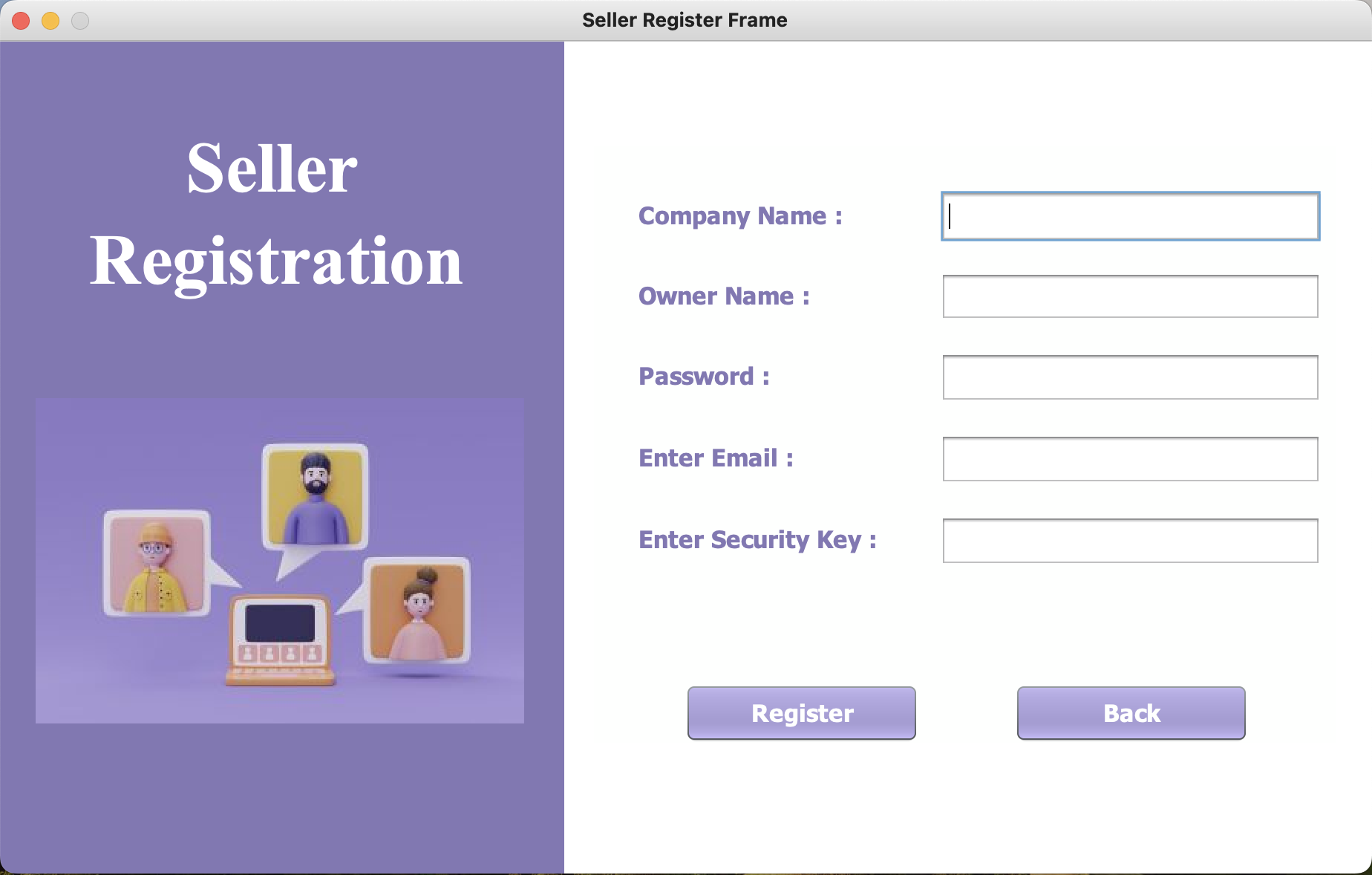
**Java SE Project**

YumXpress Application

***Seller Register Frame Documentation***

1. **SellerRegisterFrame -**

**2. STEPS TO BE DONE IN** **SellerRegi****sterFrame**

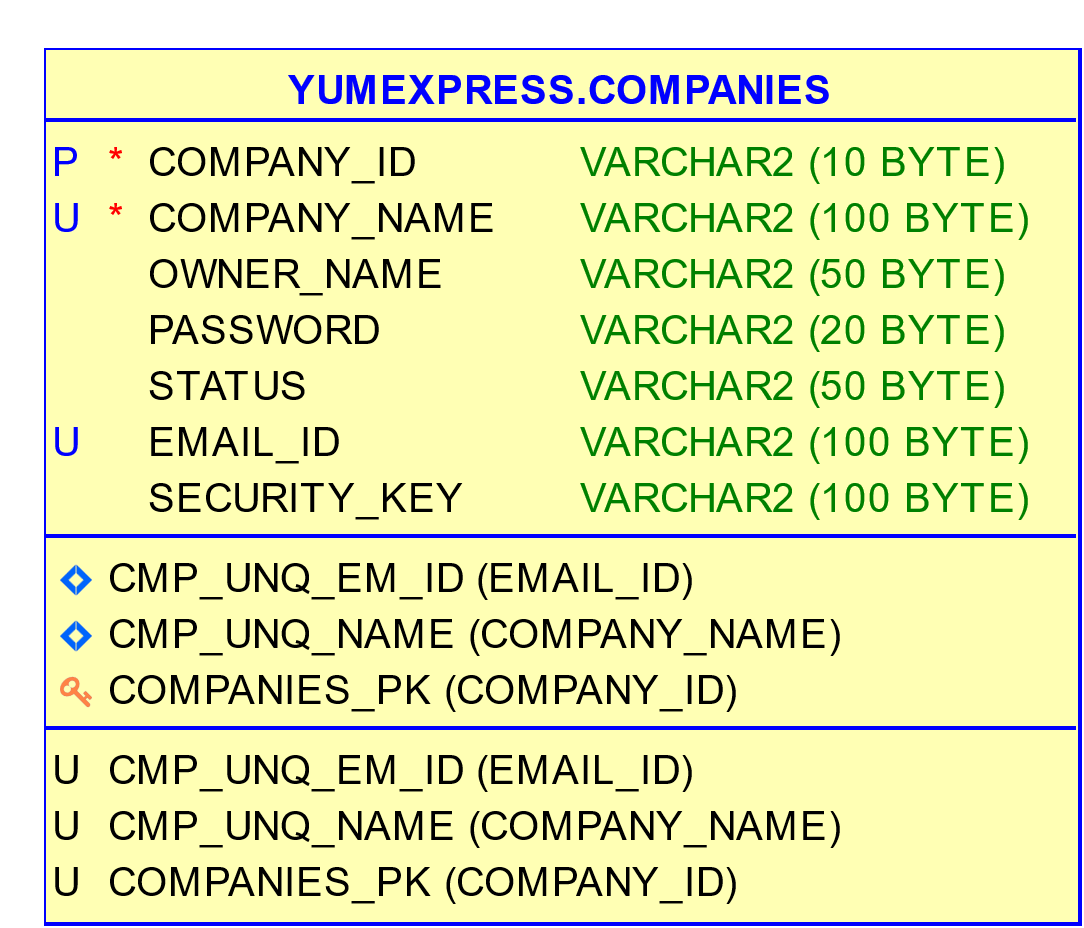
In the **SellerRegisterFrame** we need to do following steps:

1. **Creating The Companies Table**
2. **Creating CompanyPojo**
3. **Creating CompanyDAO with 2 methods: getNewId() and addSeller()**
4. **In the SellerRegisterFrame we need to define following methods:**

**validateInputs()**

* + - 1. **isValidEmail()**
      2. **Event handler for Register button**
      3. **Event handler for Back button**

**Table Structure of Companies:**



**Query of creating table :**

**CREATE TABLE COMPANIES (**

**COMPANY\_ID VARCHAR2(10) CONSTRAINT COMPANIES\_PK PRIMARY KEY,**

**COMPANY\_NAME VARCHAR2(100) CONSTRAINT CMP\_UNQ\_NAME UNIQUE,**

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

**PASSWORD VARCHAR2(20),**

**STATUS VARCHAR2(50),**

**EMAIL\_ID VARCHAR2(100) CONSTRAINT CMP\_UNQ\_EM\_ID UNIQUE,**

**SECURITY\_KEY VARCHAR2(100)**

**);**

**Creating CompanyPojo**

**package yumxpress.pojo;**

**public class CompanyPojo {**

**private String companyId;**

**private String companyName;**

**private String emailId;**

**private String ownerName;**

**private String password;**

**private String securityKey;**

**public CompanyPojo() {**

**}**

**public CompanyPojo(String companyId, String companyName, String emailId, String ownerName, String password, String securityKey) {**

**this.companyId = companyId;**

**this.companyName = companyName;**

**this.emailId = emailId;**

**this.ownerName = ownerName;**

**this.password = password;**

**this.securityKey = securityKey;**

**}**

**public String getCompanyId() {**

**return companyId;**

**}**

**public void setCompanyId(String companyId) {**

**this.companyId = companyId;**

**}**

**public String getCompanyName() {**

**return companyName;**

**}**

**public void setCompanyName(String companyName) {**

**this.companyName = companyName;**

**}**

**public String getOwnerName() {**

**return ownerName;**

**}**

**public void setOwnerName(String ownerName) {**

**this.ownerName = ownerName;**

**}**

**public String getPassword() {**

**return password;**

**}**

**public void setPassword(String password) {**

**this.password = password;**

**}**

**public String getEmailId() {**

**return emailId;**

**}**

**public void setEmailId(String emailId) {**

**this.emailId = emailId;**

**}**

**public String getSecurityKey() {**

**return securityKey;**

**}**

**public void setSecurityKey(String securityKey) {**

**this.securityKey = securityKey;**

**}**

**@Override**

**public String toString() {**

**return "CompanyPojo{" + "companyId=" + companyId + ", companyName=" + companyName + ", emailId=" + emailId + ", ownerName=" + ownerName + ", password=" + password + ", securityKey=" + securityKey + '}';**

**}**

**}**

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

**package yumxpress.dao;**

**public class CompanyDao {**

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

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

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

**ResultSet rs = st.executeQuery("Select count(\*) from companies");**

**rs.next();**

**return "CMP" + (100 + rs.getInt(1));**

**}**

**}**

**Coding for addSeller() method in CompayDAO class :**

**public static boolean addSeller(CompanyPojo seller) throws SQLException {**

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

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

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

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

**ps.setString(2, seller.getCompanyName());**

**ps.setString(3, seller.getOwnerName());**

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

**ps.setString(5, "ACTIVE");**

**ps.setString(6, seller.getEmailId());**

**ps.setString(7, seller.getSecurityKey());**

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

**return result > 0;**

**}**

**Code of PasswordEncryption class:**

The password will be stored encrypted. We will use **Base64** class to store the password as encrypted in the database. For this we will create a separate class called **PasswordEncryption** in the utility package.

**package yumxpress.utility;**

**import java.util.Base64;**

**public class PasswordEncryption {**

**public static String getEncryptedPassword(String password) {**

**Base64.Encoder encoder = Base64.getEncoder();**

**return encoder.encodeToString(password.getBytes());**

**}**

**public static String getDecryptedPassword(String password) {**

**Base64.Decoder decoder = Base64.getDecoder();**

**return new String(decoder.decode(password.getBytes()));**

**}**

**}**

**Validate user inputs and email:**

For this we will check text field is empty or not and for email we will validate email format using the utility class Validator’s static method isValidEmail() by passing it email address. It will return true if the email format is correct otherwise it will return false. Inside the method body we will use **common-validator-1.7.jar** to validate valid email format.

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

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

**EmailValidator validator =**

**EmailValidator.getInstance();**

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

**}**

1. **Concept for Register Button :**

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

CompanyPojo and stored all the data in it and then add seller details in the **COMPANIES** table of database by calling the DAO class method CompanyDao.addSeller().

**Coding for Register button:**

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

**if (validateInput()) {**

**return;**

**}**

**CompanyPojo seller = new CompanyPojo();**

**seller.setCompanyName(txtCompanyName.getText().trim().toUpperCase());**

**seller.setOwnerName(txtOwnerName.getText().trim().toUpperCase());**

**seller.setEmailId(txtEmialId.getText().trim().toUpperCase());**

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

**seller.setSecurityKey(txtSecurityKey.getText().trim());**

**System.out.println("registered details : " + seller);**

**try {**

**boolean isSellerAdded = CompanyDao.addSeller(seller);**

**if (!isSellerAdded) {**

**JOptionPane.showMessageDialog(this, "Can't Register now.\n Try Again Later", "Error", JOptionPane.ERROR);**

**return;**

**}**

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

**showFrame = new SellerLoginFrame();**

**showFrame.setVisible(Boolean.TRUE);**

**this.dispose();**

**} catch (SQLException ex) {**

**JOptionPane.showMessageDialog(null, "Email Id Already Exists.", "Error", JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

**Coding for isValidEmail() method:**

**private boolean validateInput() {**

**if (txtCompanyName.getText().trim().equals("") || txtEmialId.getText().trim().equals("") || txtOwnerName.getText().trim().equals("") || txtSecurityKey.getText().trim().equals("") || String.valueOf(txtPassword.getPassword()).trim().equals("")) {**

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

**return true;**

**}**

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

**JOptionPane.showMessageDialog(this, "Please Enter Valid Email.", "Information", JOptionPane.INFORMATION\_MESSAGE);**

**return true;**

**}**

**return false;**

**}**

1. **Coding for back button:**

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

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

**showFrame = new SellerLoginFrame();**

**showFrame.setVisible(Boolean.TRUE);**

**this.dispose();**

**}**