**IT2080 – ITP**

**Year 2, Semester 2, 2024**

**Assignment 2**



**Easy Rent Vehicle Rental And Hiring Management System**

**Group Details**

**Batch: 4.1**

Group Number: ITP24\_B1\_04

|  |  |  |
| --- | --- | --- |
|  | **Student Registration Number** | **Student Name** |
| 1 | IT22580344 | Jayasekara M.P.S.S |
| 2 | IT22286246 | Chandrasekara C.M.P.V |
| 3 | IT22137128 | Wijewantha H.D.C |
| 4 | IT22230638 | Nathasha K.D.S |
| 5 | IT22101242 | Tharuka M.S |
| 6 | IT22124876 | Silva M.W.P.R.K |
| 7 | IT22271150 | Abeysekara.W.R.G.M |
| 8 | IT22281128 | Jayasundara J.M.T.P |

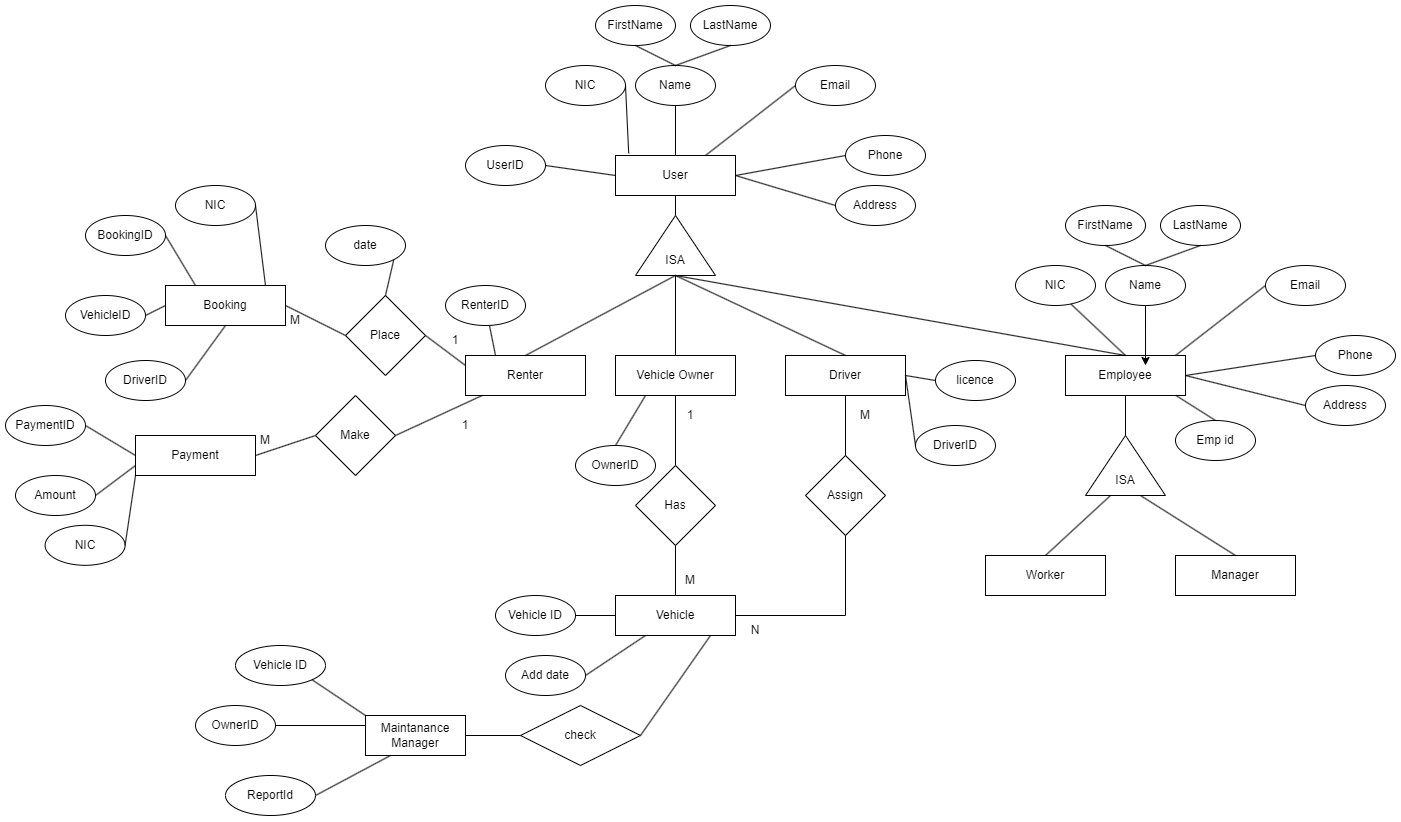
# **Progress**

|  |  |  |
| --- | --- | --- |
| Function | Completed User Stories | Incomplete User Stories |
| Customer Management System  IT22137128 | * As a new user, I want to register easily on the platform. * As a renter, I want to view vehicle ads within the system. * As a renter, I want to search for vehicles. * As a renter, I want to report vehicles if necessary. * As a renter, I want to add vehicles to my favourite list and manage it. * As a user, I want to provide feedback after service. * As a user, I want to submit complaints at any time | * As a user, I want the option to delete my account if needed. |
| Booking and Payment Management System  IT22580344 | * As a Renter, I want to create, edit, delete booking requests for selected vehicle. * As a Renter, I want to see my booking history. * As a Renter, I want to see estimate price for my trip * As a Renter, I want to search my booking history. * As a Vehicle Owner, I want to check booking request for all my vehicles. * As a Vehicle Owner, I want to search booking requests. * As a Vehicle Owner, I want to approve or reject booking requests. * As a Vehicle Owner, I want to provide a rejected message for renter when rejecting booking request. * As a Renter, I can check my booking request approved or rejected. | * As a renter, I want to do a payment * As a renter, I want to see my payment history * As a Owner, I want to add additional cost to estimated bill * As a Owner, I want to check my earning History |

## **Repository Link Of the Project**

## <https://github.com/Sudila-Jayasekara/EasyRent>

# **ER Diagram**



# High-level System Design

# Normalized Schema

# **Test Case Design**

**Test Function:** Report Generation

**Test Case Designed By:** It22137128

**Test Priority:** High

**Pre-condition:** Renter must be logged into the system.

**Test Steps:**

* Renter navigates to the "Report Generation" section.
* Selects the desired parameters for the report, such as rental duration, vehicle type, location, etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result | Description |
|  | rental duration, vehicle type, location, etc | A detailed report should be generated, including information such as rental history, expenses, vehicle usage statistics, etc. | The report is generated successfully and displays relevant data | pass | The renter successfully generates a report based on selected parameters, providing valuable insights into their rental activities |

**Test Function:** Search Vehicles Ads and Visibility

**Test Case Designed By:** It22137128

**Test Priority:** High

**Pre-condition:** Renter must be logged into the system.

**Test Steps:**

* Renter accesses the "Search Vehicles Ads" feature.
* Enters search criteria such as vehicle type, location, availability dates, etc.
* Initiates the search.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result | Description |
|  | vehicle type, location, availability dates, etc | The system should display relevant vehicle ads that match the search criteria. The ads should include detailed information about the vehicles and their availability | vehicle ads matching the search criteria are displayed, with accurate information regarding availability and features | pass | The renter successfully searches for vehicle ads and finds the desired vehicles with accurate visibility and availability information |

**Test Function:** Search Booking history

**Test Case Designed By:** IT22580344

**Test Priority:** High

**Pre-condition:** Renter must be logged into the system.

**Test Steps:**

* Vehicle Renter navigates to the "Booking" section.
* Enters search criteria such as vehicle type, location, availability dates, etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result | Description |
|  | Vehicle brand, Vehicle model, location, dates | The system should only display the relevant bookings that matching to search criteria | System display only relevant bookings the matching to search criteria | pass | The renter successfully searches for bookings |

**Test Function:** Generate Payment Report

**Test Case Designed By:** IT22580344

**Test Priority:** High

**Pre-condition:** Owner must be logged into the system.

**Test Steps:**

* Vehicle Owner navigates to the "Earning" section.
* Select a week, month or year for generate reports

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result | Description |
|  | Select week, month or year for generate report | System should generate a report for all earning for entered period of time | System display generated report for owner earnings for selected time period | pass | The owner successfully generated report for earning |

# **Individual Contribution**

**Renter Management System**

**It22137128**

|  |  |
| --- | --- |
| Completed Features | Incomplete Features |
| * Develop a user registration form with necessary fields. * Implement validation for email and password. * Create a delete account functionality with a confirmation prompt. * Design and implement a vehicle ad display page. * Develop search functionality with filters for vehicle type, location, etc. * Develop functionality to add/remove vehicles from the favourites list | * Manage User Profile * Implement saving of search results for easy access |

Pseudocode for Favourite List Component

**Initialize State**

const user = JSON.parse(localStorage.getItem('user'));

const userId = user ? user.\_id : null;

const userrole = user ? user.userType : '';

const [watchList, setWatchList] = useState([]);

const [vehicleDetails, setVehicleDetails] = useState([]);

**Define useEffect to fetch watchlist data**

useEffect(() => {

**Fetch watchlist data from the server**

if (userId) {

axios.get(`http://localhost:5556/api/${userrole}/${userId}`)

.then(response => {

console.log('Watchlist:', response.data.wishList);

setWatchList(response.data.wishList);

})

.catch(error => {

console.error('Error fetching watchlist:', error);

});

}

}, [userId, userrole]);

**Define useEffect to fetch vehicle details based on watchlist**

useEffect(() => {

**Fetch vehicle details based on watchlist from the server**

const fetchVehicleDetails = async () => {

const promises = watchList.map(vehicleId => {

return axios.get(`http://localhost:5556/api/vehicle/${vehicleId}`);

});

try {

const responses = await Promise.all(promises);

const vehicleDetailsData = responses.map(response => response.data);

setVehicleDetails(vehicleDetailsData);

} catch (error) {

console.error('Error fetching vehicle details:', error);

}

};

fetchVehicleDetails();

}, [watchList]);

**Define handleLike method to add/remove vehicles from watchlist**

const handleLike = async (vehicleId) => {

try {

**Send request to server to add/remove vehicle from watchlist**

const response = await axios.patch(`http://localhost:5556/api/renter/${userId}/${vehicleId}`);

console.log(response.data.message);

if (watchList.includes(vehicleId)) {

// If vehicle is already in watchlist, remove it

const updatedWatchList = watchList.filter(item => item !== vehicleId);

setWatchList(updatedWatchList);

} else {

// If vehicle is not in watchlist, add it

const updatedWatchList = [...watchList, vehicleId];

setWatchList(updatedWatchList);

}

} catch (error) {

console.error('Error toggling vehicle like:', error);

}

};

**Render UI with vehicle cards**

return (

<div>

{/\* UI code for displaying favourite vehicles \*/}

</div>

);

## **Flow-chart**

A diagram of a flowchart

Description automatically generated

**Booking and Payment Management System**

**IT22580344**

|  |  |
| --- | --- |
| Completed Features | Incomplete Features |
| * Develop Booking form with necessary fields for create bookings. * Develop Booking history page for see all relevant bookings for renter. * Add update and delete function for booking. * Develop Check booking page that Owner can sell all relevant bookings for their vehicles. * Develop functionality to approve or reject bookings with a message. * Generate Estimate bill for bookings | * Generate total bill with add additional cost to estimated bill * Implement payment history page. * Implement Earning page. |

## **Flow-chart**

