# ICIN BANK

SOURCE CODE:

package com.admin.controller;

import java.util.List;

import org.apache.commons.mail.EmailException;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

import com.admin.model.ChequebookRequest;

import com.admin.model.User;

import com.admin.model.UserDisplay;

import com.admin.service.impl.AdminServiceImpl;

import com.admin.service.impl.MailServiceImpl;

@RestController

@CrossOrigin

public class AdminController {

@Autowired

private AdminServiceImpl service;

@Autowired

private MailServiceImpl mailService;

@GetMapping("user/{username}/features/{featureId}")

public void setUserFeatures(@PathVariable("username") String username, @PathVariable("featureId") int featureId) {

service.setUserFeatures(username, featureId);

}

@GetMapping("user/{username}/authorize")

public void authorizeUser(@PathVariable("username") String username) {

try {

service.authorizeUser(username);

//mailService.sendAuthorizedEmail(username);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@GetMapping("user/{username}/authorize/cancel")

public void cancelAuthorization(@PathVariable("username") String username) {

try {

//mailService.sendAuthorizeCancelEmail(username);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

service.cancelAuthorization(username);

}

@GetMapping("user/unauthorized/all")

public List<User> getAllUnauthorizedUsers()

{

return service.getAllUnauthorizedUsers();

}

@GetMapping("/user/all")

public List<UserDisplay> getAllUsers()

{

return service.getAllUsers();

}

// @GetMapping("/user/transfers/{account}")

// public List<Transfer> getTransactionList(@PathVariable("account") int account)

// {

// return service.getAllTransactions(account);

// }

@GetMapping("/chequebook/request/all")

public List<ChequebookRequest> getAllChequeBookRequests()

{

return service.getAllChequebookRequests();

}

@GetMapping("/user/{accNo}/chequebook/request/confirm")

public void confirmChequeBookRequest(@PathVariable("accNo") long accNo)

{

service.acceptChequebookRequest(accNo);

}

@GetMapping("/user/{username}/enable")

public void enableUser(@PathVariable("username") String username)

{

service.enableUser(username);

}

@GetMapping("/user/{username}/disable")

public void disableUser(@PathVariable("username") String username)

{

service.disableUser(username);

}

@GetMapping("search/user/{userDetail}")

public UserDisplay searchUser(@PathVariable("userDetail") String userDetail) {

return service.searchUser(userDetail);

}

}

----------------------------------------------------------------------------------------

package com.admin.dao;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.Account;

@Repository

public interface AccountRepository extends CrudRepository<Account, Integer>{

public Account findByUsername(String username);

public Account findByAccno(long accno);

}

------------------------------------------------------------------------------------------------------------

package com.admin.dao;

import java.util.List;

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

import org.springframework.transaction.annotation.Transactional;

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

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.ChequebookRequest;

@Repository

public interface ChequeBookRequestsRepository extends CrudRepository<ChequebookRequest, Integer>{

@Modifying

@Transactional

@Query("update ChequebookRequest c set c.requestStatus=1 where c.account = ?1")

void setChequebookInfoByAccount(long accNo);

@Query("FROM ChequebookRequest c where c.requestStatus=FALSE")

public List<ChequebookRequest> findAllChequebookRequests();

}

package com.admin.dao;

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

import org.springframework.stereotype.Repository;

import com.admin.model.Saccount;

@Repository

public interface SaccountRepository extends JpaRepository<Saccount, Integer>{

public Saccount findByAccno(long accNo);

}

package com.admin.dao;

import java.util.List;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.Transfer;

@Repository

public interface TransferRepository extends CrudRepository<Transfer, Integer> {

public List<Transfer> findBySaccount(long saccount);

public List<Transfer> findByRaccount(long racoount);

}

-------------------------------------------------------------------------------------------------------------

package com.admin.dao;

import java.util.List;

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

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

import com.admin.model.User;

import com.admin.model.UserDisplay;

public interface UserDisplayRepository extends JpaRepository<User, Integer>{

@Query("SELECT new com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE u.username=a.username and u.username=s.username")

public List<UserDisplay> getAllUsers();

@Query("SELECT new com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE u.username=?1 and u.username=a.username and u.username=s.username")

public UserDisplay getUserDetailsByUsername(String userDetail);

@Query("SELECT new com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE s.accno=?1 and u.username=a.username and u.username=s.username")

public UserDisplay getUserDetailsByAccountNo(long accNo);

}

-------------------------------------------------------------------------------------------------------------

package com.admin.dao;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.admin.model.User;

import java.util.List;

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

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

@Repository

public interface UserRepository extends CrudRepository<User, Integer> {

@Query("FROM User u WHERE u.username=?1")

public User findByUsername(String username);

@Modifying

@Transactional

@Query("update User u set u.status=1 where u.username = ?1")

void disableUser(String username);

@Modifying

@Transactional

@Query("update User u set u.status=0 where u.username = ?1")

void enableUser(String username);

@Modifying

@Transactional

@Query("update User u set u.authorizationStatus=1 where u.username = ?1")

void authorizeUser(String username);

@Modifying

@Transactional

@Query("delete from User u where u.username = ?1")

void cancelAuthorization(String username);

@Query("FROM User u where u.authorizationStatus=FALSE")

public List<User> findAllUnauthorizedAccounts();

@Modifying

@Transactional

@Query("update User u set u.featureStatus=?2 where u.username = ?1")

void setUserFeatureStatus(String username, int featureId);

}

-------------------------------------------------------------------------------------------------------------

package com.admin.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.Table;

@Entity

@Table(name="account")

public class Account {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id")

private int id;

private long accno;

private int balance;

private String username;

@OneToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "user\_id")

private User user;

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public long getAccno() {

return accno;

}

public void setAccno(long accno) {

this.accno = accno;

}

public int getBalance() {

return balance;

}

public void setBalance(int balance) {

this.balance = balance;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.model;

import java.time.LocalDate;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table

public class ChequebookRequest {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

private long account;

private String accType;

private boolean requestStatus;

private LocalDate date;

private int no\_of\_pages;

public long getAccount() {

return account;

}

public void setAccount(long account) {

this.account = account;

}

public LocalDate getDate() {

return date;

}

public void setDate(LocalDate date) {

this.date = date;

}

public int getNo\_of\_pages() {

return no\_of\_pages;

}

public void setNo\_of\_pages(int no\_of\_pages) {

this.no\_of\_pages = no\_of\_pages;

}

public String getAccType() {

return accType;

}

public void setAccType(String accType) {

this.accType = accType;

}

public boolean getRequestStatus() {

return requestStatus;

}

public void setRequestStatus(boolean requestStatus) {

this.requestStatus = requestStatus;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.model;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.Table;

@Entity

@Table

public class Saccount {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

private long accno;

private int balance;

private String username;

@OneToOne(fetch = FetchType.LAZY)

@JoinColumn(name="user\_id")

private User user;

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public long getAccno() {

return accno;

}

public void setAccno(long accno) {

this.accno = accno;

}

public int getBalance() {

return balance;

}

public void setBalance(int balance) {

this.balance = balance;

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.model;

import java.time.LocalDate;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table

public class Transfer implements Comparable<Transfer>{

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

private long saccount;

private long raccount;

private int amount;

private LocalDate date;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public long getSaccount() {

return saccount;

}

public void setSaccount(long saccount) {

this.saccount = saccount;

}

public long getRaccount() {

return raccount;

}

public void setRaccount(long raccount) {

this.raccount = raccount;

}

public int getAmount() {

return amount;

}

public void setAmount(int amount) {

this.amount = amount;

}

public LocalDate getDate() {

return date;

}

public void setDate(LocalDate date) {

this.date = date;

}

@Override

public int compareTo(Transfer o) {

Integer i1=this.id;

Integer i2=o.id;

return i2.compareTo(i1);

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.model;

import java.sql.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.Table;

import com.fasterxml.jackson.annotation.JsonFormat;

@Entity

@Table(name="user")

public class User{

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id")

private int id;

private String fname;

private String lname;

private long phone;

private String address;

private String email;

private String username;

private String password;

@JsonFormat(shape=JsonFormat.Shape.STRING, pattern="dd-MM-yyyy")

private Date dob;

private String identityType;

private String identity;

@Column(columnDefinition = "boolean default false")

private boolean status;

@Column(columnDefinition = "boolean default false")

private boolean authorizationStatus;

@Column(columnDefinition = "integer default 3",nullable=false)

private int featureStatus=3;

@OneToOne(targetEntity = Account.class,mappedBy = "user",orphanRemoval = false, fetch = FetchType.LAZY)

private Account account;

@OneToOne(targetEntity = Saccount.class,mappedBy = "user",orphanRemoval = false, fetch = FetchType.LAZY)

private Saccount sAccount;

public Date getDob() {

return dob;

}

public void setDob(Date dob) {

this.dob = dob;

}

public String getIdentityType() {

return identityType;

}

public void setIdentityType(String identityType) {

this.identityType = identityType;

}

public String getIdentity() {

return identity;

}

public void setIdentity(String identity) {

this.identity = identity;

}

public int getFeatureStatus() {

return featureStatus;

}

public void setFeatureStatus(int featureStatus) {

this.featureStatus = featureStatus;

}

public boolean isStatus() {

return status;

}

public boolean isAuthorizationStatus() {

return authorizationStatus;

}

public void setAuthorizationStatus(boolean authorizationStatus) {

this.authorizationStatus = authorizationStatus;

}

public void setId(int id) {

this.id = id;

}

public void setStatus(boolean status) {

this.status = status;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

public String getLname() {

return lname;

}

public void setLname(String lname) {

this.lname = lname;

}

public long getPhone() {

return phone;

}

public void setPhone(long phone) {

this.phone = phone;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.model;

public class UserDisplay {

private String fname;

private String lname;

private long phone;

private String username;

private boolean status;

private int featureStatus;

private long primaryAccno;

private int primaryBalance;

private long savingsAccno;

private int savingsBalance;

public UserDisplay() {

}

public UserDisplay(String fname, String lname, long phone, String username, boolean status, int featureStatus, long primaryAccno,

int primaryBalance, long savingsAccno, int savingsBalance) {

super();

this.fname = fname;

this.lname = lname;

this.phone = phone;

this.username = username;

this.status = status;

this.featureStatus = featureStatus;

this.primaryAccno = primaryAccno;

this.primaryBalance = primaryBalance;

this.savingsAccno = savingsAccno;

this.savingsBalance = savingsBalance;

}

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

public String getLname() {

return lname;

}

public void setLname(String lname) {

this.lname = lname;

}

public long getPhone() {

return phone;

}

public void setPhone(long phone) {

this.phone = phone;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public boolean isStatus() {

return status;

}

public void setStatus(boolean status) {

this.status = status;

}

public long getPrimaryAccno() {

return primaryAccno;

}

public void setPrimaryAccno(long primaryAccno) {

this.primaryAccno = primaryAccno;

}

public int getPrimaryBalance() {

return primaryBalance;

}

public void setPrimaryBalance(int primaryBalance) {

this.primaryBalance = primaryBalance;

}

public long getSavingsAccno() {

return savingsAccno;

}

public void setSavingsAccno(long savingsAccno) {

this.savingsAccno = savingsAccno;

}

public int getSavingsBalance() {

return savingsBalance;

}

public void setSavingsBalance(int savingsBalance) {

this.savingsBalance = savingsBalance;

}

public int getFeatureStatus() {

return featureStatus;

}

public void setFeatureStatus(int featureStatus) {

this.featureStatus = featureStatus;

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.service.impl;

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

import org.springframework.stereotype.Service;

import com.admin.dao.AccountRepository;

import com.admin.dao.UserRepository;

import com.admin.model.Account;

@Service

public class AccountsCreationImpl {

@Autowired

AccountRepository dao;

@Autowired

private UserRepository udao;

private final String bankCode = "3914";

private final String countryCode = "91";

private final String branchCode = "820";

private final String accountcode="1";

public Account newAccount(String username,int userId) {

Account account=new Account();

account.setUsername(username);

account.setAccno(generate\_saving(userId));

account.setUser(udao.findByUsername(username));

return dao.save(account);

}

public long generate\_saving(int userId) {

String accNo = bankCode+countryCode+branchCode+accountcode+String.valueOf(userId);

return Long.parseLong(accNo);

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.service.impl;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import org.apache.commons.mail.EmailException;

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

import org.springframework.stereotype.Service;

import com.admin.dao.AccountRepository;

import com.admin.dao.ChequeBookRequestsRepository;

import com.admin.dao.SaccountRepository;

import com.admin.dao.TransferRepository;

import com.admin.dao.UserDisplayRepository;

import com.admin.dao.UserRepository;

import com.admin.model.ChequebookRequest;

import com.admin.model.Transfer;

import com.admin.model.User;

import com.admin.model.UserDisplay;

import com.admin.service.AdminService;

@Service

public class AdminServiceImpl implements AdminService{

@Autowired

private TransferRepository transDAO;

@Autowired

private ChequeBookRequestsRepository chequeBookDAO;

@Autowired

private UserRepository userDao;

@Autowired

private UserDisplayRepository displayDao;

@Autowired

private AccountRepository accDao;

@Autowired

private SaccountRepository sAccDao;

@Autowired

private MailServiceImpl mailService;

@Autowired

private AccountsCreationImpl accService;

@Autowired

private SaccountCreationImpl sAccService;

@Override

public void authorizeUser(String username) {

userDao.authorizeUser(username);

System.out.println("error here top");

User currUser = userDao.findByUsername(username);

int userId = 123;

System.out.println("error here 1");

accService.newAccount(username,userId);

System.out.println("error here 2");

sAccService.newAccount(username,userId);

}

@Override

public void cancelAuthorization(String username) {

userDao.cancelAuthorization(username);

}

@Override

public void acceptChequebookRequest(long accNo) {

String username = "";

chequeBookDAO.setChequebookInfoByAccount(accNo);

if(Long.toString(accNo).length() == 7) {

username = accDao.findByAccno(accNo).getUsername();

}

else {

username = sAccDao.findByAccno(accNo).getUsername();

}

try {

//mailService.sendChequebookConfirmedEmail(username);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@Override

public void enableUser(String username) {

userDao.enableUser(username);

}

@Override

public void disableUser(String username) {

userDao.disableUser(username);

}

@Override

public List<UserDisplay> getAllUsers() {

return displayDao.getAllUsers();

}

@Override

public List<Transfer> getAllTransactions(long accountNo) {

List<Transfer> sender=transDAO.findBySaccount(accountNo);

List<Transfer> receiver=transDAO.findByRaccount(accountNo);

List<Transfer> merged=new ArrayList<>();

merged.addAll(sender);

merged.addAll(receiver);

Collections.sort(merged);

return merged;

}

@Override

public List<ChequebookRequest> getAllChequebookRequests() {

return chequeBookDAO.findAllChequebookRequests();

}

@Override

public List<User> getAllUnauthorizedUsers() {

return userDao.findAllUnauthorizedAccounts();

}

@Override

public void setUserFeatures(String username, int featureId) {

userDao.setUserFeatureStatus(username,featureId);

}

static boolean isNumber(String s)

{

for (int i = 0; i < s.length(); i++)

if (Character.isDigit(s.charAt(i))

== false)

return false;

return true;

}

@Override

public UserDisplay searchUser(String userDetail) {

if(isNumber(userDetail)) {

return displayDao.getUserDetailsByAccountNo(Long.parseLong(userDetail));

}

else {

return displayDao.getUserDetailsByUsername(userDetail);

}

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.service.impl;

import org.apache.commons.mail.DefaultAuthenticator;

import org.apache.commons.mail.EmailException;

import org.apache.commons.mail.HtmlEmail;

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

import org.springframework.stereotype.Service;

import com.admin.dao.AccountRepository;

import com.admin.dao.UserRepository;

import com.admin.model.Account;

import com.admin.model.User;

@Service

public class MailServiceImpl {

@Autowired

UserRepository userDAO;

@Autowired

AccountRepository accountDAO;

/\*

public void sendAuthorizedEmail(String username) throws EmailException {

User user = userDAO.findByUsername(username);

String receiver = user.getEmail();

String fullName = user.getFname()+" "+user.getLname();

Account account = accountDAO.findByUsername(username);

long accountNo = account.getAccno();

int balance = account.getBalance();

HtmlEmail mail = new HtmlEmail();

mail.setSSL(true);

mail.setHostName("smtp.gmail.com");

mail.setSmtpPort(587);

mail.setAuthenticator(new DefaultAuthenticator("icinbank.noreply@gmail.com", "icin@1155"));

mail.setFrom("icinbank.noreply@gmail.com", "ICIN");

mail.addTo(receiver, "To");

mail.setSubject(user.getFname()+", Welcome to your new ICIN Bank Account");

mail.setHtmlMsg("<h1 style='font-size:27px;color:blue'>Hi "+fullName+",</h1>"

+ "<br><h2 style='font-size:21px;color:black'>Thank you for creating a ICIN Bank account. Your account has been activated,"

+ " Here are your details.</h2><br>"

+ "<p style='font-size:18px;'>Username: "+username+"<br><p style='font-size:18px;'>Account Number: "+accountNo+"<br>"

+ "<p style='font-size:18px;'>Current Balance: Rs. "+balance+"</p>");

mail.send();

System.out.println("email sent");

}

public void sendAuthorizeCancelEmail(String username) throws EmailException {

User user = userDAO.findByUsername(username);

String receiver = user.getEmail();

String fullName = user.getFname()+" "+user.getLname();

HtmlEmail mail = new HtmlEmail();

mail.setSSL(true);

mail.setHostName("smtp.gmail.com");

mail.setSmtpPort(587);

mail.setAuthenticator(new DefaultAuthenticator("icinbank.noreply@gmail.com", "icin@1155"));

mail.setFrom("icinbank.noreply@gmail.com", "ICIN");

mail.addTo(receiver, "To");

mail.setSubject("ICIN Bank Account Cancelled");

mail.setHtmlMsg("<h1 style='font-size:27px;color:red'>Hi "+fullName+",</h1>"

+ "<br><p style='font-size:19px;color:black'>Sorry, Your account with ICIN Bank with username "+username+ " has been cancelled. Please contact nearest branch for more details.</p><br><br>");

mail.send();

System.out.println("email sent");

}

public void sendChequebookConfirmedEmail(String username) throws EmailException {

User user = userDAO.findByUsername(username);

String receiver = user.getEmail();

String fullName = user.getFname()+" "+user.getLname();

HtmlEmail mail = new HtmlEmail();

mail.setSSL(true);

mail.setHostName("smtp.gmail.com");

mail.setSmtpPort(465);

mail.setAuthenticator(new DefaultAuthenticator("icinbank.noreply@gmail.com", "icin@1155"));

mail.setFrom("icinbank.noreply@gmail.com", "ICIN");

mail.addTo(receiver, "To");

mail.setSubject("ICIN Bank Chequebook Request Confirmed");

mail.setHtmlMsg("<h1 style='font-size:25px;color:green'>Hi "+fullName+",</h1>"

+ "<br><p style='font-size:19px;color:black'>Your chequebook request has been processed and confirmed it will be sent to you via courier in 4-5 business days.</p><br><br>");

mail.send();

System.out.println("email sent");

}

\*/

}

-------------------------------------------------------------------------------------------------------------

package com.admin.service.impl;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.admin.dao.SaccountRepository;

import com.admin.dao.UserRepository;

import com.admin.model.Saccount;

@Service

public class SaccountCreationImpl {

@Autowired

SaccountRepository dao;

@Autowired

UserRepository udao;

private final String bankCode = "3914";

private final String countryCode = "91";

private final String branchCode = "820";

private final String accountcode="2";

public long generate\_saving(int userId) {

String accNo = bankCode+countryCode+branchCode+accountcode+String.valueOf(userId);

return Long.parseLong(accNo);

}

public Saccount newAccount(String username, int userId) {

Saccount account =new Saccount();

account.setUsername(username);

account.setAccno(generate\_saving(userId));

account.setUser(udao.findByUsername(username));

return dao.save(account);

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin.service;

import java.util.List;

import com.admin.model.ChequebookRequest;

import com.admin.model.Transfer;

import com.admin.model.User;

import com.admin.model.UserDisplay;

public interface AdminService {

public List<UserDisplay> getAllUsers();

public List<Transfer> getAllTransactions(long accountNo);

public List<ChequebookRequest> getAllChequebookRequests();

//public void acceptChequebookRequest(int accNo);

public void enableUser(String username);

public void disableUser(String username);

public void authorizeUser(String username);

public void cancelAuthorization(String username);

public List<User> getAllUnauthorizedUsers();

public void setUserFeatures(String username, int featureId);

public UserDisplay searchUser(String userDetail);

void acceptChequebookRequest(long accNo);

}

-------------------------------------------------------------------------------------------------------------

package com.admin;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class AdminServiceApplication {

public static void main(String[] args) {

SpringApplication.run(AdminServiceApplication.class, args);

}

}

-------------------------------------------------------------------------------------------------------------

package com.admin;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class AdminServiceApplicationTests {

@Test

void contextLoads() {

}

}

-------------------------------------------------------------------------------------------------------------

package Authentication;

import org.openqa.selenium.\*;

import org.openqa.selenium.chrome.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import java.util.concurrent.TimeUnit;

import org.junit.Test;

public class Login {

@Test

public void LoginTest() throws InterruptedException {

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get("http://localhost:4200/login");

Thread.sleep(5000);

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[1]/input")).sendKeys("Novina");

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[2]/input")).sendKeys("Novina123");

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[3]/button")).click();

try

{

WebDriverWait wait=new WebDriverWait(driver, 14);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("http://localhost:4200/home")));

System.out.println("Login Successfull");

// driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[5]/div/a[2]")).click();

// wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

// System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

-------------------------------------------------------------------------------------------------------------

package UserActions;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

public class ChequeBookRequest {

@Test

public void chequeBookRequest() throws InterruptedException {

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get("http://localhost:4200/login");

Thread.sleep(5000);

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath(

"/html/body/app-root/app-login/div/form/div[1]/input")).sendKeys("Novina");

driver.findElement(By.xpath(

"/html/body/app-root/app-login/div/form/div[2]/input")).sendKeys("Novina123")

;

driver.findElement(By.xpath(

"/html/body/app-root/app-login/div/form/div[3]/button")).click();

try

{

WebDriverWait wait=new WebDriverWait(driver, 14);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(

"/html/body/app-root/app-home/div[1]/h2")));

System.out.println("Login Successfull");

driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[4]/a")).click();

driver.findElement(By.xpath("/html/body/app-root/app-cheque-book-request/div/div[2]/select")).click();

driver.findElement(By.xpath("/html/body/app-root/app-cheque-book-request/div/div[2]/select/option[1]")).click();

driver.findElement(By.xpath("/html/body/app-root/app-cheque-book-request/div/form/button")).click();

System.out.println("Cheque Book Issued Successfully!!");

// driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[5]/div/a[2]")).click();

// wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

// System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

-------------------------------------------------------------------------------------------------------------

package UserActions;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

public class EditProfile {

@Test

public void EditPorfile() throws InterruptedException {

System.setProperty("webdriver.gecko.driver",

"C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe"); WebDriver

driver = new FirefoxDriver();

driver.get("http://localhost:4200/home");

/\*

\* System.setProperty(

\* "webdriver.chrome.driver","D:\\Novina\_BNP\\Simplilearn\_Projects\\Project 4\\Chrome Driver\\chromedriver.exe"

\* ); WebDriver driver = new ChromeDriver();

\*

\* driver.get("http://localhost:4200/login");

\*/

Thread.sleep(5000);

try

{

/\*

\* WebDriverWait wait=new WebDriverWait(driver, 14);

\* wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(

\* "/html/body/app-root/app-home/div[1]/h2")));

\* System.out.println("Login Successfull");

\*/

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("//\*[@id=\"dropdown04\"]")).click();

driver.findElement(By.xpath("/html/body/app-root/nav/div/ul/li[5]/div/a[1]")).click();

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[1]/input")).sendKeys("7666854389");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/input")).sendKeys("Mumbai");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/div[1]/input")).sendKeys("novinarayudu.97@gmail.com");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/div[2]/input")).sendKeys("123P1234");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/div[3]/input")).sendKeys("123P1234");

//driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[3]/button")).click();

System.out.println("Profile edited");

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("//\*[@id=\"dropdown04\"]")).click();

driver.findElement(By.xpath("/html/body/app-root/nav/div/ul/li[5]/div/a[2]")).click();

//wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

}

}

-------------------------------------------------------------------------------------------------------------

package UserActions;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

public class Register {

@Test

public void register() throws InterruptedException {

System.setProperty("webdriver.chrome.driver", "D:\\Novina\_BNP\\Simplilearn\_Projects\\Project 4\\Chrome Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("http://localhost:4200/login");

Thread.sleep(5000);

try {

//WebDriverWait wait=new WebDriverWait(driver, 14);

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[3]/a")).click();

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[1]/input")).sendKeys("Novina");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[2]/input")).sendKeys("Rayudu");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[3]/input")).sendKeys("Novina");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[4]/input")).sendKeys("123456p");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[5]/input")).sendKeys("17/08/2020");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[6]/input")).sendKeys("7666854389");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[7]/input")).sendKeys("Mumbai");

Select id=new Select(driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[8]/select")));

id.selectByIndex(2);

WebElement fileInput = driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[9]/input"));

fileInput.sendKeys("D:\\dinos.png");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[10]/input")).sendKeys("abcde1234");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[11]/input")).sendKeys("novinarayudu.97@gmail,com");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[12]/button")).click();

Thread.sleep(5000);

//wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

System.out.println("Registration Successfull");

}

catch(Exception e) {

System.out.println("Erro in web browser\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

-------------------------------------------------------------------------------------------------------------

package UserActions;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

public class TransactionHistory {

@Test

public void TransactionHistory() throws InterruptedException {

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get("http://localhost:4200/home");

/\*

\* System.setProperty("webdriver.chrome.driver",

\* "D:\\Novina\_BNP\\Simplilearn\_Projects\\Project 4\\Chrome Driver\\chromedriver.exe"

\* ); WebDriver driver = new ChromeDriver();

\*

\* driver.get("http://localhost:4200/login");

\*/

Thread.sleep(5000);

/\*

\* driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

\* driver.findElement(By.xpath(

\* "/html/body/app-root/app-login/div/form/div[1]/input")).sendKeys("Novina");

\* driver.findElement(By.xpath(

\* "/html/body/app-root/app-login/div/form/div[2]/input")).sendKeys("Novina123")

\* ;

\*

\* driver.findElement(By.xpath(

\* "/html/body/app-root/app-login/div/form/div[3]/button")).click();

\*/

try

{

/\*

\* WebDriverWait wait=new WebDriverWait(driver, 14);

\* wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(

\* "/html/body/app-root/app-home/div[1]/h2")));

\* System.out.println("Login Successfull");

\*/

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[1]/a")).click();

System.out.println("Transaction History Displayed");

driver.findElement(By.xpath("//\*[@id=\"navbardrop\"]")).click();

driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[5]/div/a[2]")).click();

// wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

-------------------------------------------------------------------------------------------------------------

package UserActions;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

public class TransferHistory {

@Test

public void TransactionHistory() throws InterruptedException {

/\*

\* System.setProperty("webdriver.chrome.driver",

\* "D:\\Novina\_BNP\\Simplilearn\_Projects\\Project 4\\Chrome Driver\\chromedriver.exe"

\* ); WebDriver driver = new ChromeDriver();

\*

\* driver.get("http://localhost:4200/login");

\*/

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get("http://localhost:4200/home");

Thread.sleep(5000);

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

/\*

\* driver.findElement(By.xpath(

\* "/html/body/app-root/app-login/div/form/div[1]/input")).sendKeys("Novina");

\* driver.findElement(By.xpath(

\* "/html/body/app-root/app-login/div/form/div[2]/input")).sendKeys("Novina123")

\* ;

\*

\* driver.findElement(By.xpath(

\* "/html/body/app-root/app-login/div/form/div[3]/button")).click();

\*/

try

{

/\*

\* WebDriverWait wait=new WebDriverWait(driver, 14);

\* wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(

\* "/html/body/app-root/app-home/div[1]/h2")));

\* System.out.println("Login Successfull");

\*/

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[2]/a")).click();

System.out.println("Transfer History Displayed");

// driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[5]/div/a[2]")).click();

// wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

// System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

-------------------------------------------------------------------------------------------------------------

package UserActions;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

public class TransferMoney {

@Test

public void TransferMoney() throws InterruptedException{

System.setProperty("webdriver.chrome.driver", "D:\\Novina\_BNP\\Simplilearn\_Projects\\Project 4\\Chrome Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("http://localhost:4200/login");

Thread.sleep(5000);

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[1]/input")).sendKeys("Novina");

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[2]/input")).sendKeys("Novina123");

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[3]/button")).click();

try

{

WebDriverWait wait=new WebDriverWait(driver, 14);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-home/div[1]/h2")));

System.out.println("Login Successfull");

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[3]/a")).click();

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[3]/input")).sendKeys("123456789");

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[4]/input")).sendKeys("1234");

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[5]/input")).sendKeys("1000");

Thread.sleep(5000);

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[6]/button")).click();

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-home/div[1]/h2")));

System.out.println("Transfer Money Successfull");

// driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[5]/div/a[2]")).click();

// wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

// System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

-------------------------------------------------------------------------------------------------------------