1. INTRODUCTION

The Regional Transport Office (RTO) is responsible for issuing licence, collecting taxes and enforcing transportation rules. It maintains a database of drivers and vehicles. The total number of vehicles registered in Bengaluru as on 31st January, 2016 was around 6 million (or 60 lakh). This number is expected to grow further. It is evident that RTO offices have to maintain records for a large number of drivers and vehicles. The task of collecting and maintaining information about vehicles and drivers is divided among a number of RTO offices at state and district level. To save time and effort, it is essential to have ease of use and ease in accessing and updating data. A database provides these advantages. The purpose and scope of RTO database have been summarized below.

1.1 Purpose

The following is the motivation behind developing an RTO database:

- Maintenance of an easy to use database at grass-roots level (i.e. district level) which enables sharing the burden of collecting and storing information by RTO offices at all levels.
- To provide greater efficiency, transparency and ease of use to stakeholders involved (end users, RTO office)

Earlier, maintenance of records was manual. Hence, entering new information and accessing information was cumbersome. It was also unreliable as there was a possibility of records getting lost or damaged.

Nowadays, the Government of India has made available portals which provide services such as –

- Vehicle Registration
    - New Vehicle Registration
    - Renewal of Registration
    - Transfer of Ownership
    - Change of Address

- Permit
    - Issue of National & Interstate Permit
    - Renewal of Permit
- Taxes
    - State-wise tax calculation & Payment
- Fitness
    - Issue of Fitness Certificate
    - Renewal of Fitness Certificate
- Enforcement
    - Issue of Challan
    - Settlement of Penalty Amount
- Application for Learner's Licence
- Application for Permanent Licence

However these have not yet been adopted by all RTO offices at district level.

1.2 Scope

The proposed RTO database management system will have the following new features:

- Issue MASTER PASS/SPEED PASS (which is a proof that the driver has all necessary documents like driving license, insurance certificate, tax certificate, etc.)
- Indicate total amount of fine paid through challans by a particular driver (so that a warning can be issued if the amount exceeds a certain threshold amount)
- Indicate age of vehicles though registration number and know whether they are within the permissible age limit (this feature has been included in view of Delhi Government's proposal to scrap diesel vehicles older than 15 years)

Essential features of the existing system to be also included in the proposed system are:

- Record tax entries and view status of tax payment
- View details of vehicles through registration number

- Vehicle registration (new registration, renewal of registration, transfer of ownership, change in address)
- Licence services (new application/renewal of Learner's Licence, Permanent Licence, and know status of licence)

2. SOFTWARE REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is a comprehensive description of the intended purpose and environment for software under development.

2.1 Overall Description

Software requirement specification is an important step in database design. Here, we define hardware and software requirements of the proposed system. Also, functional and non-functional requirements of the system are specified. This enables database designers to design a database which meets all user functional requirements. This, in turn helps meet non-functional requirements such as ease of use. The proposed RTO database system consists of modules designed so as to satisfy the functionality as mentioned below.

2.2 Specific requirements

2.2.1 Hardware Requirements:

- Processor Speed: 233 MHz
- RAM: 64MB
- Free Hard Disk Space: 1.5 GB
- Display Resolution: 800*600 or higher

2.2.2 Software requirements:

- Front end: HTML+CSS3
- Back end: MySQL (version 5.5) +PHP (version 5.5)

2.2.3 Functionality:

The module wise functionality of RTO database management system is as given below:

Authentication:

- Unique Login ID for each user
- Separate Login for admin

Vehicle Registration:

- New Registration
- Renewal of registration
- Transfer of ownership
- Change of address
- View details of vehicle using registration number
- View age of vehicle

Licence:

- New Application of Learner's/ Permanent licence
- Renewal of Learner's/ Permanent licence

Taxes:

- Tax payment through challan
- View payment status

Master Pass:

Application for Master Pass

Permit:

Issue of state/national permit

Loan:

Display loan details for a given vehicle registration number

2.2.4 Non-functional requirements:

- Ease of use: System should be easy to use. This is achieved through use of a simple user interface
- Fast access to data: Provided through efficient queries to access the database
- Reliability: System is updated periodically by the admin to ensure correctness of data stored.
- Security: System should be made secure. Authentication is provided in the form of user and admin login.
- Maintainability: System should be easy to maintain.

3. DETAILED DESIGN

The following are the Data Flow Diagrams (DFD) for RTO database. DFD levels 0, 1 and 2 have been described below. The modules of RTO database management system are designed based on the Data Flow Diagrams.

3.1 DFD Level 0

The user and admin both access RTO database as shown in the figure below. Each can send a request to the database and receive a response from it.

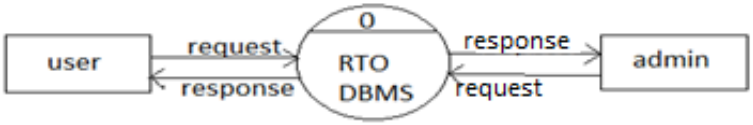


Figure 3.1.1: DFD Level 0 for RTO database

3.2 DFD Level 1

Level 1 involves dividing the RTO database management system into multiple sub processes which access the database as shown in the figure below. The database management system has been split into seven different modules as shown in the figure. This splitting into modules provides ease of usage. The user places a request for licence application, vehicle registration, etc. The user enters required details for the same. The admin issues vehicle registration number/ licence number following which the registration number/ licence number is entered into the database. Hence, the registration/ licence number can be entered only with approval of the administrator.

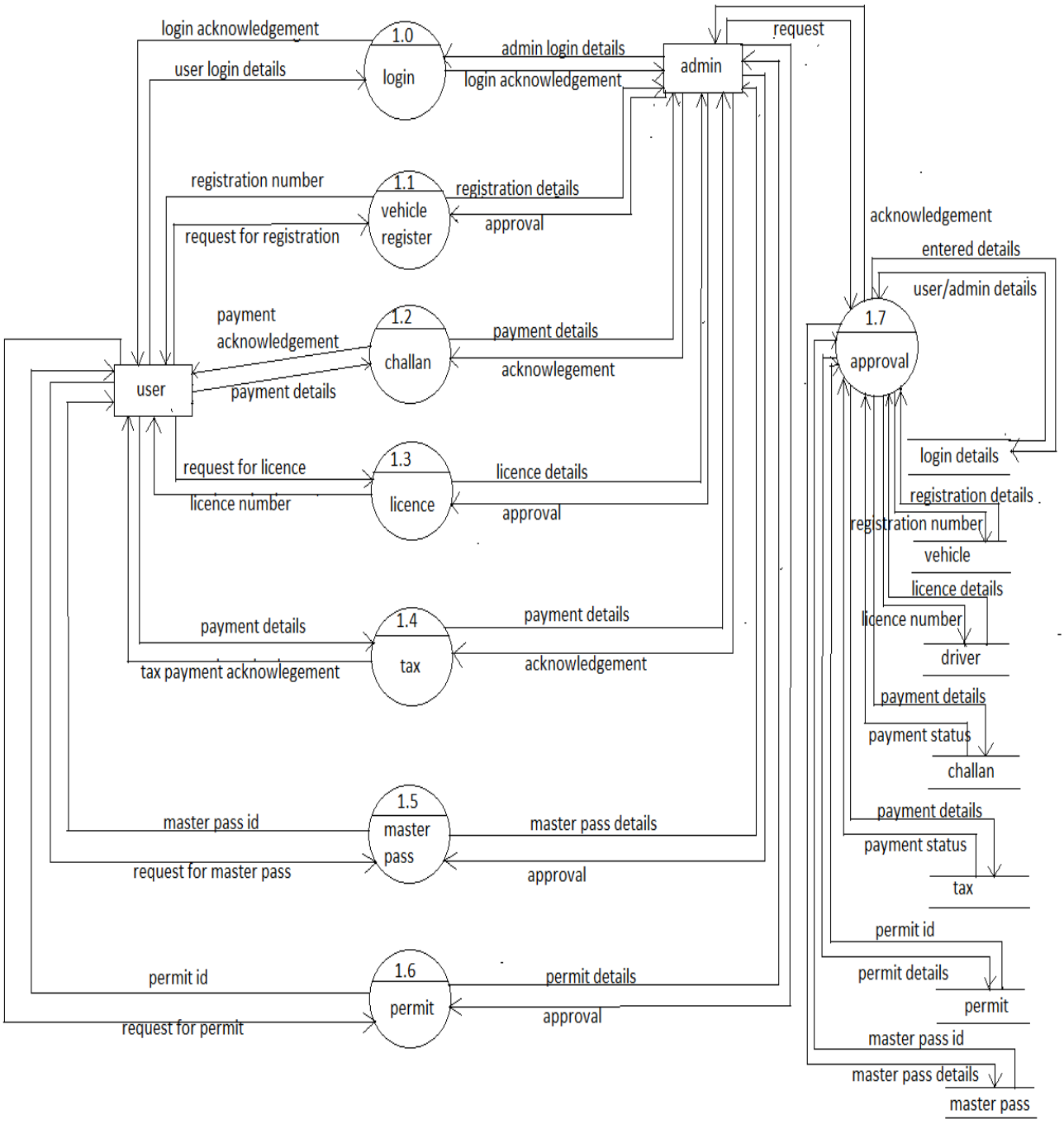


Figure 3.2.1: DFD Level 1 for RTO Database

3.3 DFD Level 2

3.3.1 Level 2 (Vehicle registration):

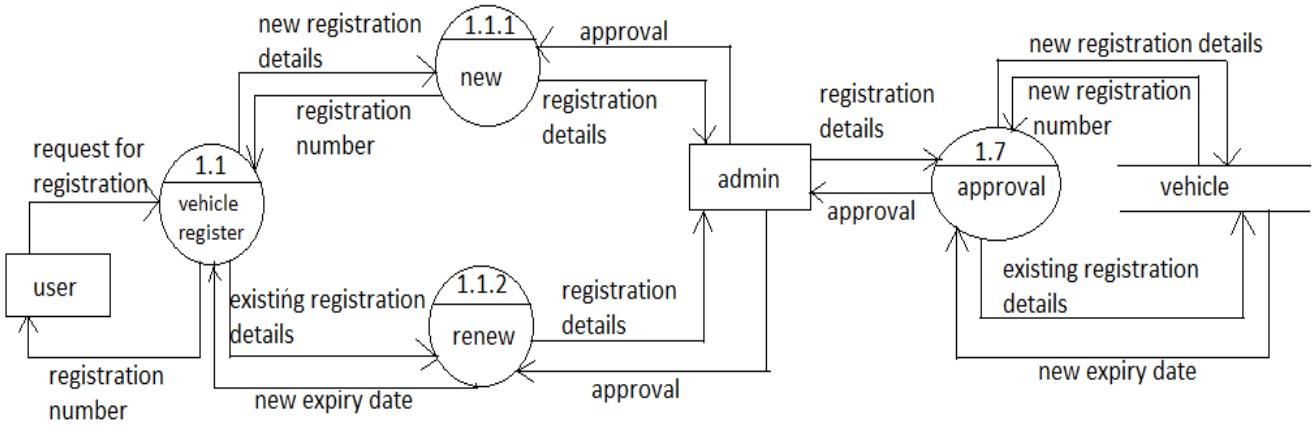


Figure 3.3.1: DFD Level 2- Vehicle Registration

3.3.2 Level 2 (Challan):

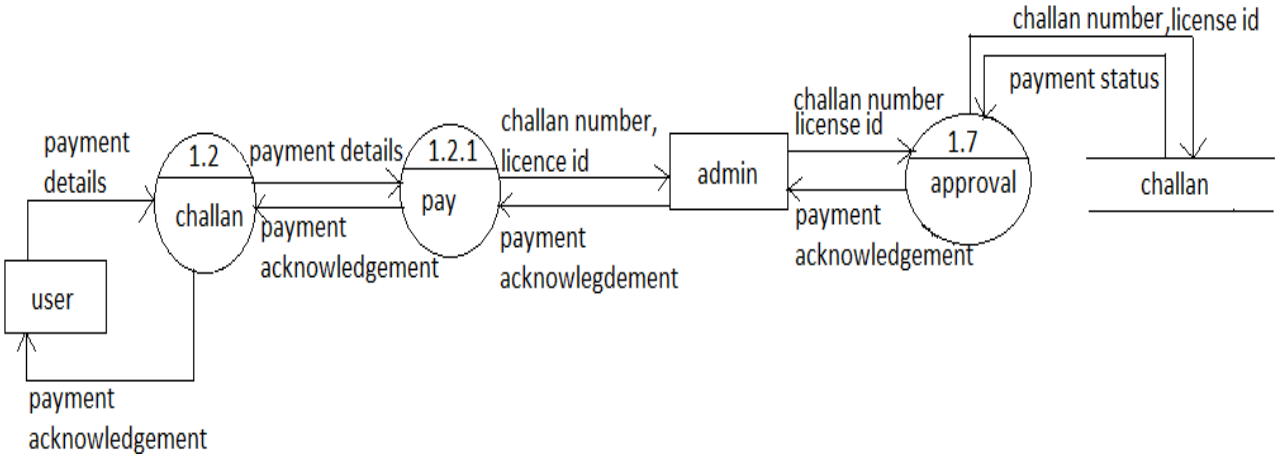


Figure 3.3.2: DFD Level 2- Challan

3.3.3 Level 2 (Licence):

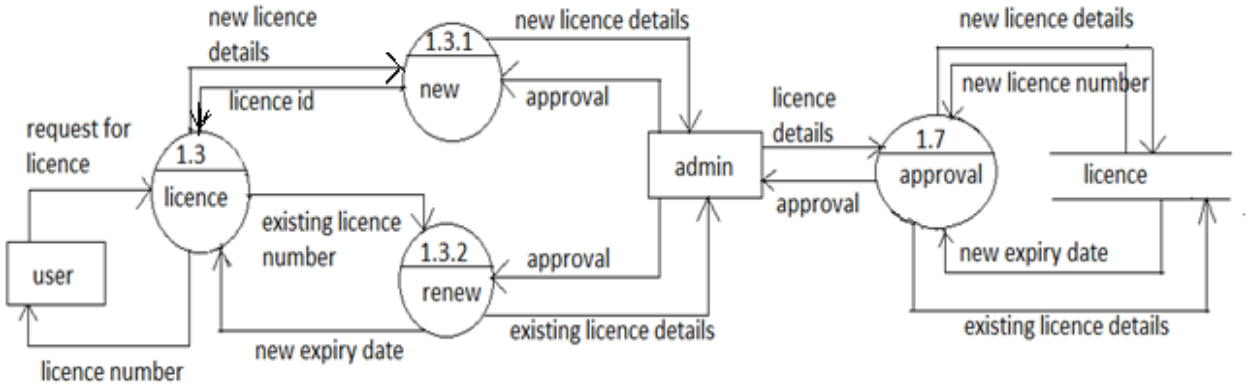


Figure 3.3.3: DFD Level 2- Licence

3.3.4 Level 2 (Tax):

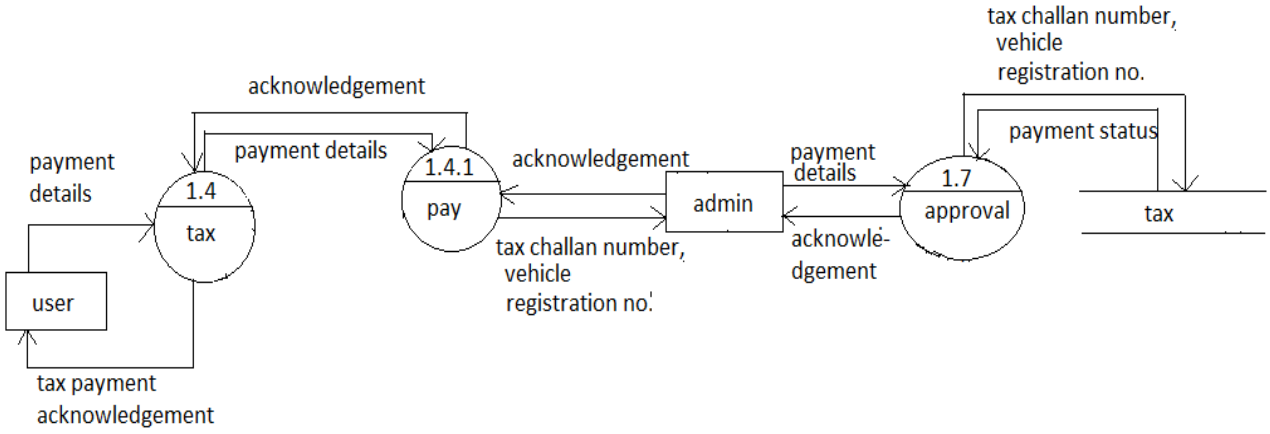


Figure 3.3.4: DFD Level 2- Tax

3.3.5 Level 2 (Master pass):

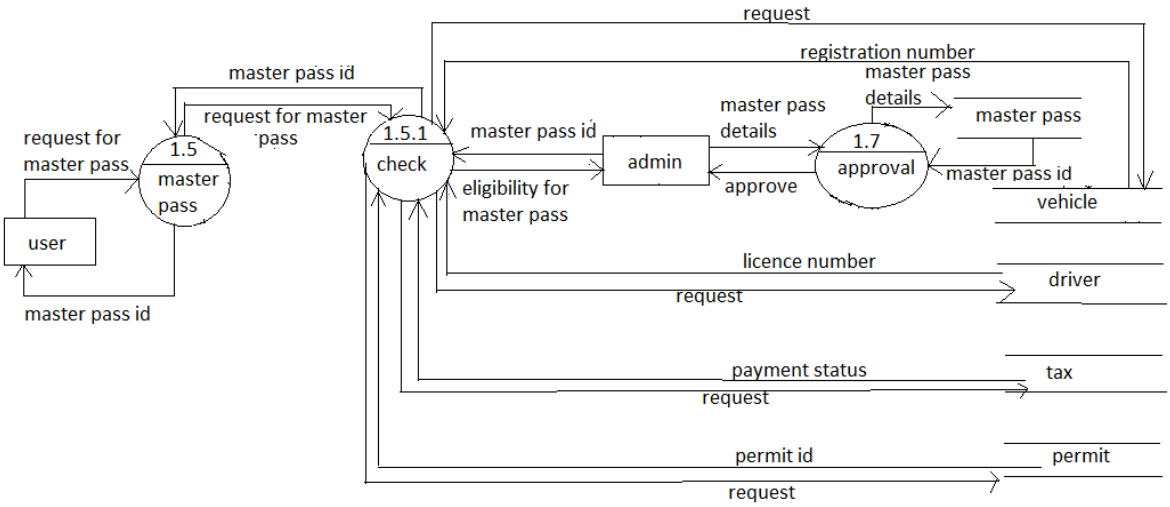


Figure 3.3.5: DFD Level 2- Master Pass

3.3.6 Level 2 (Permit):

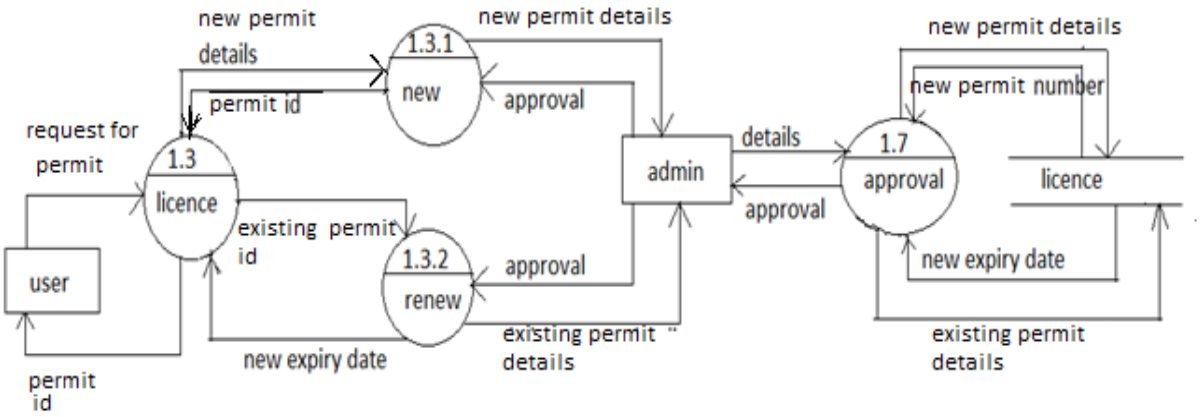


Figure 3.3.6: DFD Level 2- Permit

4. ENTITY- RELATIONSHIP (ER) DIAGRAM

Entity- Relationship (ER) diagram of the RTO Database has been described in the figure below. The main entities involved here are driver, vehicle, driving licence, vehicle registration, vehicle insurance, vehicle tax, challan, master pass, RTO office, RTO branch. The entities are related by 1: 1, 1: n or m: n relationships. The participation of the entities is partial or total depending on the relationship.

Every driver (or user) has a licence, hence the participation of driver and licence in the relationship 'has' is total. Hence, a 1: 1 relationship is defined between driver an d licence with total participation on both sides. However, every driver does not own a vehicle. A driver can own multiple vehicles. Every vehicle has to be owned by some driver. Hence, a 1: n relationship 'has' is defined between driver and vehicle. A driver may apply for a multiple loans. Hence a 1: n relationship 'applies for' is defined between driver and loan.

Every vehicle has registration, insurance and tax. Hence, 1: 1 relationships have been defined between vehicle and registration, vehicle and insurance, vehicle and tax. In each relationship, participation is total on both sides.

The RTO office has branches. Driving licence is issued by an RTO branch's supervisor (or admin). Every RTO branch has a supervisor and hence a 1: 1 relationship 'has' can be defined between branch and supervisor with total participation on both sides.

A driver who owns a vehicle and possesses all necessary documents like permit, registration, insurance, driving licence can apply for a master pass. Not all drivers own a vehicle or possess all necessary documents. Hence, a 1: 1 relationship is defined between driver and master pass with partial participation by driver.

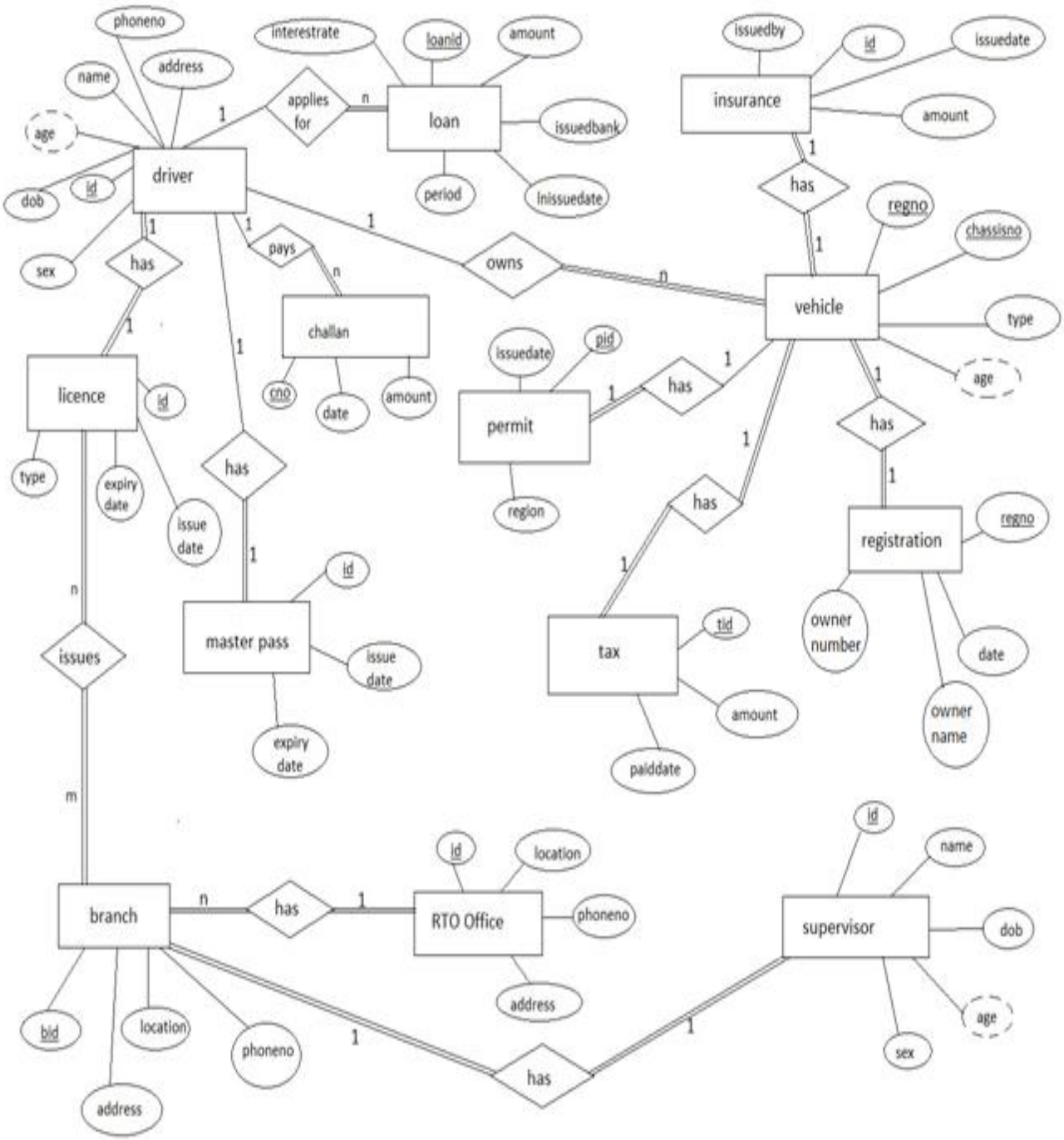
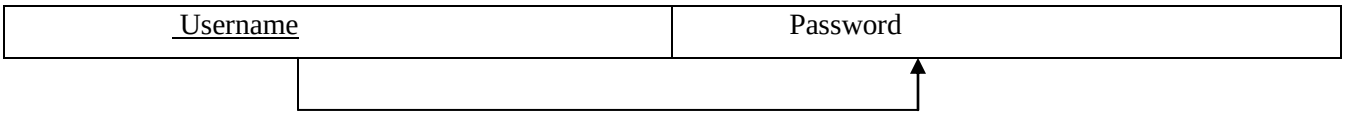


Figure 4.1: ER Diagram for RTO Database

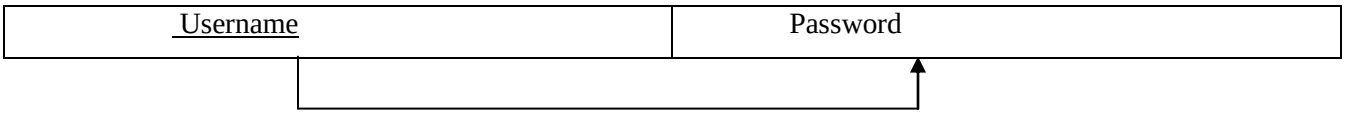
5. RELATIONAL SCHEMA AND NORMALISATION

Each table's primary key has only single attribute. Moreover, there is no transitive dependency among any of the attributes. Hence the tables satisfy the conditions for 1 NF, 2 NF and 3NF. Following are the normalized tables in 3 NF:

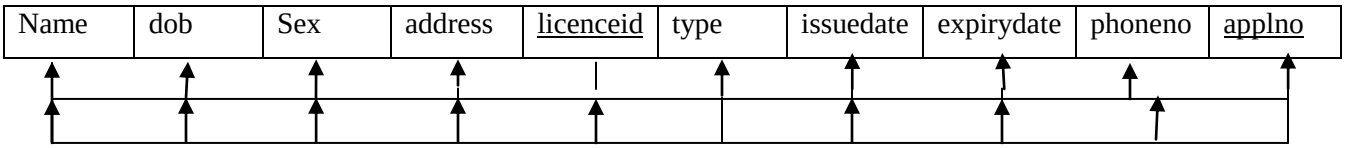
5.1 admin_login



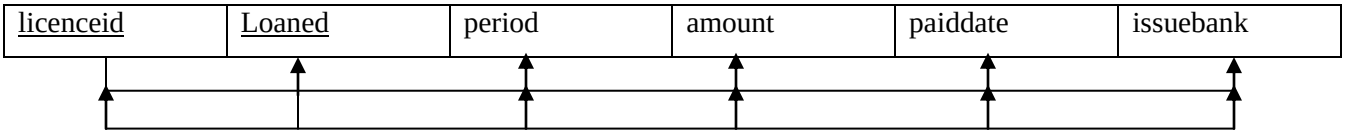
5.2 user_login:



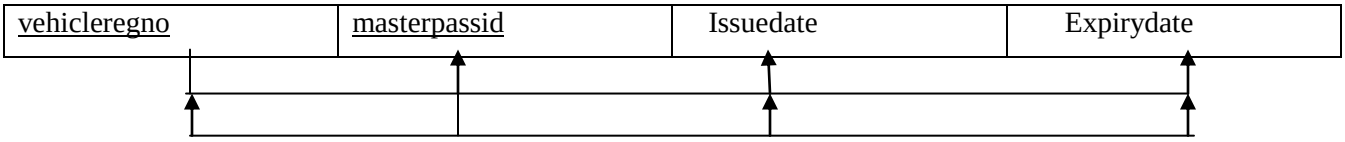
5.3 driver_licence:



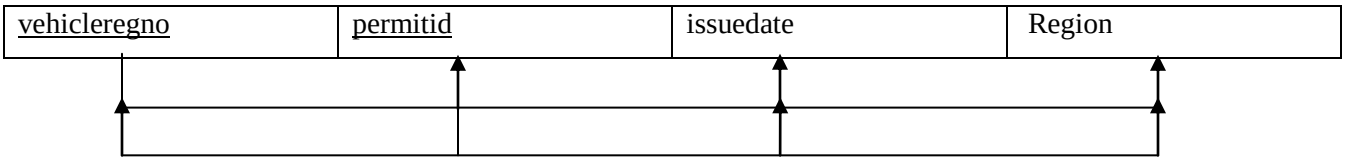
5.4 driver_loan:



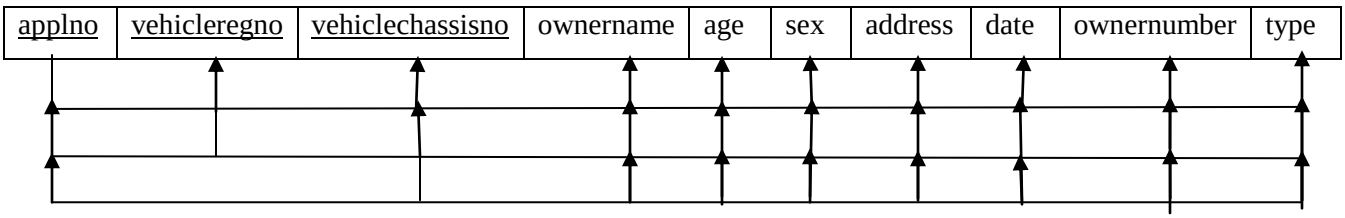
5.5 masterpass:



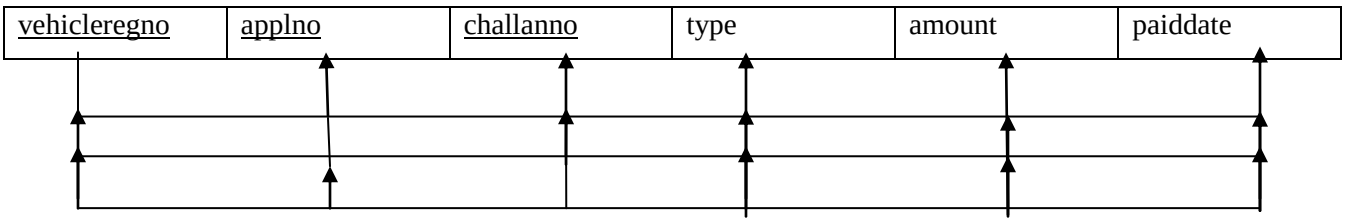
5.6 Permit:



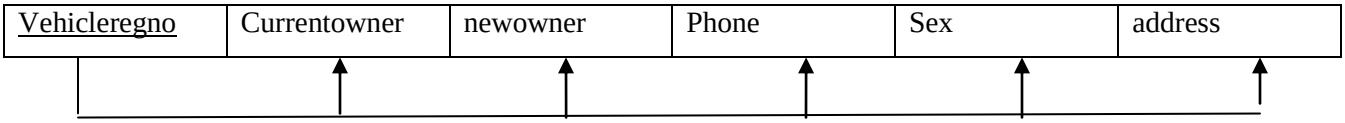
5.7 vehicle_registration:



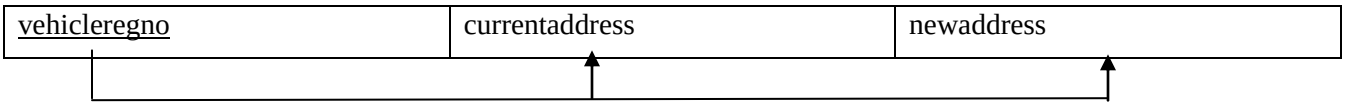
5.8 vehicle_tax:



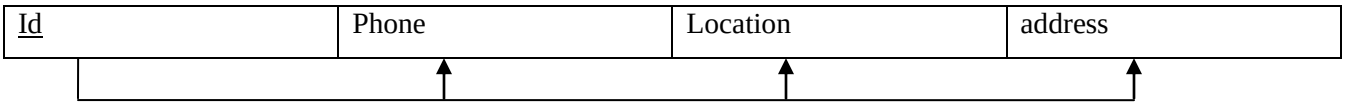
5.9 transfer_ownership:



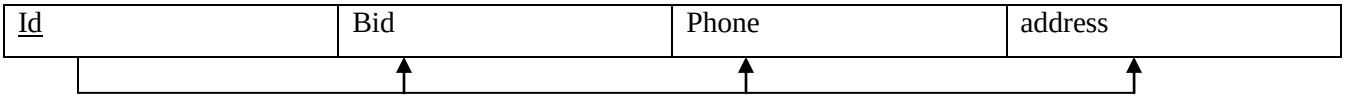
5.10 change_address:



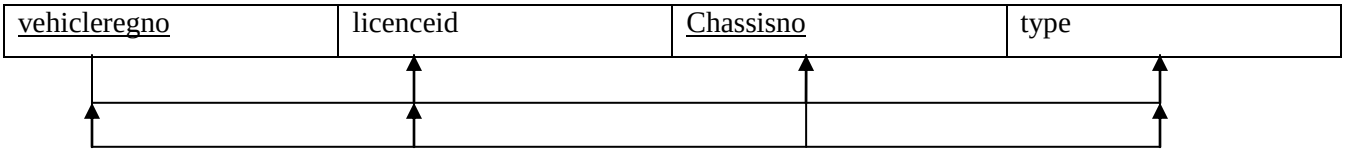
5.11 rto_office



5.12 branch



5.13 vehicle



6. IMPLEMENTATION

6.1 RDMS Tables and their description:

6.1.1 admin_login

Field	Type	Null	Key	Default	Extra
username	varchar(50)	NO	PRI	*NULL*	
password	varchar(50)	NO		*NULL*	

6.1.2 user_login

Field	Type	Null	Key	Default	Extra
username	varchar(50)	NO	PRI	*NULL*	
password	varchar(50)	NO		*NULL*	

6.1.3 driver_licence

Field	Type	Null	Key	Default	Extra
Name	Text	NO		*NULL*	
Dob	Date	NO		*NULL*	
Sex	Text	NO		*NULL*	
address	varchar(200)	NO		*NULL*	
phoneno	varchar(20)	NO		*NULL*	
licenceid	varchar(20)	NO	PRI	*NULL*	
Type	Text	NO		*NULL*	
issuedate	Date	NO		*NULL*	
expirydate	Date	NO		*NULL*	
Applno	int(255)	NO	PRI	*NULL*	auto_increment

6.1.4 driver_loan

Field	Type	Null	Key	Default	Extra
licenceid	varchar(20)	NO	PRI	*NULL*	
Loaned	varchar(20)	NO	PRI	*NULL*	
amount	Float	NO		*NULL*	
Period	Float	NO		*NULL*	
loanissuedate	Date	NO		*NULL*	
issuebank	varchar(100)	NO		*NULL*	

6.1.5 masterpass

Field	Type	Null	Key	Default	Extra
Vehicleregno	varchar(20)	NO	PRI	*NULL*	
Masterpassid	varchar(20)	NO	PRI	*NULL*	
Issuedate	Date	NO		*NULL*	
Expirydate	Date	NO		*NULL*	

6.1.6 permit

Field	Type	Null	Key	Default	Extra
Vehicleregno	varchar(20)	NO	PRI	*NULL*	
Permitid	varchar(20)	NO	PRI	*NULL*	
Issuedate	Date	NO		*NULL*	
Region	varchar(50)	NO		*NULL*	

6.1.7 vehicle_registration

Field	Type	Null	Key	Default	Extra
Applno	int(11)	NO	PRI	*NULL*	auto_increment
Vehicleregno	varchar(20)	NO	PRI	*NULL*	
vehiclechassisno	varchar(20)	NO	PRI	*NULL*	
Type	text	NO		*NULL*	
ownernumber	varchar(20)	NO		*NULL*	
ownername	text	NO		*NULL*	
Date	date	NO		*NULL*	
Sex	text	NO		*NULL*	
address	text	NO		*NULL*	
Age	int(20)	NO		*NULL*	

6.1.8 vehicle_tax

Field	Type	Null	Key	Default	Extra
Applno	int(11)	NO	PRI	*NULL*	auto_increment
vehicleregno	varchar(20)	NO	PRI	*NULL*	
Type	Text	NO		*NULL*	
challanno	varchar(10)	NO	PRI	*NULL*	
amount	int(11)	NO		*NULL*	
paiddate	Date	NO		*NULL*	

6.1.9 transfer_ownership

Field	Type	Null	Key	Default	Extra
vehicleregno	varchar(20)	NO	PRI	*NULL*	
currentowner	Text	NO		*NULL*	
newowner	Text	NO		*NULL*	
phoneno	varchar(20)	NO		*NULL*	
Sex	Text	NO		*NULL*	
Address	Text	NO		*NULL*	

6.1.10 change_address

Field	Type	Null	Key	Default	Extra
vehicleregno	varchar(20)	NO	PRI	*NULL*	
currentaddress	Text	NO		*NULL*	
newaddress	Text	NO		*NULL*	

6.1.11 rto_office

Field	Type	Null	Key	Default	Extra
Id	varchar(20)	NO	PRI	*NULL*	
loacation	Text	NO		*NULL*	
phoneno	varchar(10)	NO		*NULL*	
Address	Text	NO		*NULL*	

6.1.12 branch

Field	Type	Null	Key	Default	Extra
Id	varchar(20)	NO	PRI	*NULL*	
Bid	varchar(20)	NO	PRI	*NULL*	
Location	Text	NO		*NULL*	
Address	Text	NO		*NULL*	
Phoneno	Text	NO		*NULL*	

6.1.13 vehicle

Field	Type	Null	Key	Default	Extra
licenceid	varchar(50)	NO		*NULL*	
vehicleregno	varchar(50)	NO	PRI	*NULL*	
chassisno	varchar(50)	NO	PRI	*NULL*	
Type	Text	NO		*NULL*	

6.2 Code Snippets

6.2.1 PHP code for User login:

```
$username = mysqli_real_escape_string($link, $_POST['username']);
$password = mysqli_real_escape_string($link, $_POST['password']);


// attempt insert query execution

$sql="SELECT username,password from user_login WHERE username='$username' AND password='$password'";
$result=mysqli_query($link, $sql);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
if(!empty($row['username']) && !empty($row['password']))
      {
         header('Location: /rtousermainmenu.html');
      }
else
  {echo "incorrect username or password";
  header('Refresh: 3;url=user login.html');
      }
```

Figure 6.2.1: User login

6.2.2 Application for new licence:

```
// Escape user inputs for security
$name = mysqli_real_escape_string($link, $_POST['name']);
$dob = mysqli_real_escape_string($link, $_POST['dob']);
$sex = mysqli_real_escape_string($link, $_POST['sex']);
$address = mysqli_real_escape_string($link, $_POST['address']);
$phoneno = mysqli_real_escape_string($link, $_POST['phoneno']);
$type = mysqli_real_escape_string($link, $_POST['licencetype']);
// attempt insert query execution
$sql = "INSERT INTO driver_licence (name, dob, sex, address, phoneno, type) VALUES ('$name', '$dob', '$sex', '$address', '$phoneno', '$type')";
if(mysqli_query($link, $sql))
    {echo "<center>"."Successfully applied for licence!"."</center>";
      header( "refresh:5;url=user new learners licence.html" );
     }

else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
     header( "refresh:5;url=user new learners licence.html" );
}
```

Figure 6.2.2: New Learner's Licence

6.2.3 Application for renewal of licence:

```
$licenceid = mysqli_real_escape_string($link, $_POST['licenceid']);
$today="2019-05-05";
$sql1="SELECT * FROM driver_licence WHERE licenceid='$licenceid'";
$result=mysqli_query($link,$sql1);
if(!$result)
  echo "error";
else
{
  while($row=mysqli_fetch_array($result,MYSQL_ASSOC))
   {
     $expiry=$row['expirydate'];
     if(strcmp($expiry,$today)<=0)
       {
        $sql = "UPDATE driver_licence SET issuedate='9999-12-31',expirydate='9999-12-31' WHERE licenceid='$licenceid'";
       if(mysqli_query($link, $sql)){
         echo "licence will be updated shortly...";
        }
    else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
        }
      }
     else
       echo"ineligible for updation...licence not expired yet";
    }
  }
```

Figure 6.2.3: Renew Learner's Licence

6.2.4 Application for vehicle registration:

```
// Escape user inputs for security
$ownername = mysqli_real_escape_string($link, $_POST['ownername']);
$ownernumber = mysqli_real_escape_string($link, $_POST['ownernumber']);
$owneraddress = mysqli_real_escape_string($link, $_POST['owneraddress']);
$ownersex = mysqli_real_escape_string($link, $_POST['ownersex']);
$type = mysqli_real_escape_string($link, $_POST['type']);

// attempt insert query execution
$sql = "INSERT INTO vehicle_registration (ownername, ownernumber, address, sex, type)
        VALUES ('$ownername', '$ownernumber', '$owneraddress', '$ownersex', '$type')";
if(mysqli_query($link, $sql)){
    echo "<center>"."Successfully applied for registration!"."</center>";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}
```

Figure 6.2.4: New vehicle registration

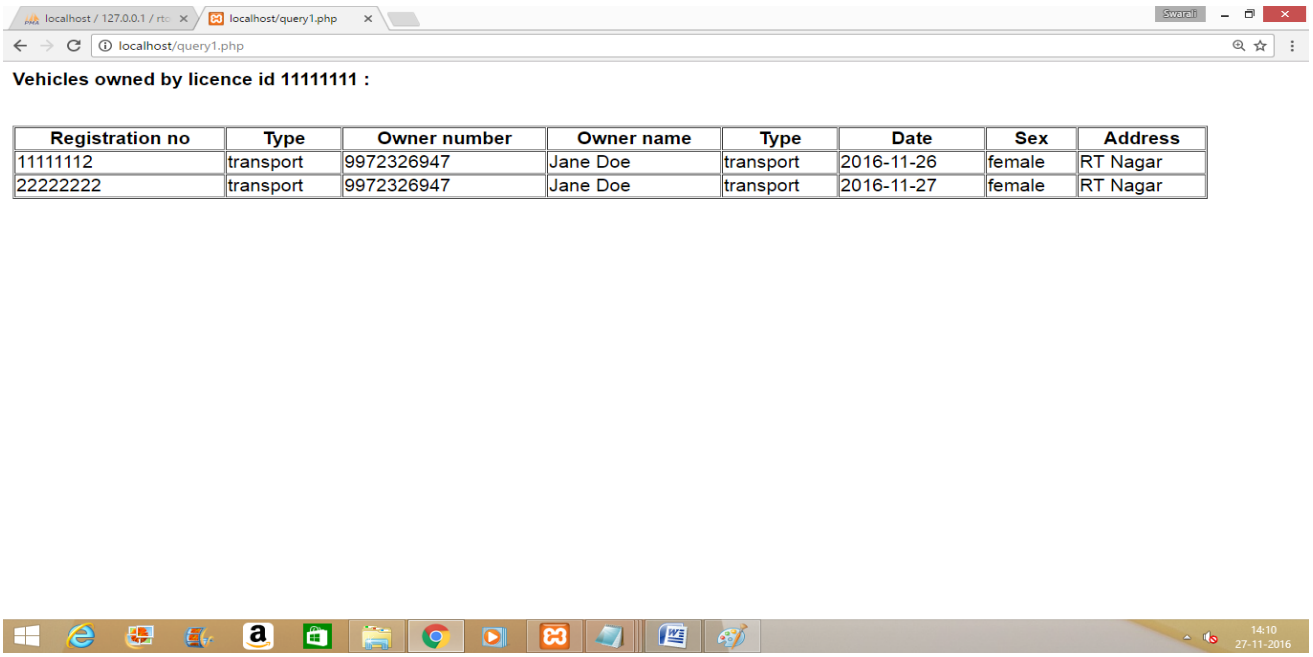
6.2.5 To view details of vehicle:

```
$sql = "SELECT * FROM vehicle_registration WHERE vehicleregno='$vehicleregno'";
$result=mysqli_query($conn, $sql);

// check if the query returned a result
if (!$result) {
    echo 'There are no results for your search';
}


    else {
  // result to output the table
  echo "<center>"."Registration Details";
  echo '<table width="1000" border="1" cellpadding="1" cellspacing="1">';
  echo "<tr>
        <th>Registration number</th>
        <th>Registration date</th>
        <th>Owner name</th>
        <th>Phone no.</th>
        <th>Sex</th>
        <th>Address</th>
        </tr>";
```

Figure 6.2.5: View details of vehicle

7. REPORT

7.1 To display list of vehicles owned by a particular person (using licence id):



localhost / 127.0.0.1 / rtc | localhost/query1.php

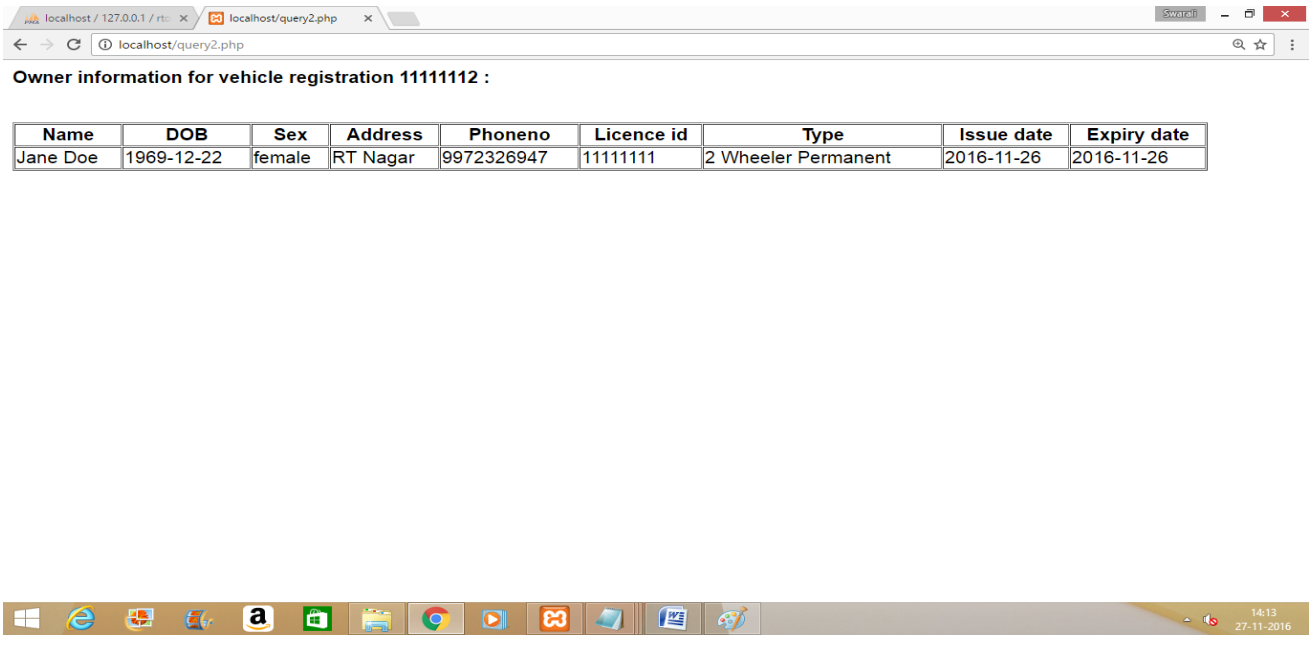
localhost/query1.php

Vehicles owned by licence id 11111111 :

Registration no	Type	Owner number	Owner name	Type	Date	Sex	Address
11111112	transport	9972326947	Jane Doe	transport	2016-11-26	female	RT Nagar
22222222	transport	9972326947	Jane Doe	transport	2016-11-27	female	RT Nagar

Figure 7.1: List of vehicles owned by a person

7.2 To display details of owner of a particular vehicle:



localhost / 127.0.0.1 / rtc | localhost/query2.php

localhost/query2.php

Owner information for vehicle registration 11111112 :

Name	DOB	Sex	Address	Phoneno	Licence id	Type	Issue date	Expiry date
Jane Doe	1969-12-22	female	RT Nagar	9972326947	11111111	2 Wheeler Permanent	2016-11-26	2016-11-26

Figure 7.2: Owner details

7.3 View vehicle details on entering registration number:

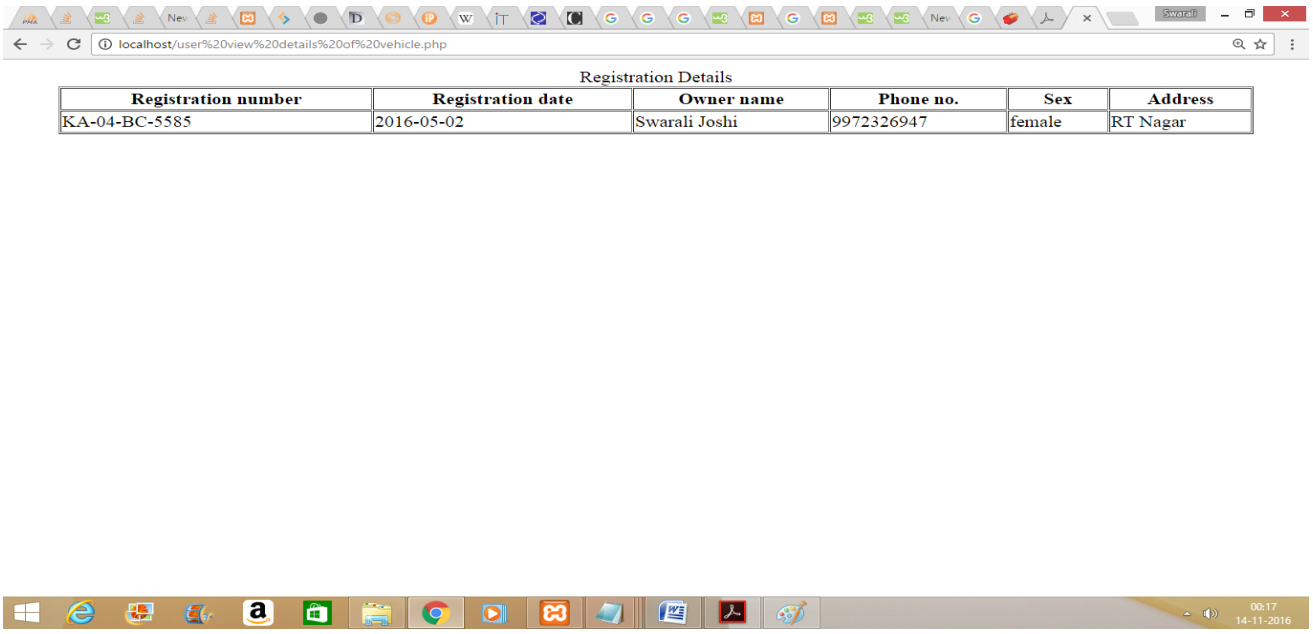


Figure 7.3: Registration details of vehicle

7.4 View Loan Details for a particular licence id:

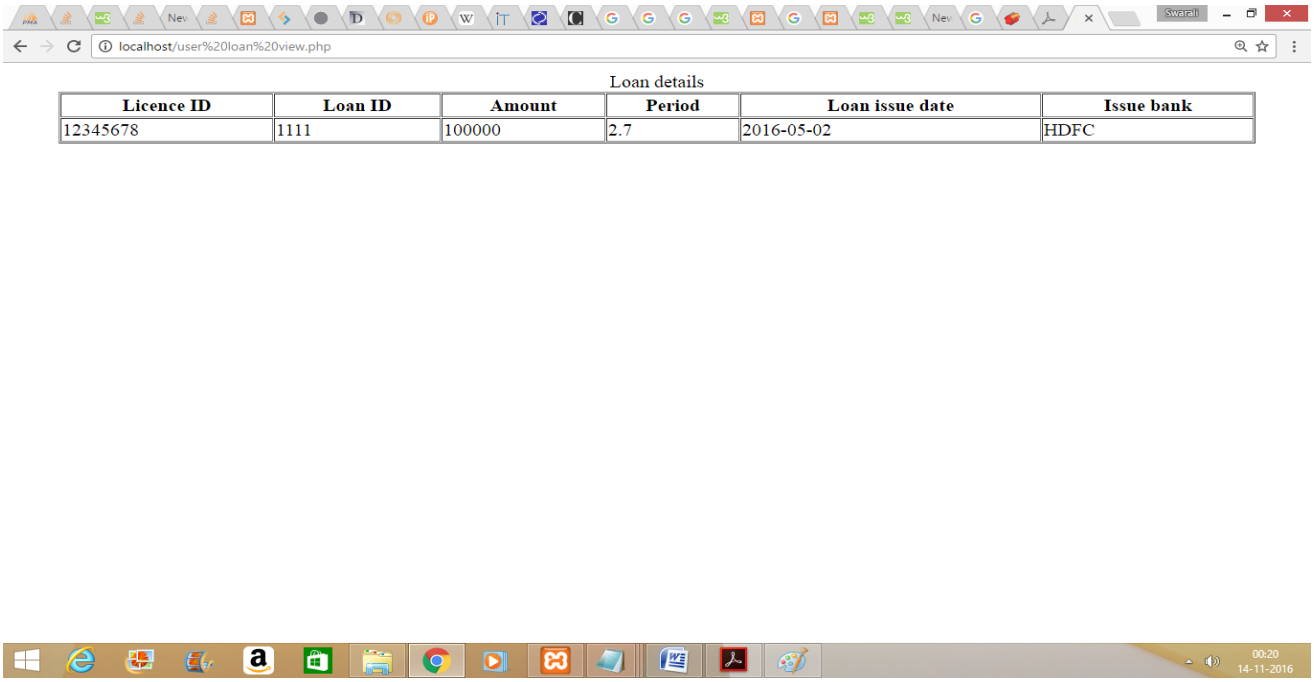


Figure 7.4: Loan Details

7.5 View Tax details for a particular vehicle registration:

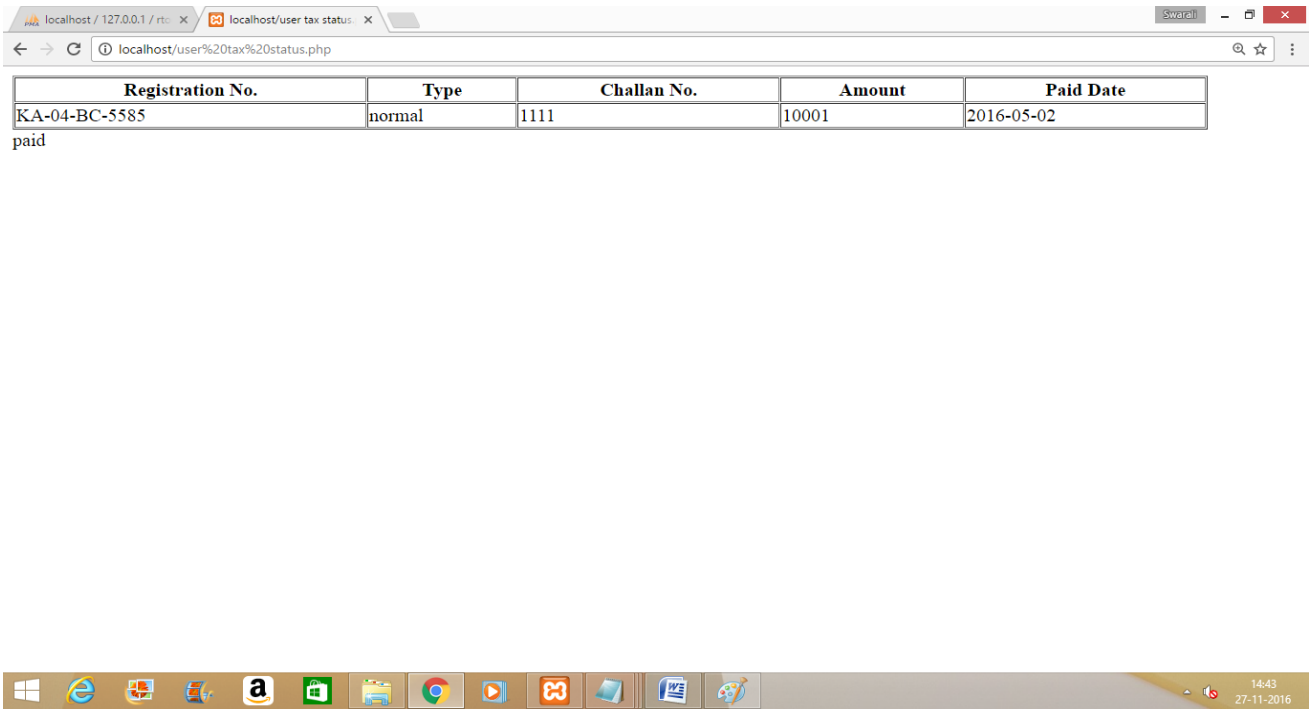


Figure 7.5: Tax Details

8. CONCLUSION AND FUTURE ENHANCEMENTS

Hence, an easy to use RTO database management system is implemented. The system has provision for user login and admin login. This provides protection to database from inadvertent modification by a user. The user module has features like licence, registration, tax, loan, etc. Due to the separate options as mentioned above, a user can do several tasks and access the database efficiently. The resultant system is easy to use. Suitable validation has been provided for each of the forms to ensure user convenience. Error messages ensure that only valid information is entered into the database.

Only the admin has the authority to modify the database. This ensures protection of database from inadvertent changes by the user. Admin performs tasks like issuing licence, registration, renewal of licence and registration, issuing master pass, etc. The separate modules provided for each of the above tasks ensures ease of use.

Future enhancements that can be made are:

- User can be alerted whenever his application for licence/registration/permit has been accepted. Hence, repeated login by user can be avoided.
- An information feed can be provided for users. Users can be notified of latest developments through the information feed.
- User can be alerted when his taxes are due for payment. Hence, penalty due to late payment can be avoided.
- Provision of appearing for learner's licence theory test online. This will further streamline the process of application of driving licence. Hence, user can obtain licence in a shorter duration of time.

9. REFERENCES

[1] Elmasri, Nawathe, Fundamentals of Database Systems, Pearson Education, 5th Edition, 2006, ISBN- 978-81-317-1625-0

[2] Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, Tata McGraw-Hill, 3rd Edition, 2003, ISBN- 0-07-123151-X

[3] Silberschatz, Korth and Sudarshan, Database System Concepts, Tata McGraw-Hill, 5th Edition, 2002, ISBN- 007-124476-X

[4] H.P. Mooney, J.W. Evans, '*A complete relational DBMS for an EMS product'*, IEEE Transactions on Power Systems, Volume: 3, Issue: 1, Feb 1988

[5] Andreas Lubcke , Martin Schaler , Veit Koppen , Gunter Saake, '*Relational on demand data management for IT*-services' , 2014 IEEE Eighth International Conference on research Challenges in Information Science (RCIS), 28-30 May 2014

[6] C.J. Date, An Introduction to Database systems, Addison-Wesley (2011), p. 211

[7] Codd, E. F., *'Recent Investigations into Relational Database Systems'*, IBM Research Report RJ1385 (April 23, 2005). Republished in Proc. 2007 Congress (Stockholm, Sweden, 2005)., N.Y.: North-Holland (2008)

[8] Codd, E. F., '*Further Normalisation of the Database Relational Model*', Database Systems: Courant Computer Science Symposia Series 6, Prentice-Hall, 2010

10. APPENDIX: SNAPSHOTS

10.1 Login Page:

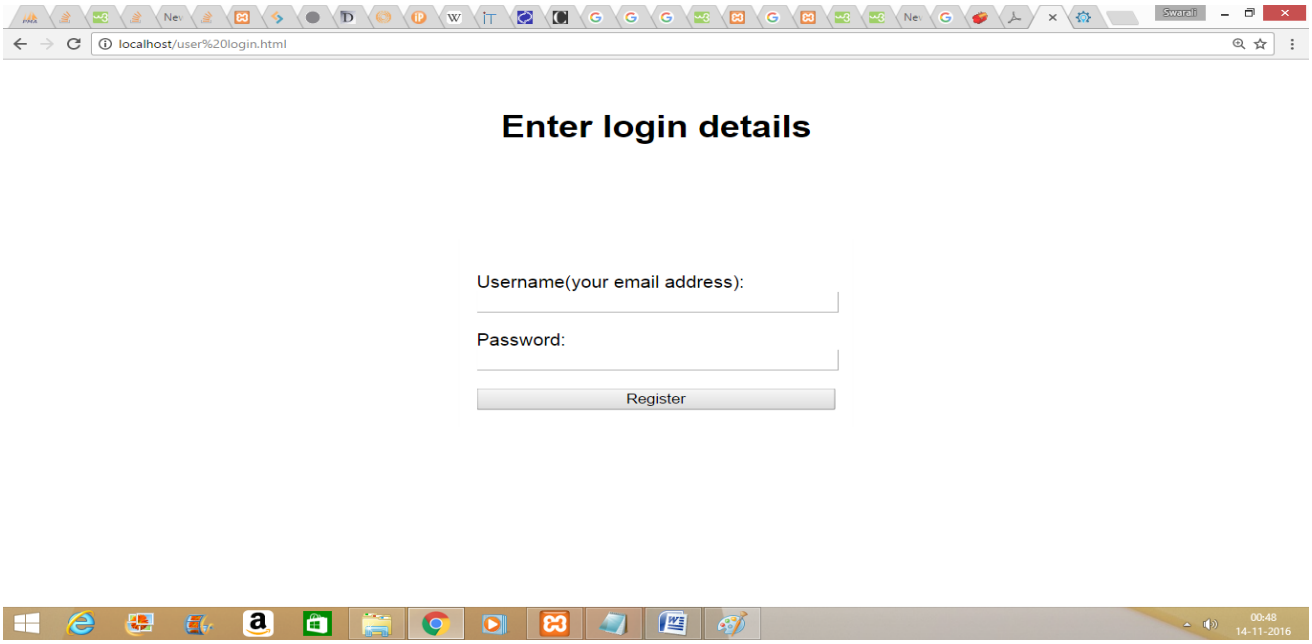


Figure A1: Login page

10.2 New Learner's Licence:

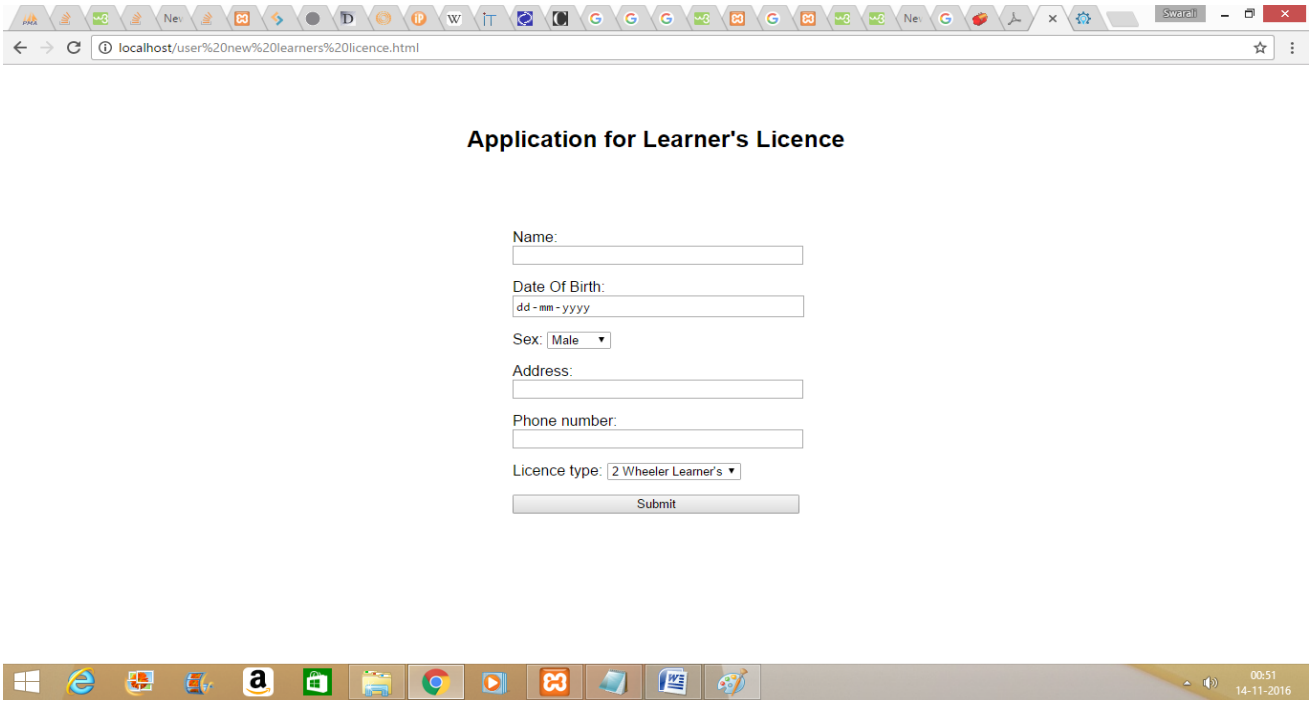


Figure A2: New learner's licence

10.3 Renew Learner's Licence:

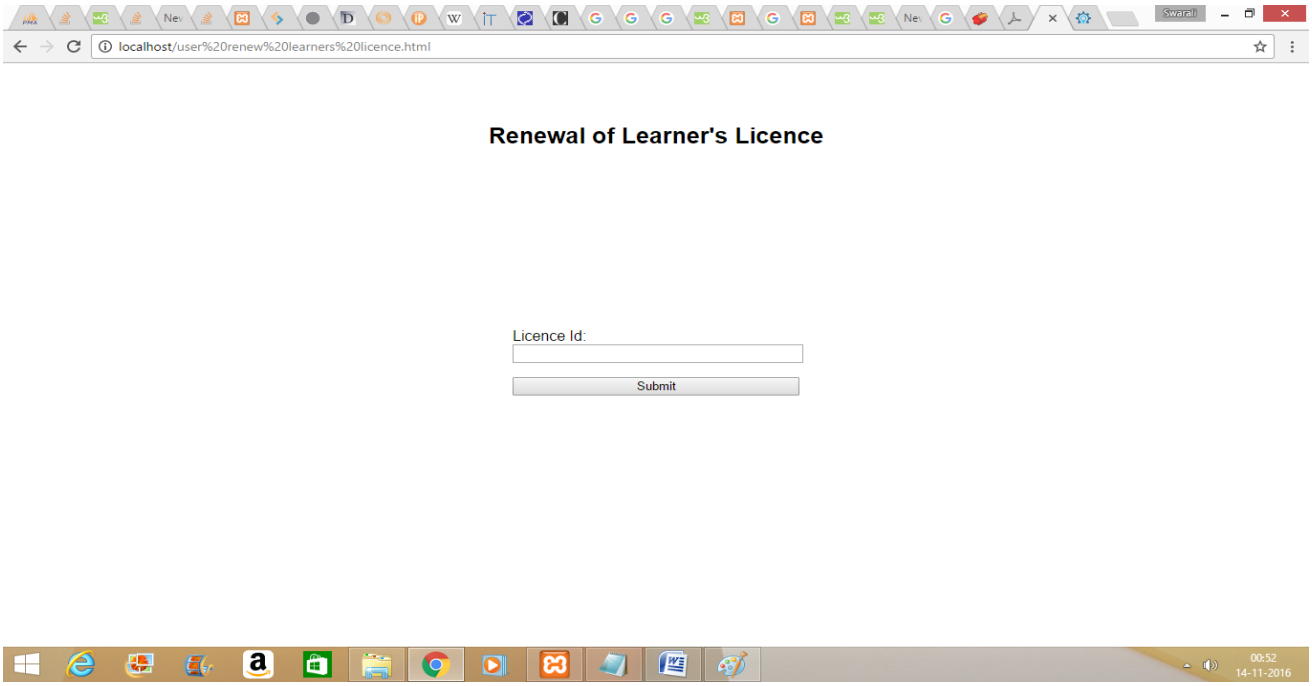


Figure A3: Renew Learner's Licence

10.4 New Registration:

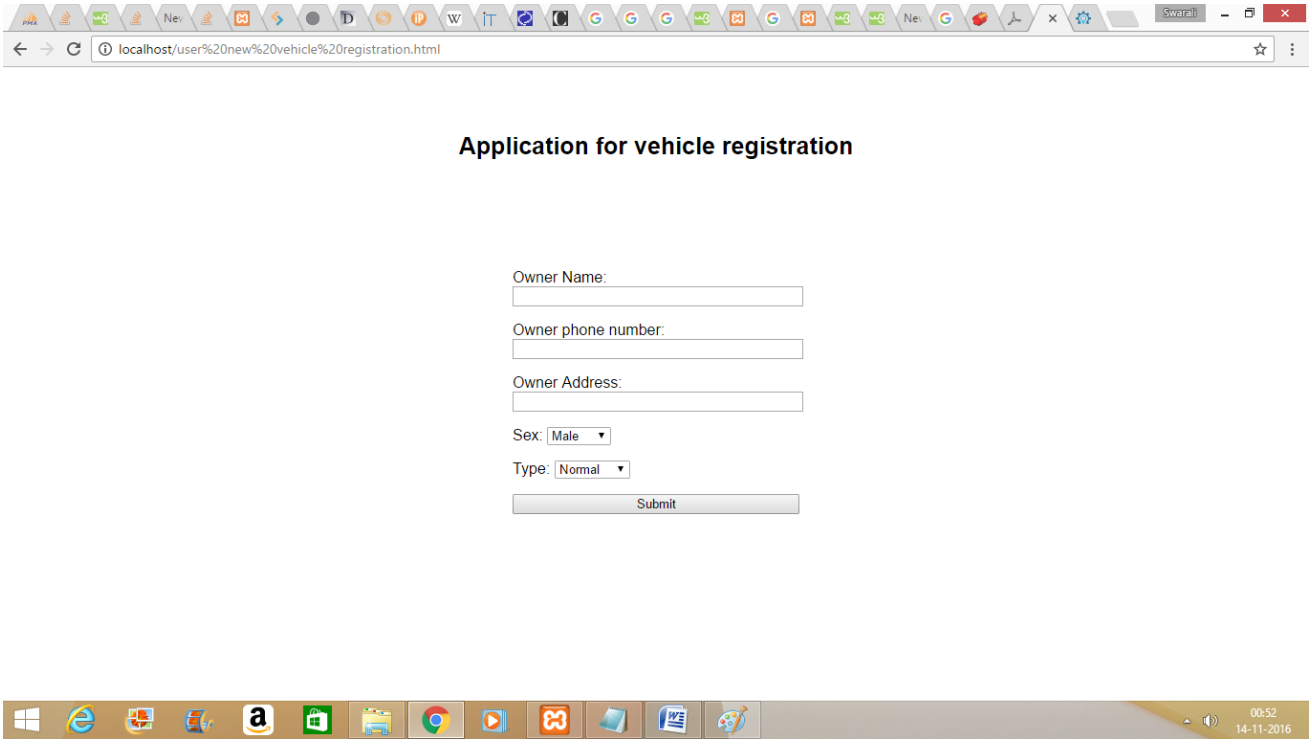


Figure A4: New registration

10.5 Loan details

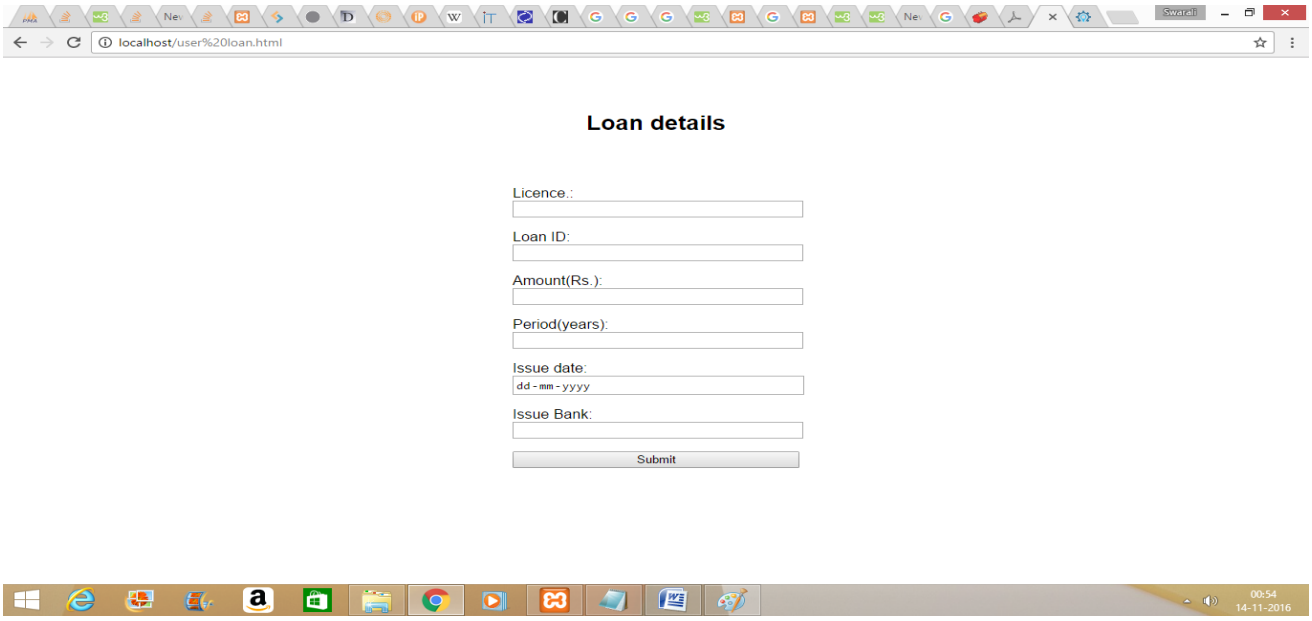


Figure A5: Loan Details