Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

OOSD Semester Two 2024

EV Charging System using Java Swing

Submission Due Date: 30th April 2024

Student Name: Abigail Murray

Student Number: C00260073

Table of Contents

Description

This project was developed using Java Swing and MySQL. It is an EV Charging Management System which the customer can interact with through the GUI.
The aim of this system is to provide a user friendly interface which streamlines the process of locating a charger, starting a session, ending a session, reserving a charger, managing payment methods and account details. The target for this application would be anyone driving an electric vehicle. This application would be for charging stations that any customer can use, not an at home EV charging management system.

This system has many tables, many of which are linked using inner joins. The tables included in this project are Chargers, charging stations, charging transactions, reservations, customer accounts and payment methods.

On successful login or registration the customer should be taken to a customer dashboard where they are given 6 different options to choose from.
After selection one option, in many instances the user is then taken to a submenu where they can choose to add, delete, update or view based off the item they selected. All submenus across the system have a similar look to ensure consistency across the application.

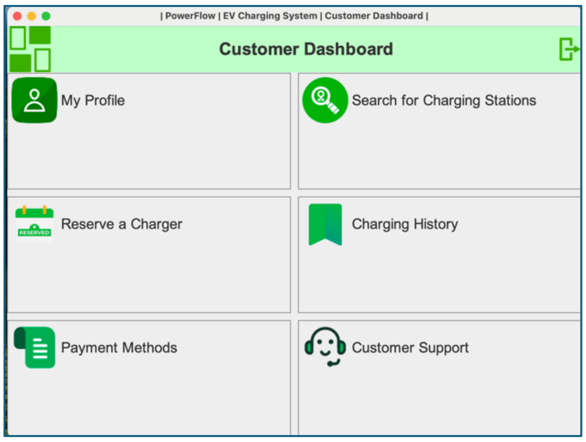


Figure 1:Customer Dashboard

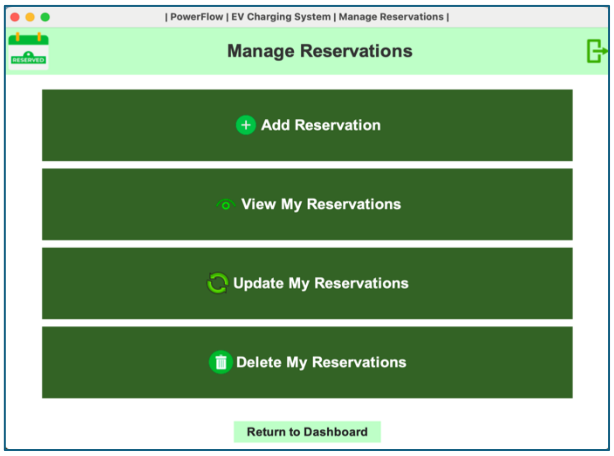


Figure 2: Sub Menu

Requirements

The requirements for this application can be broken down into functional and non-functional requirements. The main non-functional requirements focused on in this application were security and usability. A customer support section was added to provide instructions to users. The overall design such as colours, text and images used throughout were carefully considered to ensure the system was easy to use. There are only 4(mint green, black, white, grey) colours used throughout to ensure that the design stays consistent and clear. Images were used in many places alongside text to improve the user experience for customers. In terms of security, the password is stored in a secure way using hashed passwords and salts. This improves the overall security of the application.

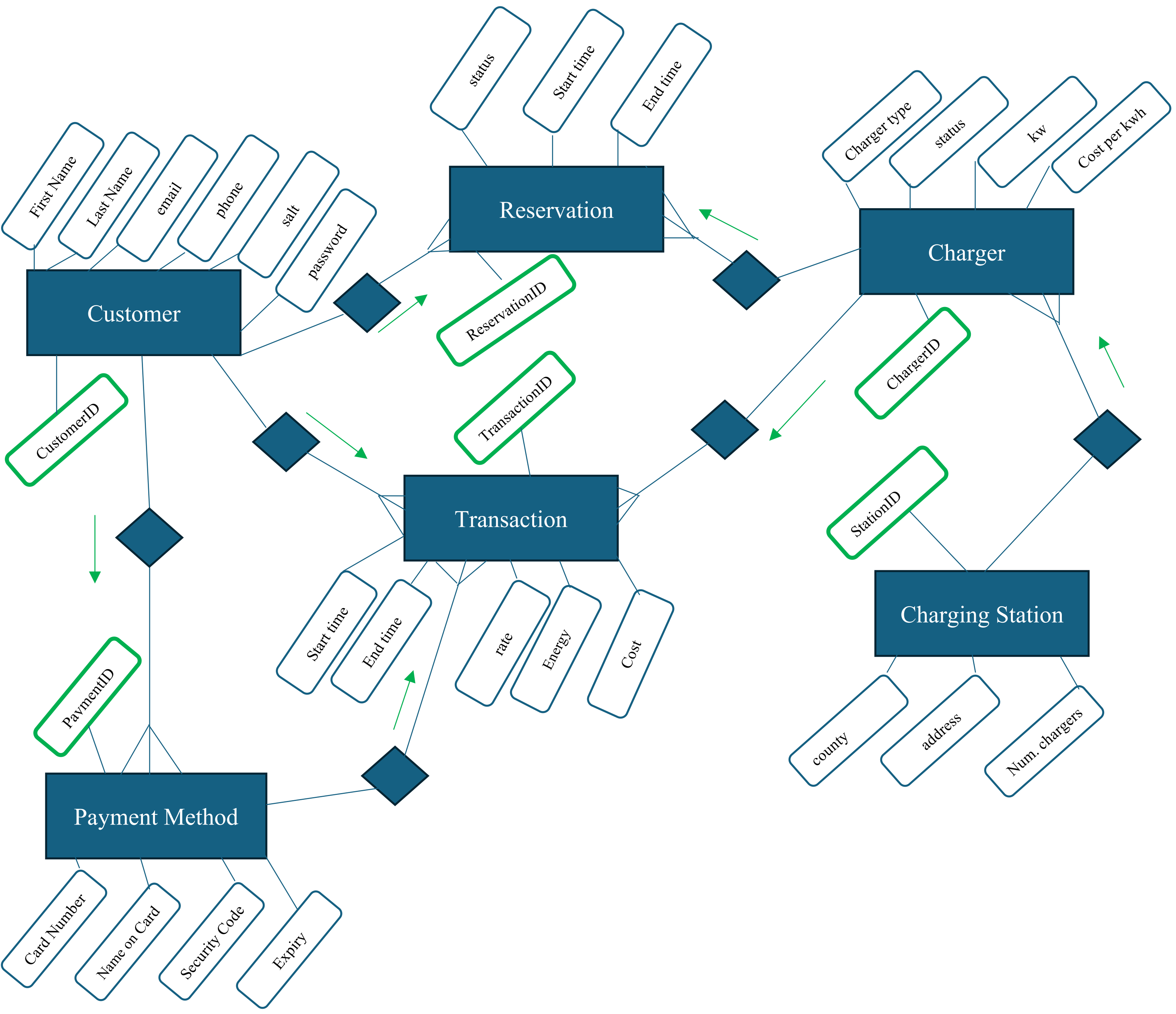
In relation to functional requirements, the system should be able to handle requests to register as a new customer or login as an existing customer. Customers should be able to manage their profile and have the option of viewing their details, updating their details and deleting their account.
The system should also offer an area where the user can add, delete, view and update their payment methods. Only card is accepted, the payments would then be processed through the use of another platform.
The reservation functionality should enable the user to add a reservation, update a reservation (in the future), view reservations of past, present and future and also cancel future reservations.

Finding charging stations should be made easier through the search by county drop down box where a list of stations in the chosen county is then generated. The user can then select any station and see what chargers are at that station. Details such as charger type, status and cost per kwh are displayed for each charger. Beside each charger the user then has the option to click start session. To increase the user experience, when the user starts a session a timer is started so they know how long they have been charging for. Once the user stops the session, the total cost of the transaction is then displayed to them.

Requirements

Requirement Name	Description	Use cases Linked	Priority
Create Customer account	Can create a new account and login using email and password	Register	High
View Customer Details	Customers can view their account details	View Account	Medium
Update Customer Details	Customer can update first name, last name and phone	Update Account	Medium
Delete Customer Account	The customer can delete their account, deleting all their details from the database	Delete Account	Low
Add Reservation	The Customer can add a reservation for future time	Add Reservation	High
Update Reservation	The customer can edit future reservation for future date/time	Update Reservation	High
View Reservations	The customer can view all reservations made	View Reservation	Medium
Delete Reservations	Customer can cancel a reservation which has a date/time in the future	Delete Reservation	High
Add payment method	Can add any card payment method	Add payment method	High
View Payment methods	Can view all payment methods	View payment method	Medium
Update Payment Methods	Can update the card number or name on card	Update payment method	High
Delete Payment Methods	Can delete any payment method	Delete payment method	Medium
Find charging station	Can find a charging station based off county selected	Find station	High
View chargers	Can view all chargers in a particular station	Find chargers	High
Start Transaction	Can begin a charging session if the charger is available	Start session	High
End Transaction	Can end a session when the press stop, total cost is then calculated	End session	High
View charging history	Can view all previous transactions	View Transactions	Low

ER Diagram



Fields in green are posted keys, green arrow gives a representation of where the keys are posted to.

Database Tables

Charger Table

Structure

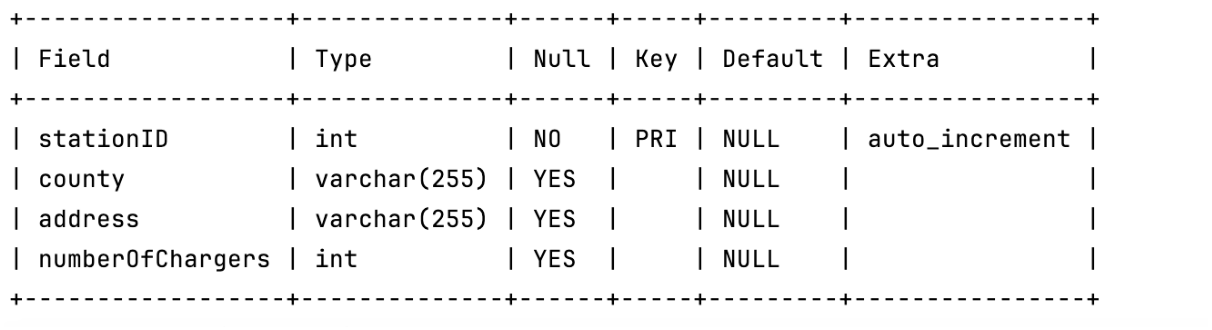
```
+------------------+---------------------------------------------------+------+-----+---------+----------------+
| Field            | Type                                              | Null | Key | Default | Extra          |
+------------------+---------------------------------------------------+------+-----+---------+----------------+
| chargerID        | int                                               | NO   | PRI | NULL    | auto_increment |
| chargerType      | varchar(50)                                       | NO   |     | NULL    |                |
| stationID        | int                                               | NO   | MUL | NULL    |                |
| status           | enum('Available','In-Use','Under Repair','Reserved') | YES  |     | NULL    |                |
| kw               | int                                               | NO   |     | NULL    |                |
| costPerKWH       | decimal(10,3)                                     | YES  |     | NULL    |                |
| sessionStartTime | datetime                                          | YES  |     | NULL    |                |
| currentUserId    | int                                               | YES  | MUL | NULL    |                |
+------------------+---------------------------------------------------+------+-----+---------+----------------+
```

Data Populated

```
+-----------+-------------+-----------+-----------+-----+------------+---------------------+---------------+
| chargerID | chargerType | stationID | status    | kw  | costPerKWH | sessionStartTime    | currentUserId |
+-----------+-------------+-----------+-----------+-----+------------+---------------------+---------------+
|         1 | ccs         |         3 | Reserved  | 150 |      0.682 | NULL                |          NULL |
|         2 | chademo     |         3 | Available |  70 |      0.682 | NULL                |          NULL |
|         4 | ccs         |         2 | Available | 200 |      0.682 | NULL                |          NULL |
|         7 | chademo     |         2 | Reserved  |  70 |      0.682 | NULL                |          NULL |
|         8 | ccs         |         2 | Available |  70 |      0.682 | NULL                |          NULL |
|         9 | ccs         |         2 | Reserved  | 200 |      0.682 | NULL                |          NULL |
|        10 | ccs         |         4 | In-Use    |  50 |      0.682 | 2024-04-23 13:52:49 |            12 |
|        11 | ccs         |         5 | Available | 100 |      0.682 | NULL                |          NULL |
|        12 | chademo     |         5 | Available | 100 |      0.682 | NULL                |          NULL |
|        13 | ccs         |         5 | Available |  70 |      0.682 | NULL                |          NULL |
|        14 | chademo     |         5 | Reserved  |  70 |      0.682 | NULL                |          NULL |
|        15 | chademo     |         6 | Available |  50 |      0.682 | NULL                |          NULL |
|        16 | ccs         |         6 | Available | 100 |      0.682 | NULL                |          NULL |
|        17 | ccs         |         7 | Available | 100 |      0.682 | NULL                |          NULL |
|        18 | chademo     |         7 | Available |  50 |      0.682 | NULL                |          NULL |
|        19 | ccs         |         8 | Available | 100 |      0.682 | NULL                |          NULL |
|        20 | chademo     |         8 | Available |  70 |      0.682 | NULL                |          NULL |
|        21 | chademo     |         9 | Reserved  |  70 |      0.682 | NULL                |          NULL |
|        22 | ccs         |         9 | Reserved  | 100 |      0.682 | NULL                |          NULL |
+-----------+-------------+-----------+-----------+-----+------------+---------------------+---------------+
```

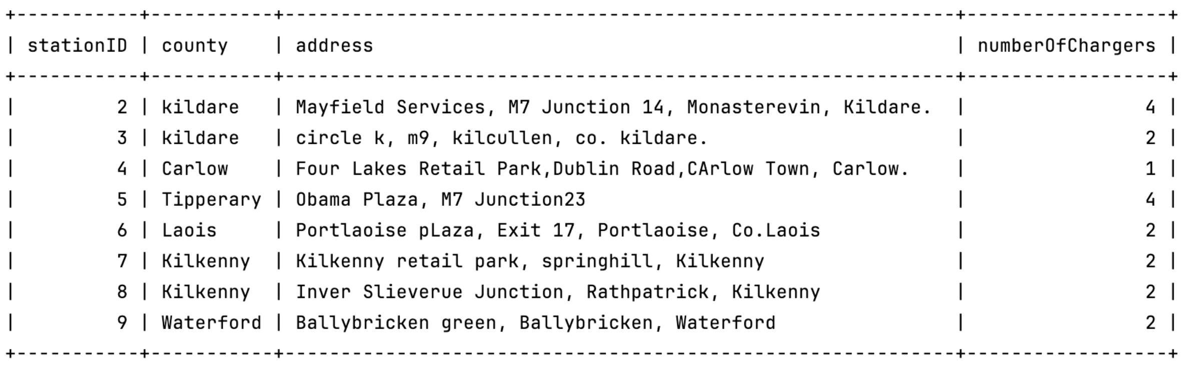
Charging Stations Table

Structure



```
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| stationID        | int          | NO   | PRI | NULL    | auto_increment |
| county           | varchar(255) | YES  |     | NULL    |                |
| address          | varchar(255) | YES  |     | NULL    |                |
| numberOfChargers | int          | YES  |     | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
```

Data Populated



```
+-----------+-----------+-------------------------------------------------------------+------------------+
| stationID | county    | address                                                     | numberOfChargers |
+-----------+-----------+-------------------------------------------------------------+------------------+
|         2 | kildare   | Mayfield Services, M7 Junction 14, Monasterevin, Kildare.   |                4 |
|         3 | kildare   | circle k, m9, kilcullen, co. kildare.                       |                2 |
|         4 | Carlow    | Four Lakes Retail Park,Dublin Road,CArlow Town, Carlow.     |                1 |
|         5 | Tipperary | Obama Plaza, M7 Junction23                                  |                4 |
|         6 | Laois     | Portlaoise pLaza, Exit 17, Portlaoise, Co.Laois             |                2 |
|         7 | Kilkenny  | Kilkenny retail park, springhill, Kilkenny                  |                2 |
|         8 | Kilkenny  | Inver Slieverue Junction, Rathpatrick, Kilkenny             |                2 |
|         9 | Waterford | Ballybricken green, Ballybricken, Waterford                 |                2 |
+-----------+-----------+-------------------------------------------------------------+------------------+
```

Charging Transactions Table

Structure

```
+----------------+---------------+------+-----+---------+----------------+
| Field          | Type          | Null | Key | Default | Extra          |
+----------------+---------------+------+-----+---------+----------------+
| transactionID  | int           | NO   | PRI | NULL    | auto_increment |
| startTime      | datetime      | YES  |     | NULL    |                |
| endTime        | datetime      | YES  |     | NULL    |                |
| energyConsumed | decimal(10,2) | YES  |     | NULL    |                |
| rate           | decimal(10,3) | YES  |     | NULL    |                |
| totalCost      | decimal(10,3) | YES  |     | NULL    |                |
| chargerID      | int           | YES  |     | NULL    |                |
| customerID     | int           | YES  | MUL | NULL    |                |
+----------------+---------------+------+-----+---------+----------------+
```

Data Populated

```
+---------------+---------------------+---------------------+----------------+-------+-----------+-----------+------------+
| transactionID | startTime           | endTime             | energyConsumed | rate  | totalCost | chargerID | customerID |
+---------------+---------------------+---------------------+----------------+-------+-----------+-----------+------------+
|             2 | 2024-02-19 10:00:00 | 2024-02-19 10:30:00 |          16.00 | 0.682 |    10.990 |         1 |          8 |
|             4 | 2024-04-18 09:20:48 | 2024-04-18 09:32:56 |          14.00 | 0.682 |     9.550 |         7 |          8 |
|             5 | 2024-04-18 18:51:52 | 2024-04-18 19:01:53 |           8.50 | 0.682 |     5.800 |        10 |          8 |
|             7 | 2024-04-18 19:48:16 | 2024-04-18 19:51:42 |           3.00 | 0.682 |     2.046 |        10 |          8 |
|             9 | 2024-04-18 20:03:04 | 2024-04-18 20:10:02 |           6.00 | 0.682 |     4.092 |        10 |          8 |
|            15 | 2024-04-19 16:09:51 | 2024-04-19 16:10:58 |           1.40 | 0.682 |     0.955 |         7 |          8 |
|            17 | 2024-04-20 09:59:57 | 2024-04-20 10:04:57 |          16.00 | 0.682 |    10.912 |         9 |          8 |
|            19 | 2024-04-20 10:26:48 | 2024-04-20 10:28:39 |           1.50 | 0.682 |     1.023 |        10 |          8 |
|            20 | 2024-04-20 12:02:20 | 2024-04-20 12:02:37 |           0.00 | 0.682 |     0.000 |         7 |          8 |
|            21 | 2024-04-20 12:07:37 | 2024-04-20 12:09:44 |           8.00 | 0.682 |     5.456 |         4 |          8 |
|            22 | 2024-04-20 12:29:19 | 2024-04-20 12:39:37 |          11.90 | 0.682 |     8.116 |         2 |          8 |
|            24 | 2024-04-20 13:37:41 | 2024-04-20 13:44:25 |           5.50 | 0.682 |     3.751 |        10 |          8 |
|            25 | 2024-04-20 15:46:00 | 2024-04-20 15:46:15 |           0.00 | 0.682 |     0.000 |        10 |          8 |
|            26 | 2024-04-21 16:10:33 | 2024-04-21 16:11:45 |           1.40 | 0.682 |     0.955 |         2 |          8 |
|            27 | 2024-04-21 18:13:58 | 2024-04-21 18:43:53 |          25.00 | 0.682 |    17.050 |        18 |          8 |
|            28 | 2024-04-23 11:00:22 | 2024-04-23 11:00:41 |           1.00 | 0.682 |     0.682 |        19 |          8 |
|            29 | 2024-04-23 11:26:06 | 2024-04-23 11:26:21 |           0.00 | 0.682 |     0.000 |        10 |          8 |
|            30 | 2024-04-23 11:56:51 | 2024-04-23 12:02:21 |           6.30 | 0.682 |     4.297 |        13 |          7 |
|            31 | 2024-04-23 13:50:12 | 2024-04-23 13:51:40 |           2.00 | 0.682 |     1.364 |        16 |         12 |
|            32 | 2024-04-23 13:52:49 | 2024-04-23 13:55:52 |           2.50 | 0.682 |     1.705 |        10 |         12 |
+---------------+---------------------+---------------------+----------------+-------+-----------+-----------+------------+
```

Customer Accounts Table

Structure

```
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| customerID | int          | NO   | PRI | NULL    | auto_increment |
| firstName  | varchar(50)  | NO   |     | NULL    |                |
| lastName   | varchar(50)  | NO   |     | NULL    |                |
| email      | varchar(50)  | NO   |     | NULL    |                |
| phone      | varchar(50)  | NO   |     | NULL    |                |
| password   | varchar(255) | YES  |     | NULL    |                |
| salt       | varchar(255) | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
```

Data Populated

```
mysql> select * from customer_accounts;
+------------+-----------+----------+------------------+--------------+------------------------------------------------------------------+--------------------------+
| customerID | firstName | lastName | email            | phone        | password                                                         | salt                     |
+------------+-----------+----------+------------------+--------------+------------------------------------------------------------------+--------------------------+
|          7 | sally     | o'brien  | sallyb@mail.com  | 086 4528977  | e93a76f0a58158a17b2801aafada8e4a6fd053a5d4955d5c9c4aea21040ba36b | TP/OocI7gHhrdHLwS+X64g== |
|          8 | Frank     | Martin   | frank@mail.com   | 087 543 6710 | 9445600a36ea7e001c53e99b30be949357af32065781ce56d901b91faacd2246 | naLSs6ahaXRn7W7sSUHZJw== |
|         12 | majella   | murphy   | majella@mail.com | 087 5342182  | b4f7e3806c1a18983719148dadacd2bd935c85345cd5199f3b2d02390572f164 | /gAm03v+dAFkumYPuRP4cg== |
|         15 | john      | murphy   | johnm@mail.com   | 087 635 277  | fd7b5cd74463f51f4e66b809848c939ead6c415fe5c87ff15c385d12a68aa9c5 | UtYunohsRdpMin4BI81QYg== |
|         16 | sarah     | halloran | sarah1@mail.com  | 085 652 1773 | 006b7197abec3240d311d5c09a095e50c0d1647211858044323becf60990206e | NNzUIiXbc+czC6xt//HiYQ== |
|         17 | Lucy      | Jones    | LucyJ@mail.com   | 087 456 3772 | f084c433db98f6a3ec439af023e43080821860f90eb189ecb47a2176dcd2239e | nKhcN92fEoVJQVXVLVIHKA== |
+------------+-----------+----------+------------------+--------------+------------------------------------------------------------------+--------------------------+
```

Payment methods table

Structure

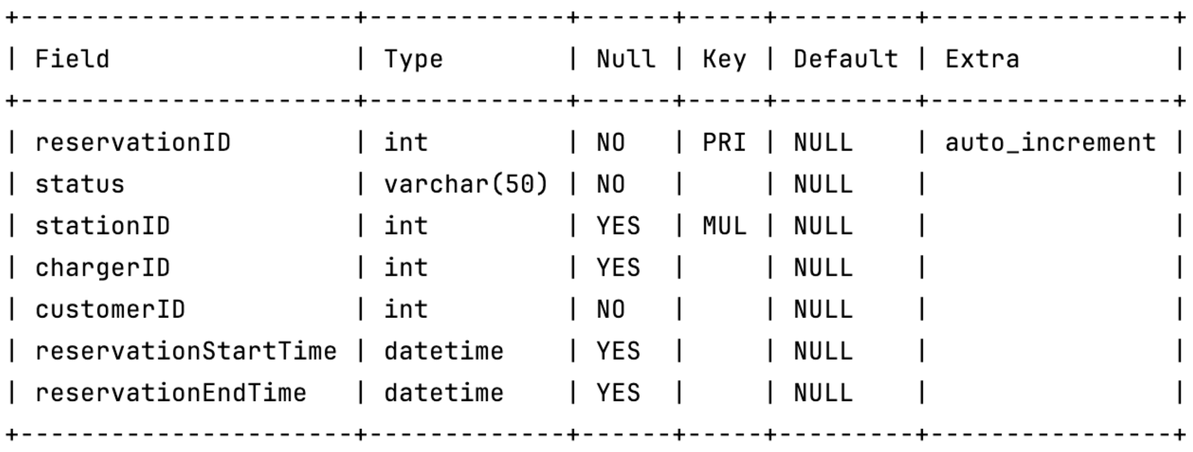
```
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| PaymentMethodID | int          | NO   | PRI | NULL    | auto_increment |
| CustomerID      | int          | NO   | MUL | NULL    |                |
| CardNumber      | varchar(16)  | YES  |     | NULL    |                |
| Expiry          | varchar(7)   | YES  |     | NULL    |                |
| SecurityCode    | varchar(4)   | YES  |     | NULL    |                |
| NameOnCard      | varchar(100) | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
```

Data Populated

```
+-----------------+------------+------------------+--------+--------------+---------------+
| PaymentMethodID | CustomerID | CardNumber       | Expiry | SecurityCode | NameOnCard    |
+-----------------+------------+------------------+--------+--------------+---------------+
|               9 |          8 | 9873427541234842 | 08/24  | 694          | frank martin  |
|              10 |          8 | 6785567889765432 | 08/26  | 678          | frank         |
|              13 |          8 | 7564738298765432 | 03/26  | 1234         | f             |
|              14 |          8 | 8765432187654321 | 04/26  | 9864         | frank         |
|              15 |          8 | 5463723891393748 | 01/28  | 435          | frank m       |
|              16 |          8 | 1235873294619435 | 06/27  | 582          | frank m       |
|              17 |          8 | 7564738567345321 | 04/28  | 234          | frank         |
|              18 |          7 | 6767676767676767 | 03/36  | 376          | sally o'brien |
|              19 |          7 | 1234567891298765 | 01/27  | 653          | Sally         |
|              21 |         12 | 5655555555555656 | 09/27  | 999          | m             |
|              22 |         12 | 8921217642919320 | 05/26  | 847          | majella       |
+-----------------+------------+------------------+--------+--------------+---------------+
```

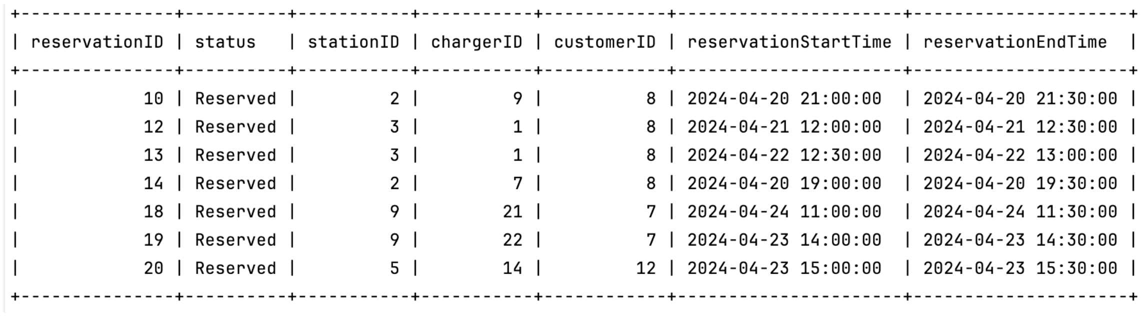
Reservations Table

Structure



```
+----------------------+------------+------+-----+---------+----------------+
| Field                | Type       | Null | Key | Default | Extra          |
+----------------------+------------+------+-----+---------+----------------+
| reservationID        | int        | NO   | PRI | NULL    | auto_increment |
| status               | varchar(50) | NO   |     | NULL    |                |
| stationID            | int        | YES  | MUL | NULL    |                |
| chargerID            | int        | YES  |     | NULL    |                |
| customerID           | int        | NO   |     | NULL    |                |
| reservationStartTime | datetime   | YES  |     | NULL    |                |
| reservationEndTime   | datetime   | YES  |     | NULL    |                |
+----------------------+------------+------+-----+---------+----------------+
```

Data Populated

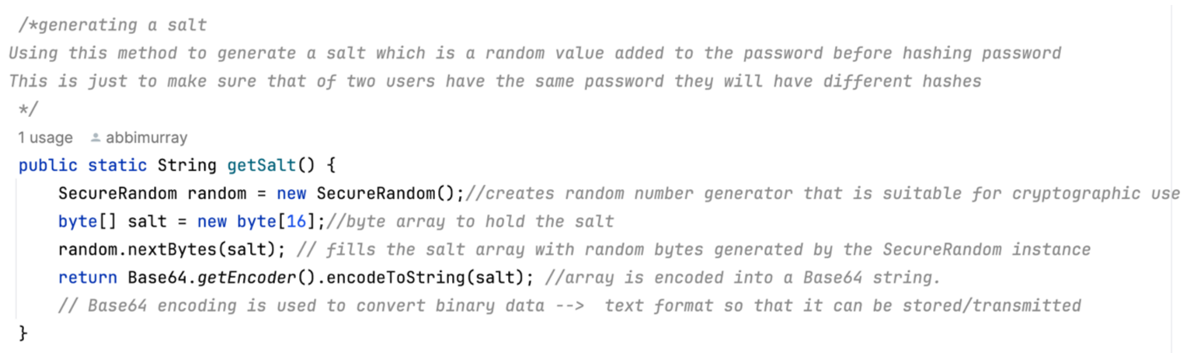


```
+---------------+----------+-----------+-----------+------------+----------------------+---------------------+
| reservationID | status   | stationID | chargerID | customerID | reservationStartTime | reservationEndTime  |
+---------------+----------+-----------+-----------+------------+----------------------+---------------------+
|            10 | Reserved |         2 |         9 |          8 | 2024-04-20 21:00:00  | 2024-04-20 21:30:00 |
|            12 | Reserved |         3 |         1 |          8 | 2024-04-21 12:00:00  | 2024-04-21 12:30:00 |
|            13 | Reserved |         3 |         1 |          8 | 2024-04-22 12:30:00  | 2024-04-22 13:00:00 |
|            14 | Reserved |         2 |         7 |          8 | 2024-04-20 19:00:00  | 2024-04-20 19:30:00 |
|            18 | Reserved |         9 |        21 |          7 | 2024-04-24 11:00:00  | 2024-04-24 11:30:00 |
|            19 | Reserved |         9 |        22 |          7 | 2024-04-23 14:00:00  | 2024-04-23 14:30:00 |
|            20 | Reserved |         5 |        14 |         12 | 2024-04-23 15:00:00  | 2024-04-23 15:30:00 |
+---------------+----------+-----------+-----------+------------+----------------------+---------------------+
```

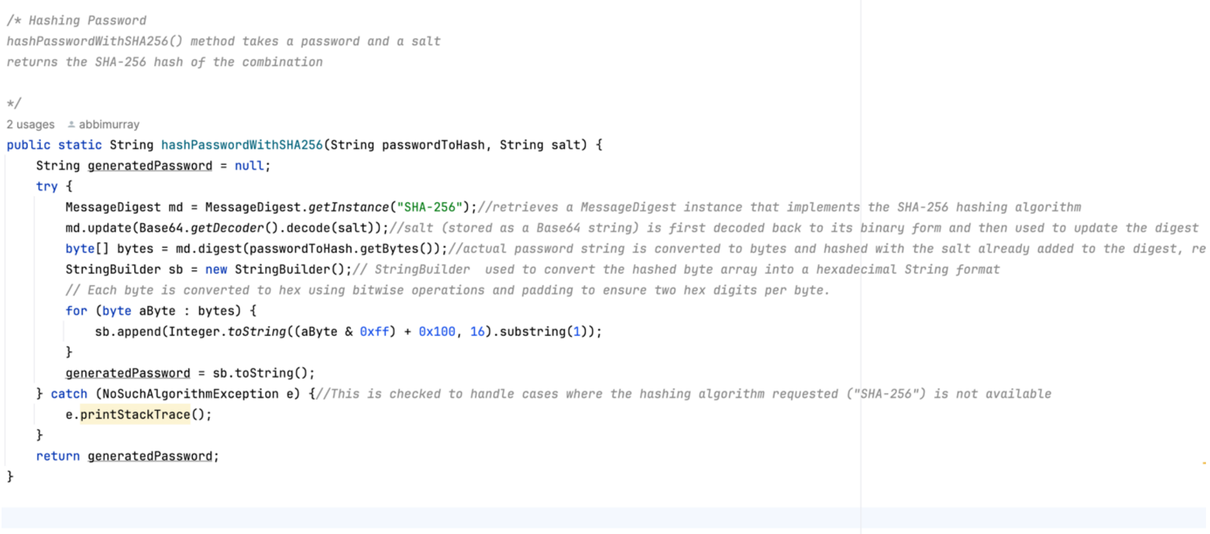
Source Code Snippets

The first code snippet is my password hashing function to ensure that password are stored safely. Firstly, a salt which is a random value added to the password is generated. Then the password and salt is taken and using the hashPasswordWithSHA256 method, the SHA256 hash of the combination is returned.

In the Hashing Utils class



```
 /*generating a salt
Using this method to generate a salt which is a random value added to the password before hashing password
This is just to make sure that of two users have the same password they will have different hashes
 */
 1 usage   abbimurray
 public static String getSalt() {
     SecureRandom random = new SecureRandom();//creates random number generator that is suitable for cryptographic use
     byte[] salt = new byte[16];//byte array to hold the salt
     random.nextBytes(salt); // fills the salt array with random bytes generated by the SecureRandom instance
     return Base64.getEncoder().encodeToString(salt); //array is encoded into a Base64 string.
     // Base64 encoding is used to convert binary data -->  text format so that it can be stored/transmitted
 }
```

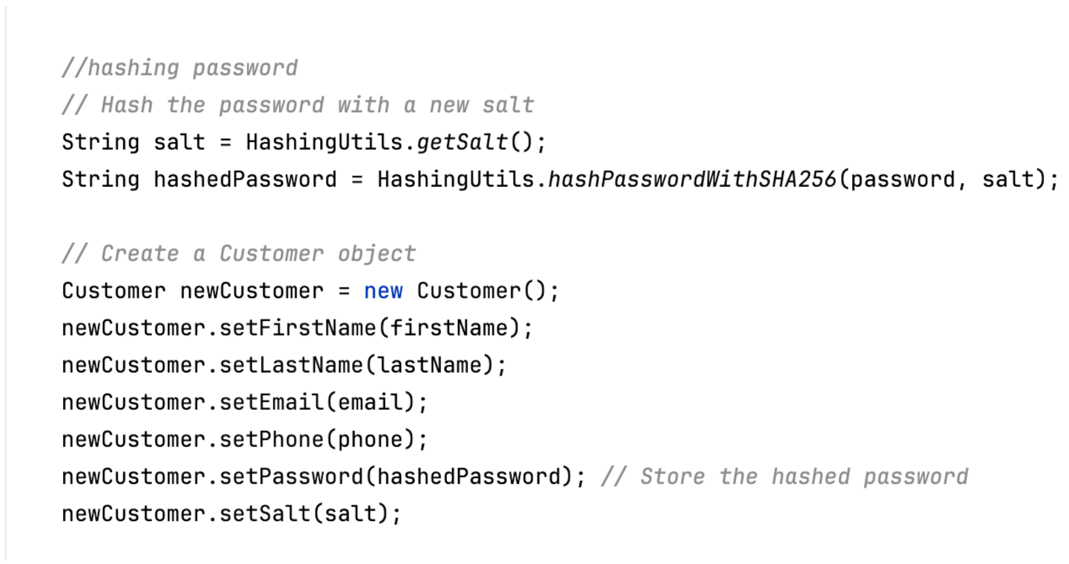


```
/* Hashing Password
hashPasswordWithSHA256() method takes a password and a salt
returns the SHA-256 hash of the combination

*/
2 usages   abbimurray
public static String hashPasswordWithSHA256(String passwordToHash, String salt) {
    String generatedPassword = null;
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-256");//retrieves a MessageDigest instance that implements the SHA-256 hashing algorithm
        md.update(Base64.getDecoder().decode(salt));//salt (stored as a Base64 string) is first decoded back to its binary form and then used to update the digest
        byte[] bytes = md.digest(passwordToHash.getBytes());//actual password string is converted to bytes and hashed with the salt already added to the digest, re
        StringBuilder sb = new StringBuilder();// StringBuilder  used to convert the hashed byte array into a hexadecimal String format
        // Each byte is converted to hex using bitwise operations and padding to ensure two hex digits per byte.
        for (byte aByte : bytes) {
            sb.append(Integer.toString((aByte & 0xff) + 0x100, 16).substring(1));
        }
        generatedPassword = sb.toString();
    } catch (NoSuchAlgorithmException e) {//This is checked to handle cases where the hashing algorithm requested ("SHA-256") is not available
        e.printStackTrace();
    }
    return generatedPassword;
}
```

Hashing Password continued

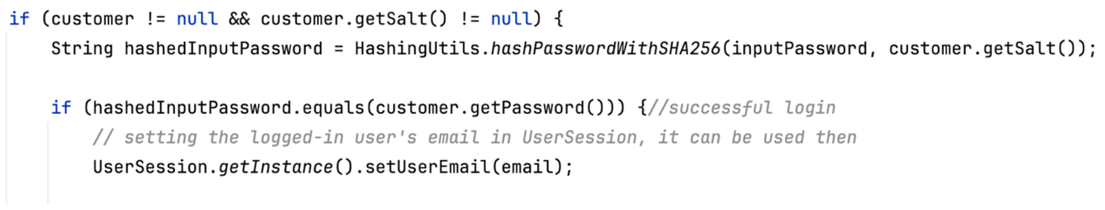
In the registration class



```
//hashing password
// Hash the password with a new salt
String salt = HashingUtils.getSalt();
String hashedPassword = HashingUtils.hashPasswordWithSHA256(password, salt);

// Create a Customer object
Customer newCustomer = new Customer();
newCustomer.setFirstName(firstName);
newCustomer.setLastName(lastName);
newCustomer.setEmail(email);
newCustomer.setPhone(phone);
newCustomer.setPassword(hashedPassword); // Store the hashed password
newCustomer.setSalt(salt);
```

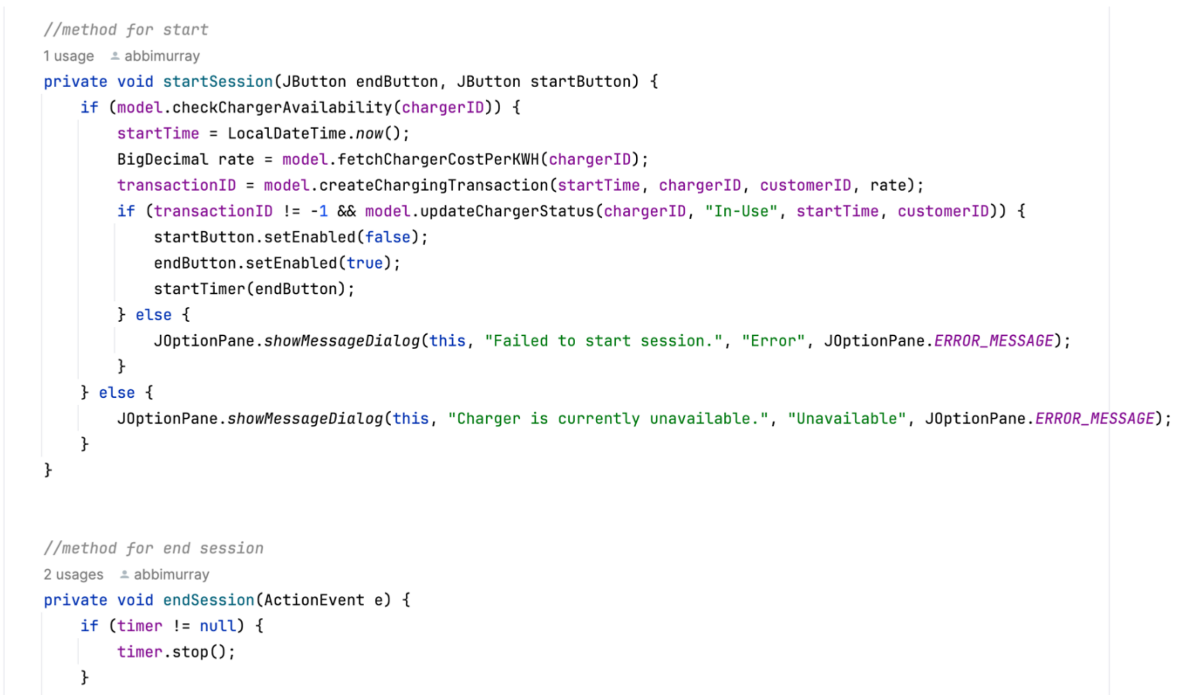
In the log in class



```
if (customer != null && customer.getSalt() != null) {
    String hashedInputPassword = HashingUtils.hashPasswordWithSHA256(inputPassword, customer.getSalt());

    if (hashedInputPassword.equals(customer.getPassword())) {//successful login
        // setting the logged-in user's email in UserSession, it can be used then
        UserSession.getInstance().setUserEmail(email);
```

The following is a code snippet of the start session which uses a timer to track the customers charging transaction.



```
//method for start
1 usage   abbimurray
private void startSession(JButton endButton, JButton startButton) {
    if (model.checkChargerAvailability(chargerID)) {
        startTime = LocalDateTime.now();
        BigDecimal rate = model.fetchChargerCostPerKWH(chargerID);
        transactionID = model.createChargingTransaction(startTime, chargerID, customerID, rate);
        if (transactionID != -1 && model.updateChargerStatus(chargerID, "In-Use", startTime, customerID)) {
            startButton.setEnabled(false);
            endButton.setEnabled(true);
            startTimer(endButton);
        } else {
            JOptionPane.showMessageDialog(this, "Failed to start session.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Charger is currently unavailable.", "Unavailable", JOptionPane.ERROR_MESSAGE);
    }
}


//method for end session
2 usages   abbimurray
private void endSession(ActionEvent e) {
    if (timer != null) {
        timer.stop();
    }
```

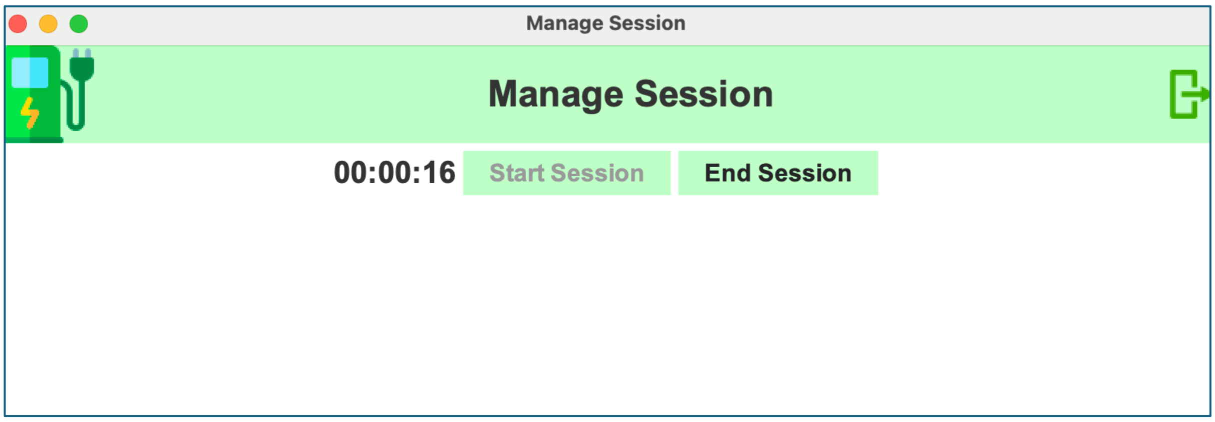
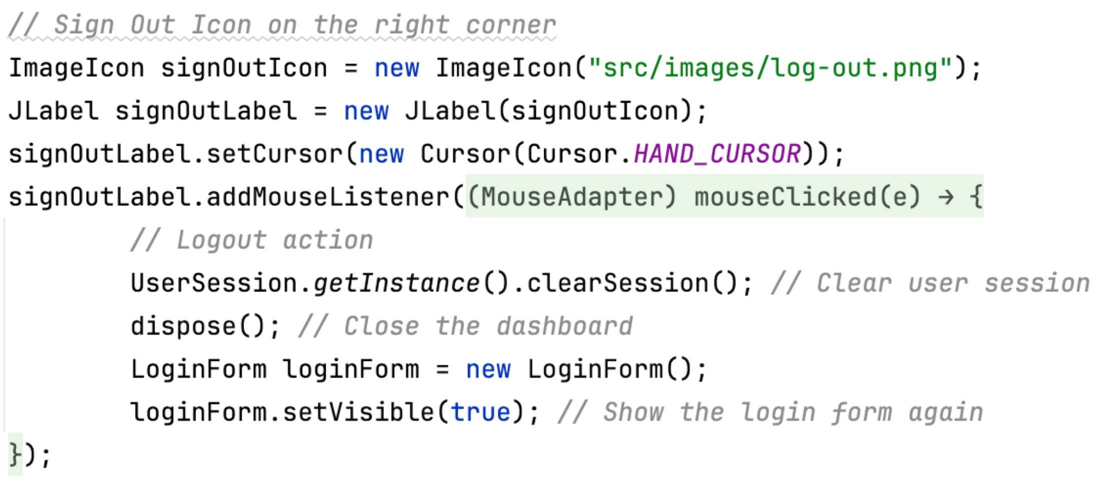


Figure : Showing the timer for start session

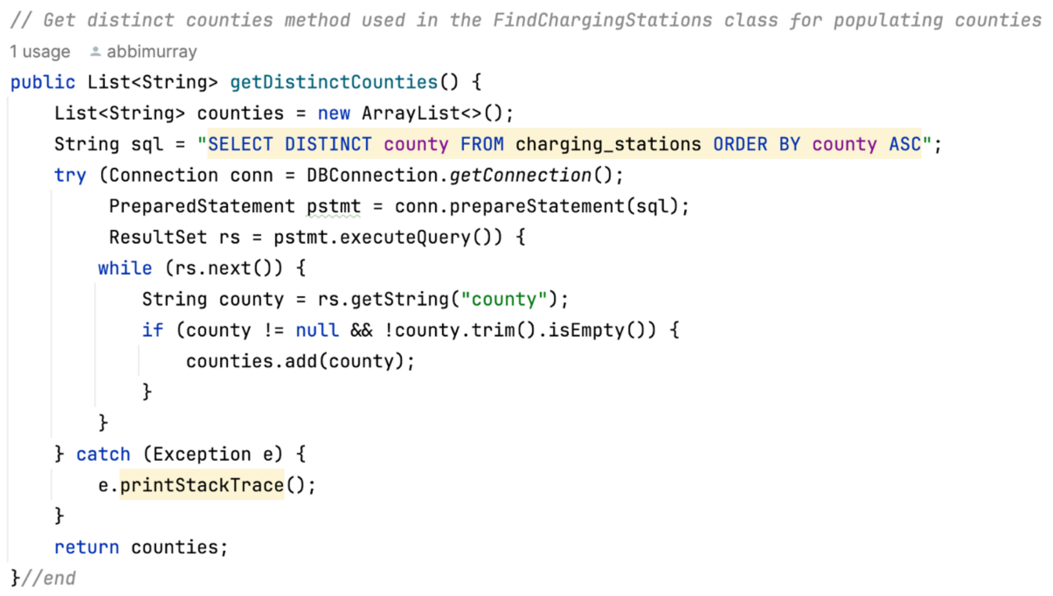
This code snippet is used in most of the pages. It lets the user log out of their account by just pressing the log out icon at the top right corner of each page. I thought this was an accessible and easy way to let users log out.



```java
// Sign Out Icon on the right corner
ImageIcon signOutIcon = new ImageIcon("src/images/log-out.png");
JLabel signOutLabel = new JLabel(signOutIcon);
signOutLabel.setCursor(new Cursor(Cursor.HAND_CURSOR));
signOutLabel.addMouseListener((MouseAdapter) mouseClicked(e) → {
        // Logout action
        UserSession.getInstance().clearSession(); // Clear user session
        dispose(); // Close the dashboard
        LoginForm loginForm = new LoginForm();
        loginForm.setVisible(true); // Show the login form again
});
```

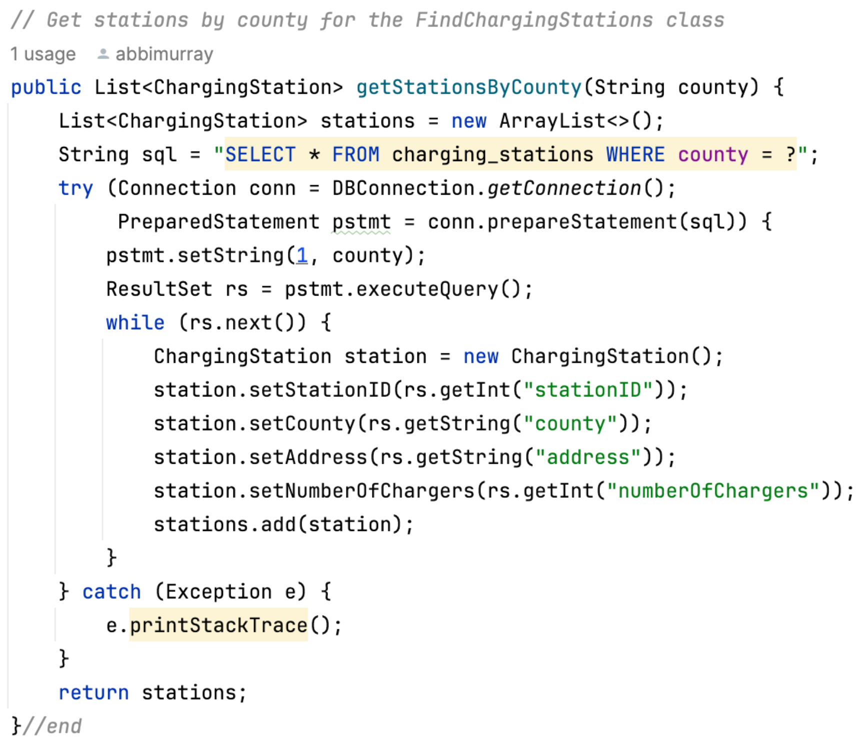
Figure: Sign out icon

The next snippet is methods used to ensure the customer can select charging stations by county. After selecting the county, they then select a station from a list of generated stations. When the desired station is selected they are brought to another page with all chargers at that station. From here they can start/end session or just see information about all the chargers at that station.

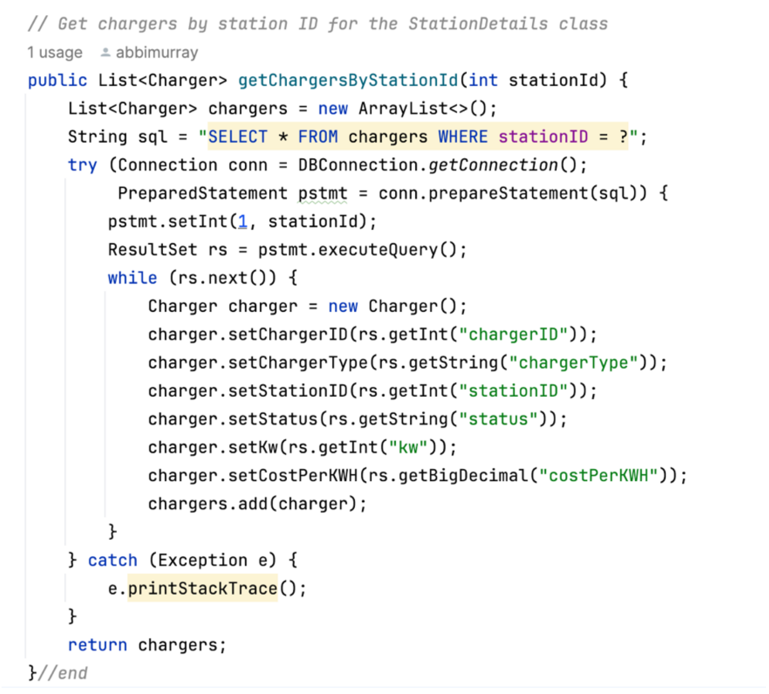


```java
// Get distinct counties method used in the FindChargingStations class for populating counties
1 usage  abbimurray
public List<String> getDistinctCounties() {
    List<String> counties = new ArrayList<>();
    String sql = "SELECT DISTINCT county FROM charging_stations ORDER BY county ASC";
    try (Connection conn = DBConnection.getConnection();
         PreparedStatement pstmt = conn.prepareStatement(sql);
         ResultSet rs = pstmt.executeQuery()) {
        while (rs.next()) {
            String county = rs.getString("county");
            if (county != null && !county.trim().isEmpty()) {
                counties.add(county);
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return counties;
}//end
```

method to select stations for dropdown box

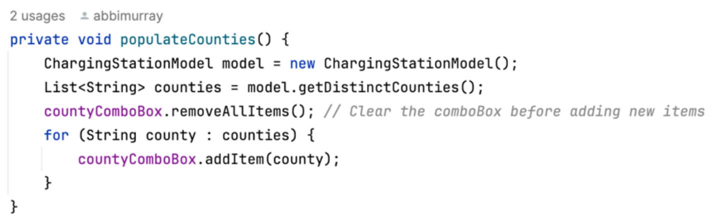


```java
// Get stations by county for the FindChargingStations class
1 usage  abbimurray
public List<ChargingStation> getStationsByCounty(String county) {
    List<ChargingStation> stations = new ArrayList<>();
    String sql = "SELECT * FROM charging_stations WHERE county = ?";
    try (Connection conn = DBConnection.getConnection();
         PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setString(1, county);
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            ChargingStation station = new ChargingStation();
            station.setStationID(rs.getInt("stationID"));
            station.setCounty(rs.getString("county"));
            station.setAddress(rs.getString("address"));
            station.setNumberOfChargers(rs.getInt("numberOfChargers"));
            stations.add(station);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return stations;
}//end
```

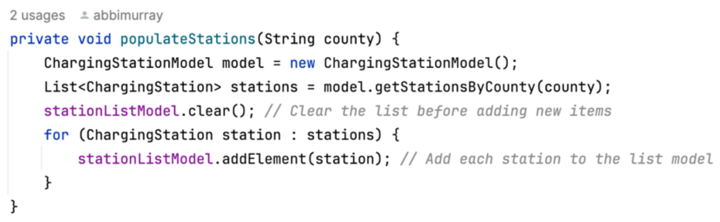


```java
// Get chargers by station ID for the StationDetails class
1 usage   abbimurray
public List<Charger> getChargersByStationId(int stationId) {
    List<Charger> chargers = new ArrayList<>();
    String sql = "SELECT * FROM chargers WHERE stationID = ?";
    try (Connection conn = DBConnection.getConnection();
         PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setInt(1, stationId);
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            Charger charger = new Charger();
            charger.setChargerID(rs.getInt("chargerID"));
            charger.setChargerType(rs.getString("chargerType"));
            charger.setStationID(rs.getInt("stationID"));
            charger.setStatus(rs.getString("status"));
            charger.setKw(rs.getInt("kw"));
            charger.setCostPerKWH(rs.getBigDecimal("costPerKWH"));
            chargers.add(charger);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return chargers;
}//end
```

Methods used to populate the droboxes in FindChargingStation class

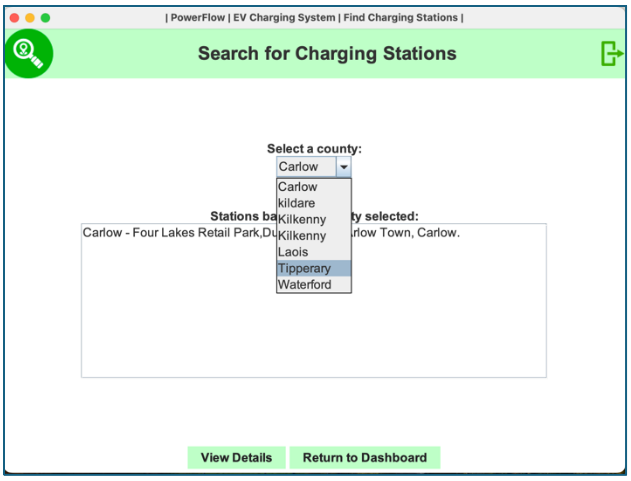


```java
2 usages   abbimurray
private void populateCounties() {
    ChargingStationModel model = new ChargingStationModel();
    List<String> counties = model.getDistinctCounties();
    countyComboBox.removeAllItems(); // Clear the comboBox before adding new items
    for (String county : counties) {
        countyComboBox.addItem(county);
    }
}
```



```java
2 usages   abbimurray
private void populateStations(String county) {
    ChargingStationModel model = new ChargingStationModel();
    List<ChargingStation> stations = model.getStationsByCounty(county);
    stationListModel.clear(); // Clear the list before adding new items
    for (ChargingStation station : stations) {
        stationListModel.addElement(station); // Add each station to the list model
    }
}
```

1. show the search based off county selection


| PowerFlow | EV Charging System | Find Charging Stations |

Search for Charging Stations

Select a county:

Carlow

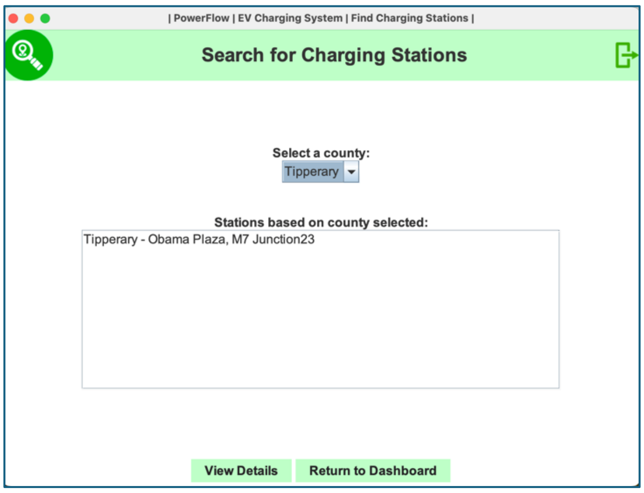
Carlow
kildare
Kilkenny
Kilkenny
Laois
Tipperary
Waterford

Stations ba... ty selected:

Carlow - Four Lakes Retail Park,Du... rlow Town, Carlow.

View Details | Return to Dashboard

2. Populate based off the county selected


| PowerFlow | EV Charging System | Find Charging Stations |

Search for Charging Stations

Select a county:

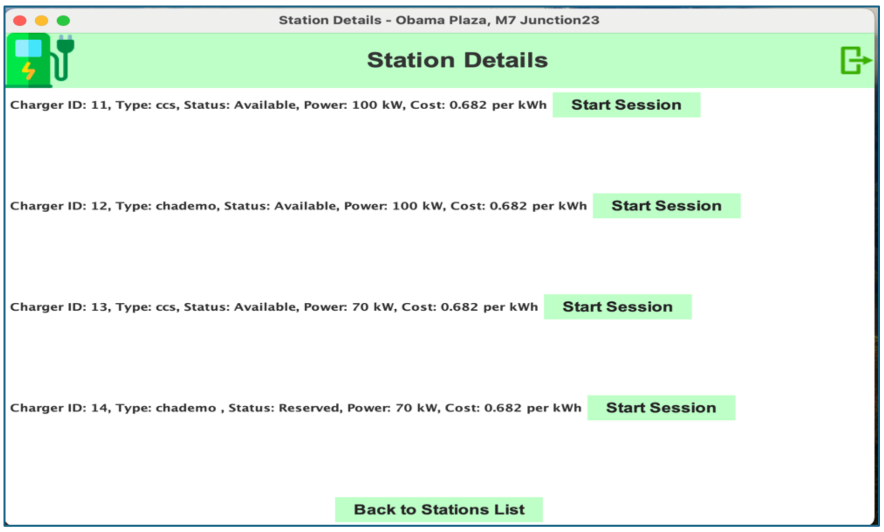
Tipperary

Stations based on county selected:

Tipperary - Obama Plaza, M7 Junction23

View Details | Return to Dashboard

3. Display details of chargers based off the selected station


Station Details - Obama Plaza, M7 Junction23

Station Details

Charger ID: 11, Type: ccs, Status: Available, Power: 100 kW, Cost: 0.682 per kWh | Start Session

Charger ID: 12, Type: chademo, Status: Available, Power: 100 kW, Cost: 0.682 per kWh | Start Session

Charger ID: 13, Type: ccs, Status: Available, Power: 70 kW, Cost: 0.682 per kWh | Start Session

Charger ID: 14, Type: chademo , Status: Reserved, Power: 70 kW, Cost: 0.682 per kWh | Start Session

Back to Stations List

Tests

For Unit testing Junit and Mockito were used. Mockito provides a mocking framework that enables the creation of mock objects. This allowed components to be tested in isolation. The tests focused on the logic of the components without the implementation of their dependencies.

Test Number	Test Name	Result	Pass / Fail
Test 1	testUpdateCustomerDetailsValid	Updated details	P
Test 2	testUpdateCustomerDetailsInvalid	Didn't update details, gave error message	P
Test 3	testGetCustomerByEmail	Fetched customer by email	P
Test 4	testSaltGeneration()	Generated salt correctly	P
Test 5	testHashPasswordWithSHA256()	Generated correctly	P
Test 6	testAddPaymentMethod_ValidDetails_ReturnsSuccess	Added payment details correctly	P
Test 7	testAddPaymentMethod_InvalidDetails_ReturnsErrorMessage	Didn't add details, gave error message	P
Test 8	testGetPaymentMethodsForCustomer	Fetched payment methods	P
Test 9	testUpdatePaymentMethod_Successful	Updated payment methods	P
Test 10	testGetReservationsForCustomer	Fetched Reservations	P
Test 11	testAddReservation_ChargerAvailable	Added Reservation	P
Test 12	testAddReservation_ChargerNotAvailable	Didn't add reservation, gave message	P
Test 13	testDeleteReservation_Successful	Deleted Reservation	P
Test 14	testDeleteReservation_Failure	Could not delete reservation	P
Test 15	testUpdateReservation_ChargerAvailable	Updated Successfully	P
Test 16	testUpdateReservation_ChargerNotAvailable	Couldn't update reservation	P
Test 17	testValidEmail	Valid email passed	P

Test 18	testInvalidEmail	Invalid email caught	P
Test 19	testValidPassword	Valid Password passed	P
Test 20	testInvalidPassword	Invalid password caught	P
Test 21	testValidName	Valid name passed	P
Test 22	testInvalidName	Invalid name caught	P
Test 23	testValidPhone	Valid phone passed	P
Test 24	testInvalidPhone	Invalid phone caught	P