Payment Management: IT22099518
Hemapriya H.A.N.S

The module, which the member is working on

In our Cab Service Management System, the "Payment Management" function enables users (customers, drivers, and managers) to handle various payment-related tasks. Customers and drivers can create, update, or delete their payment details.

And here, managers must submit their payment reports. Therefore this system has an interface to report submission.

All the things are handled by the payment manager. The payment manager creates driver payments and the system ensures payments are directed to the drivers(It can be bank payment or cash payment) with business commissions transferred to the company's account.

The completion level of the features

- Backend CRUD routes
- Frontend and Backend Validations
- Some frontend UIs
- Pushed completed works to the branch

Work not completed and to be completed by the final

Work in progress	To be completed
Create payment details	Access home page
View / Retrieve payment details	Report generating
Update payment details	Access payment gateway
Delete payment details	Calculations parts
Some payment Manager UI's	
Create report submission	
View/ Retrieve report	
Update and delete the report	
Create drivers payment	
Update drivers payment	
Delete drivers payment	

SQL queries used

reports model

```
const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const reportSchema = new Schema({

    paymentType : {
```

```
        type : String,
        required : true
    },
    department : {
        type : String,
        required : true
    },
    date : {
        type : Date,
        required : true
    },
    time : {
        type : String,
        required : true
    },
    document: {
        type: Buffer, // Assuming you want to store the file as binary data
        required: true
    }

})

const Reports = mongoose.model("Reports", reportSchema);

module.exports = Reports;
```

reports database


```
[
    {
        "_id": "66214c071379044b0f6cd31b",
        "paymentType": "Bank Payment",
        "department": "Order",
        "date": "2024-04-10T00:00:00.000Z",
        "time": "10:00 AM",
        "document": {
            "type": "Buffer",
            "data": [
                117,
                112,
                108,
                111,
                97,
                100,
                115,
                92,
                49,
                55,
                49,
```

```
                51,
                52,
                53,
                56,
                49,
                56,
                51,
                51,
                49,
                57,
                45,
                73,
                84,
                49,
                48,
                51,
                48,
                32,
                77,
                32,
                102,
                111,
                114,
                32,
                67,
                46,
                112,
                100,
                102
            ]
        },
        "__v": 0
    },
    {
        "_id": "66214c8c1379044b0f6cd31d",
        "paymentType": "Bank Payment",
        "department": "Inventory",
        "date": "2024-02-12T00:00:00.000Z",
        "time": "11:00 AM",
        "document": {
            "type": "Buffer",
            "data": [
                117,
                112,
                108,
                111,
                97,
                100,
                115,
```

```
                92,
                49,
                55,
                49,
                51,
                52,
                53,
                56,
                51,
                49,
                54,
                53,
                57,
                54,
                45,
                65,
                84,
                77,
                32,
                40,
                49,
                50,
                48,
                41,
                46,
                112,
                100,
                102
            ]
        },
        "__v": 0
    },
    {
        "_id": "66214cdc1379044b0f6cd31f",
        "paymentType": "Cash Payment",
        "department": "Administrator",
        "date": "2024-02-12T00:00:00.000Z",
        "time": "11:00 AM",
        "document": {
            "type": "Buffer",
            "data": [
                117,
                112,
                108,
                111,
                97,
                100,
                115,
                92,
```

```
                49,
                55,
                49,
                51,
                52,
                53,
                56,
                51,
                57,
                54,
                50,
                54,
                49,
                45,
                65,
                84,
                77,
                32,
                40,
                49,
                50,
                48,
                41,
                46,
                112,
                100,
                102
            ]
        },
        "__v": 0
    }
]
```

paymentdetails model

```
const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const paymentdetailsSchema = new Schema({

    paymentType : {
        type : String,
        required : true
    },
    amount : {
        type : Number,
        required : true
    },
    paymentDescription : {
```

```
        type : String,
        required : true
    }
})

const paymentDetails = mongoose.model("paymentDetails", paymentdetailsSchema);

module.exports = paymentDetails;
```

paymentdetails database

```
[
    {
        "_id": "66160a33fcdfb5dfdcf0279b",
        "paymentType": "Bank Payment",
        "amount": 2000,
        "paymentDescription": "Driver's payment",
        "__v": 0
    },
    {
        "_id": "661f58e4a4f6c7b49413d232",
        "paymentType": "Bank Payment",
        "amount": 2400,
        "paymentDescription": "Tour Payment",
        "__v": 0
    },
    {
        "_id": "661f5a11a4f6c7b49413d234",
        "paymentType": "Cash Payment",
        "amount": 1000,
        "paymentDescription": "Driver Payment",
        "__v": 0
    }
]
```

driverpayments model

```
const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const driverpaymentsSchema = new Schema({

    name : {
        type : String,
        required : true
    },
    date : {
        type : Date,
        required : true
```

```
    },
    amount : {
        type : Number,
        required : true
    },
    companycommission : {
        type : Number,
        required : true
    },
    finalsalary : {
        type : Number,
        required : true
    }

})

const driverPayments = mongoose.model("driverPayments", driverpaymentsSchema);

module.exports = driverPayments;
```

driverpayments database


```
[
    {
        "_id": "6616668bc5948c74a7406f6e",
        "name": "Saman",
        "date": "2024-09-12T00:00:00.000Z",
        "amount": 3000,
        "companycommission": 0,
        "finalsalary": 3000,
        "__v": 0
    },
    {
        "_id": "66214f451379044b0f6cd324",
        "name": "Kamal",
        "date": "2024-02-12T00:00:00.000Z",
        "amount": 3600,
        "companycommission": 1000,
        "finalsalary": 2600,
        "__v": 0
    }
]
```

Algorithm
Flow chart, pseudocode

Algorithm

Step 1: Start
Step 2: Input login credentials
Step 3: Validate the credentials:
    a. If the credentials are valid:
      -Display “Login Successful”
      -Proceed to step 4
    b. If the credentials are not valid:
      -Display “Invalid Username or Password”
      -Back to step 2
Step 4: Display HOME Dashboard
Step 5: Click “Add Payment Details” and Fill the form
Step 6: Click the “Submit” button
    a. If Successful:
      -Display “Payment Details Added”
    b. If submission failed:
      -Proceed to Step 6
Step 7: View the submitted payment details
Step 8: Update the submitted payment details
    a. If Successful:
      -Display “Payment Details Updated”
    b. If the update failed:
      -Proceed to Step 8
Step 9: Delete the submitted payment details
    a. If Successful:

-Display “Payment Details Deleted”

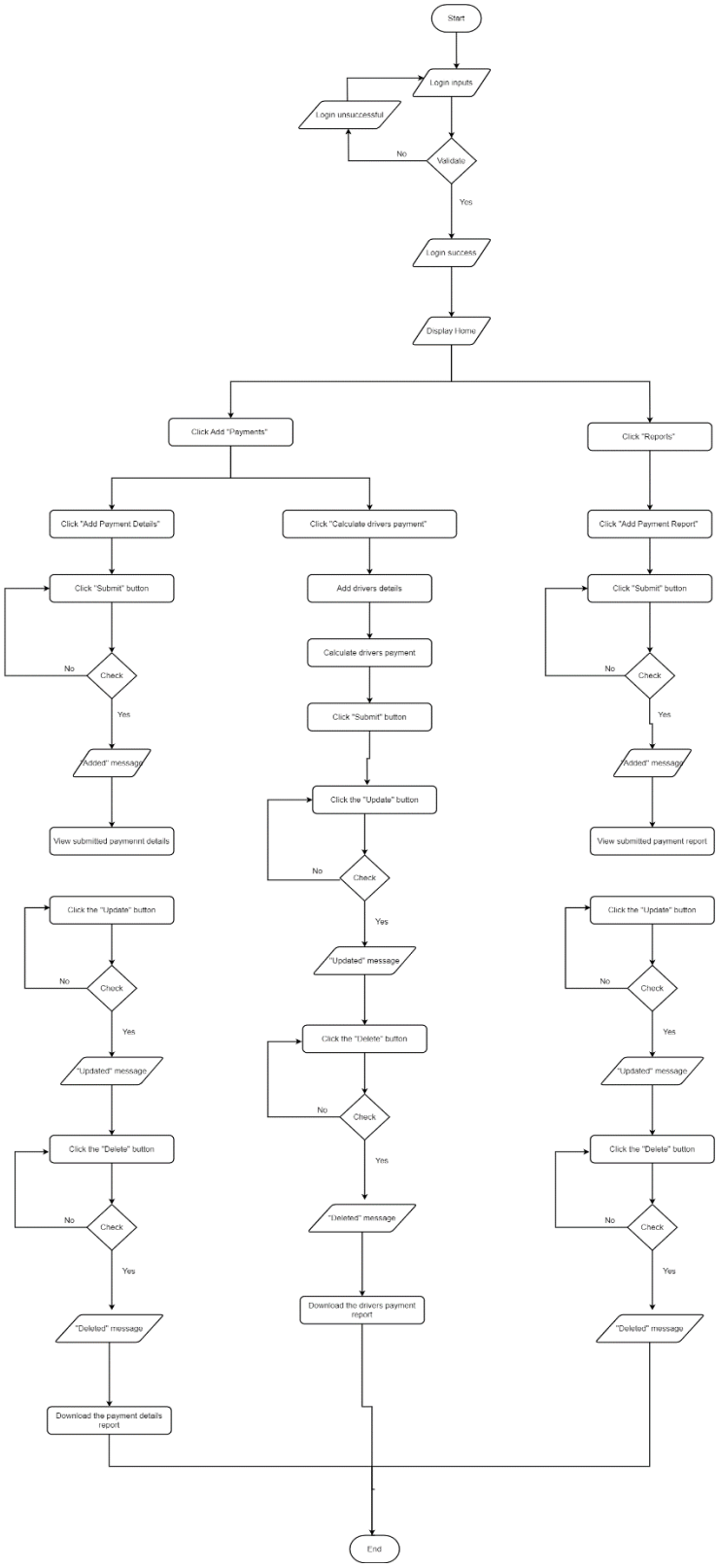
b. If delete failed:

-Proceed to Step 9

Step 10: Download the payment details report

Step 11: End

Flow Chart


Start
Login inputs
Login unsuccessful
No
Validate
Yes
Login success
Display Home
Click Add "Payments"
Click "Reports"
Click "Add Payment Details"
Click "Calculate drivers payment"
Click "Add Payment Report"
Click "Submit" button
Add drivers details
Click "Submit" button
No
Check
Yes
Calculate drivers payment
Click "Submit" button
"Added" message
Click the "Update" button
View submitted paymennt details
View submitted payment report
Click the "Update" button
"Updated" message
Click the "Delete" button
"Updated" message
Click the "Delete" button
"Deleted" message
Download the drivers payment report
"Deleted" message
Download the payment details report
End

Pseudocode

```
Start
Input login credentials
Validate the credentials
   If valid:
      Display "Login Successful"
      Proceed to HOME Dashboard
   Else:
      Display "Invalid Username or Password"
      Go back to Input login credentials
HOME Dashboard
Click "Add Payment Details" and Fill the form
Click "Submit" button
   If submission is successful:
      Display "Payment Details Added"
   Else:
      Go back to Click "Add Payment Details" and Fill the form
View the submitted payment details
Update the submitted payment details
   If update is successful:
      Display "Payment Details Updated"
   Else:
      Go back to Update the submitted payment details
Delete the submitted payment details
   If delete is successful:
      Display "Payment Details Deleted"
   Else:
      Go back to Delete the submitted payment details
Download the payment details report
End
```