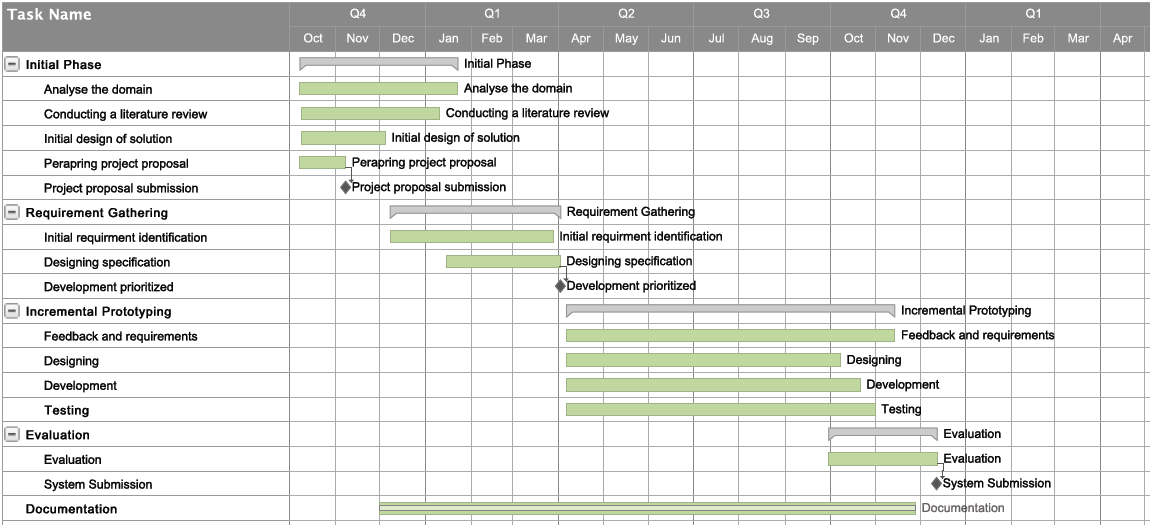
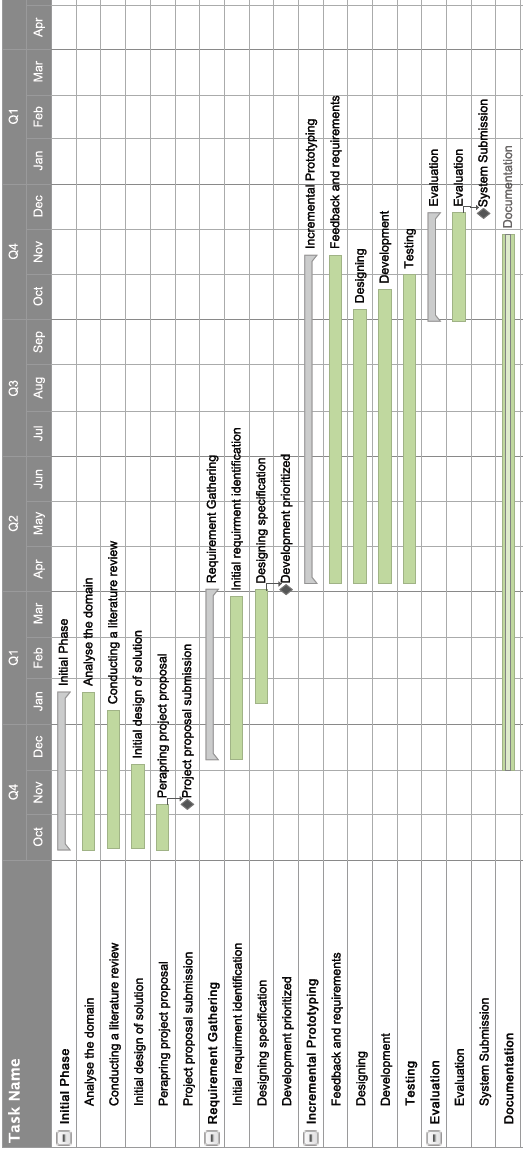
# Appendix A

**Planned Project Plan**

****

# **Actual Project Plan**



# Major change that was seen between planned and actual plan was the project methodology. Initially it was planned to do the project in a methodology that is lean more towards the traditional approach. But after realizing the importance in agile methodology projects requirement identification, designing development, testing were done concurrently. Constant user feedback was a big advantage for the developers.

As this is an individual project, first six month was conducted while studying at the university. In that period also there were allocating time for development and documentation while doing loads of assessment works and exams. After exams were over industrial trainings started. After going to the industry also time was more limited to the project. Slight deviations in the plan are caused by those factors. But as an overall project was conducted within the time period very successfully.

# Appendix B

**Data Gathering and Analysis**

**Pre Prepared Question Plan for Interviews**

|  |  |
| --- | --- |
| Question | Resource Person Type |
| What are you responsibilities? Can you explain your task briefly? | Claim Agent |
| What are the difficulties you face when travelling to the accident location? | Claim Agent |
| What kind of information and equipment you carry when you travel to accident locations. | Claim Agent |
| How do you validate the client information with company database? | Claim Agent |
| What kind of difficulties do you face when handling claim requests? | Operating Staff |
| What do you think about current practiced conducted when assessing claims? | Managerial Level |
| Do you think automating the claim assessment to some extent will help to increase profit margins? | Managerial Level |

Table 1: Interview question plan

Source: Author

**Summary of Fact Findings**

Conducting interviews with insurance sector employees from time to time in the development process helped me to understand the types of employees, agents and their current performance as average, power of human resource, level of technological skills of employees and their general attitude and capability of being flexible for any type of techno sophisticated environment. Further it was discussed about the steps of creating vehicle insurance policy, customer registration, detail description about the agreements, legal situation, especially how they give approval to the claim currently and existing verification methods and current annual and premium payment methods.

Drawbacks of the current procedure were found out by interviewing customers and internals stakeholders. As an example customer have to spend lot of time in order to collect their claiming amount, and sometimes customers are receiving lesser amount than what they deserve due to inaccurate assessment performed by the claiming insurance agent. Company also face so many problems by using the current procedure, like, when the assessed claiming amount is more than the actual claiming amount due to the manual calculation performed by the agent.

After several interviews with claim agents, their main roles and responsibilities were identified. They discussed their difficulties when handling with documents and other equipment’s such as camera as well. After getting friendly with them it led to the conduct of observation technique as well. As the developer I got a chance to witness a real time claim assessment procedure.

Apart from interviews and observations other main source of data were the document reviews. Certain insurance registrations were provided. Not only that claim assessment forms also were supplied for authors reference which helped author to design the database. Since the resource persons have asked not to publish the forms sample are not attached.

# Appendix C

**System Design**

**Database Tables**

This table contains the payment details related to a claim record. Payment id act as a foreign key to claim table. The below tables driver table, spare part payment table, third party details tables also contain details related to a particular claim. Therefore they are also connected to claim tables via foreign key.

|  |  |  |
| --- | --- | --- |
| Claim Payment Table | | |
| Attribute | **Data Type** | **Length** |
| Paymentid | Int | - |
| Garagecost | Float | - |
| Othercosts | Float | - |
| Deductions | Float | - |
| InsurancePercentage | Float | - |

Table 2: Claim Payment Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Driver Table | | |
| Attribute | **Data Type** | **Length** |
| Driverid | Int | - |
| Drivername | nvarchar | 50 |
| Licenceno | nvarchar | 20 |
| Expirationdate | DateTime | 15 |
| Drivernic | nvarchar | 20 |

Table 3: Driver Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Spare part Payment Table | | |
| Attribute | **Data Type** | **Length** |
| Paymentid | Int | - |
| Sparepartid | Int | - |
| Sparepartqty | Float | - |
| Sparepartcost | Float | - |

Table 4: Spare part Payment Table:

Source: Author

|  |  |  |
| --- | --- | --- |
| Third Party Details Table | | |
| Attribute | **Data Type** | **Length** |
| Thirdpartydetailid | Int | - |
| Vehicleregistrationno | nvarchar | 50 |
| Ownername | nvarchar | 50 |
| Owneraddress | nvarchar | 100 |
| Reneweldate | DateTime | 10 |
| Specialnotes | nvarchar | Max |
| Victimname | nvarchar | 50 |
| Victimaddress | nvarchar | 100 |
| Damagenature | nvarchar | 100 |
| Thirdpartyclaimant | nvarchar | 50 |
| Claimamount | Float | - |

Table 5: Third Party Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Vehicle Table | | |
| Attribute | **Data Type** | **Length** |
| Vehicleid | Int | - |
| Vehicletypeid | Int | - |
| Manufactureid | Int | - |
| Model | nvarchar | 25 |
| Makeyear | DateTime | 10 |
| Fueltype | nvarchar | 10 |
| Enginecpacity | nvarchar | 20 |
| Seatingcapacity | nvarchar | 10 |
| Carryingcapacity | nvarchar | 10 |
| Presentvalue | Float | - |
| Dutyfreevalue | Float | - |

Table 6: Vehicle Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Customer Vehicle Table | | |
| Attribute | **Data Type** | **Length** |
| Vehicleid | Int | - |
| Customerid | Int | - |
| Registrationno | nvarchar | 15 |
| Color | nvarchar | 20 |
| Engineno | nvarchar | 50 |
| Chassisno | nvarchar | 50 |
| Currentdamages | nvarchar | Max |
| Absoluteowner | nvarchar | 10 |
| Financialrights | nvarchar | 50 |
| Extrafittins | nvarchar | Max |
| Usage | nvarchar | Max |

Table 7: Customer Vehicle Table

Source: Author

Claim request table holds claim request data which are sent by the clients. When a new record is added state will be always pending. After an agent respond to it they can change it to responded state.

|  |  |  |
| --- | --- | --- |
| Claim Request Table | | |
| Attribute | **Data Type** | **Length** |
| Id | Int | - |
| Policyid | Int | - |
| Gps | nvarchar | 50 |
| State | nvarchar | 50 |
| Submittime | DateTime | - |
| Respondtime | DateTime | - |

Table 8: Claim Request Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Manufacturer Table | | |
| Attribute | **Data Type** | **Length** |
| ManufactureId | Int | - |
| ManufactureName | nvarchar | 50 |

Table 9: Manufacturer Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Spare part Category Table | | |
| Attribute | **Data Type** | **Length** |
| Sparecategoryid | Int | - |
| Sparecategoryname | nvarchar | 50 |

Table 10: Spare part Category Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Spare part Table | | |
| Attribute | **Data Type** | **Length** |
| Sparepartid | Int | - |
| Sparepartname | nvarchar | 50 |
| Sparepartcategory | Int | - |
| Sparepartmanufacturer | Int | - |
| Spareparmanufacyear | nvarchar | 10 |
| Sparepartunitcost | Float | - |

Table 11: Spare part Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Vehicle Type Table | | |
| Attribute | **Data Type** | **Length** |
| VehicleTypeID | Int | - |
| VehicleTypeName | nvarchar | 50 |

Table 12: Vehicle Type

Source: Author

|  |  |  |
| --- | --- | --- |
| Garage Table | | |
| Attribute | **Data Type** | **Length** |
| GarageID | Int | - |
| GarageName | nvarchar | 50 |
| GarageLocation | nvarchar | 50 |
| GarageTP | nvarchar | 20 |
| Email | nvarchar | 100 |

Table 13: Garage Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Tow Truck Table | | |
| Attribute | **Data Type** | **Length** |
| Id | Int | - |
| Name | nvarchar | 50 |
| Location | nvarchar | 50 |
| Telephone | nvarchar | 20 |
| Email | nvarchar | 100 |

Table 14: Tow Truck Table

Source: Author

|  |  |  |
| --- | --- | --- |
| Employee Table | | |
| Attribute | **Data Type** | **Length** |
| EmpId | Int | - |
| EmpName | nvarchar | 50 |
| EmpPhone | nvarchar | 20 |
| EmpEmail | nvarchar | 100 |
| EmpBranch | nvarchar | 20 |

Table 15: Employee Table

Source: Author

|  |  |  |
| --- | --- | --- |
| User Accounts Table | | |
| Attribute | **Data Type** | **Length** |
| Account\_Id | Int | - |
| Employee\_id | Int | - |
| Username | nvarchar | 50 |
| Password | nvarchar | 50 |
| Type | nvarchar | 20 |

Table 16: User Account Table

Source: Author

**Interfaces of the Online Claiming System (Web Application)**

Interface for viewing garages. You can also select add/edit/remove options from this view. You can filter by location also in this view.

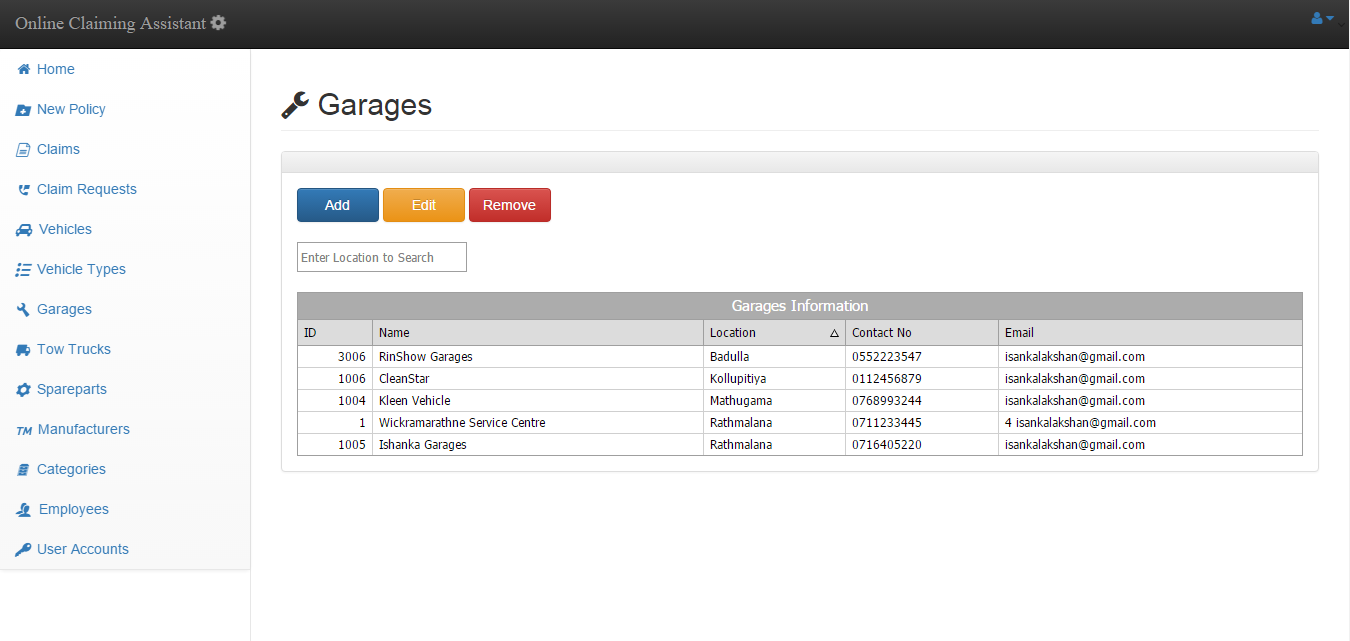


Figure 1: Garages Interface

Source: Author

Interface to select add a new garage.

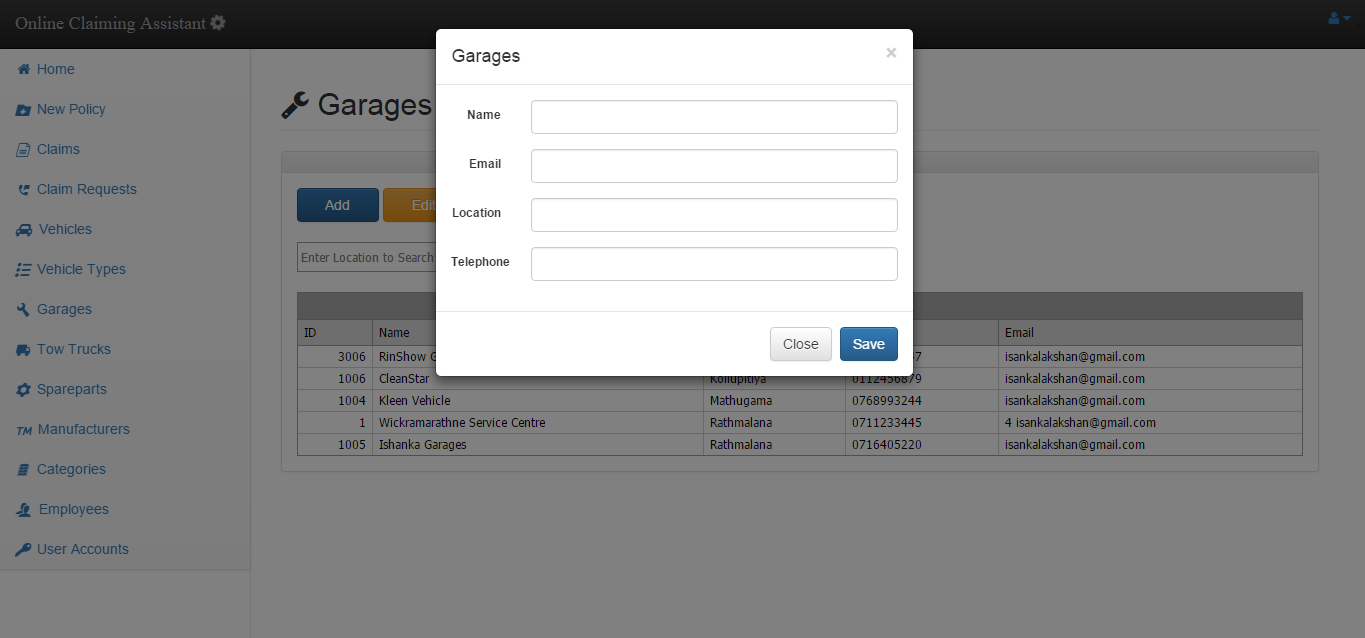


Figure 2: Add New Garage Interface

Source: Author

Interface of editing a garage. First you have to select a record from the data grid and click edit. Then it will be loaded like shown.

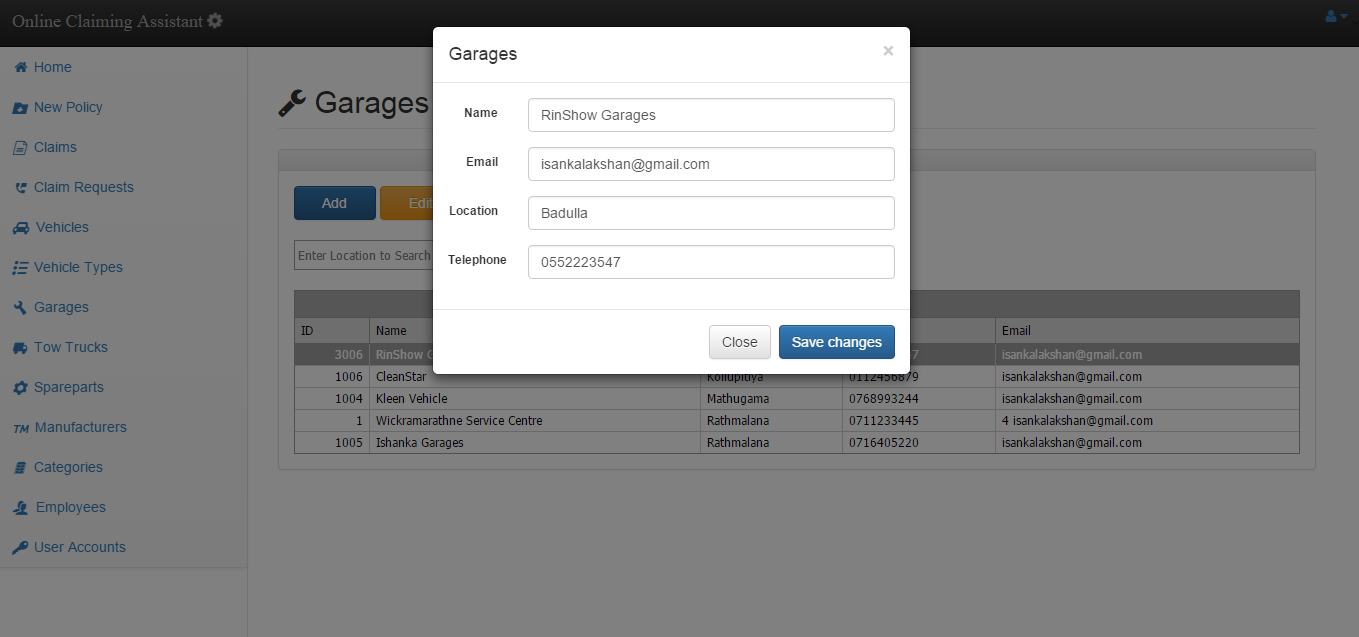


Figure 3: Edit Garage Interface

Source: Author

Interface for viewing vehicles. You can also select add/edit/remove options from this view. You can filter by several parameters in this view.

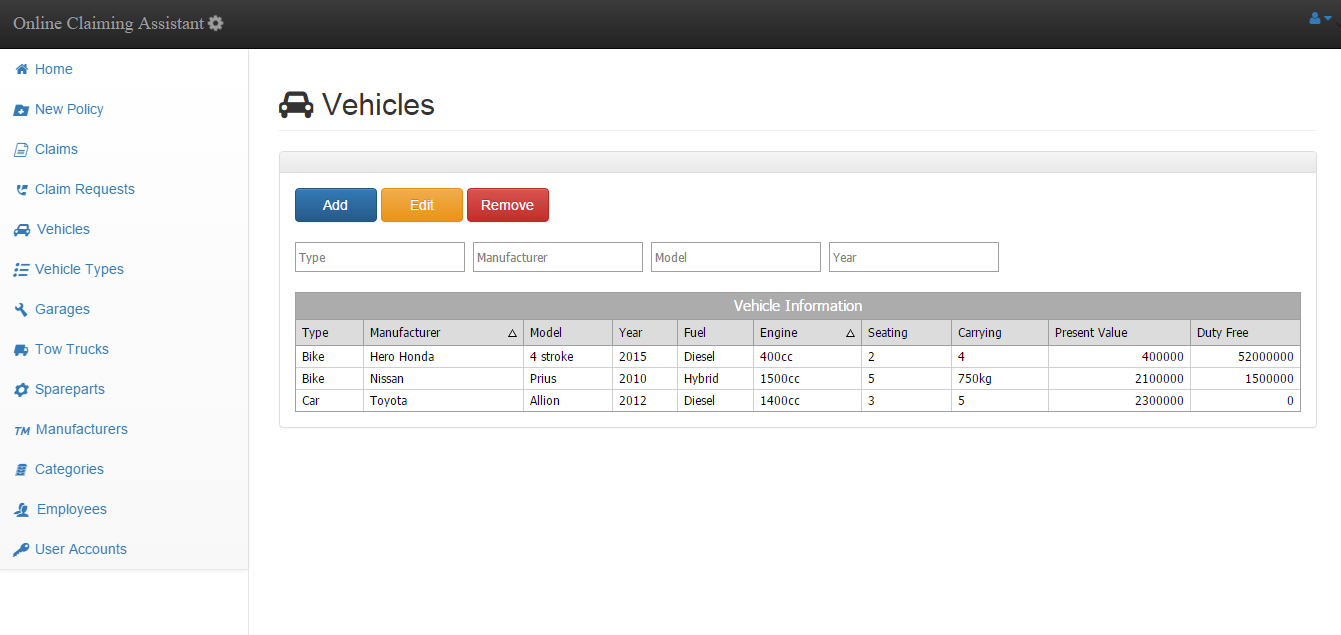


Figure 4: Vehicles Interface

Source: Author

Interface for viewing manufacturers. You can also select add/edit/remove options from this view. You can filter by name also in this view.

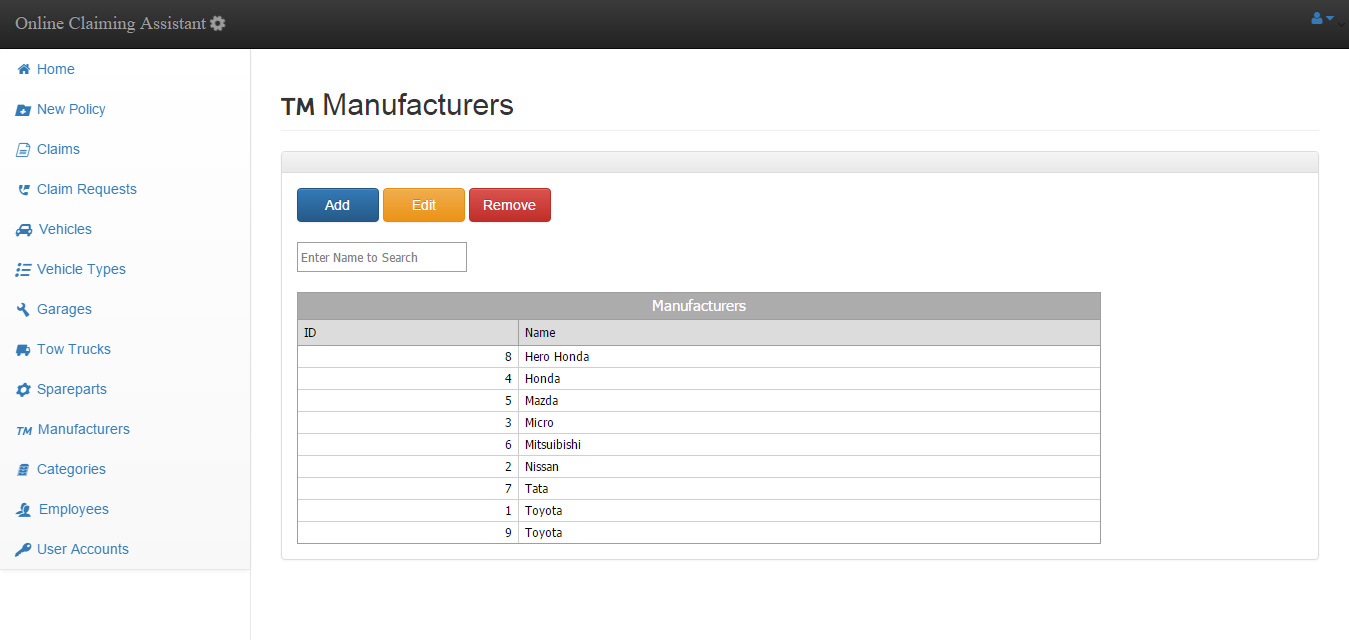


Figure 5: Manufacturers Interface

Source: Author

Interface for viewing spare part Category. You can also select add/edit/remove options from this view. You can filter by name also in this view.

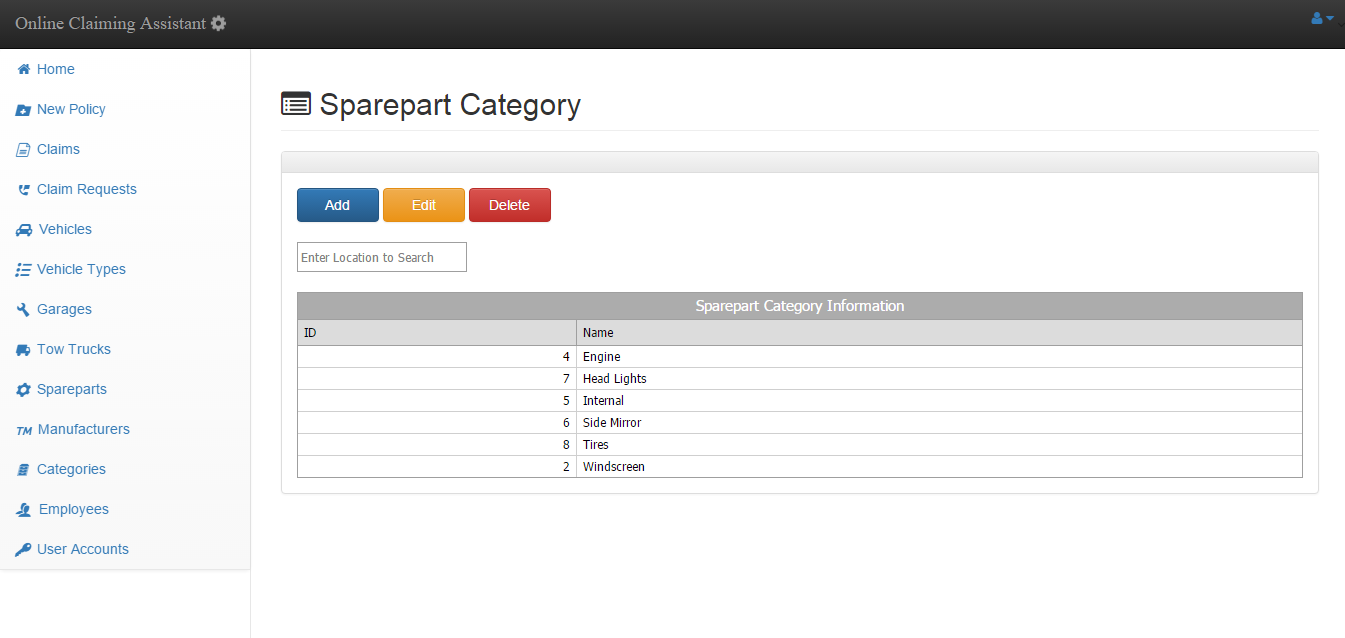


Figure 6: Spare part Category Interface

Source: Author

Interface for viewing garages. You can also select add/edit/remove options from this view. You can filter by location also in this view.

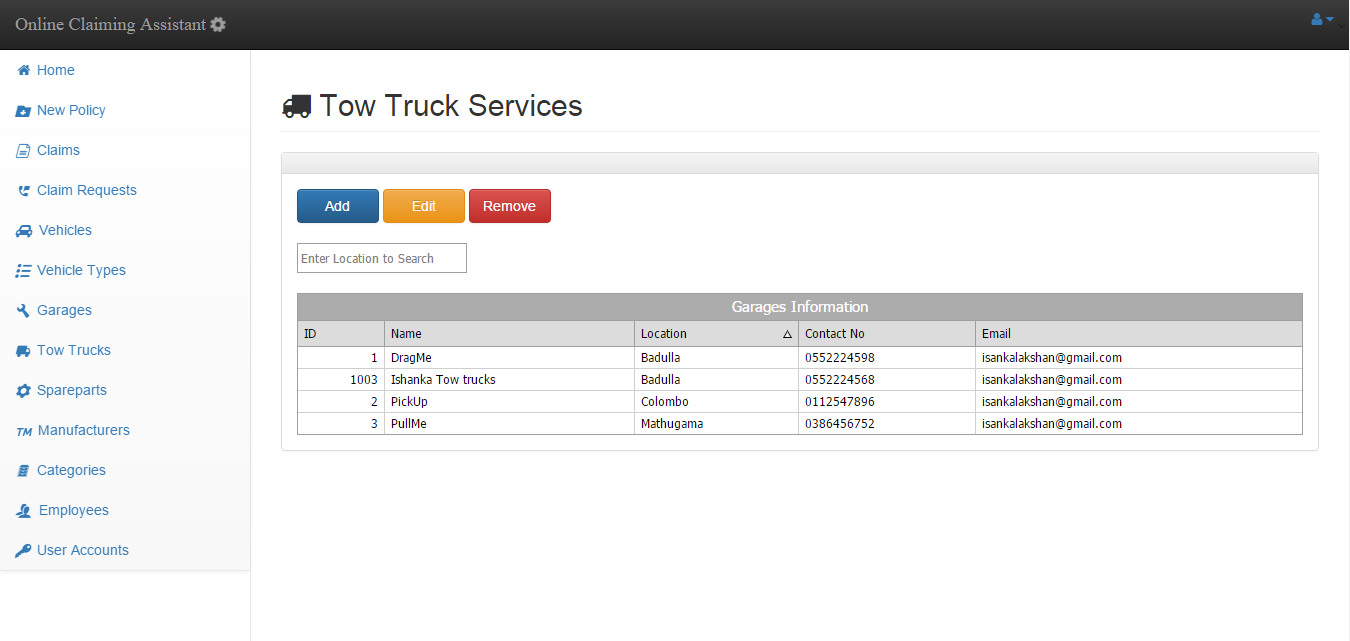


Figure 7: Tow Truck Services

Source: Author

Interface for viewing employees. You can also select add/edit/remove options from this view. You can filter by name and branch in this view.

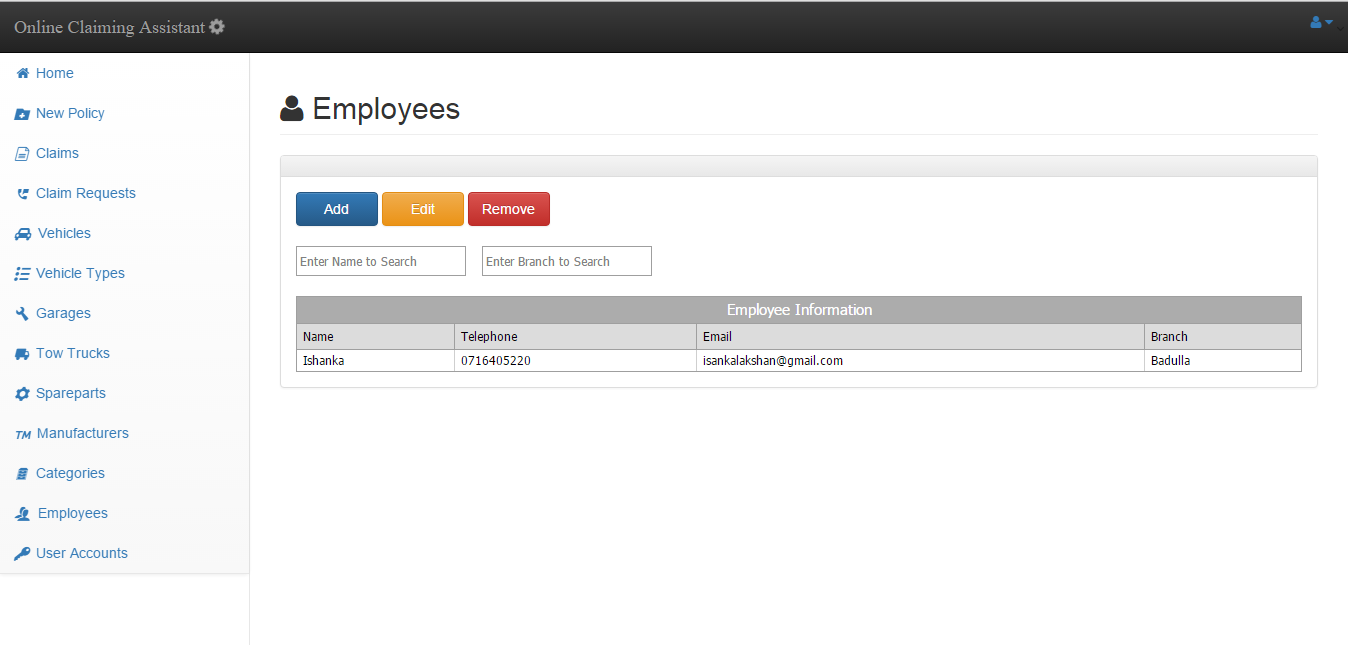


Figure 8: Employee Interface

Source: Author

# Appendix D

**Testing and Implementation**

Test log for the web application is attached in the next page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Test Case | Expected Result | Steps | Actual Result | Remark |
| Login Module | Test the result when login button pressed without giving credentials | Show enter username & password error message | 1. Enter the URL and enter to the system 2. Press login without providing credentials | Error message showed | Success |
| Test the result when tried to login with invalid credentials | Show invalid username or password error message | 1. Enter the URL and enter to the system 2. Enter some invalid credentials 3. Press login button | Error message showed | Success |
| Test the result when tried to login with valid credentials | Forward to the home/main menu | 1. Enter the URL and enter to the system 2. Enter some valid credentials 3. Press login button | Main menu loaded | Success |
| Test whether session has being created & session variables are initiated | Employee name should appear in top right | 1. Enter the URL and enter to the system 2. Enter some valid credentials 3. Press login button | Name appeared correctly | Success |
| New Policy Module | When New policy tile clicked form for new policy should appear | New policy form modal should pop up | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on New policy tile | New policy modal loaded | Success |
| All the required fields must be entered before submitting | Should display provide all required fields error message | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on New policy tile 4. Leave empty some required fields and submit | Success error message displayed | Success |
| Ability to insert a new policy successfully | Should display successfully entered message | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on New policy tile 4. Fill all required fields and submit | successfully entered message displayed | Success |
| Claims Module | When clicked claims tile load the claims submitted summary view | Should display the claims interface with data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on claims tile | Claims interface loaded | Success |
| Test the view button function | Should display detailed view of claims | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on claims tile 4. Select a claim record from data grid 5. Press view button | Detailed view of claim was displayed on a modal | Success |
| Test the approve button function in detailed view of a claim | Should display the approved message. | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on claims tile 4. Select a claim record from data grid 5. Press view button 6. Press approve button | Approved message displayed | Success |
| Claim Requests Module | When clicked claim requests tile load the respective interface | Should display the claims requests interface with data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on claim requests tile | Claims requests interface loaded with data grid | Success |
| Load the coordinates on a map | Should display a map pinpointing the exact GPS location | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on claim requests tile 4. Select a request from data grid 5. Press Map button | Map loaded | Success |
| Change the state of a claim request to responded | State should be changed to responded | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on claim requests tile 4. Select a request from data grid 5. Press Responded button | State changed to responded | Success |
| Vehicles  Module | Click the vehicles tile to load interface with filled data grid | Should display the vehicles details in data grid data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicles tile | Interface displayed with loaded data grid | Success |
|  | Add a new vehicle | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicles tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit vehicle details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicles tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a vehicle | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicles tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add a new spare part | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on spare part tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit spare part details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on spare part tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a spare part | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on spare part tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add a new spare part category | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on category tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit spare part category details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on category tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a spare part category | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on spare part category tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add a new manufacturer | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on manufacturer tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit manufacturer details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on manufacturer tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a manufacturer | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on manufacturer tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add a new vehicle type | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicle type tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit vehicle type details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicle type tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a vehicle type | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on vehicle type tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add a new garage | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on garage tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit garage details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on garage tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a garage | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on garage tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add a new tow truck service | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on tow truck service tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit tow truck service details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on tow truck service tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a tow truck service | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on tow truck service tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |
|  | Add an employee | Should add to database and display in data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on employee tile 4. Click add button 5. Fill the details and press save | Added and displayed in the data grid | Success |
|  | Edit employee details | Should submit to database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on employee tile 4. Click a record on data grid and press edit 5. Fill the details and press save changes | Submitted to database and updated the data grid | Success |
|  | Remove a employee | Should remove from database and update the data grid | 1. Enter the URL and enter to the system 2. Login to the system 3. Click on employee tile 4. Click a record on data grid and press Remove 5. Confirm by clicking remove. | Removed from database and refreshed the data grid | Success |

Table 17: Test Log for Web Application

Source: Author