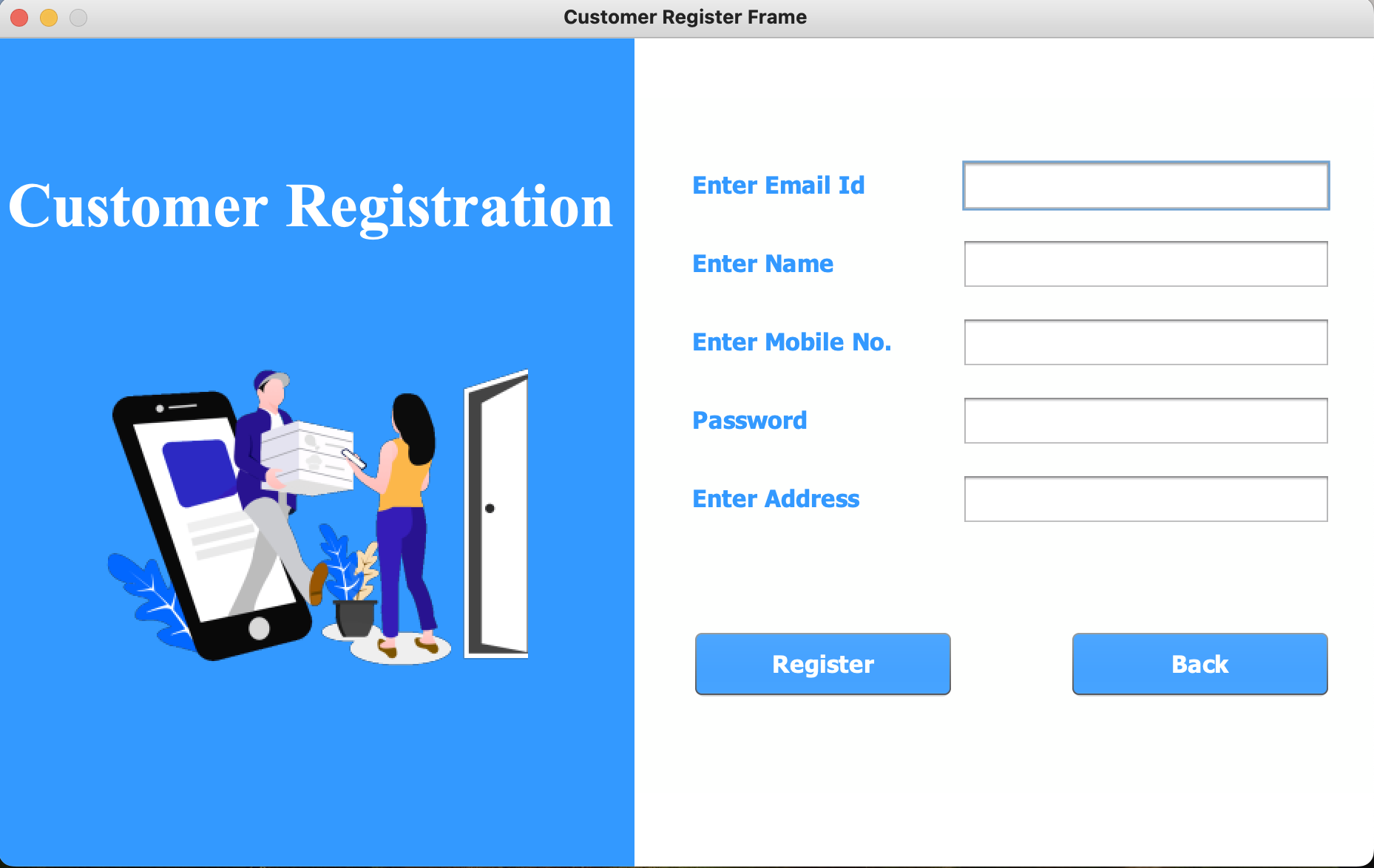
**Java SE Project**

YumXpress Application

***CustomerRegisterFrame Documentation***

1. **CustomerRegisterFrame –**



**2. STEPS TO BE DONE IN** **CustomerRegisterFrame**

In the **CustomerRegisterFrame** we need to do following steps:

1. **Creating The Customers Table**
2. **Creating CustomerPojo**
3. **Creating CustomerDAO with 2 methods: getNewId() and addCustomer()**
4. **Define a method called sendMail() in Mailer class for sending Welcome Email to customer with account details.**
5. **In the Validator class we need 2 methods isValidEmail() and isValidMobileNo() for validating email and mobile no entered.**
6. **In the CustomerRegisterFrame we need to define following methods:**

**validateInputs()**

* + - 1. **Event handler for Register button**
      2. **Event handler for Back button**

**Query of creating table :**

**CREATE TABLE CUSTOMERS (**

**CUSTOMER\_ID VARCHAR2(10) CONSTRAINT CUSTOMERS\_PK PRIMARY KEY,**

**CUSTOMER\_NAME VARCHAR2(50),**

**EMAIL\_ID VARCHAR2(30) CONSTRAINT CUSTOMERS\_EM\_NN NOT NULL,**

**PASSWORD VARCHAR2(20),**

**MOBILE\_NO VARCHAR2(10),**

**ADDRESS VARCHAR2(150)**

**);**

**Creating CustomerPojo**

**package yumxpress.pojo;**

**public class CustomerPojo {**

**private String customerId;**

**private String customerName;**

**private String emailId;**

**private String password;**

**private String mobileNo;**

**private String address;**

**// Constructors , setter and getters**

**}**

**Coding for getNewId () method in CustomerDAO class :**

**package yumxpress.dao;**

**public class CustomerDao {**

**public static String getNewId()throws SQLException{**

**Connection conn=DBConnection.getConnection();**

**Statement st=conn.createStatement();**

**ResultSet rs=st.executeQuery("Select max(customer\_id) from customers");**

**rs.next();**

**String id=rs.getString(1);**

**String custId="";**

**if(id!=null){**

**id=id.substring(4);**

**compId="CUS-"+(Integer.parseInt(id)+1);**

**}else{**

**custId="CUS-101";**

**}**

**return custId;**

**}}**

**Coding for addCustomer() method in CustomerDAO class :**

**public static boolean addCustomer(CustomerPojo customer) throws SQLException {**

**Connection conn = DBConnection.getConnection();**

**String qry = "INSERT INTO customers VALUES (?,?,?,?,?,?)";**

**PreparedStatement ps = conn.prepareStatement(qry);**

**ps.setString(1, getNewId());**

**ps.setString(2, customer.getCustomerName());**

**ps.setString(3, customer.getEmailId());**

**ps.setString(4, customer.getPassword());**

**ps.setString(5, customer.getMobileNo());**

**ps.setString(6, customer.getAddress());**

**int result = ps.executeUpdate();**

**return result > 0;**

**}**

**Coding for isValidEmail() method of Validator class:**

**public static boolean isValidEmail(String email) {**

**EmailValidator validator =**

**EmailValidator.getInstance();**

**return validator.isValid(email);**

**}**

**Coding for isValidMobileNo() method of Validator class:**

**public static boolean isValidMobileNo(String mobileNo){**

**if(mobileNo.length()!=10)**

**return false;**

**char[]mob=mobileNo.toCharArray();**

**for(char ch:mob){**

**if(Character.isDigit(ch)==false)**

**return false;**

**}**

**return true;**

**}**

**Coding for sendMail() method of Mailer class:**

**public static void sendMail(Map<String,String>emailCredentials,CustomerPojo customer)throws MessagingException{**

**final String username = emailCredentials.get("emailId");**

**final String password = emailCredentials.get("security");**

**MyAuthenticator1 myAuth=new MyAuthenticator1(username,password);**

**Session session = Session.getInstance(prop,myAuth);**

**Message message= new MimeMessage(session);**

**InternetAddress[]recipients=InternetAddress.parse(customer.getEmailId());**

**message.setRecipients(**

**Message.RecipientType.TO,**

**recipients**

**);**

**String emailSubject = "Welcome to YumXpress App - Your Ultimate Food Ordering Companion!";**

**String emailBody = "Dear "+ customer.getCustomerName() +",\n"**

**+ "\n"**

**+ "Welcome to YumXpress App! We're excited to have you as a new member of our food-loving community. "**

**+ "Here are your account details:\n"**

**+ "Your ID: " + customer.getCustomerId()+ "\n"**

**+ "Email ID: " + customer.getEmailId() + "\n"**

**+ "Your password: " + PasswordEncryption.getDecryptedPassword(customer.getPassword()) + "\n"**

**+ "\n"**

**+ "Happy ordering!\n"**

**+ "\n"**

**+ "Best Regards,\n"**

**+ "The YumXpress App Team";**

**Transport.send(message);**

**System.out.println("Mail Sent Successfully!");**

**}**

**Concept for Register Button :**

On the click of Register button we will validate input details after that create an object of

**CustomerPojo** and store all the data in it and then add customer details in the **CUSTOMERS** table of database by calling the DAO class method **addCustomer().**

**Coding for validateInput() method:**

**private boolean validateInput() {**

**if (txtEmail.getText().trim().equals("") || txtAddress.getText().trim().equals("") || txtName.getText().trim().equals("") || txtMobileNo.getText().trim().equals("") || String.valueOf(txtPassword.getPassword()).trim().equals("")) {**

**JOptionPane.showMessageDialog(this, "Please Fill All The Fields", "Error", JOptionPane.ERROR\_MESSAGE);**

**return false;**

**}**

**if (!Validator.isMobileNoValid(txtMobileNo.getText().trim())) {**

**JOptionPane.showMessageDialog(null, "please input valid mobile number!", "Error", JOptionPane.ERROR\_MESSAGE);**

**return false;**

**}**

**if (!Validator.isValidEmail(txtEmail.getText().trim())) {**

**JOptionPane.showMessageDialog(null, "please input valid Email Id!", "Error", JOptionPane.ERROR\_MESSAGE);**

**return false;**

**}**

**return true;**

**}**

**Coding for Register button:**

**private void btnRegisterCustomerActionPerformed(java.awt.event.ActionEvent evt) {**

**if (!validateInput()) {**

**return;**

**}**

**CustomerPojo customer = new CustomerPojo();**

**customer.setEmailId(txtEmail.getText().trim().toUpperCase());**

**customer.setCustomerName(txtName.getText().trim().toUpperCase());**

**customer.setMobileNo(txtMobileNo.getText().trim());**

**customer.setPassword(PasswordEncryption.getEncryptedPassword(String.valueOf(txtPassword.getPassword()).trim()));**

**customer.setAddress(txtAddress.getText().trim().toUpperCase());**

**try {**

**boolean isCustomerAdded = CustomerDao.addCustomer(customer);**

**if (!isCustomerAdded) {**

**JOptionPane.showMessageDialog(null, "Can't Register The Customer!\nPlease Try Again Later", "Error", JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**JOptionPane.showMessageDialog(null, "Register Successfully.", "Success", JOptionPane.INFORMATION\_MESSAGE);**

**//To send welcome message to the customer**

**Map<String,String>emailCredentials=new HashMap<>();**

**emailCredentials.put("emailId", "yumxpress2@gmail.com");**

**emailCredentials.put("securityKey", "tbtl nqjd vyxj knml");**

**YumxpressMailer.sendMail(emailCredentials,customer);**

**JOptionPane.showMessageDialog(null, "Mail sent successfully to "+customer.getEmailId(), "Success", JOptionPane.INFORMATION\_MESSAGE);**

**showFrame = new CustomerLoginFrame();**

**showFrame.setVisible(true);**

**this.dispose();**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(null, "DB ERROR in CustomerRegistrationFrame.", "Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}catch(Exception ex){**

**ex.printStackTrace();**

**}**

**Coding for back button:**

By clicking on back button we will redirect to login option of Customer.

**private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {**

**showFrame = new CustomerLoginFrame();**

**showFrame.setVisible(true);**

**this.dispose();**

**}**