Connection Details

Homepage URL: https://zeno.computing.dundee.ac.uk/2018-ac32006/team10/

Login details

Customer

Email: annahutch@gmail.com

Password: 1234qwer

Sales Staff

Email: cowdeng145@gmail.com

Password: 1234qwer

Branch Manager

Email: traywray@gmail.com

Password: 1234qwer

Executive

Email: emchopey142@gmail.com

Password: 1234qwer
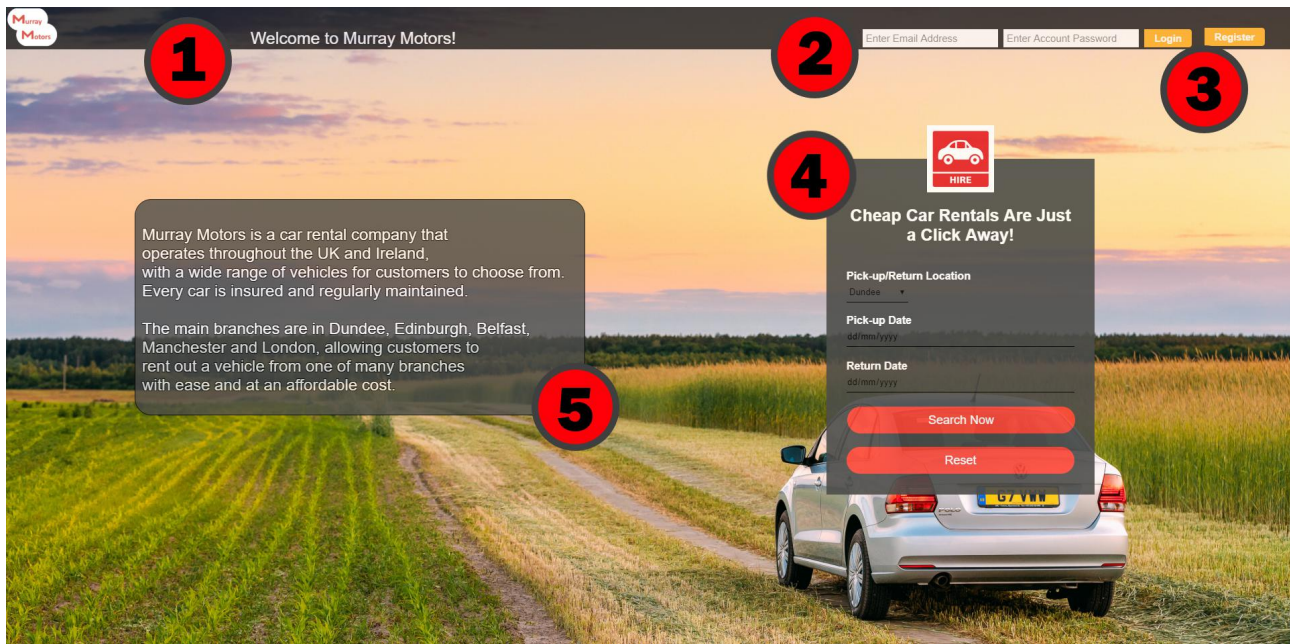
User Guide

Customer View

Figure 1 - Home Page

1. This is a responsive message which will give feedback correspondingly based on customers' actions. This will be explained in the following sections.

2. Login with the email address and password of customers or staff members. If login details are incorrect, the responsive message will display "Username and/or password incorrect".

3. Alternatively, if they do not have an account yet, they can click 'Register' to create one.

4. Customers can choose where they want to rent a car from as well as choose the date range for the rental. Click 'Search now' to start browsing cars. If customers choose an invalid date range, i.e. if the pick up date is in the past, the feedback message will become "Invalid date range". Click 'Reset' to clear the form.
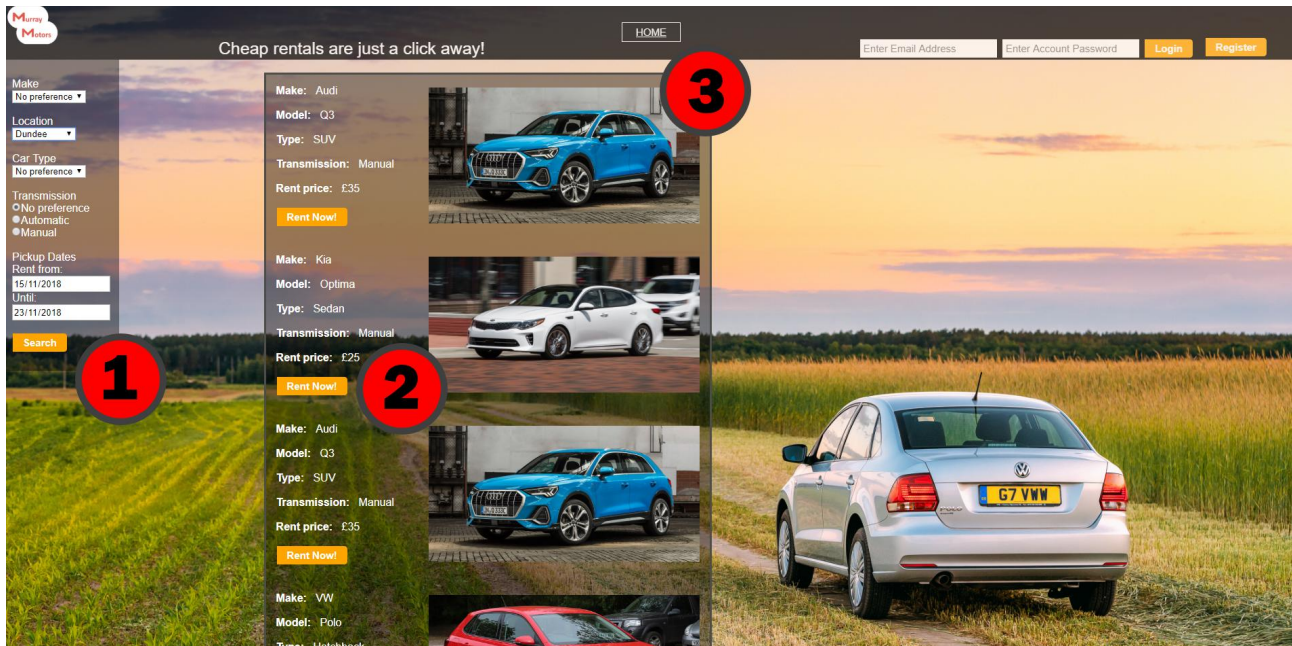
5. This is information about the company.

`Figure 2 - Car Browse Page

1. Customers can use this filter to narrow down the car options.
2. Click 'Rent Now' when they decided which car to rent. The details of the car being rented can be seen here as well.
3. Click 'Home' to redirect to home page.

Customers have to login before proceeding to payment. When they try to book a car without logging in, they will be redirected to the **Login/Register Page**:
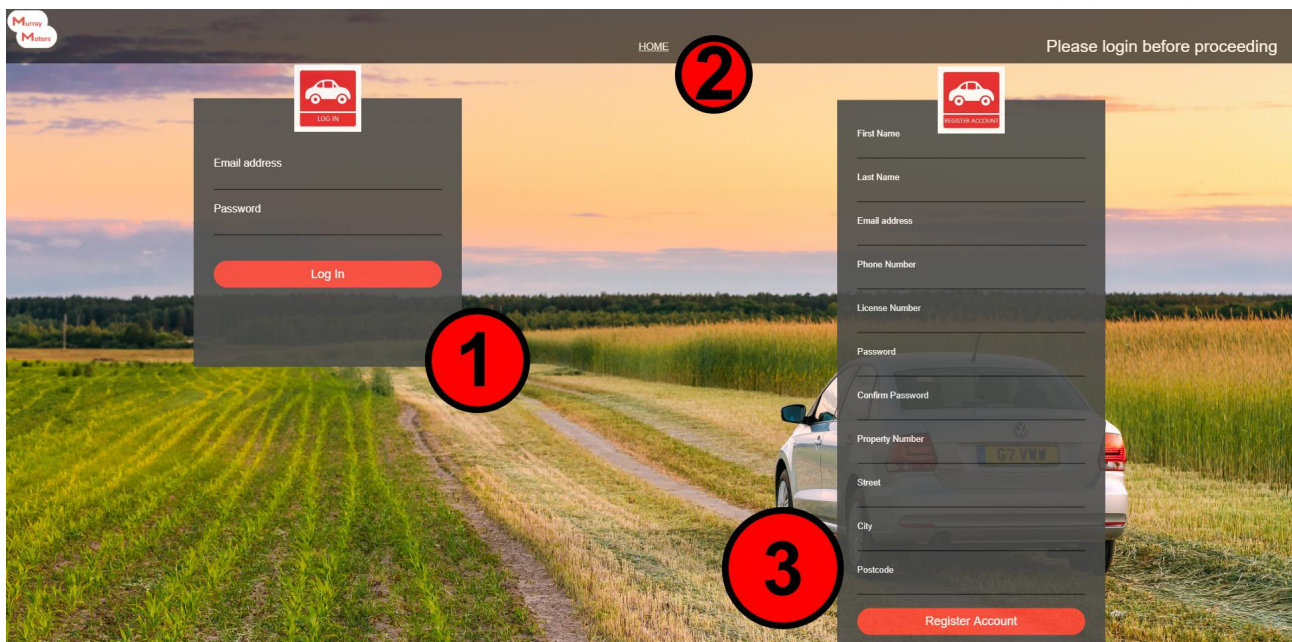

Figure 3 - Login/Register Page

1. Log into customer account.
2. Redirect to home page
3. Fill out the form to register a new account.

If the user does not already have an account, they can create one from the home page by clicking the 'register' button. This page is identical to [Figure 3], with the exception of 3.1 and that 3.3 is centered.

Once customers are logged in, they will enter into the **Booking Page**:

Figure 4 - Booking Page

1. Details about the car which the customers chose, rental details, and the branch location where they can double check the details are correct.
2. Put in credit card details in order to complete booking.
3. Once the booking is complete, the user will be redirected back to the home page and the responsive message will inform the user that the booking succeeded. The company system will generate a lease for the rental and update the chosen car as not available for other customers to rent in the database. If customers click 'home' the system will NOT generate a lease. Customers need to go through the whole process again in order to successfully rent a car.

Once customers login, they can click 'Profile' from the home page to enter the **Profile Page**:

Figure 5 - Profile Page

1. This will provide customers feedback based on their editing action.

2. Customers can only see their own information and details about their car preferences.
3. If they want to update any part of their information, just click 'Edit'.
4. Click 'Logout' when they want to leave the company website.

Once our staff for sales, managers, and executives log in from the home page:

Sales View

Figure 6 - Staff Profile Page

1. Navigation sidebar. Clicking any of these buttons will change what table is being displayed. For example, clicking 'Valet' will show a table of Valet services.
2. Edit bars. Input information here to change personal information.
3. A normal sales staff can only view his or her own personal information. Click the 'Edit' button to reveal or conceal the edit bars.

Figure 7 - Staff Car Stock Page

1. Use the options in this bar to refine the results by different filters. This can be used, for example, to only show cars that are currently available or only manual cars.
2. Navigation sidebar.

3. Database table. Displays information of the appropriate table along with any filters set by the user. In the case of a sales staff member, only data relevant to their branch will be shown.

Manager View
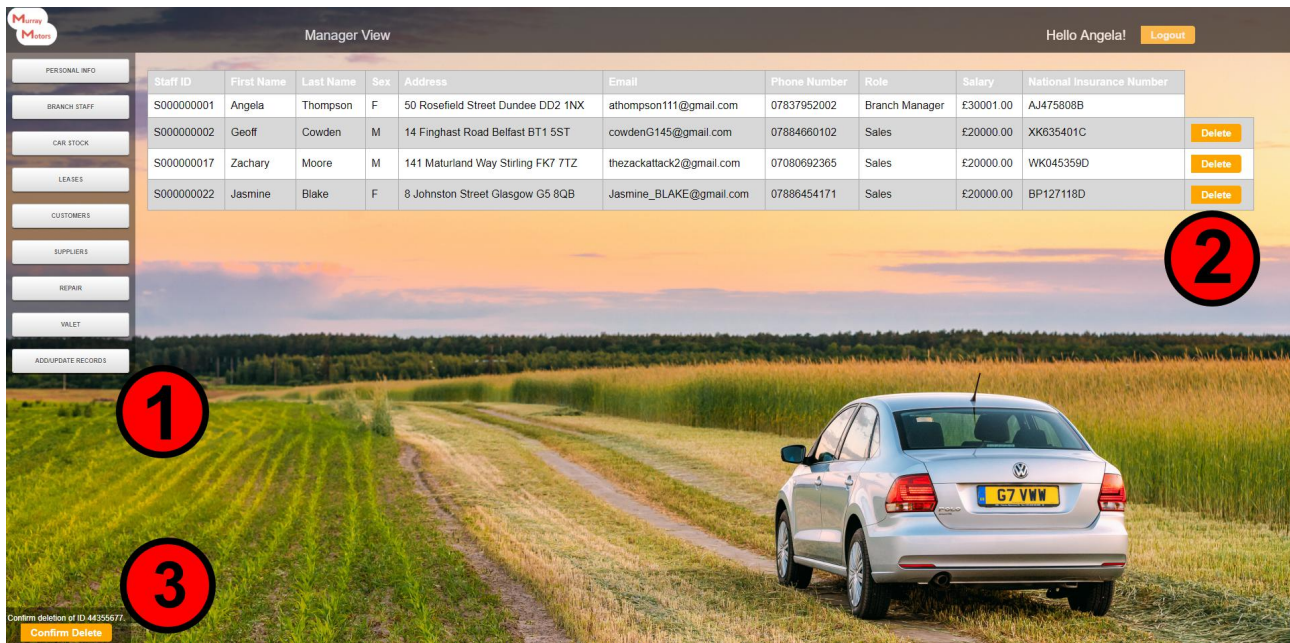
Figure 8 - Manager Overview

1. Similar navigation sidebar to view data only relevant to their branch. Unlike sales staffs, managers can add or edit record to tables.
2. Managers can also delete data by clicking 'Delete'.
3. Once 'Delete' has been clicked, this prompt will appear to confirm the deletion.
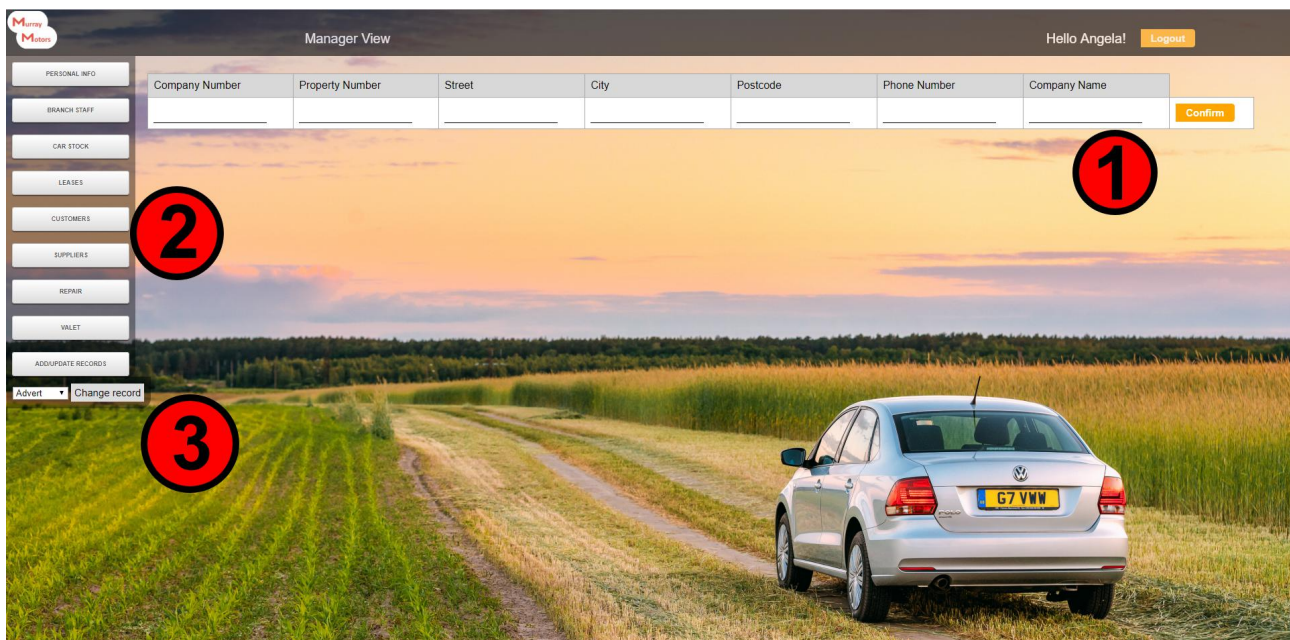
Figure 9 - Manager Add/Edit Record

1. Enter information here to add or edit a record. To edit a record, the left-most field must match an already existing record. To add a new record, the left-most field needs to be unique.
2. Navigation sidebar.
3. In order to change what record is being added/edited, use the drop down menu to select the type of record to add or edit in the database. Click 'Change record' to switch

Executive View

Figure 10 - Executive View

1. Similar to manager view, executives can also use the navigation sidebar and add or edit records.
2. Unlike managers who can only view information related to their own branch, executives can see all information across branches.
3. Narrow down the results of staff information by applying specific filters.

Technical Annex

Branch View:

```sql
SELECT
    COUNT(DISTINCT `staff`.`staff_number`) AS `staff`,
    COUNT(DISTINCT `car`.`VIN`) AS `cars`,
    COUNT(DISTINCT `lease`.`lease_number`) AS `leases`,
    (SELECT
            SUM(`staff`.`salary`)
        FROM
            `staff`
        WHERE
            (`staff`.`BranchNumber` = `branch`.`branch_number`)) AS `salary`,
    `address`.`property_number` AS `property_number`,
    `address`.`street` AS `street`,
    `address`.`city` AS `city`,
    `address`.`postcode` AS `postcode`,
    `branch`.`branch_number` AS `branch_number`,
    `branch`.`Manager_StaffNumber` AS `managerid`
FROM
    (((((`branch`
    LEFT JOIN `staff` ON ((`branch`.`branch_number` = `staff`.`BranchNumber`)))
    LEFT JOIN `address` ON ((`branch`.`AddressID` = `address`.`address_id`)))
    LEFT JOIN `car` ON ((`branch`.`branch_number` = `car`.`BranchNumber`)))
    LEFT JOIN `lease` ON ((`car`.`VIN` = `lease`.`CarVIN`)))
GROUP BY `branch`.`branch_number`
```

Count and distinct are used to get a total number of unique staff members, cars, and leases across the database. A sub-select sum is used to get the total salary of all staff members per branch. The

view left joins the branch table to staff, address, car and lease. The view is indexed by branch number and is only accessible by executive users.

Payroll View:

```sql
SELECT
    `staff`.`BranchNumber` AS `BranchNumber`,
    `address`.`property_number` AS `property_number`,
    `address`.`street` AS `street`,
    `address`.`city` AS `city`,
    `address`.`postcode` AS `postcode`,
    `staff`.`staff_number` AS `staff_number`,
    `staff`.`first_name` AS `first_name`,
    `staff`.`last_name` AS `last_name`,
    `staff`.`sex` AS `sex`,
    `staff`.`email` AS `email`,
    `staff`.`phone_number` AS `phone_number`,
    `staff`.`role` AS `role`,
    `staff`.`national_insurance_number` AS `nin`,
    CAST((`staff`.`salary` / 12) AS DECIMAL (10 , 2 )) AS `Monthly_Salary`
FROM
    (`staff`
    JOIN `address` ON ((`staff`.`AddressID` = `address`.`address_id`)))
```

Monthly_Salary is returned as the salary of each staff member divided by 12 in order to get the salary per month. It is casted as a decimal with 2 decimal places to round accurately to real currency. The view joins the staff table to address. The view is indexed by staff number, and is only accessible by executive users.

Customer View:

```sql
SELECT
    `customer`.`customer_number` AS `customer_number`,
    `customer`.`first_name` AS `first_name`,
    `customer`.`last_name` AS `last_name`,
    `customer`.`email` AS `email`,
    `customer`.`phone_number` AS `phone_number`,
    `customer`.`license_number` AS `license_number`,
    `address`.`property_number` AS `property_number`,
    `address`.`street` AS `street`,
    `address`.`city` AS `city`,
    `address`.`postcode` AS `postcode`,
    COALESCE(`preferences`.`make`, 'Any') AS `make`,
    COALESCE(`preferences`.`model`, 'Any') AS `model`,
    COALESCE(`preferences`.`type`, 'Any') AS `type`,
    COALESCE(`preferences`.`transmission`, 'Any') AS `transmission`,
    (CASE
        WHEN
            EXISTS( SELECT
                    1
                FROM
                    `lease`
                WHERE
                    (`lease`.`CustomerNumber` = `customer`.`customer_number`))
        THEN
            'Yes'
        ELSE 'No'
    END) AS `lease`,
    (CASE
        WHEN (`customer`.`membership` = 1) THEN 'Yes'
        ELSE 'No'
    END) AS `membership`
FROM
    (((`customer`
    JOIN `address` ON ((`customer`.`Address_ID` = `address`.`address_id`)))
    LEFT JOIN `preferences` ON ((`customer`.`customer_number` = `preferences`.`Customer_Number`)))
    LEFT JOIN `lease` ON ((`customer`.`customer_number` = `lease`.`CustomerNumber`)))
```

COALESCE function is used to return "Any" where a preference is null. "lease" is returned as a "Yes" or "No" by using an exists on a subselect that checks if a customer and lease share the same customer number. If they do, "Yes" is returned to indicated that customer has a lease. "membership" is returned as a "Yes" or "No" for values 1 and 0 respectively. The view joins customer to address and left joins customer to preferences and lease. The view is indexed by customer number, and is accessible by all users. A customer can only see their own data whereas sales, manager and executive views can see all customer data.

Car Stock View:

```sql
SELECT
    `car`.`VIN` AS `VIN`,
    `car`.`BranchNumber` AS `BranchNumber`,
    `car`.`InsuranceNumber` AS `InsuranceNumber`,
    `supplier`.`company_name` AS `company_name`,
    `car`.`type` AS `type`,
    `car`.`make` AS `make`,
    `car`.`model` AS `model`,
    `car`.`transmission` AS `transmission`,
    `car`.`colour` AS `colour`,
    `car`.`availability` AS `availability`,
    `car`.`rent_per_day` AS `rent_per_day`,
    `car`.`last_checkup` AS `last_checkup`,
    `insurance`.`provider` AS `provider`,
    `insurance`.`insurance_type` AS `insurance_type`,
    (CASE
        WHEN (`car`.`availability` = 1) THEN 'Yes'
        ELSE 'No'
    END) AS `available`,
    (CASE
        WHEN
            EXISTS( SELECT
                    1
                FROM
                    `lease`
                WHERE
                    (`lease`.`CarVIN` = `car`.`VIN`))
        THEN
            'Yes'
        ELSE 'No'
    END) AS `leased`,
    (CASE
        WHEN
            EXISTS( SELECT
                    1
                FROM
                    `adverts`
                WHERE
                    (`adverts`.`CarVIN` = `car`.`VIN`))
        THEN
            'Yes'
        ELSE 'No'
    END) AS `advertised`
FROM
    ((((`car`
    LEFT JOIN `lease` ON ((`car`.`VIN` = `lease`.`CarVIN`)))
    JOIN `supplier` ON ((`car`.`Supplier_CompanyNumber` = `supplier`.`company_number`)))
    LEFT JOIN `adverts` ON ((`car`.`VIN` = `adverts`.`CarVIN`)))
    JOIN `insurance` ON ((`car`.`InsuranceNumber` = `insurance`.`insurance_number`)))
```

"available" is returned as "Yes" or "No" when availability is 1 or 0 respectively. "leased" and "advertised" are returned as "Yes" through an exists sub-select statement if it is found that a car shares its VIN with a lease or advert. Otherwise, "No" is returned. The view joins the car table with supplier and insurance, and left joins the car table with lease and adverts. The view is only accessible by sales, manager and executive users.