  
        return checkedIn;  
    }  
    public void setCheckedIn(boolean checkedIn) {  
        this.checkedIn = checkedIn;  
    }  
    public int getNumberOfBags() {  
        return NumberOfBags;  
    }  
    public void setNumberOfBags(int numberOfBags) {  
        NumberOfBags = numberOfBags;  
    }  
    public Passenger getPassenger() {  
        return passenger;  
    }  
    public void setPassenger(Passenger passenger) {  
        this.passenger = passenger;  
    }  
    public Flight getFlight() {  
        return flight;  
    }  
    public void setFlight(Flight flight) {  
        this.flight = flight;  
    }  
}
```

RESERVATION UPDATE REQUEST.JAVA

package com.CheckIn.integration.dto;

```
public class ReservationUpdateRequest {  
  
    private Long id;  
    private int numberOfBags;  
    private boolean checkInStatus;  
    public Long getId() {  
        return id;  
    }  
    public void setId(Long id) {
```

```

        this.id = id;
    }
    public int getNumberOfBags() {
        return numberOfBags;
    }
    public void setNumberOfBags(int numberOfBags) {
        this.numberOfBags = numberOfBags;
    }
    public boolean isCheckInStatus() {
        return checkInStatus;
    }
    public void setCheckInStatus(boolean checkInStatus) {
        this.checkInStatus = checkInStatus;
    }
}

```

APPLICATION.PROPERTIES FILE

#Suffix & Prefix

```

spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp

```

```

server.servlet.context-path=/checkin

```

```

server.port=9090

```

POM.XML FILE

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.4</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.CheckIn</groupId>
    <artifactId>CheckIn</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>CheckIn</name>
    <description>Flight CheckIn</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
    </dependencies>

```

```

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.apache.tomcat.embed</groupId>
            <artifactId>tomcat-embed-jasper</artifactId>
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>

```

VIEW LAYERS

CONFIRMRESERVATION.JSP

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <h2>BORDING IS COMPLETED</h2>
</body>
</html>

```

DISPLAY RESERVATION.JSP

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>RESERVATION DETAILS</title>
</head>
<body>
    <h2> RESERVATION DETAILS....</h2>
    <pre>
        Passenger Name:${reservation.getPassenger().firstName}<br/>
        Passenger Email:${reservation.getPassenger().email}<br/>
    </pre>

```

```
Passenger Phone:${reservation.getPassenger().phone}<br/>
Operating Airlines:${reservation.getFlight().operatingAirlines}<br/>
Flight Number:${reservation.getFlight().flightNumber}<br/>
Departure City:${reservation.getFlight().departureCity}<br/>
Arrival City:${reservation.getFlight().arrivalCity}<br/>
Date Of Departure:${reservation.getFlight().dateOfDeparture}<br/>
<h2>UPDTAE NO. OF BAGS & STATUS</h2>
<form action = "proceedToCheckIn" method = "post">
<pre>
    Reservation Id      <input type = "text" name = "id" value = "${reservation.id}"/>
    Number Of Bags:     <input type = "text" name = "numberOfBags"/>
    <input type = "submit" value = "Confirm"/>
</pre>
</form>
</pre>
</body>
</html>
```

START CHECKIN.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>START CHECK IN</title>
</head>
<body>
    <h2>CHECKIN</h2>
    <form action = "proceedCheckIn" method = "post">
        <pre>
            ENTER YOUR RESERVATION ID<input type = "text" name = "id"/>
            <input type = "submit" value = "Start"/>
        </pre>
    </form>
</body>
</html>
```

=====