**Customer Account**

1. View existing business or private customers

**Preconditio**n: User is successfully logged into the system.

**Test steps:**

* In the search bar
* Type “business” or “individual” and press enter on the keyboard.

**Expected Result:** business or individual customers should be listed on the table.

**Actual result:** The list of either business or individual customers is listed.

**Pass/fail:** PASS

2. View existing business and private customers

**Precondition**: User search to view business or individual customers

**Test Steps**:

* User should clear search bar and press enter.

**Expected Result**: A list of all existing business and individual customers should be displayed on the table.

**Actual Result**: The list of all existing customers is displayed.

**Pass/fail**: PASS

3. Search for a business customer, using partial name, and show contact details and list of vehicles.

**Precondition**: Search bar is clear and customer table has a list of existing business customers.

**Test steps**:

* User should type part of the customer name (i.e. First name or last name) and press enter.
* Click on the customer in the table to show the contact details and vehicles owned by the customer.

**Expected Result**: Customer name and details should be listed in the table, contact details and vehicle details text area.

**Actual Result**: Customer name and details are listed in the table, contact details text area and vehicle owned listview.

**Pass/fail**: PASS

4. Search for individual customer, using partial name, and show contact details and list of vehicles.

**Precondition**: Search bar is clear and all existing customers are listed in the table.

**Test steps**: Type part of customer name in the search bar (first name or last name) and press enter on the keyboard. Click on the customer in the table to show the contact details and vehicles owned.

**Expected Result**: Customer name and details should be listed in the table, contact details text area and vehicle details listview.

**Actual Result**: Customer name and details are listed in the table, contact details text area and vehicles listview.

**Pass/fail**: PASS

5. For an existing customer, show bills for past completed bookings and show settlement status.

**Precondition**: All existing customers are listed on the table.

**Test Steps**:

* Search customer using partial name or ID,
* Select customer on the table
* Click on the account button to view bookings, cost and settlement status.

**Expected result**: Clicking the account button should open up a pop up listing the all the customer’s bookings and settlement status.

**Actual Result**: Account button opens a pop listing all customer bills, bookings and settlement status.

**Pass/fail**: PASS

6. Changing settlement status for an existing customer’s bookings

**Preconditions**: An existing customer has been selected from the table.

**Test Steps**:

* Click on the account button
* Select a bill/booking with settle status unpaid,
* Click on the payment status choice box
* Selected “paid” then click on the save button.

**Expected Result**: The bill/booking selected should have the settlement status updated from “unpaid” to “paid”

**Actual Result**: The settlement status for the bill/booking selected is updated.

**Pass/fail**: PASS

7. Edit a customer record

**Precondition**: All existing customers are listed on the table.

**Test Steps**:

* Select a customer
* Click on the edit button
* Edit information
* Click on save.

**Expected Result**: The customer details should be updated and shown on the table.

**Actual Result**: The customer details is updated and shown on the table.

**Pass/fail**: PASS

8. For an existing customer show vehicle records

**Precondition**: All existing customers are listed on the table.

**Test Steps**:

* Select a customer from on the table
* Click on vehicles in vehicles owned list view.
* Click on booking in bookings list view.

**Expected Result**: All vehicles owned by the customer should be shown in the vehicles listview, vehicle details should be shown in the vehicle details text area, all bookings and parts used should also be shown.

**Actual Result**: All vehicles owned by the customer is listed, the vehicle details are also listed. All bookings and parts used are also listed.

**Pass/fail**: PASS

9. Delete a customer record

**Precondition**: Customer already exists

**Test Steps**:

* Select a customer from the table
* Click on delete
* Click “ok” when alert message pops up.

**Expected Result**: Customer and customer records should be deleted from the table.

**Actual Result**: Customer and customer record is deleted.

**Pass/fail**: PASS

10. Create new customer record

**Precondition**: User is logged into the system

**Test Steps**:

* Click on add button
* Enter all fields correctly
* Click on create button.

**Expected Result**: A new customer should be added to the table.

**Actual Result**: A new customer is added to the table.

**Pass/fail**: PASS

11. Logout, close application and log back in. Show created record is persistently stored.

**Precondition**: User is logged currently logged into the system.

**Test steps**:

* Click on logout on the top left of the system
* Close the application
* Reopen the application
* Enter valid userID and password of an existing user into the respective fields
* Click login button
* Click on customer account tab
* Check that all records created are persistently stored.

**Expected result**: The currently logged in user should be logged out, then the application should be closed successfully, user should reopen the application, enter the valid userID and password and should be logged in successfully. When the user clicks on the customer tab, all information previously stored should be persistent and showed.

**Actual Result**: The currently logged in user is logged out successfully and has closed the application and reopened it successfully. The new user logs in successfully after entering the valid userID and password in the fields and clicking log in. The user is able to click on the customer tab and view all stored details.

**Pass/fail**: PASS