**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Content** | **Page No.** |
| 1 | Introduction | 2 |
| 2 | Proposed System | 2 |
| 3 | Module Specification | 4 |
| 4 | Hardware Specification | 5 |
| 5 | Software Specification | 6 |
| 6 | Data Dictionary | 7 |
| 7 | E-R Diagrams | 12 |
| 8 | UML Diagrams |  |
|  | * Class Diagram | 13 |
|  | * Object Diagram | 14 |
|  | * Use Case Diagram | 15 |
|  | * Sequence Diagram | 16 |
|  | * Deployment Diagram | 17 |
| 9 | Snapshots | 18 |
|  | * Admin Side Screens | 18 |
|  | * Driver Side Screens | 21 |
|  | * Customer Side Screens | 23 |
| 10 | Bibliography | 32 |
|  |  |  |

**INTRODUCTION**

Now-a-days our life becoming faster and this fast life we need vehicle to reach some destination. But everyone can’t afford to buy a vehicle due to any reason. Well no need to worry about that because we are working on project which will solve the vehicle problem and that project is nothing but

**“E-Auto”**

E-Auto is a web site to book an auto online on fare. You just need to book your ride (Auto) online and auto will came with driver within a minutes in front of you. See how easy it is?

What you need to do, firstly you have to register yourself on website then and then you can book your ride. This can also provide features like tracking auto, online payment etc. so this what about the project.

**Proposed System**

Since auto’s are available but they are not always available next to your door if you want to go out with auto, you need to move out before 10 to 15 min from your house which takes more time.

But you can book your private auto from just before 5 min leaving from home this will save our time. And it is easily works.

**What we are going to do?**

We are going to develop two applications

1. For customer
2. For auto driver

Using GPS

1. **For customer -**

In this application customer can book auto after registration. Customer can see available auto’s surrounding. Customer can also track the auto for satisfaction.

During booking he /she can see the auto driver’s details like name ,auto number. After book auto customer can call auto through our system for any query .Customer can also track the auto for satisfaction. After ride finished, customer needs to give feedback about ride in terms of star. But for all this, customer should have a mobile device and data connection, GPS available in that.

1. **For auto driver-**

For receiving bookings from the customer driver should have mobile device. If not then company can provide the device on fare.

In this application driver will receive the notification about booking then he/ she has to accept that booking. Driver will also receive the details about customer.

As we are using GPS driver get the Navigation panel to reach to the customer.

This application will provide the all updates about application and booking rate, booking packages.

Bill will also generate on drivers application like customers application.

After finished ride, if driver wants to stop or continue the receiving booking help she will get notification about that.

1. **GPS -**

GPS is helpful for both driver and customer for tracking each other.

**MODULE SPECIFICATION**

* **Choose source and destination address:**

For hiring auto passenger/customer have to enter appropriate source and destination address

* **Auto hired by system automatically:**

The system will automatically allocate nearest auto of source address of passenger/customer.

* **Pay your tariff:**

After reaching destination address passenger/customer will pay the tariff by various ways like cash, credit/debit card/paytm wallet etc.

* **Tracking:**

Passenger/Customer can easily track auto for security purpose.

* **Give Feedback :**

The customer will give the feedback to the admin.

**Hardware Requirement**

**Client side:**

|  |  |
| --- | --- |
| RAM | 512MB |
| Hard Disk | 10GB |
| Processor | 1.0GHz |

**Server side:**

|  |  |
| --- | --- |
| RAM | 1GB |
| Hard Disk | 20GB |
| Processor | 2.0GHz |

**Software Requirement**

**Client Side:**

|  |  |
| --- | --- |
| Web Browser | Mozilla Firefox, Google Chrome, Internet Explorer |
| Operating System | Windows ,Linux or any equivalent OS |

**Server side:**

|  |  |
| --- | --- |
| Web Server | Apache Http server |
| Framework | .Net with C# |
| Database Server | PostgreSql database |
| Web browser | Internet Explorer |
| Operating system | Fedora Server 8 |

**Data Dictionary**

**List of relations**

Schema | Name | Type | Owner

--------+--------------+-------+-------

public | admin | table | auto

public | bookauto1 | table | auto

public | complaint | table | auto

public | custregister | table | auto

public | driver | table | auto

public | driverlogin | table | auto

public | payment | table | auto

(7 rows)

**----------------------------------------------------------------------------**

1. **Admin Table:**

postgres=# \d admin;

Table "public.admin"

Column | Type | Modifiers

------------+-----------------------+-----------

emp\_id | character varying(20) | not null

emp\_name | character varying(20) |

contact\_no | numeric(10,0) | not null

email | character varying(30) |

gender | character varying(6) |

password | character varying(30) |

Indexes:

"admin\_pkey" PRIMARY KEY, btree (emp\_id)

1. **BookAuto Table:**

postgres=# \d bookauto1;

Table "public.bookauto1"

Column | Type | Modifiers

-------------+------------------------+-----------

contact | numeric(10,0) |

pickdate | date |

from\_loc | text |

to\_loc | text |

no\_of\_seats | integer |

batch\_no | character varying(15) |

time | time without time zone |

km | integer |

fare | money |

Foreign-key constraints:

"bookauto1\_batch\_no\_fkey" FOREIGN KEY (batch\_no) REFERENCES driver(batch\_no) ON UPDATE CASCADE ON DELETE CASCADE

"bookauto1\_contact\_fkey" FOREIGN KEY (contact) REFERENCES custregister(mobile\_no) ON UPDATE CASCADE ON DELETE CASCADE

1. **Complaint Table:**

postgres=# \d complaint;

Table "public.complaint"

Column | Type | Modifiers

----------------------+-----------------------+-----------

comp\_id | character varying(20) | not null

contact\_no | numeric(10,0) |

subject\_of\_complaint | text |

city | character varying(20) |

details | text |

upfile | character varying(20) |

status | character varying(10) |

Indexes:

"complaint\_pkey" PRIMARY KEY, btree (comp\_id)

Check constraints:

"complaint\_status\_check" CHECK (status::text = ANY (ARRAY['accepted'::character varying, 'pending'::character varying, 'solved'::character varying]::text[]))

Foreign-key constraints:

"complaint\_contact\_no\_fkey" FOREIGN KEY (contact\_no) REFERENCES custregister(mobile\_no) ON UPDATE CASCADE ON DELETE CASCADE

**4) Customer-register Table:**

postgres=# \d custregister;

Table "public.custregister"

Column | Type | Modifiers

----------------+-----------------------+-----------

customer\_name | character varying(20) | not null

address | character varying(50) | not null

age | character varying(10) |

gender | character varying(6) |

mobile\_no | numeric(10,0) | not null

email\_id | character varying(40) | not null

disable\_status | character varying(3) | not null

password | character varying(20) |

Indexes:

"custregister\_pkey" PRIMARY KEY, btree (mobile\_no)

**5)Driver Table:**

postgres=# \d driver;

Table "public.driver"

Column | Type | Modifiers

-----------------+-----------------------+-----------

dr\_name | character varying(20) |

address | text |

date\_of\_joining | date |

age | character varying(10) |

gender | character varying(6) |

batch\_no | character varying(20) | not null

lic\_no | character varying(15) |

autono | character varying(10) |

contact | numeric(10,0) |

email | character varying(30) |

password | character varying(20) |

status | character(3) |

Indexes:

"driver\_pkey" PRIMARY KEY, btree (batch\_no)

"driver\_contact\_key" UNIQUE, btree (contact)

Check constraints:

"driver\_status\_check" CHECK (status = ANY (ARRAY['on'::bpchar, 'off'::bpchar]))

1. **driverlogin Table:**

Table "public.driverlogin"

Column | Type | Modifiers

----------+-----------------------+-----------

batch | character varying(20) |

password | character varying(20) |

Foreign-key constraints:

"driverlogin\_batch\_fkey" FOREIGN KEY (batch) REFERENCES driver(batch\_no) ON UPDATE CASCADE ON DELETE CASCADE

1. **payment Table:**

Table "public.payment"

Column | Type | Modifiers

--------------+-----------------------+-----------

c\_name | character varying(20) |

cust\_phone | numeric(10,0) |

driver\_name | character varying(20) |

driver\_phone | numeric(10,0) |

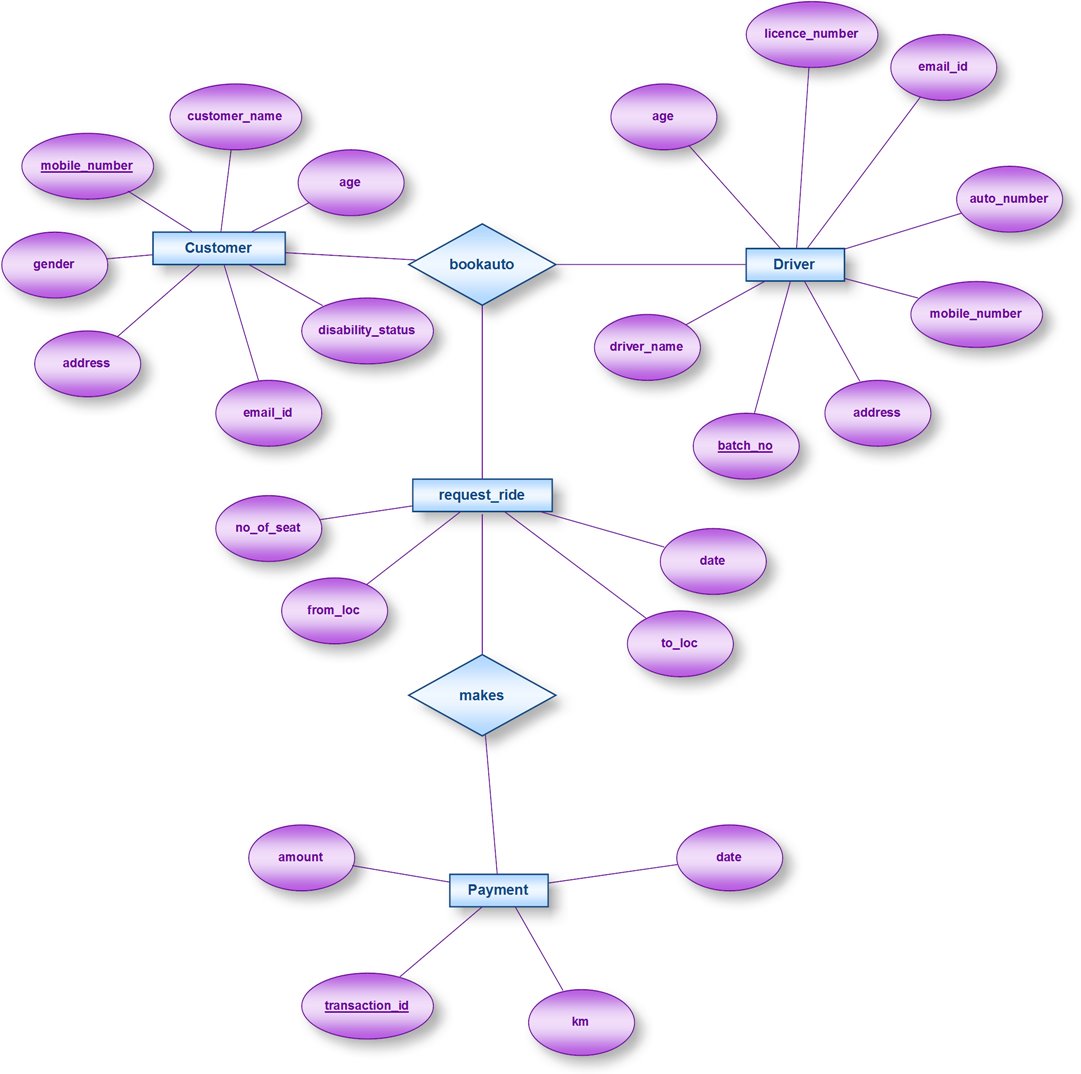
payment\_date | date |

km | double precision |

amount | double precision |

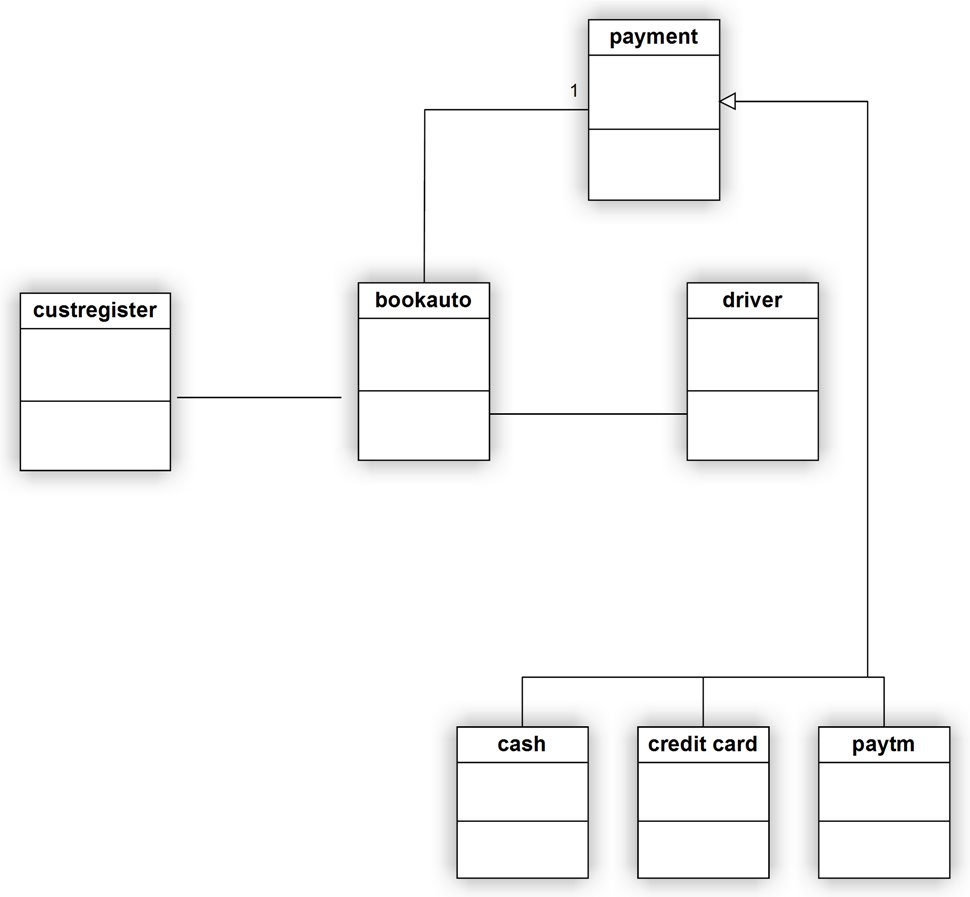
payment\_mode | character varying(20) |

**Entity –Relationship Diagram**

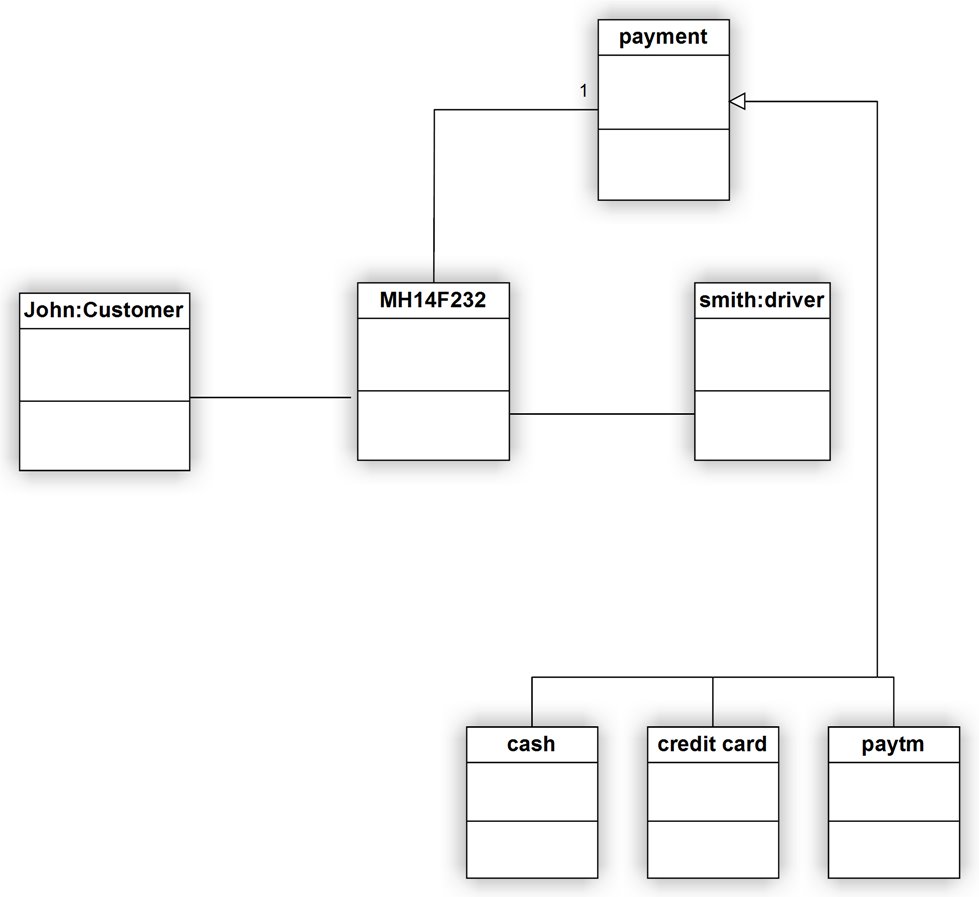


**UML DIAGRAMS**

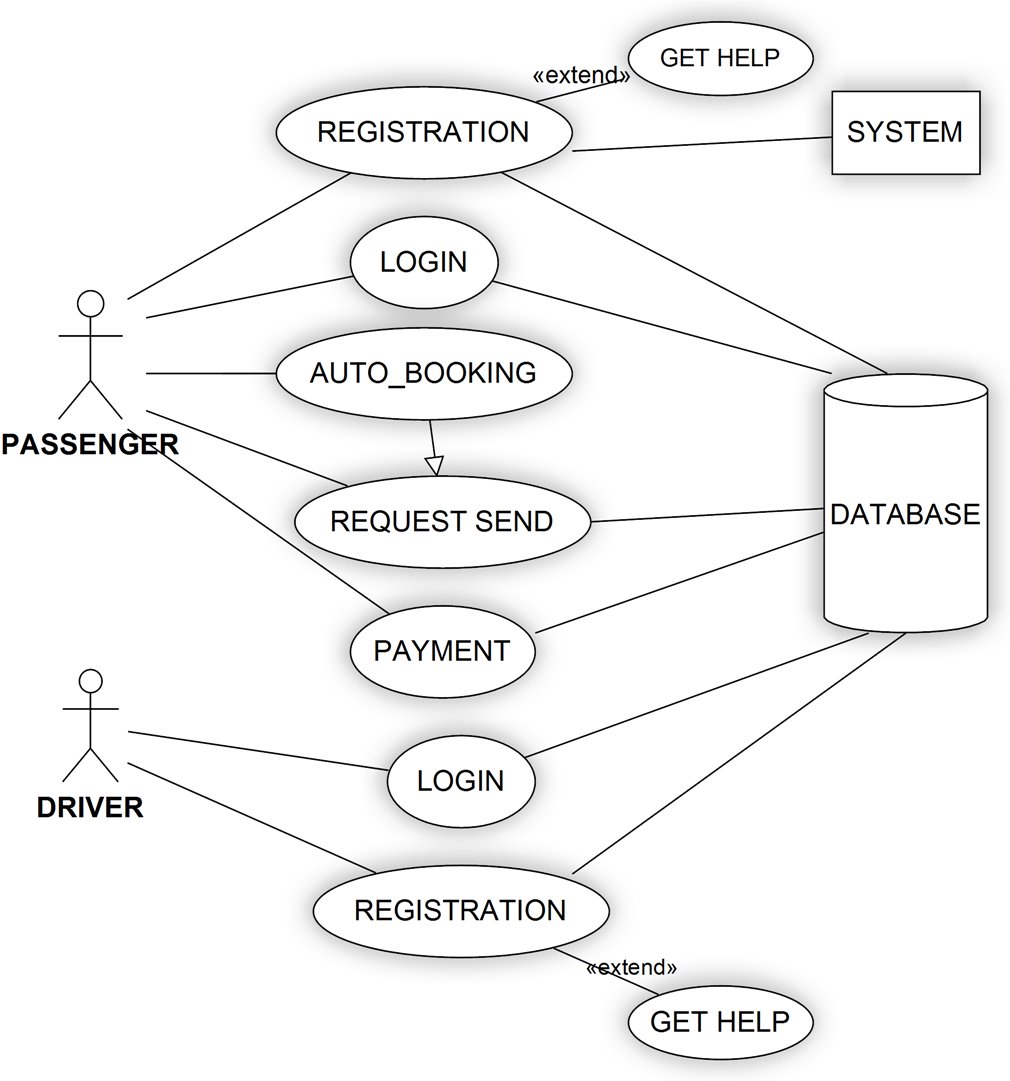
1. CLASS DIAGRAM:



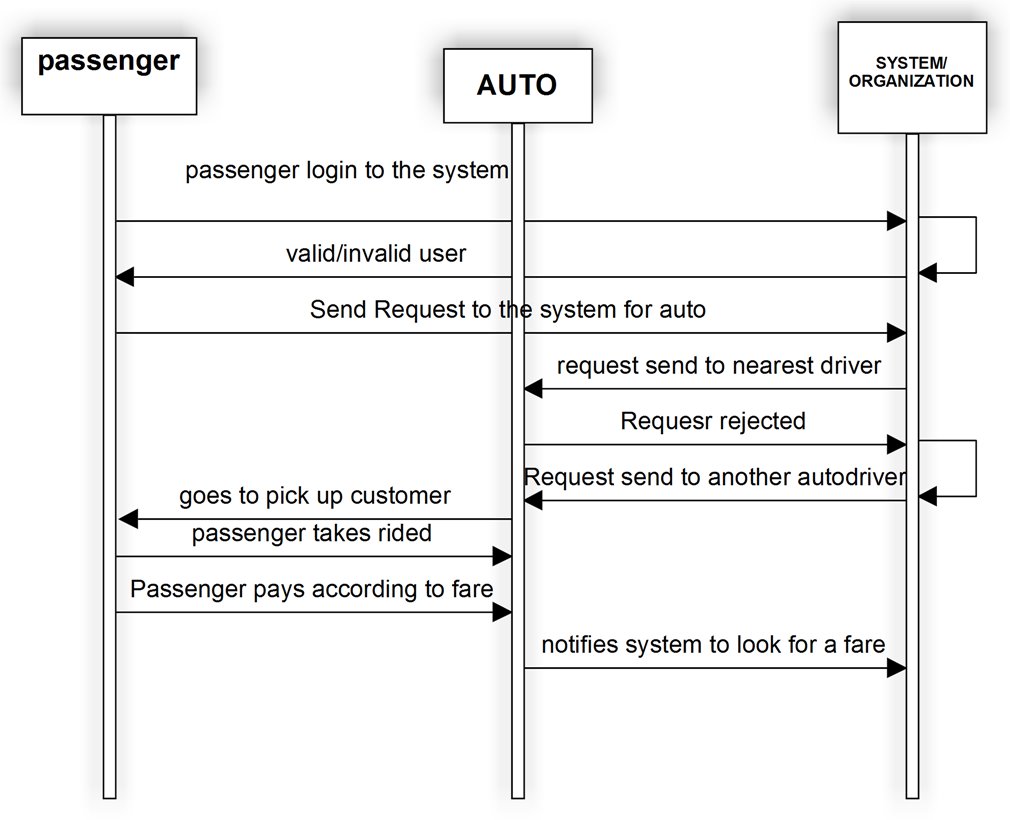
1. OBJECT DIAGRAM:



1. USE CASE DIAGRAM



1. SEQUENCE DIAGRAM:



1. DEPLOYMENT DIAGRAM

