Final Year Project Report
Drive legally

Project Advisor:
Mr.Asif Subhani

Submitted By:

Waleed Mustafa	12019020032
Muhammad Rizwan Amin	12019020051
Tanveer Rasheed	12019020052

Session:
Fall 2014
&
Spring 2015

University of Management and Technology

Dedication

This project and work done by our team is dedicated to Mr. Asif Subhani project advisor he helped us in all the possible manner. We appreciate the efforts of the Dean -------- Computer Science department and we Thank Director Dr.Tahir Ijaz and Director projects Mr. Farooq Ali. All the respected faculty members of UMT are also appreciated for their support and believe in us.
This project is dedicated to my Parents, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my Teachers, who taught me that even the largest task can be accomplished if it is done one step at a time.

Final Approval

Panel of Examiners

Head of Department _____
Dr. Tahir Ejaz
Department of Computer Science
UMT Lahore

Program Director (Final Year Projects) _____
Syed Farooq Ali
Department of Computer Science
UMT Lahore

Supervisor _____
Muhammad Asif Subhani
Department of Computer Science
UMT Lahore

Co-Supervisor _____
Abu Bakar
Department of Computer Science
UMT Lahore

Acknowledgment

I am highly indebted to the entire team of faculty of University of management and technology and specially Mr Asif Subhani (our advisor), my friend and fellows for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

Project Title:	Drive Legally
Objective	Increase number of educated and licensced driviers
Undertaken by:	Waleed Mustafa Muhammad Rizwan Amin Tanveer Rasheed
Supervised by:	Mr. Asif Subhani
Starting Date:	6-Jan-2015
Completion Date:	3-Mar-2016
Tools Used:	Android Studio, MS Office, Wamp, Dreamweaver.
Operating System:	Microsoft Windows 10

Abstract

The biggest challenge that we envision in developing this project will be coming up with a good design including an efficient and friendly user interface. A good design early on will make the development process easier by allowing team members to be assigned well defined areas of responsibility. Reduced coupling in the design will allow for better software verification, and also improve flexibility in case implementation has to be modified at a later stage. We will be designing a new platform for Drivers to drive in a safe zone knowing all rules regulation and all other stuff that could guide them while they are on the roads of Pakistan.

TABLE OF CONTENTS

Project/Product Feasibility Report

Drive legally is an android based application which will be help for the drivers on Pakistani roads. This consists of a simple visuals and text easy to understandable for a lay man. We have sufficient technology and technical person available for that project.

Following are the types of feasibilities:

1. Operational
2. Economic
3. Legal and Ethical
4. Technical
5. Schedule
6. Information
7. Specification
8. Motivational

Operational Repot of Feasibility

Calculating the technical abilities of crew to run the system is the basic target of repot for operational feasibility stats. Working feasibility decides if the human resources are available to run the system once it's installed. Users who do not want a new system would to resist the system from being operationally feasible. So we take that type of users which can operate easily.

Economic Repot of Feasibility

We have cost and time available for that project. We have available software, hardware and new equipment.

Legal & Ethical Repot of Feasibility

This project is legal and ethical. Drive legally does work ethical and legal.

Technical Repot of Feasibility

Technical repot of feasibility deals with asking whether the system can be developed or not. Technical feasibility measures whether the current technical resources are enough for the new system or not. We have sufficient technology for this project and we have also technical person available for that project which can do this project easily.

Schedule Repot of Feasibility

Time factor matters the most in calculating the schedule repot of feasibility. Calculating and making estimations about finishing of project with the provided crew members and resources, on given interval is really important. Achieving milestones and beating deadlines simultaneously must be kept in mind. The Gantt chart is given below in which deadline and milestones are given.

Information Repot of Feasibility

The information feasibility must be of high accuracy regarding its finishing, reliability, and relevance.

Specification Repot of Feasibility

Requirements are clear and certain. The scope boundaries are clearly examined and defined.

Motivational Repot of Feasibility

Assessment of the client crew related the motivational tasks to perform the essential steps must occur correctly.

Project/Product Costing:

Cost estimation is being done utilizing only one method.

Project Cost Estimation by Function Point Analysis

Functional facts are being calculated by filling the table in figure given below. These are five different entities being calculated and are provided in table and results are obtained.

Type of Component	Complexity of Components			
	Low	Average	High	Total
External Inputs	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
External Outputs	___ x 4 = ___	___ x 5 = ___	___ x 7 = ___	
External Inquiries	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
Internal Logical Files	___ x 7 = ___	___ x 10 = ___	___ x 15 = ___	
External Interface Files	___ x 5 = ___	___ x 7 = ___	___ x 10 = ___	
			Total Number of Unadjusted Function Points	
			Multiplied Value Adjustment Factor	
			Total Adjusted Function Points	

Information domain values are defined in the following manner:
Number of user iputs: In our project external input is 4
Number of user outputs: In our project external output is 30
Number of user inquiries: In our project external inquiries is 40
Number of files: In our project external logical file is 1
Number of external interfaces: In our project external Interface files is 0
Total number of Unadjusted Function Points=259

To compute function points (FP), the following relationship is used:

FP est. = 259 * [0.65 + 0.01 * (32)]=251.23

The value adjustment factor is as follows

Number	Activity	Value
1	Distributed Data Processing	0
2	Performance	2
3	Heavily Used Configuration	0
4	Transaction Rate	1
5	Online Data Entry	4
6	End User Efficiency	3
7	Complex Processing	2
8	Reuse Ability	1
9	Installation Ease	2
10	Operational Ease	1
11	Multiple Sites	1
12	Facilitate Change	3
13	Data Communications	0
14	Online Update	1

Finally, Total Project Cost and Total Project Effort are calculated given the average productivity parameter for the system.

The formulae are given as follows:

As calculated early, developers in our association average 18 functional points each month 259/18=14.38 Man-Month
Labor rate=$5000
Total Project Cost = $5000 * (14.38) =71900

Effort = 14man-month
TDEV = (2.5) $(14)^{0.38}$ = 6.81 months
Staff required = 14/6.81 = 2.05 people

Estimated cost of Project using COCOMO

Organic: Comparatively teams with less team members use to make such products for a house environment. Mostly team members are familiar with working with such systems.

Basic Pricing

Type	Effort	Schedule
Organic	PM= 2.4 (7.285)1.05=18.35	TD=2.5(18.35)0.38=17.4325

PM= person's Monthly (effort)
KLOC= Code of lines in thousands
TD= Time estimated in moths for development (duration)

Intermediate Pricing

Type	Effort
Organic	Effort = EAF *A (KSLOC) ^{exp} = ((1) (3.2) $(7.285)^{0.38}$) = 6.8manmonths

- EAF = Nominal, use 1
- A = constant based on developmental mode - 3.2
- Exp = constant based on developmental mode - 0.38

– TDEV = (2.5)$(6.8)^{0.38}$ = 5.17 months
– Staff Required = 6.8/5.17 = 1.312 people

Critical Path Method

Steps in CPM Project Planning

1. Individual's Specification of activities.
2. Calculation of activities sequence.
3. Estimate the completion time for each activity.
4. Activity Completion time estimate.
5. CPM Diagram Update.

Individual's Specification of Task

1st task is project proposal (A) or project definition. 2nd activity is planning of project (B) 3rd is project specification (C)4th activity is project costing(D) 5th activity is making project schedule(E) 6th activity is designing (F)7th activity is top level designing (G) 8th activity is prototyping(H) 9th activity is detailed designing (I) 10th activity is program implementation. (J) 11th task is testing (K) 12th task is component testing (L), 13 task is system testing.(M) , 14 activity is integration testing(N) 15 activity is System delivery= (O)

Determination of Activities Sequence

Activity B depend on Activity A , Activity C, D, E depend on Activity B , Activity F depend on Activity C ,D ,E , Activity G , H , I depend on activity f , J activity depend on G ,H ,I. Activity K is depend on activity J ,activity N depend on activity K ,activity M depend on activity N , activity O depend on activity M

Activity Completion Time Estimate

The estimate activity completion time is given below
The critical path is given below

CPM Diagram Update

As the making of project proceeds, the time required to complete the project will be known and that information can be implemented to network diagram and network diagram can be updated. New critical paths could be formed and there may be some structural changes too in network diagram.

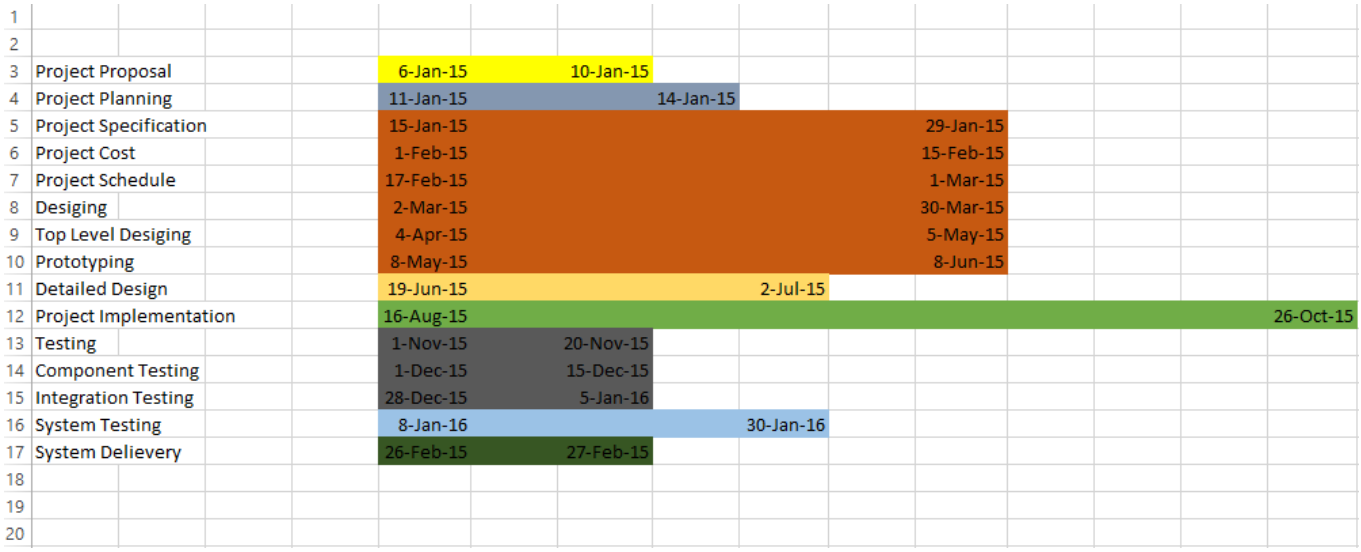
Example:

Activity	Immediate Predecessor	Duration (Weeks)
A	None	5
B	A	3
C	B	8
D	B	7
E	B	6
F	C,D,E	4
G	F	5
H	F	3
I	F	4
J	G ,H ,I	3
K	J	2
N	K	2
M	N	2
O	M	2

Activity	Duration	ES	EF	LS	LF	TS
A1	5	0	5	0	5	0
B1	3	5	8	5	8	0
C1	8	8	16	8	16	0
D1	7	8	15	9	16	1
E1	6	8	14	10	16	2
F1	4	16	20	16	20	0

G1	5	20	25	20	25	0
H1	3	20	23	22	25	2
I1	4	20	24	21	25	1
J1	3	25	28	25	28	0
K1	2	28	30	28	30	0
N1	2	30	32	30	32	0
M1	2	32	34	32	34	0
O1	2	34	36	34	36	0

Gant Chart



The Stacks and Parameters are calculated:

Following is the critical path:

A-> B-> F-> G-> J -> K-> N-> M-> O

Project Overview

Drive legally would guide common public about the traffic laws. It would help common people about learning different types of traffic signs, guide about violation and fines and where to pay fines, how to get driving license, when the tests are about to be conducted for license, the procedure, date and time for licenses tests for different locations of Punjab.

Problem Statement

Most of the people in Pakistan do not get the driving license and yet they are very good drivers and those who have license there mostly reason is this that they have someone well known in the authorities or they have paid the bribe for it. As far as people who can drive well do not know about the traffic signs what they mean. When someone is fined his 1st concern is not this that he has to pay the fine his tension is to find the bank where fine is to be paid and then find the collection center and mostly in finding the places too much time is spent that when they reach the spot than time is up.

User

Most common users for this application are common people of Pakistan. Initial targeted users are from Lahore and some other main cities of Punjab like Multan, Faisalabad, Okara. One who want to get new licenses renew them. And mostly those are fined and want to know the location of respective bank where fine is to be paid and from which collection center they are supposed to get their document.

Affected Groups

Those impacted in a positive manner by the deployment of the system are

- Public
- Traffic Police Departments

Public would have a great helping impact by getting almost every information regarding fines and where to pay them, from where to collect their document and how to get new license or re new them. Traffic Police Department would not be in any need to argue to anyone about any disobedience of law.

Assumptions

It would help people to obey the traffic rules:

- People would get driving license when they will come to know how easy is to get it

- Traffic might get some better when they will be knowing about laws.

Dependencies/ External Systems

As application is in Android java language and built for all Android devices and it will be available on Google playstore.

Goals

Main goal is to have educated drivers and maximize the number of drivers having license and help out people where to pay the fines and collect documents

1 Users/stakeholders:

1.1 Primary

Primary user for Drive Legally would be common people. It is an application which will be educating common people about basic traffic laws, signs and fines against them. One who is going for a driving license would get directions how to apply for a new one till the end of process. It would be helping common people to know about rules of driving.

1.2 Secondary

Secondary user for this application would be people having the driving license. People who are not aware of some scenarios of diving can get help from it. Like how someone can renew his driving license. Someone who has never been to motorway can now the requirements that are must to be full filled before entering the motorway.

1.3 Tertiary

There are not going to be any specific tertiary users of this application but if Law enforcement Agencies could get any help which is immediately required and they can get it instead looking for that in books at that time, than this application can help them to some extent.

2 Types of System Interactors:

2.1 Novice User:

Novice user are common person who wants to know about traffic laws and get license. He will be using searching mechanism and categorized details to get to his required information and he can get help about his fine payment where are they meant to be paid and from where he could collect his documents.

2.2 Casual User:

Casual user could be a driver who is a good driver want to have driving license. He will be using this application to boost his learning and understanding process regarding license application. When and where the test is going to held and how much tickets would cost him at which point he has to paste the tickets for how much cost and where to pay the fines and collect documents, against which laws how much fine is to be paid.

2.3 Frequent User:

Frequent user could be a driver who want to know about the traffic laws. Frequent user will be facilitated by keeping track of the search he is making. He could just jump to those results instead of searching them again and again.

2.4 Expert User:

Expert user could just directly look into the application about their required information.

3 Product Requirements:

3.1 Data Requirements:

The data would be persistence until there is an amendment made in the rules provided and data would be accurate.

.

3.2 Environmental Requirements:

This application could be used anywhere, where legal help is required. So it could be used in public places like markets, roadsides, courts and it can be used in personal

3.3 Organizational Requirements:

This application targets a common person so it cannot be providing a good support to any organization. But it can provide support to some extent to traffic police department in licensing field and managing traffic.

.

3.4 Social Requirements:

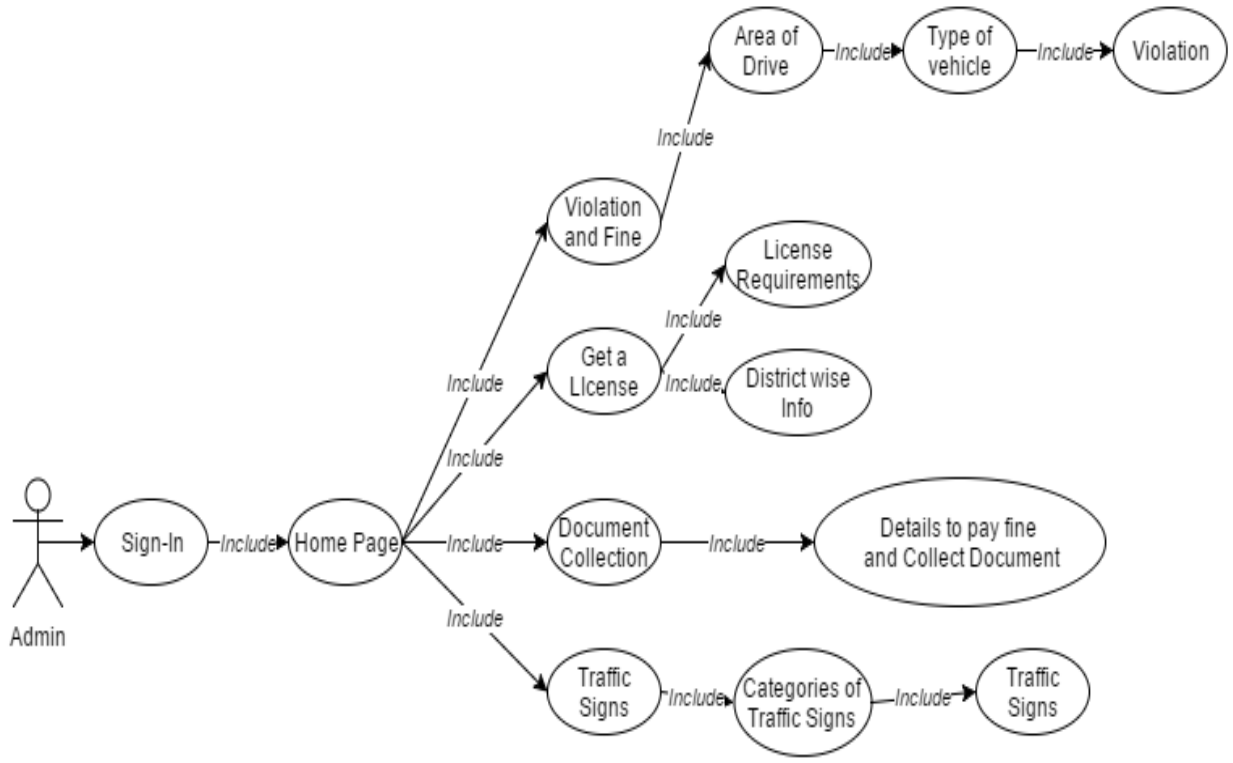
There are no social requirements because the data provided would not be shared

3.5 Technical Requirements:

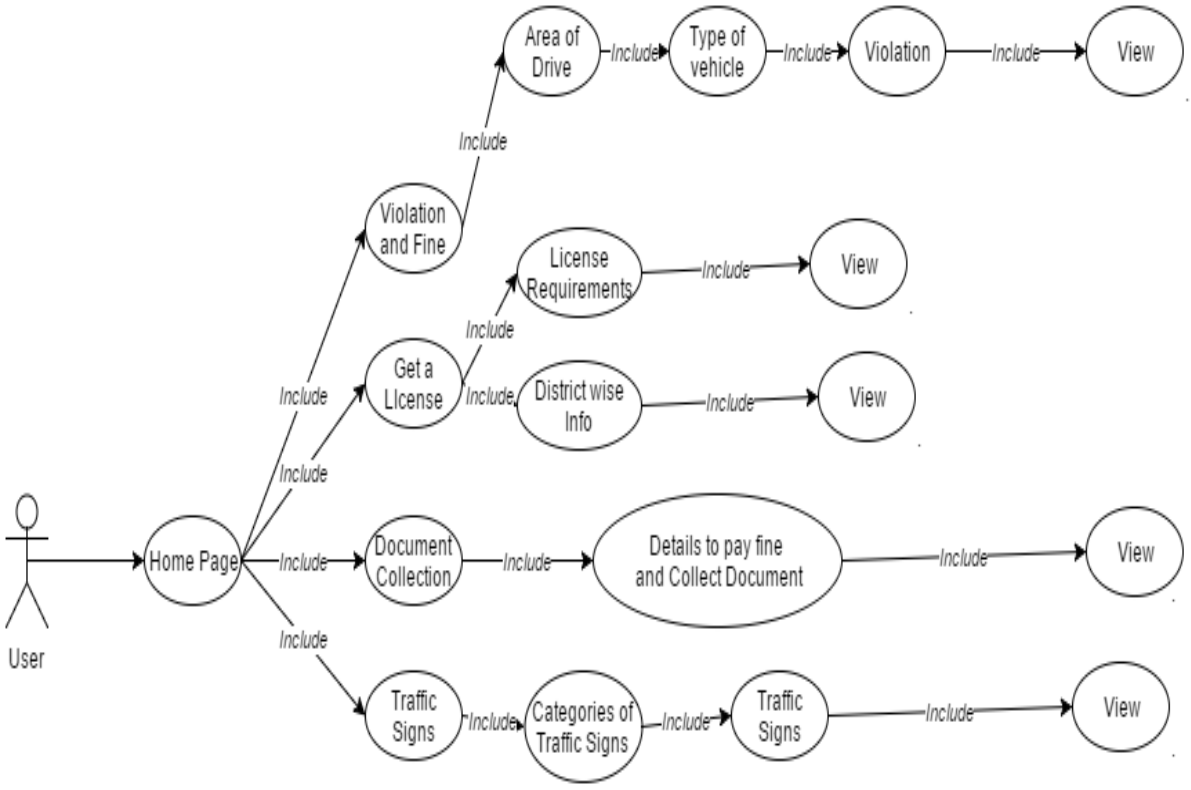
Application would be running on android platform and would be requiring almost 60Mb of free space on the device at least 500Mb of RAM and most important of all internet connection will be required. Any device fulfilling these preconditions can support this application

4 UseCases:

4.1 Admin Use Case Diagram



4.2 User Use Case Diagram



4.3 UseCase : UC_Traffic_Signs

Primary Actor(s):

1. User

Brief Description:

This document describes that user can get information about traffic signs.

Stake Holders and their Interest List:

User: Wants to know about the traffic signs

Pre-Conditions:

1. User must be in Traffic signs section.

Success Guarantees (Post-Conditions):

1. User gets the traffic sign details which he wants.

Main Success Scenario or Basic Flow:

1. User opens the application and from 1st menu enters traffic signs section
2. Different types of traffic signs sections like mandatory or informative are shown in next menu
3. Selects any of them which he likes
4. Traffic signs with and picture in a table view are shown
5. Selects the sign which he finally wants to view and can see the traffic sign fully in a single view.

Extensions or Alternative Flows:

1. User is not in traffic signs.
2. User can not view the right traffic sign which he wants to.

4.4 UseCase : UC_Voilations

Primary Actor(s):

1. User

Brief Description:

This document describes that user can get information about violations.

Stake Holders and their Interest List:

User: Wants to know about violations of different type of vehicles

Pre-Conditions:

1. User must be connected to internet connection
2. User must be in violation and fines section.

Success Guarantees (Post-Conditions):

1. User gets the violation details which he wants.

Main Success Scenario or Basic Flow:

1. User enters the violation and fine menu from home screen.
2. User selects the violations.
3. Selects the area for which he is looking up. Like Motorway or NHA.
4. Selects the type of vehicle and HTV or LTV or Motorcycle
5. List related to violations of that specific type of vehicle and area is shown.

Extensions or Alternative Flows:

1. User is not in the violation and fines section.
2. Information shown is not related to the type of vehicle.
3. Information shown is not related to that specific area

4.5 Use Case : UC_Motor_Vehicle_Ordinance

Primary Actor(s):

1. User

Brief Description:

This document describes that user can get legal information from Motor Vehicle Ordinance Code.

Stake Holders and their Interest List:

User: Wants to know for which sections he could be charged against a violation

Pre-Conditions:

1. User must be have selected Violations and fines menu from home secreed.

Success Guarantees (Post-Conditions):

1. User gets the section details against a violation in accordance to Motor Vehicle Ordinance.

Main Success Scenario or Basic Flow:

1. User enters the violation and fine menu from home screen.
2. User selects the violations Motor Vehicle Ordinance.
3. Selects it and can read the ordinances categorized.

Extensions or Alternative Flows:

1. User is not in the violation and fines section.
2. User can not view a specific section.

4.6 UseCase : UC_Fine_Payment_and_Document_Collection

Primary Actor(s):

1. User

Brief Description:

This document describes that user can get information about where is the bank which will collect the fine for that area.

Stake Holders and their Interest List:

User: Wants to know the address of bank and timings for the fine payment

Pre-Conditions:

1. User must be in fines and violations section and select fine payment than select city and respective area of that city and can get directions.

Success Guarantees (Post-Conditions):

1. User gets the directions to bank to document collection center.

Main Success Scenario or Basic Flow:

1. User selects fine payment from home screen.
2. Looks for the city
3. Looks up the area from the city where he was fined
4. Than in description views the bank where fine is to be paid and address of document collection center.

Extensions or Alternative Flows:

1. User is not in the violation and fines section.
2. User can not view the address of bank.
3. User can not view the address of document collection center

4.7 UseCase : UC__Licensing_Requirements

Primary Actor(s):

1. User

Brief Description:

This document describes that user can get information about applying for the license.

Stake Holders and their Interest List:

User: Wants to know how to apply for license.

Pre-Conditions:

1. User must be in licensing information section.

Success Guarantees (Post-Conditions):

1. User gets the details for licensing.

Main Success Scenario or Basic Flow:

1. User enters the licensing information menu from home screen.
2. Selects the license requirements.
3. User selects the type of license.
4. Information related to the process for driving license will be shown.

Extensions or Alternative Flows:

1. User is not in the licensing information section.
2. User has not selected the right type of License.

4.8 UseCase : UC_District_Wise_License

Primary Actor(s):

1. User

Brief Description:

This document describes that user can get information about applying for the license.

Stake Holders and their Interest List:

User: Wants to know how to apply for license.

Pre-Conditions:

1. User must be in licensing information section.

Success Guarantees (Post-Conditions):

1. User gets the details for licensing.

Main Success Scenario or Basic Flow:

1. User enters the licensing information menu from home screen.
2. Selects the license requirements.
3. User selects the type of license.
4. Information related to the process for driving license will be shown.

Extensions or Alternative Flows:

1. User is not in the licensing information section.
2. User has not selected the right type of License.

4.9 Use Case : UC_Admin_Panel

Primary Actor(s):

1. Admin

Brief Description:

This document describes that Administrator can change and edit through server in app.

Stake Holders and their Interest List:

Admin: Wants to update the whole application.

Pre-Conditions:

1. Admin is permitted to edit or update the application.

Success Guarantees (Post-Conditions) :

1. Admin updates and edits the in app information.

Main Success Scenario or Basic Flow:

1. Admin enters the admin panel.
2. Attempts the multiple updates in an app.
3. The app is successfully updated.

Extensions or Alternative Flows:

1. Admin has not been permitted.
2. Server does not update.

4.10 Admin Use Case: UC_Traffic_Signs

Primary Actor(s):

1. Admin

Brief Description:

This document describes that Admin can get information about traffic rule and update or edit the rules.

Stake Holders and their Interest List:

Admin: Wants to edit or update the traffic rules

Pre-Conditions:

1. Admin must be already login
2. Admin must be in Traffic rules section.

Success Guarantees (Post-Conditions):

1. Admin gets or updates the traffic sign details which he wants.

Main Success Scenario or Basic Flow:

1. Admin opens the application from admin panel.
2. Different traffic rules sections like mandatory or informative are shown in table.
3. Selects any of them which he likes
4. Traffic rules with name and text info in a table view are shown
5. Selects the rules which he finally wants to update or edit.

Extensions or Alternative Flows:

1. Updates made are not viewed.
2. Admin is not in traffic rules section.
3. Admin is not authenticated.

4.11 Admin Use Case: UC_Violations

Primary Actor(s):

1. Admin

Brief Description:

This document describes that Admin can get information about traffic Violations and update or edit those violations.

Stake Holders and their Interest List:

Admin: Wants to edit or update the traffic violations

Pre-Conditions:

1. Admin must be already login
2. Admin must be in Traffic violation section.

Success Guarantees (Post-Conditions):

1. Admin gets or updates the traffic violation details which he wants.

Main Success Scenario or Basic Flow:

1. Admin opens the application from admin panel.
2. Different type of traffic violation are shown in table.
3. Selects types of vehicle which he likes
4. Selects area of drive
5. Traffic violations are shown in a table.
6. Selects the violation which he finally wants to update or edit.

Extensions or Alternative Flows:

1. Admin is not in traffic violation section.
2. Admin is not authenticated.

4.12 Admin Use Case: UC_DRIVING_LICENSE

Primary Actor(s):

1. Admin

Brief Description:

This document describes that Admin can change information regarding driving license requirements and district wise licensing information

Stake Holders and their Interest List:

Admin: Wants to edit or update the traffic violations

Pre-Conditions:

1. Admin must be already login.
2. Admin must be in Traffic violation section.

Success Guarantees (Post-Conditions):

1. Admin gets or updates the driving license details which he wants.

Main Success Scenario or Basic Flow:

1. Admin opens the application from admin panel.
2. Opens “Get Driving License”.
3. Selects Licensing Requirements or Licensing centers.
4. Makes updates into it.

Extensions or Alternative Flows:

1. Admin is not in driving license section.
2. Admin is not authenticated.

4.13 Admin Use Case: UC_MVO

Primary Actor(s):

1. Admin

Brief Description:

This document describes that Admin can get and update information about Motor Vehicle Ordinance (MVO).

Stake Holders and their Interest List:

Admin: Wants to edit or update the MVO

Pre-Conditions:

1 Admin must be already login.
2. Admin must be in Traffic violation section.

Success Guarantees (Post-Conditions):

1. Admin gets or updates the MVO details which he wants.

Main Success Scenario or Basic Flow:

1. Admin opens the application from admin panel.
2. Opens “Violation”.
3. Selects Motor Vehicle Ordinance (MVO).
4. Makes updates into it.

Extensions or Alternative Flows:

1. Admin is not in Violations menu.
2. Admin is not authenticated.

4.14 Admin Use Case: UC_Fine_Payment

Primary Actor(s):

1. Admin

Brief Description:

This document describes that Admin can Update the information related to fine payment and document collection.

Stake Holders and their Interest List:

Admin: Wants to edit or update the Fine Payment process

Pre-Conditions:

1. Admin must be already login.
2. Admin must be in Fine Payment section.

Success Guarantees (Post-Conditions):

1. Admin gets or updates the Fine Payment details which he wants.

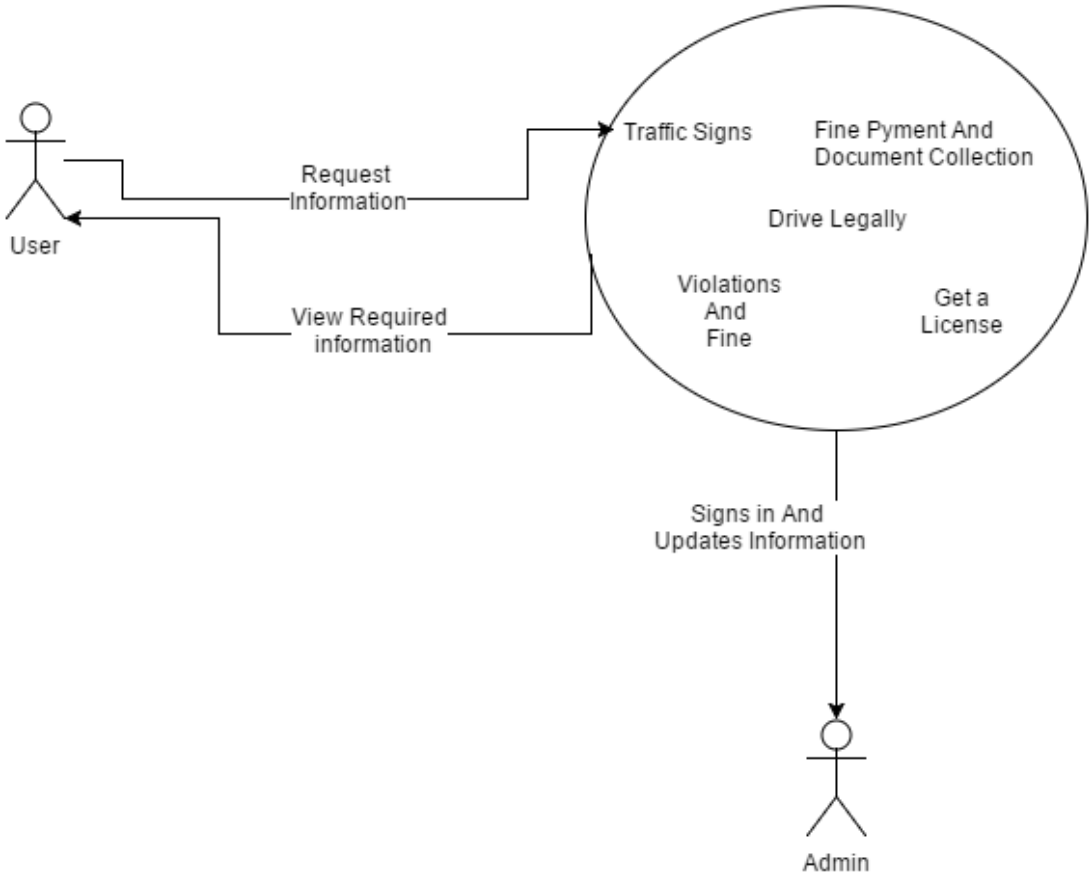
Main Success Scenario or Basic Flow:

1. Admin opens the application from admin panel.
2. Opens “Fine Payment And Document Collection”.
3. Selects field where updates are meant to be made.
4. Makes updates into it.

Extensions or Alternative Flows:

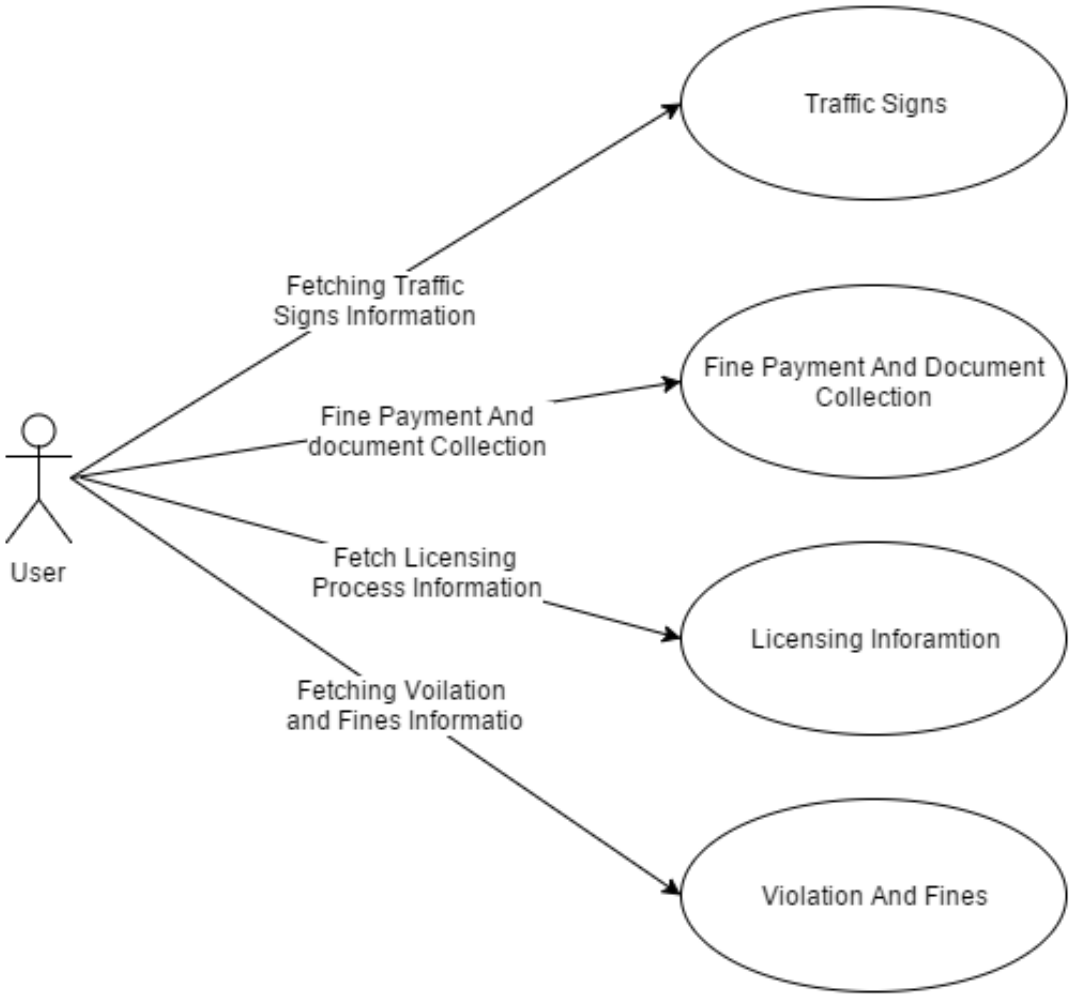
1. Admin is not in driving license section.
2. Admin is not authenticated.

5 Context Level Diagram

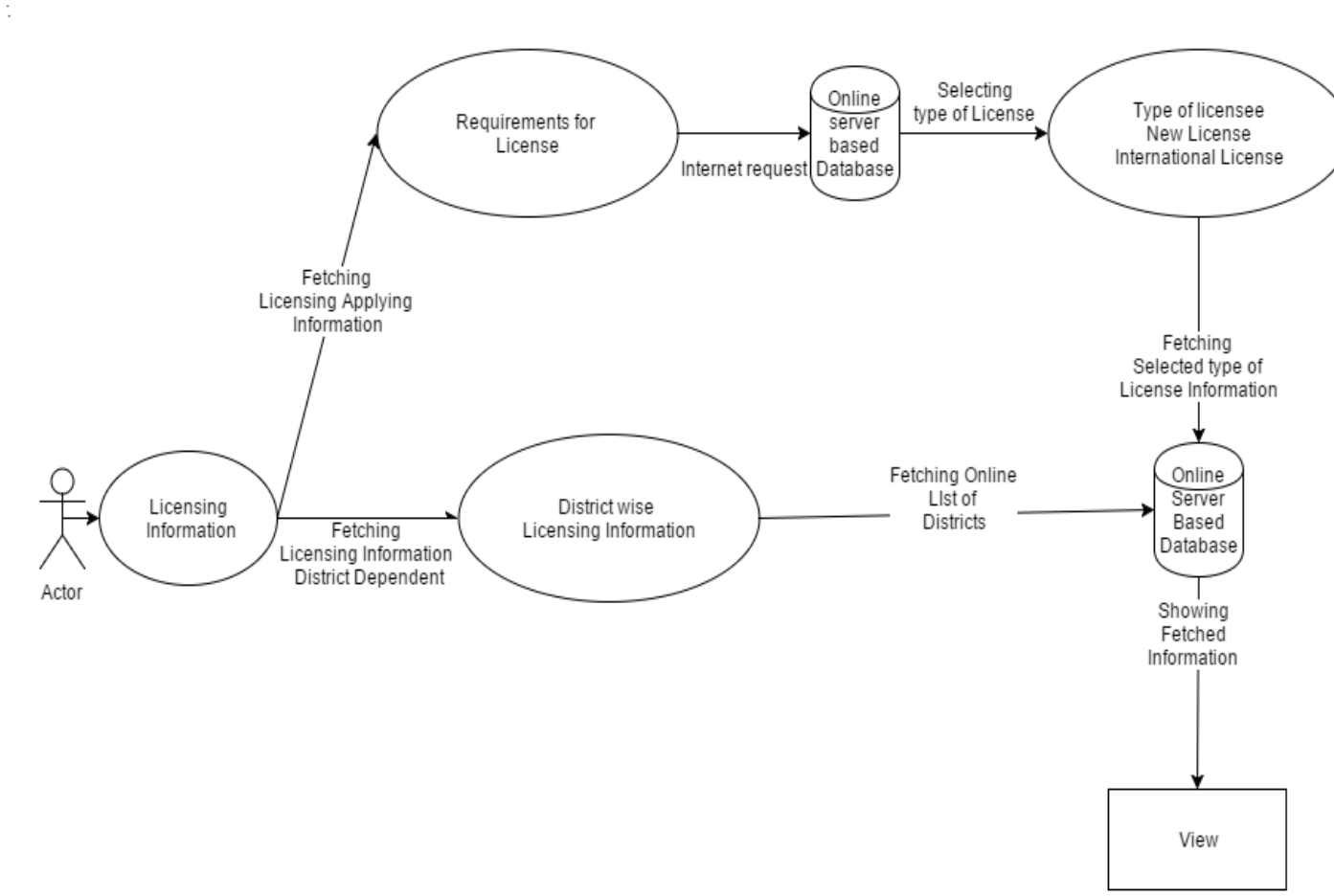


6 Data Flow Diagrams

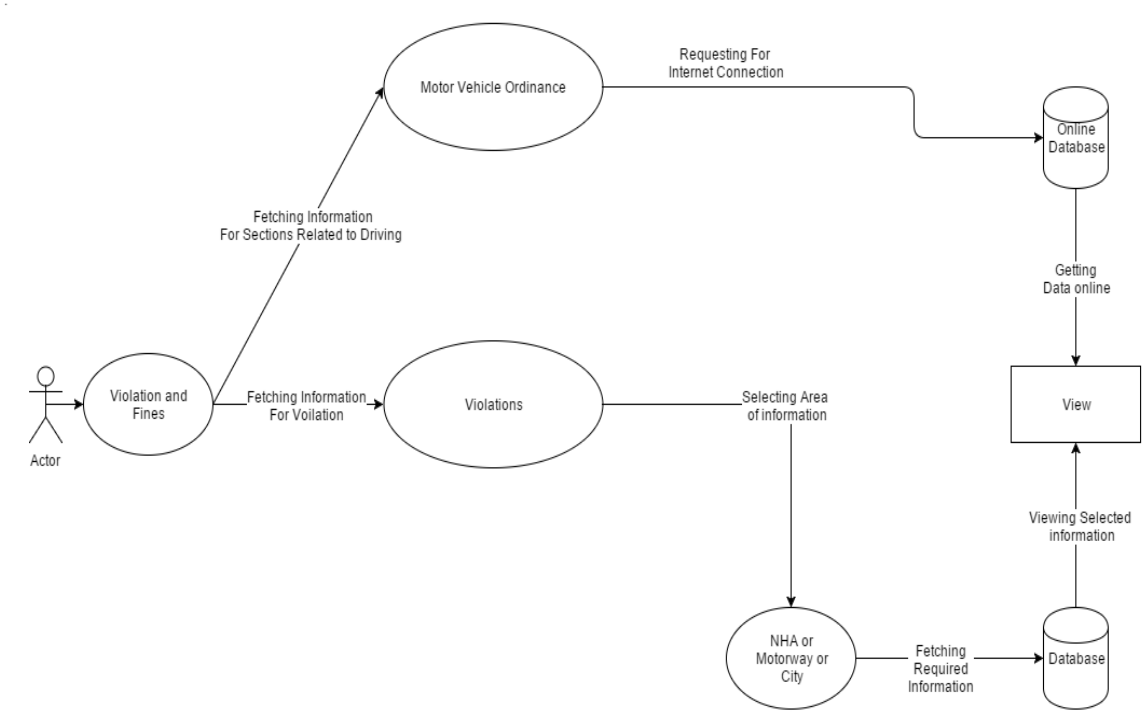
6.1 DATA FLOW DIAGRAM Level 1



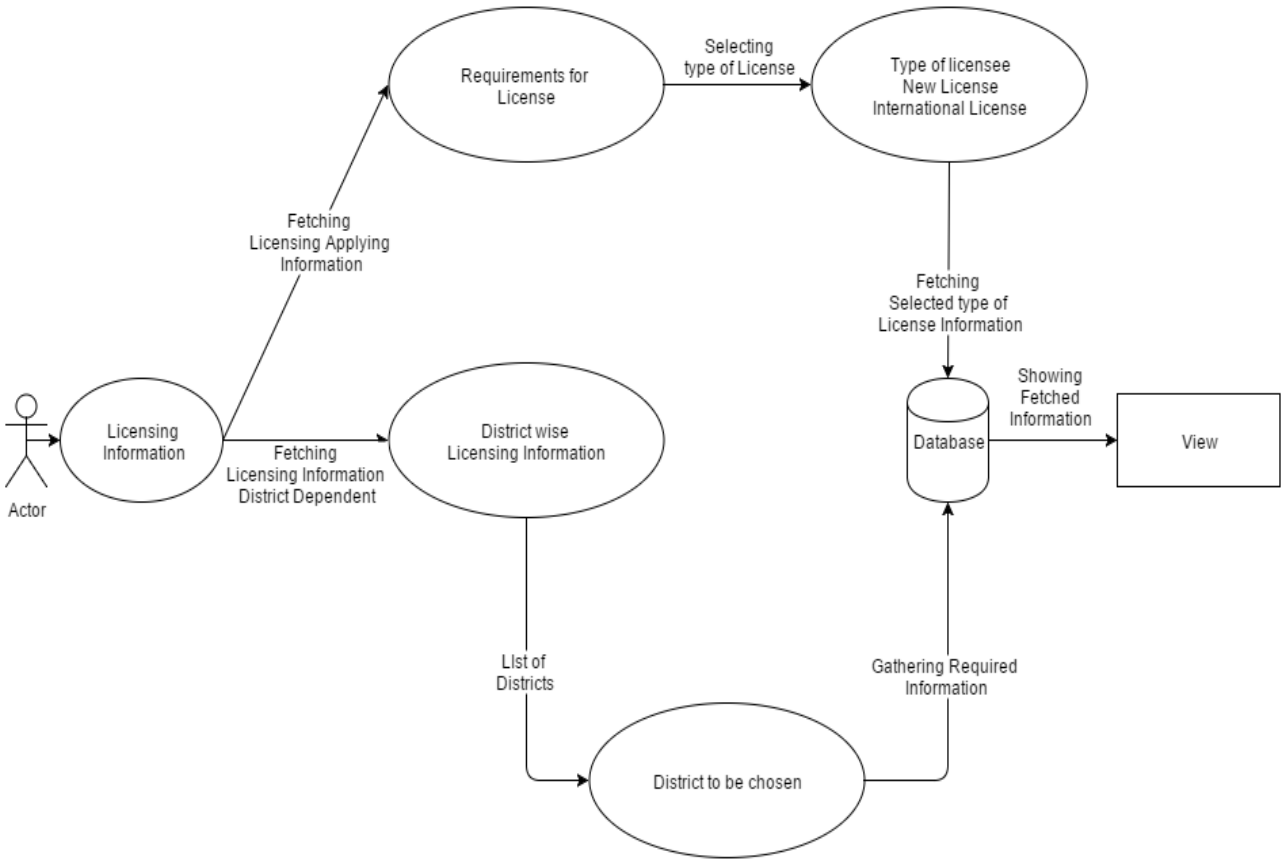
6.2 DATA FLOW DIAGRAM Level 2



6.3 DATA FLOW DIAGRAM Level 3

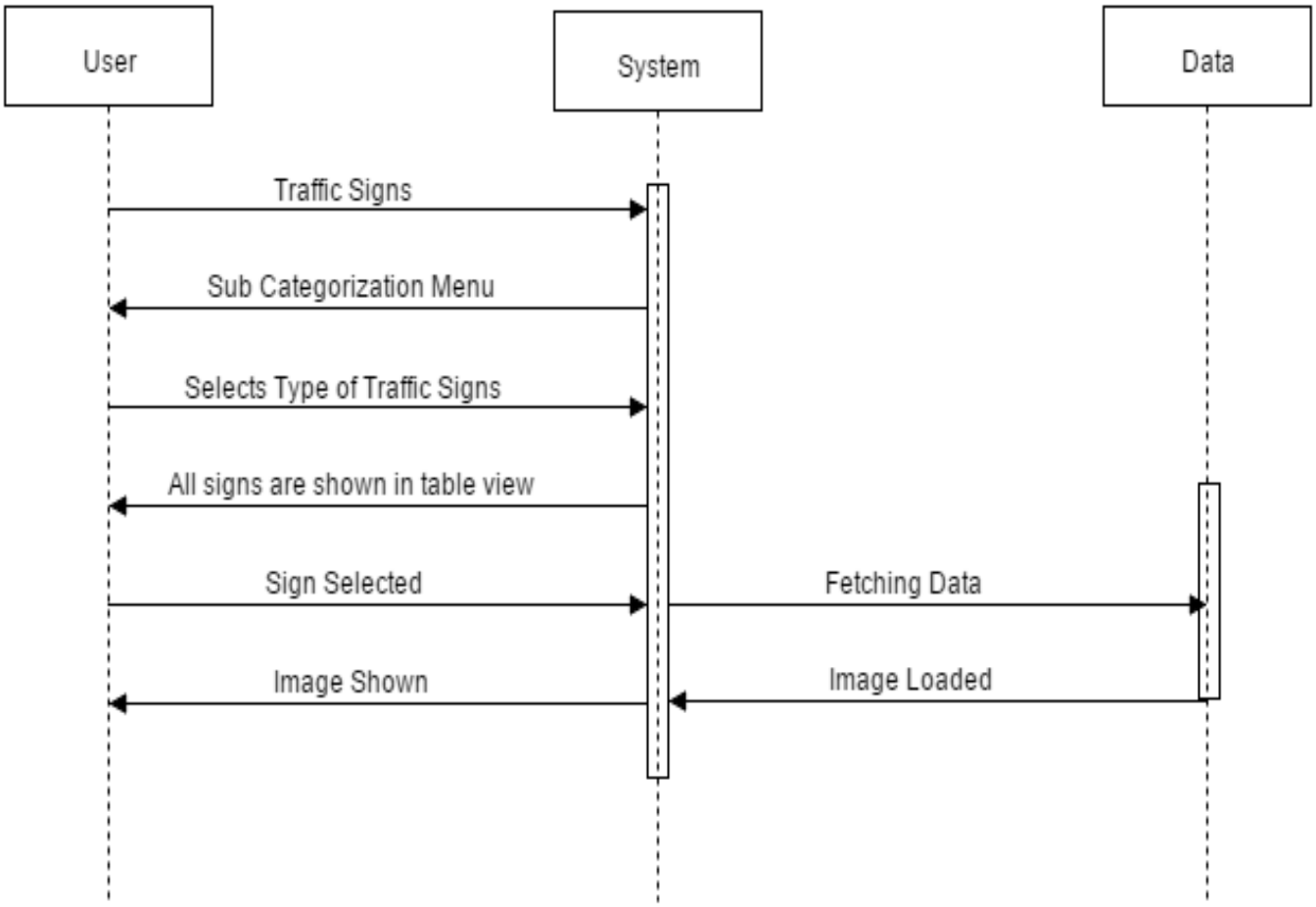


6.4 DATA FLOW DIAGRAM Level 4

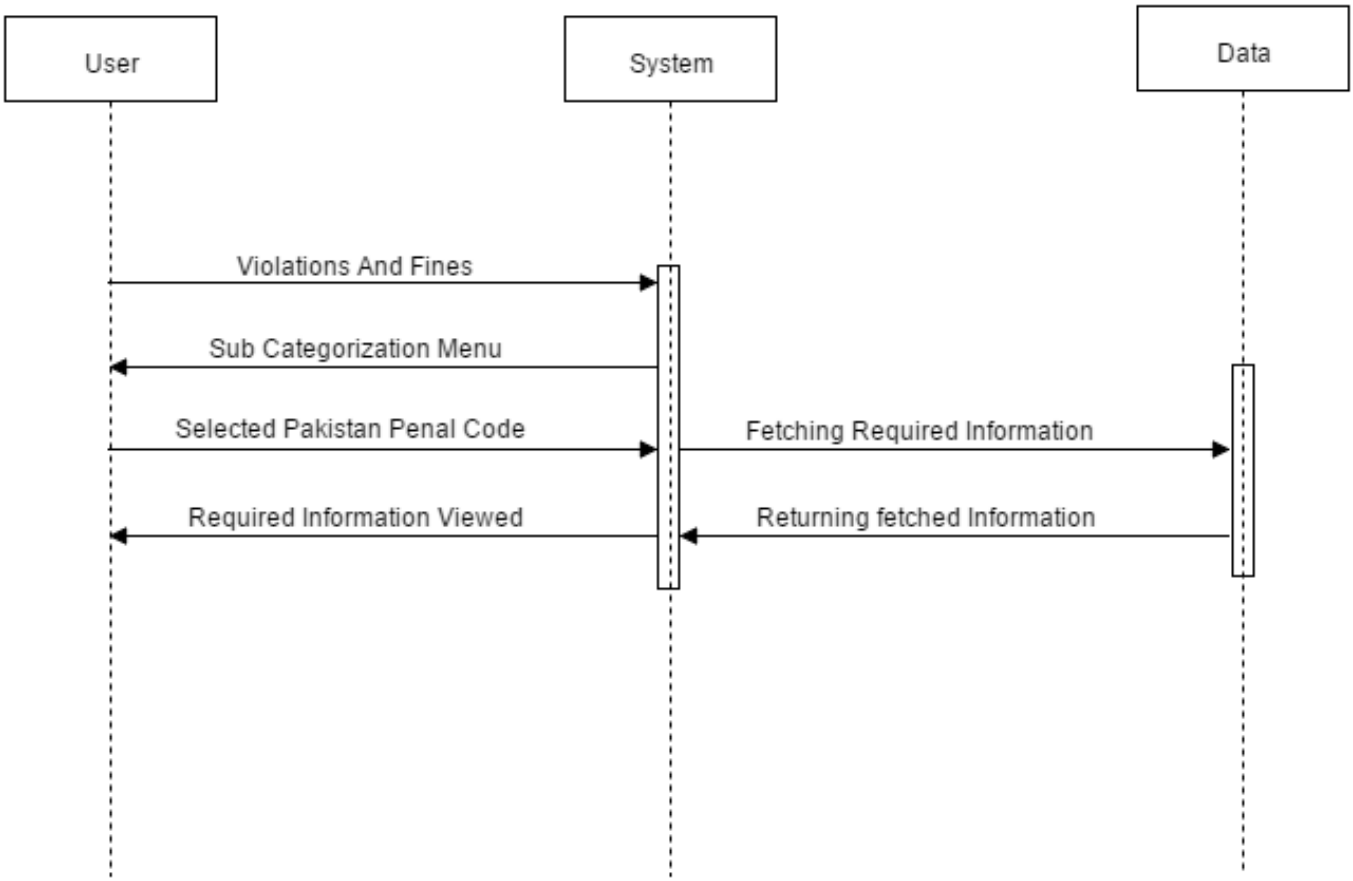


7 System Sequence Diagram

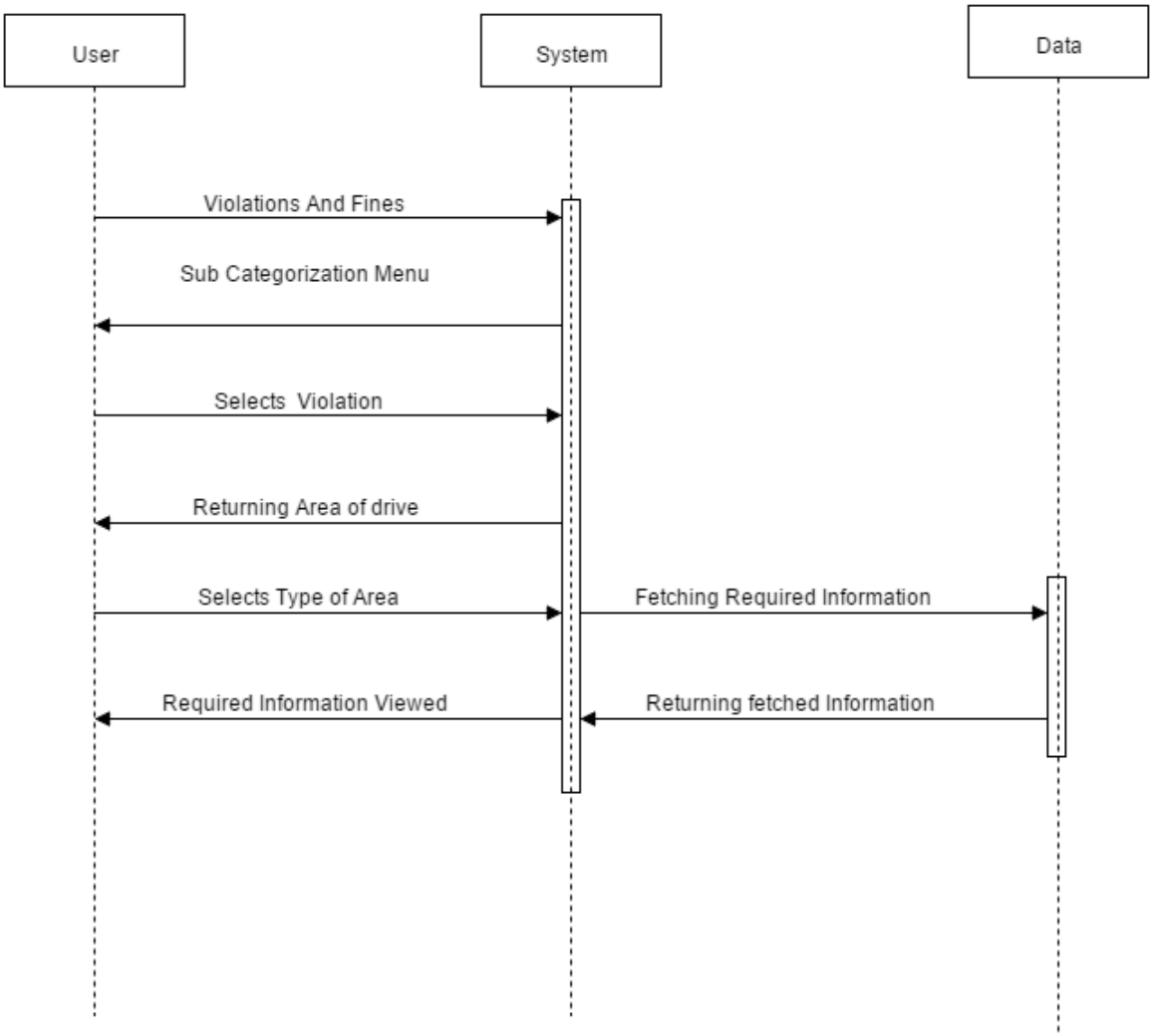
7.1 System Sequence Traffic Signs



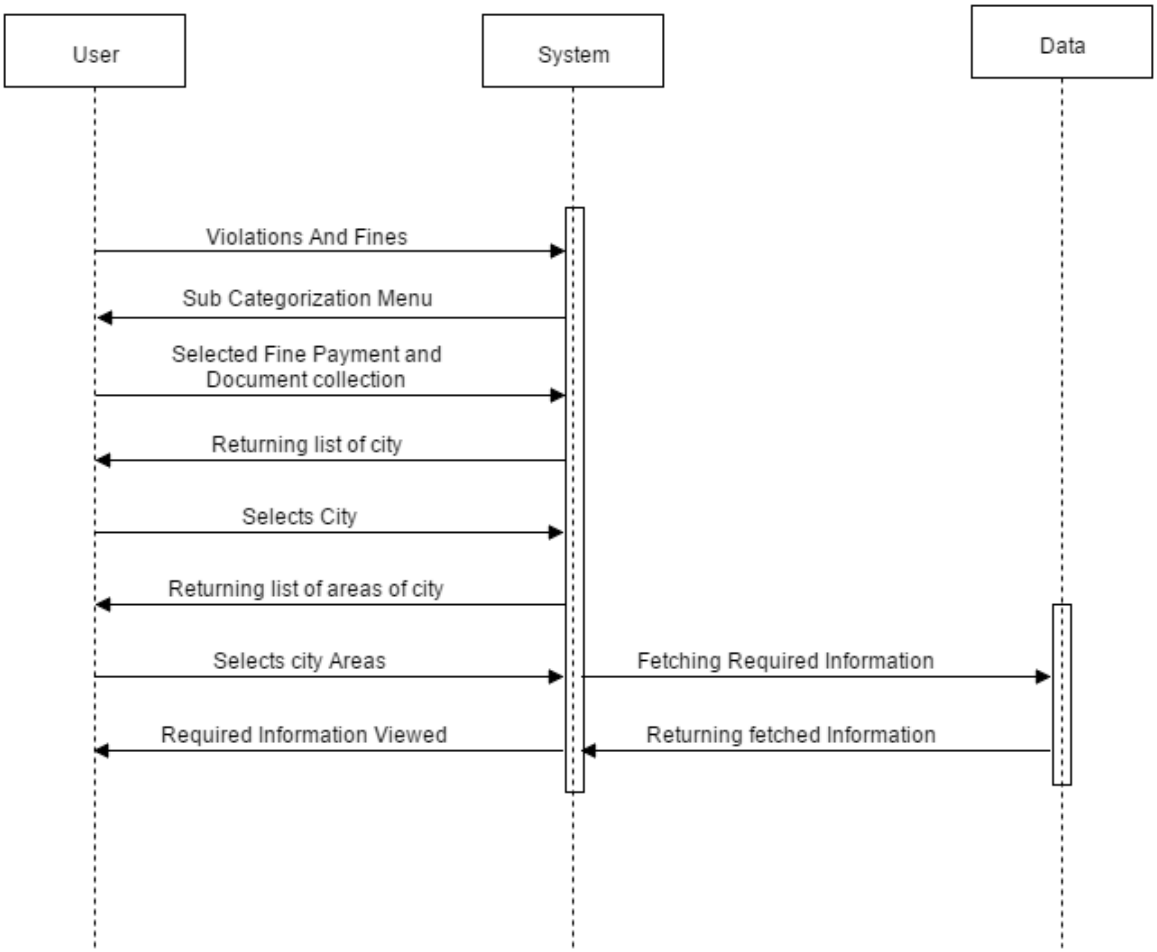
7.2.1 System Sequence Violations and Fine 1



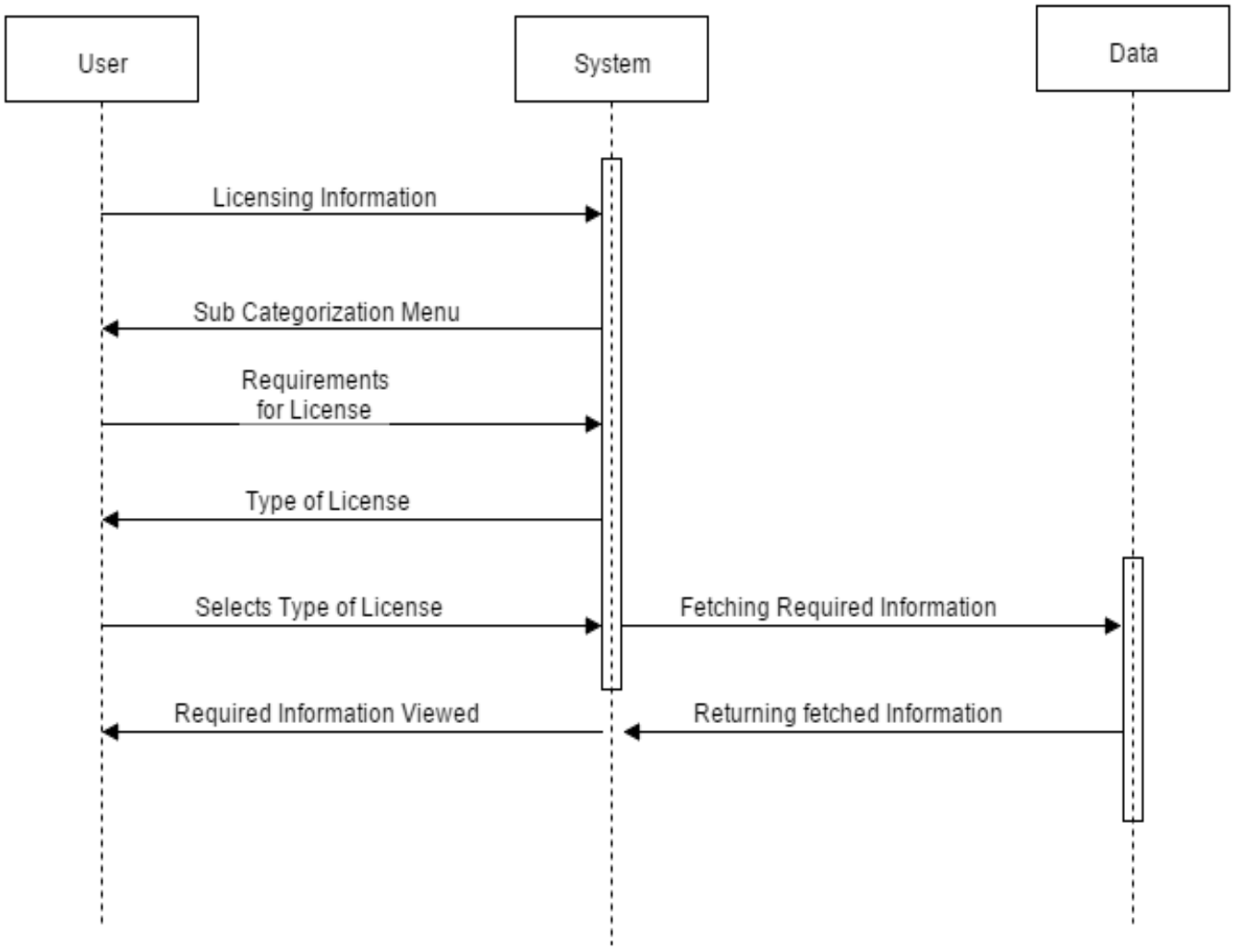
7.2.2 System Sequence Violations and Fine 2



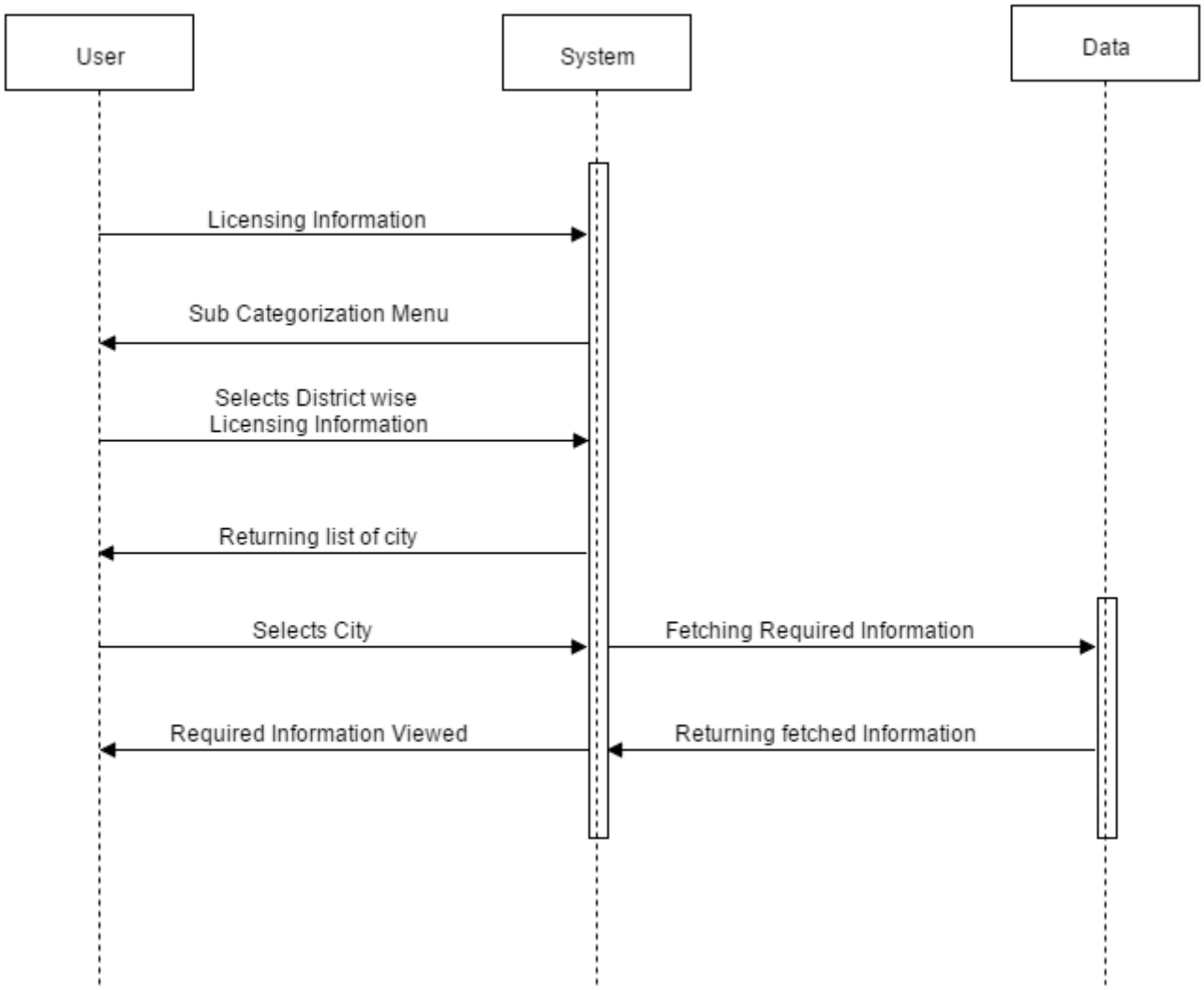
7.2.3 System Sequence Violations and Fine 3



7.3.1 System Sequence Licensing Information 1



7.3.2 System Sequence Licensing Information 2



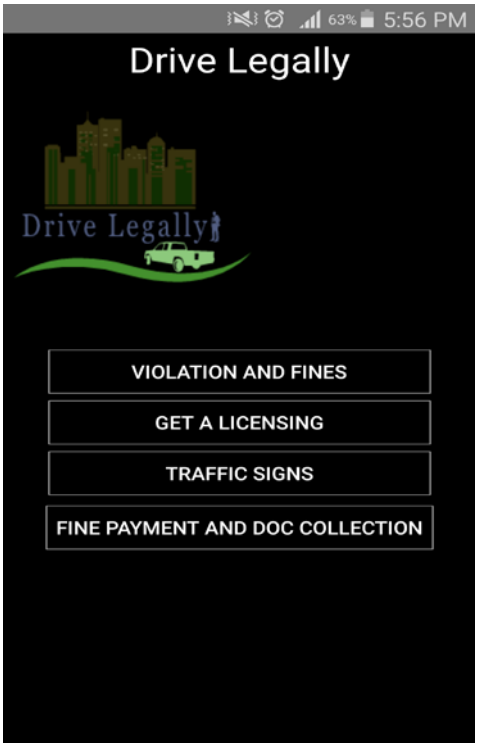
8. User Manual

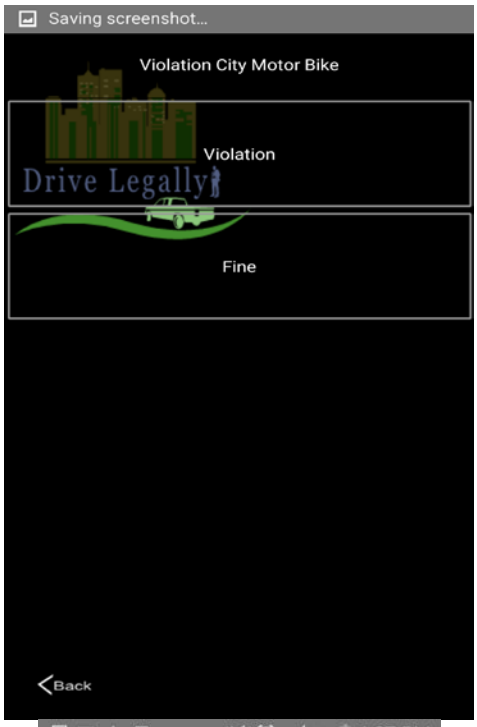
The user manual will discuss about the flow of the application used in the mobile phone for the sake of user so that application could be run successfully. This part of the documentation contains screen shot of the original application and the backend admin panel.

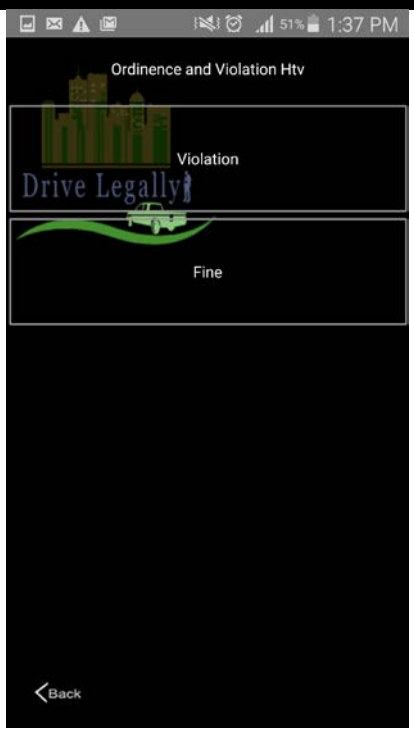
Hierarchy Of the mobile app screens:

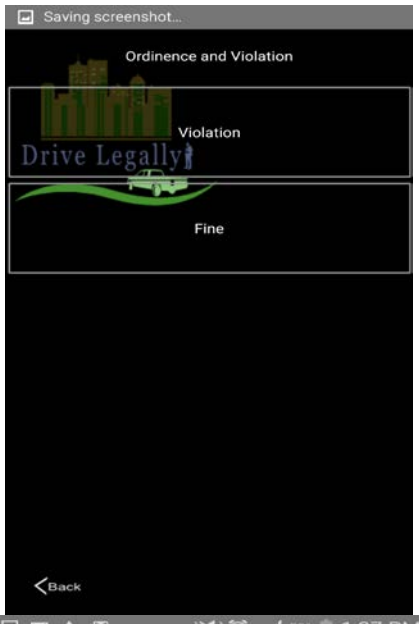
Home Screen:

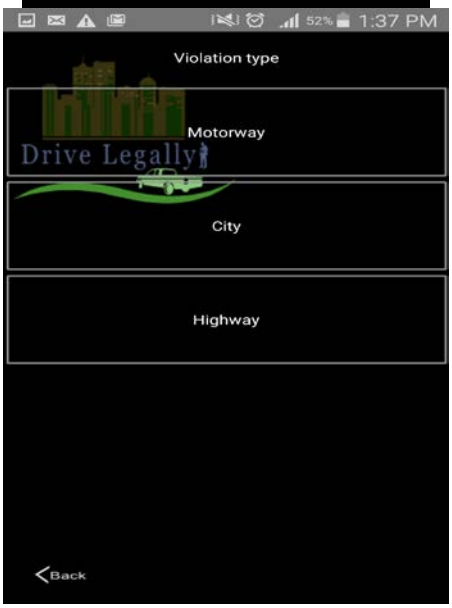
It contains the main screen of the project with four men buttons.

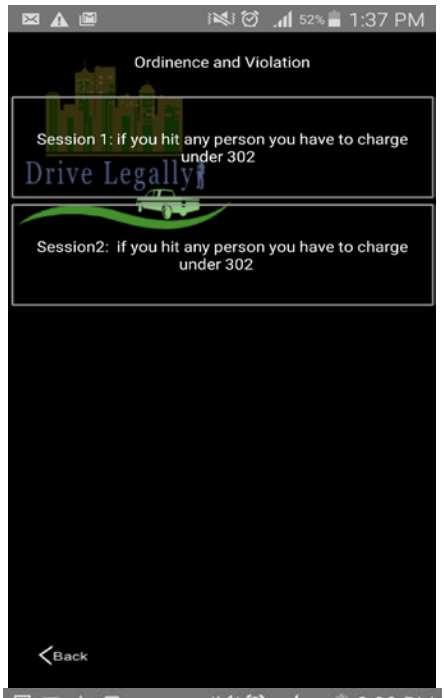


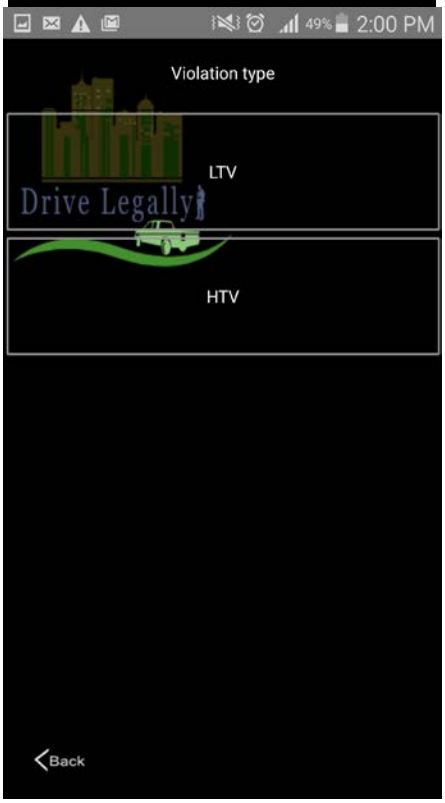


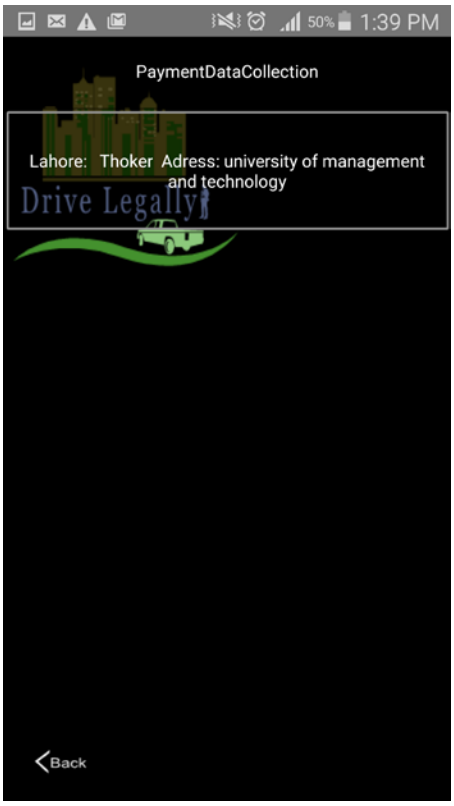


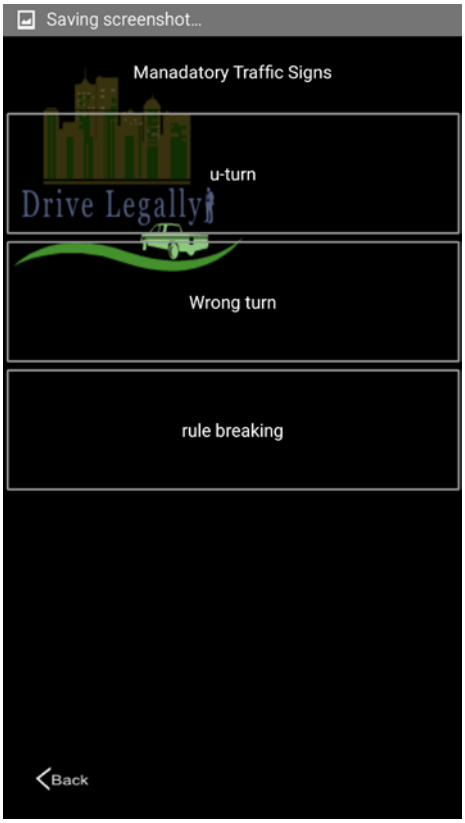


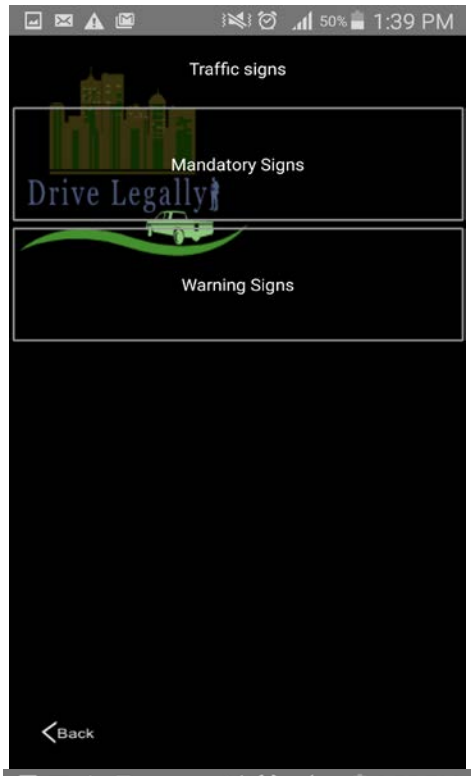


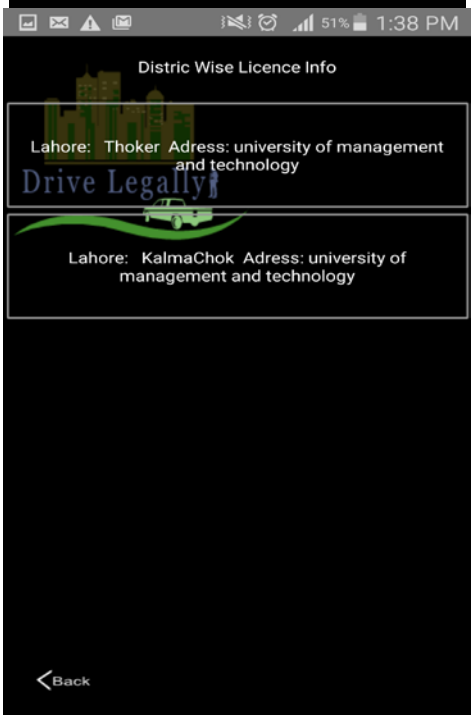


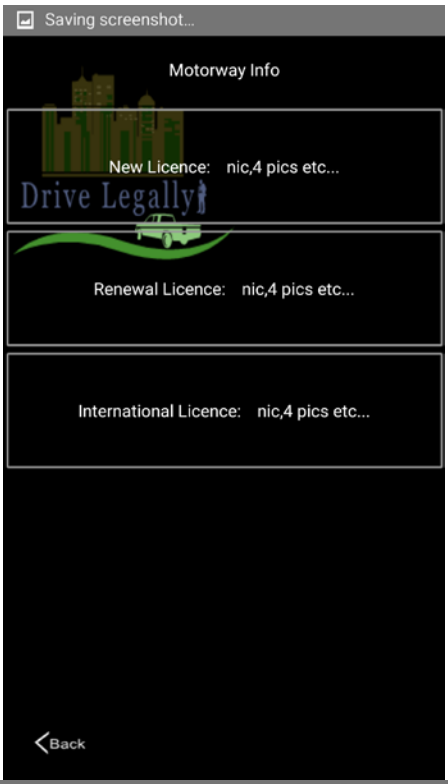


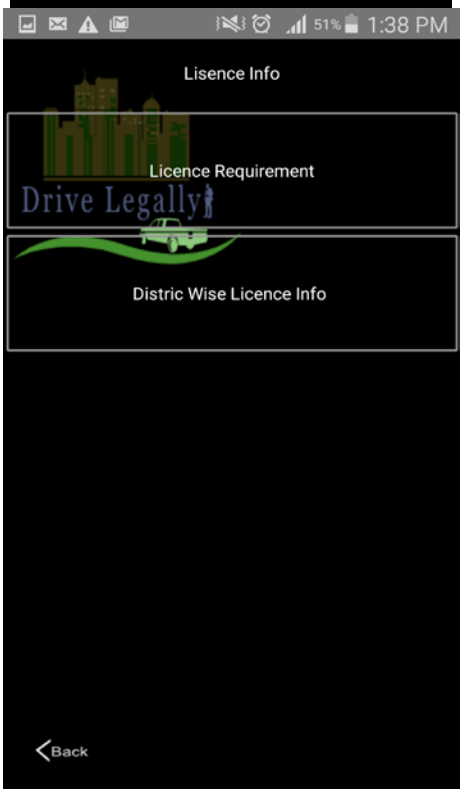


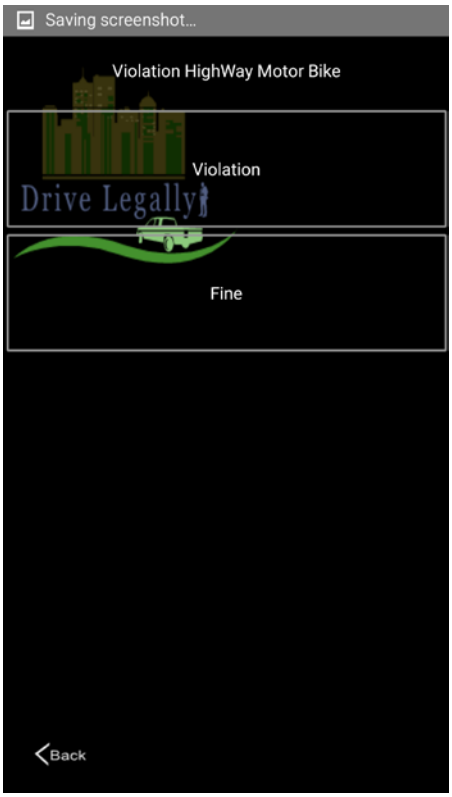


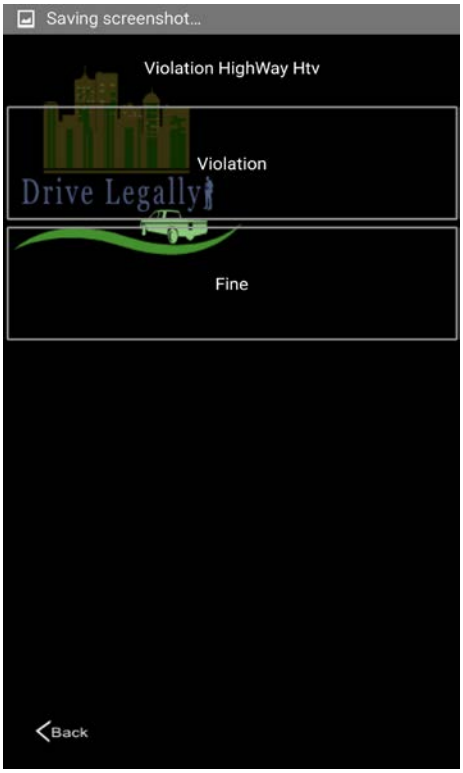


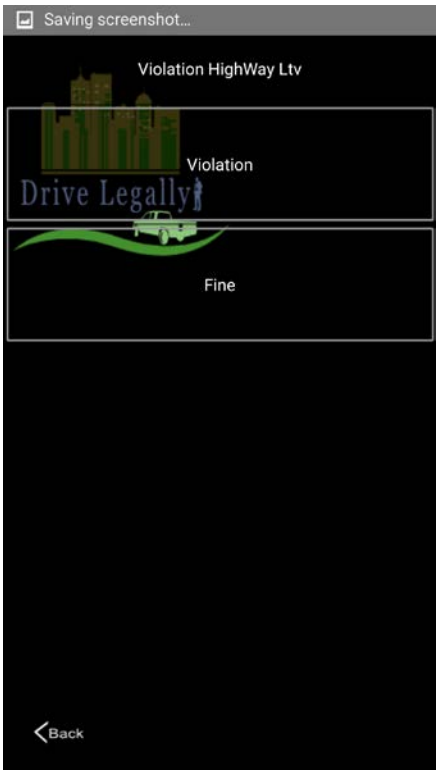


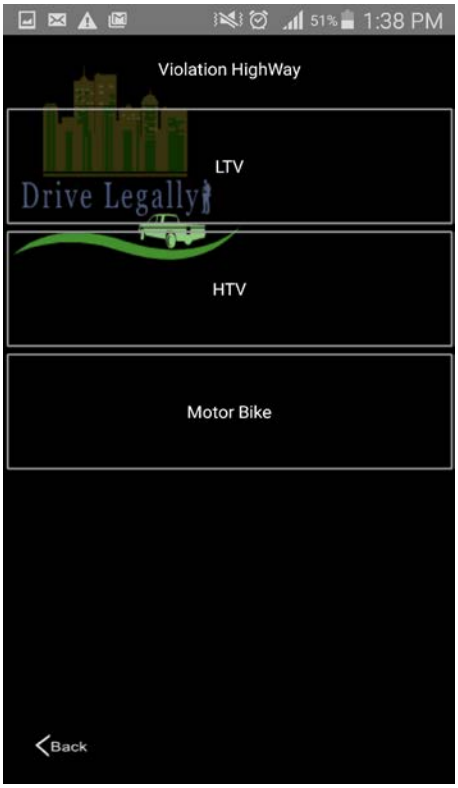








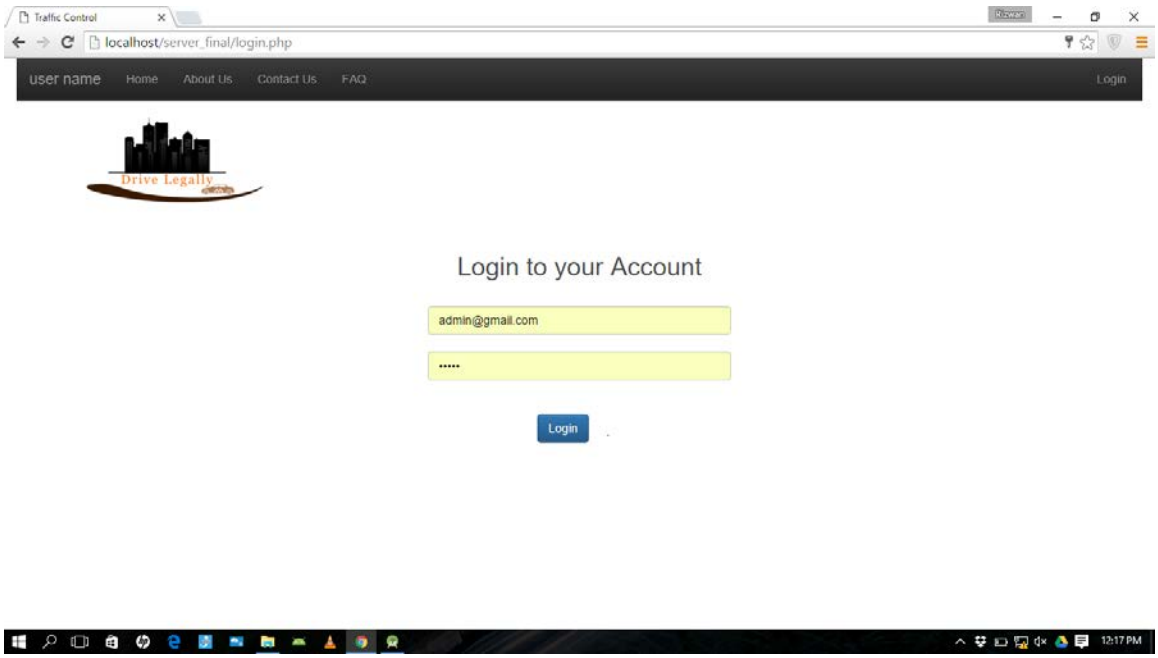




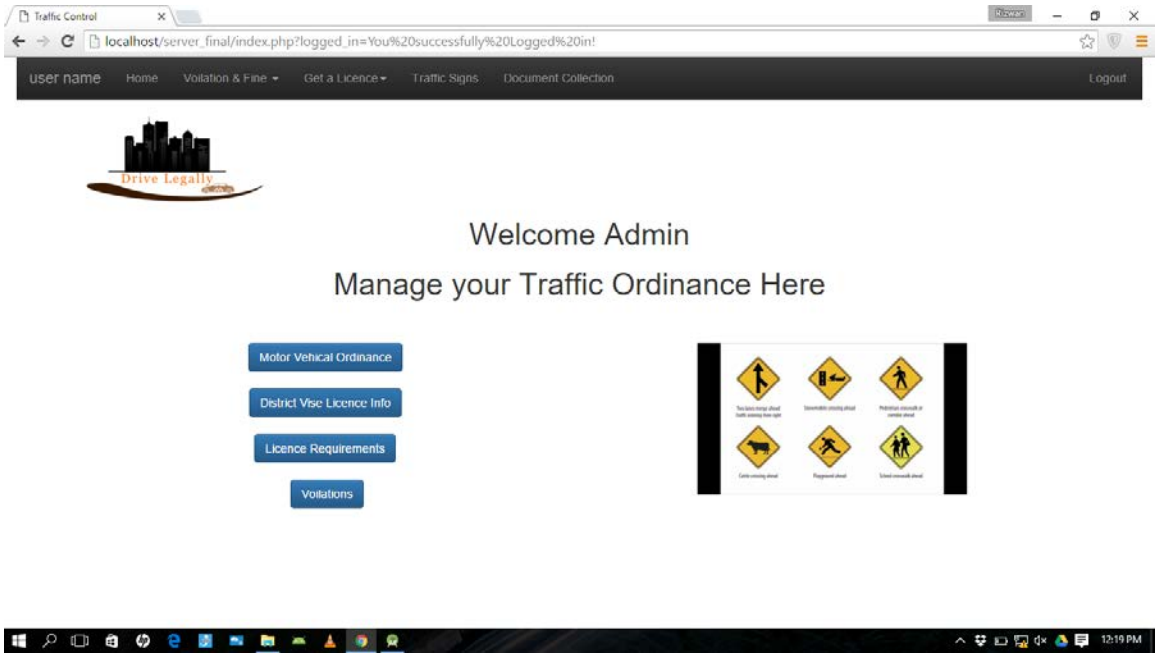
Admin Panel Screens:

This part of the documentation contain the admin panel screens of the application which is design in PHP and MySQL databases.

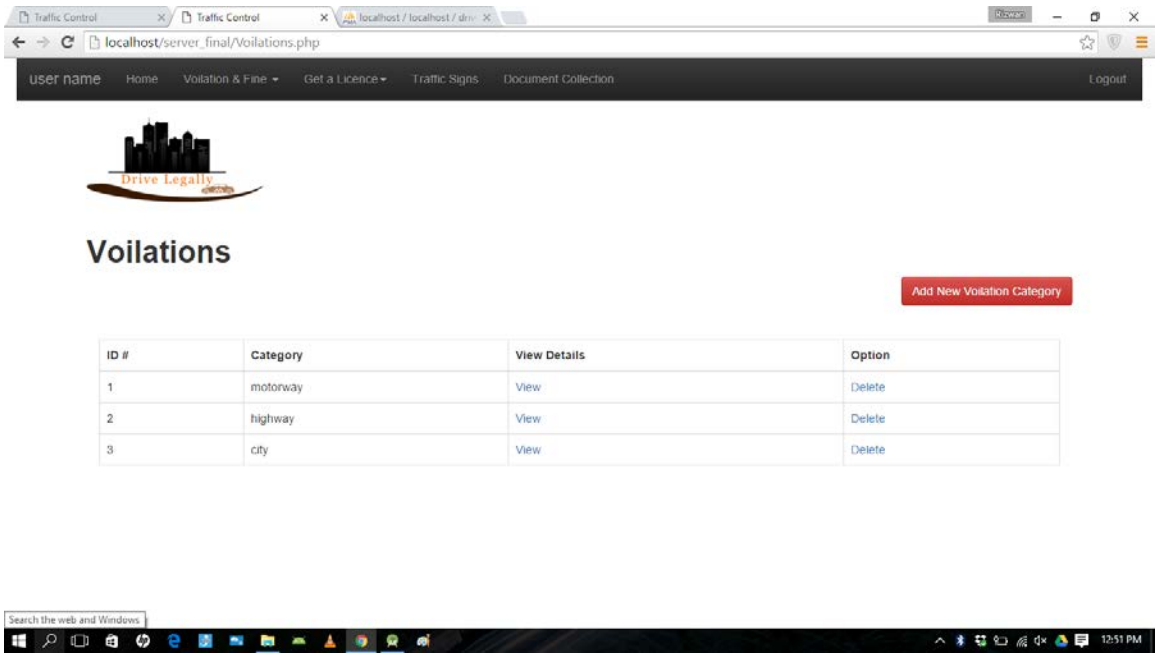
8.1 Login Screen



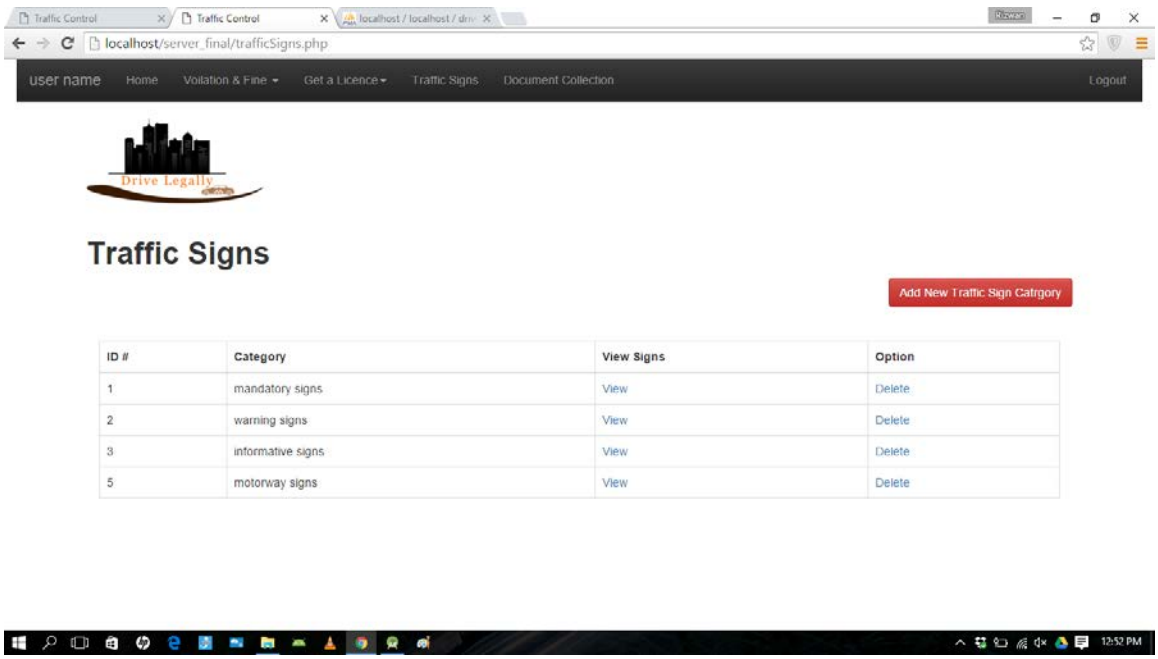
8.2 Home Screen



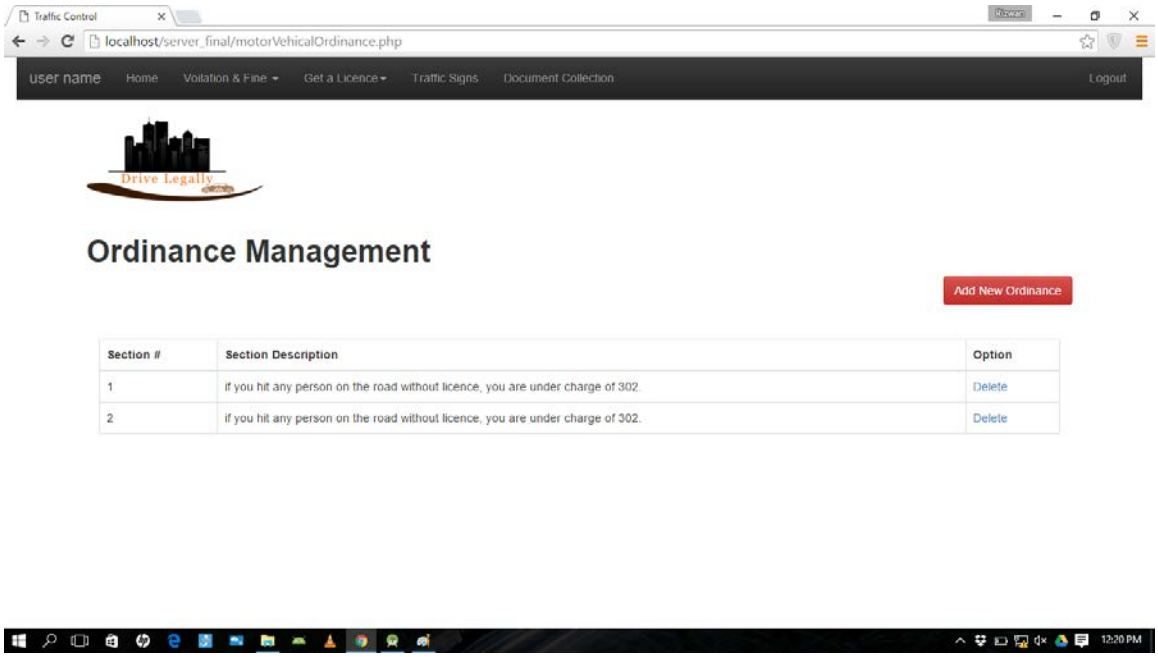
8.3 Violations Screen



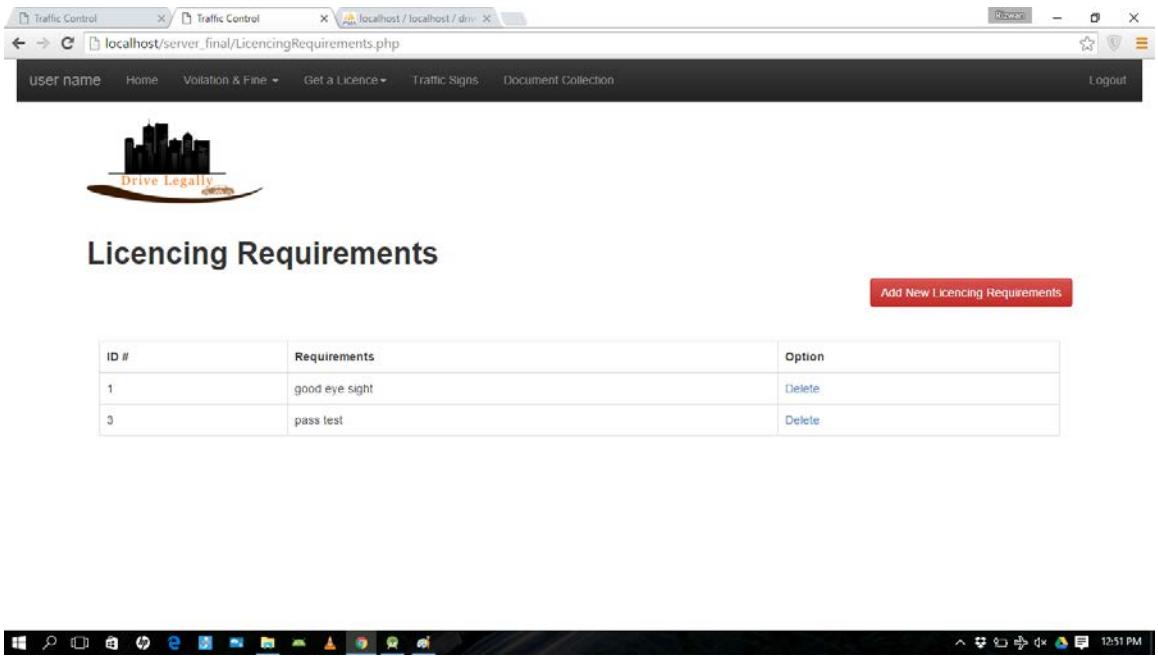
8.4 Traffic Signs Screen



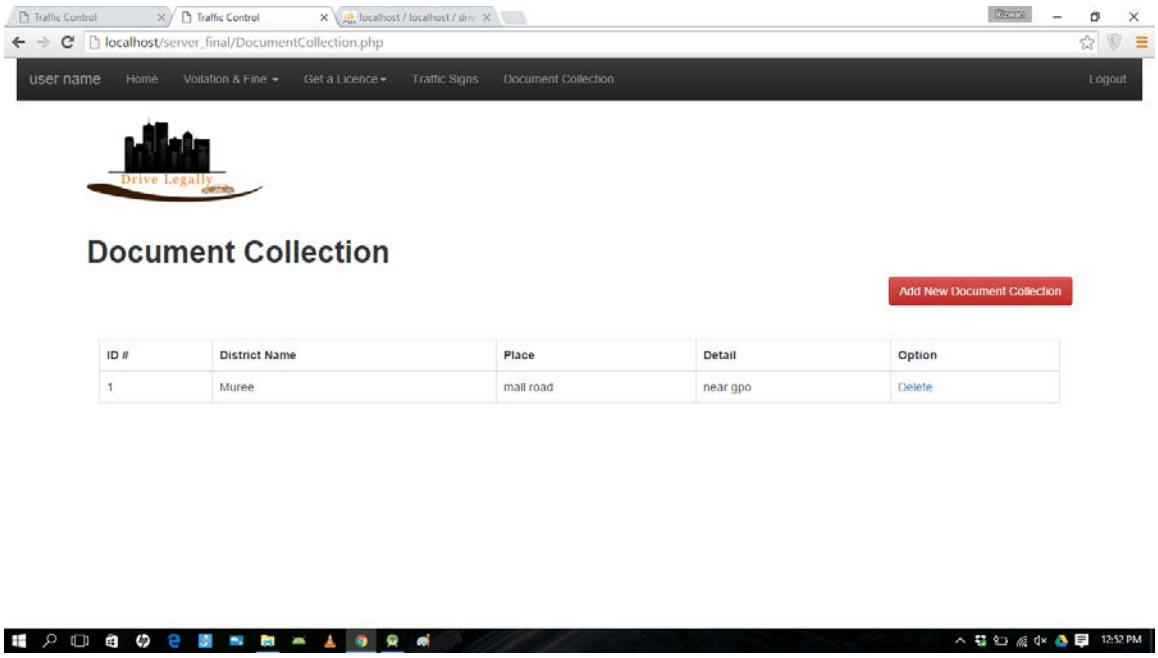
Ordinance Management Screen



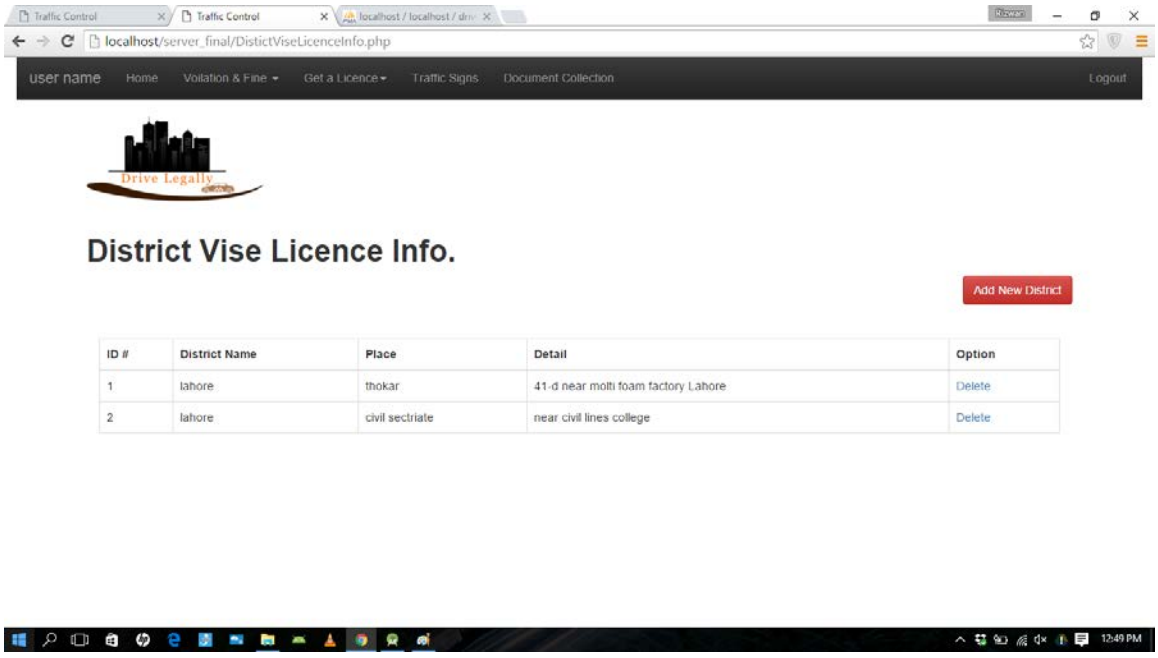
8.5 Licensing Management Screen



8.6 Document Collection Screen



8.7 District vise Licensing Info Screen



9. Code:

Android Code

Main Activity xml code

```
package easyshop.com.easyshopstore;

import java.sql.Connection;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Fragment;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.Toast;
import android.widget.ViewAnimator;

public class MainActivity extends Activity {
    public  boolean isInternetConnection()
    {
        boolean connected = false;
        ConnectivityManager connectivityManager = (ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
        if(connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
        if(connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() == NetworkInfo.State.CONNECTED ||
                connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() == NetworkInfo.State.CONNECTED) {
            //we are connected to a network
            return connected = true;
        }
        return  connected = false;
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        Button violation = null;
        Button lic = null;
        Button signs = null;
        Button finePayments = null;
        final AlertDialog alertDialog ;
        alertDialog = new AlertDialog.Builder(MainActivity.this).create();
        alertDialog.setTitle("Net checker");
        alertDialog.setMessage("pha gee net te chla lo :-) ");
        alertDialog.setButton("wadi meher bani", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getApplicationContext(), "You clicked on OK", Toast.LENGTH_SHORT).show();
            }
        });
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        violation = (Button)findViewById(R.id.violation);
        lic = (Button)findViewById(R.id.licence);
        signs = (Button)findViewById(R.id.traffic);
        finePayments = (Button)findViewById(R.id.finepayment);

        violation.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this, Violation.class);
            startActivity(intent);
            overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);


        }
    });
    lic.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this, Licence.class);
            startActivity(intent);
            overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);
        }
    });

    signs.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (isInternetConnection()) {
            Intent intent=new Intent(MainActivity.this,Traffic_Signs.class);
            startActivity(intent);
            overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);
            }
            else
                alertDialog.show();
        }});
```

```
    finePayments.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (isInternetConnection()) {
                Intent intent = new Intent(MainActivity.this, PaymentDataCollection.class);
                startActivity(intent);
                overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);
            }
            else
                alertDialog.show();
            }
            });


}
}
```

Send Data Class

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.widget.Toast;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;

import java.util.ArrayList;


public class SendDataTask extends AsyncTask<String, String, String> {


    Context context;
    String response;
    String userName;
    String password;
    float   balance;
    Activity activity;

    public SendDataTask(Context context, String userName, String password, Activity activity)
    {
        this.context = context;
        this.userName = userName;
        this.password = password;
        this.activity = activity;
    }
```

```
@Override
protected String doInBackground(String... params) {



    ArrayList<NameValuePair> postParameters = new ArrayList<~>();
    postParameters.add(new BasicNameValuePair("user", userName));
    postParameters.add(new BasicNameValuePair("password", password));

    try
    {

        response = CustomHttpClient.executeHttpPost("http://192.168.1.106/myjob/fatch.php", postParameters);


    }
    catch (Exception e) {

        e.printStackTrace();
    }


    return null;

}
```

```
    @Override
    protected void onPostExecute(String result) {

        if (response.equals("not found\n")) {

            Toast.makeText(context, "Not found", Toast.LENGTH_SHORT).show();
        }
        else
        {
           // Toast.makeText(context, response, Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(context,MainMenu.class);
            intent.putExtra("balance",balance);
            activity.startActivity(intent);
        }
    }


}
```

Main Menu xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:background="#ffffff"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="fill_parent"
        android:orientation="horizontal"
        android:padding="5dp"
        android:layout_margin="5dp"
        android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:text="welcome"
        android:textColor="#000000"
        android:layout_marginTop="10dp"
        android:textSize="22sp"
        android:layout_height="wrap_content" />

     <LinearLayout
         android:layout_width="fill_parent"
         android:gravity="right"
         android:layout_height="wrap_content">

            <ImageView
                android:layout_width="50dp"
                android:src="@drawable/buy"
```

```
            android:layout_height="50dp" />
        <TextView
            android:layout_width="wrap_content"
            android:text="1000"
            android:textColor="#000000"
            android:layout_marginTop="15dp"
            android:textSize="22sp"
            android:layout_height="wrap_content" />


</LinearLayout>


</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

<ImageView
    android:layout_width="300dp"
    android:src="@drawable/logo"
    android:layout_height="400dp" />

 <LinearLayout
     android:layout_width="wrap_content"
     android:orientation="vertical"
     android:gravity="center"
     android:layout_margin="5dp"
```

```
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this, Violation.class);
            startActivity(intent);
            overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);


        }
    });
    lic.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this, Licence.class);
            startActivity(intent);
            overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);
        }
    });

    signs.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (isInternetConnection()) {
            Intent intent=new Intent(MainActivity.this,Traffic_Signs.class);
            startActivity(intent);
            overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);
            }
            else
                alertDialog.show();
        }});
```

Image View Activity xml code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:background="@drawable/bck"
    android:gravity="center"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="60dp">
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:text="Image View Activity"
            android:gravity="center"
            android:textColor="@color/white"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:gravity="center"
        android:layout_height="300dp">

        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:background="@drawable/trafficlightssign"
            android:id="@+id/imageView" />
```

```
    </LinearLayout>

    <RelativeLayout
        android:layout_width="fill_parent"
        android:gravity="bottom"
        android:layout_height="60dp">
        <Button
            android:layout_width="60dp"
            android:layout_marginLeft="20dp"
            android:layout_marginBottom="10dp"
            android:layout_height="50dp"
            android:id="@+id/licence_back"
            android:background="@drawable/errow"
            android:layout_alignParentBottom="true" />
    </RelativeLayout>


</LinearLayout>
```

```
  </LinearLayout>

  <LinearLayout
      android:layout_width="fill_parent"
      android:orientation="horizontal"
      android:gravity="center"
      android:layout_height="wrap_content">

      <Button
          android:layout_width="wrap_content"
          android:text="left"
          android:id="@+id/left"
          android:layout_height="wrap_content" />


      <Button
          android:layout_width="wrap_content"
          android:text="right"
          android:id="@+id/right"
          android:layout_marginLeft="100dp"
          android:layout_height="wrap_content" />
```

Violation xml code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:background="@drawable/bck"
    android:gravity="center"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="60dp">
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:text="Violation"
            android:textSize="14dp"
            android:gravity="center"
            android:textColor="@color/white"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:gravity="center"
        android:layout_height="500dp">

        <ListView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:id="@+id/listViewordinence" />

    </LinearLayout>
    <RelativeLayout
        android:layout_width="fill_parent"
        android:gravity="bottom"
        android:layout_height="60dp">
        <Button
            android:layout_width="60dp"
            android:layout_marginLeft="20dp"
            android:layout_marginBottom="10dp"
            android:layout_height="50dp"
            android:id="@+id/violation_back"
            android:background="@drawable/errow"
            android:layout_alignParentBottom="true" />
    </RelativeLayout>
</LinearLayout>
```

Custom Client Class

```
package easyshop.com.easyshopstore;

import ...

public class CustomHttpClient {
    /** The time it takes for our client to timeout */
    public static final int HTTP_TIMEOUT = 30 * 1000; // milliseconds

    /** Single instance of our HttpClient */
    private static HttpClient mHttpClient;

    /**
     * Get our single instance of our HttpClient object.
     *
     * @return an HttpClient object with connection parameters set
     */
    private static HttpClient getHttpClient() {
        if (mHttpClient == null) {
            mHttpClient = new DefaultHttpClient();
            final HttpParams params = mHttpClient.getParams();
            HttpConnectionParams.setConnectionTimeout(params, HTTP_TIMEOUT);
            HttpConnectionParams.setSoTimeout(params, HTTP_TIMEOUT);
            ConnManagerParams.setTimeout(params, HTTP_TIMEOUT);
        }
        return mHttpClient;
    }

    /**
     * Performs an HTTP Post request to the specified url with the
     * specified parameters.
     *
     * @param url The web address to post the request to
```

```
 */
public static String executeHttpPost(String url, ArrayList<NameValuePair> postParameters) throws Exception {
    BufferedReader in = null;
    try {
        HttpClient client = getHttpClient();
        HttpPost request = new HttpPost(url);
        UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(postParameters);
        request.setEntity(formEntity);
        HttpResponse response = client.execute(request);
        in = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));

        StringBuffer sb = new StringBuffer("");
        String line = "";
        String NL = System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();

        String result = sb.toString();
        return result;
    } finally {
        if (in != null) {
            try {
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
public static String executeHttpGet(String url) throws Exception {
    BufferedReader in = null;
    try {
        HttpClient client = getHttpClient();
        HttpGet request = new HttpGet();
        request.setURI(new URI(url));
        HttpResponse response = client.execute(request);
        in = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));

        StringBuffer sb = new StringBuffer("");
        String line = "";
        String NL = System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();

        String result = sb.toString();
        return result;
    } finally {
        if (in != null) {
            try {
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Violation Class

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

/**
 * Created by HP 8460p on 2/8/2016.
 */
public class DistricWiseLicence extends Activity {

    //scriptdistricwiselic
    ListView lv1;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        Button back = null;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.districwiselicence);
        back = (Button)findViewById(R.id.distric_licence_back);


       // String lv_arr[]={"Lahore:   Thoker  Adress: university of management and technology","Lahore:   KalmaChok  Adress: university
        lv1=(ListView)findViewById(R.id.listViewdistricwiselicence);
        //lv1.setAdapter(new ArrayAdapter<String>(this, R.layout.textview, lv_arr));

        SendDataTask sendDataTask = new SendDataTask(getApplicationContext(),"","",this);
        sendDataTask.execute();



        lv1.setOnItemClickListener((parent, view, position, id) -> {

                switch (position) {

            }
        });

        back.setOnClickListener((v) -> {

                Intent intent = new Intent(getApplicationContext(), Licence.class);
                startActivity(intent);
                overridePendingTransition(R.anim.card_flip_left_in, R.anim.card_flip_left_out);


        });
```

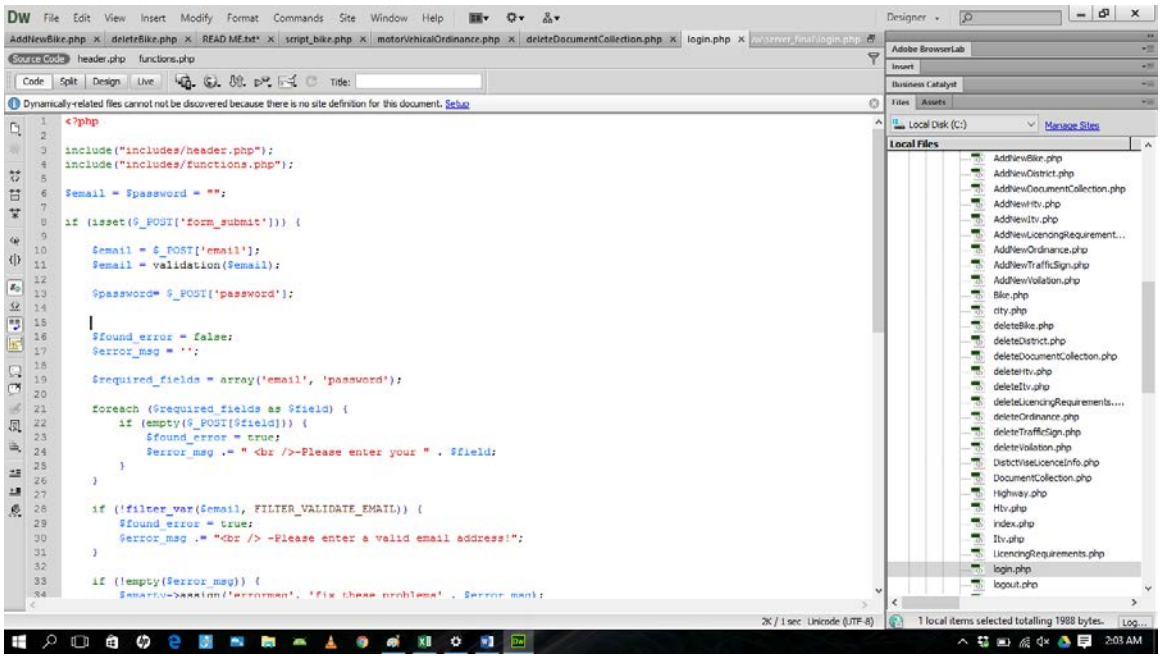
```
public class SendDataTask extends AsyncTask<String, String, String> {


    Context context;
    String response;
    String userName;
    String password;
    float   balance;
    Activity activity;

    public SendDataTask(Context context, String userName, String password, Activity activity)
    {
        this.context = context;
        this.userName = userName;
        this.password = password;
        this.activity = activity;
    }


    @Override
    protected String doInBackground(String... params) {



        ArrayList<NameValuePair> postParameters = new ArrayList<~>();
        postParameters.add(new BasicNameValuePair("user", userName));
        postParameters.add(new BasicNameValuePair("password", password));
```

```
        try
        {

            String str = "192.168.1.10";
            response = CustomHttpClient.executeHttpPost("http://"+str+"/server_final/script_dist_licence_req.php", postParameters);


        }
        catch (Exception e) {

            e.printStackTrace();
        }


        return null;

    }

    @Override
    protected void onPostExecute(String result) {


        if (!response.equals("")) {

            ArrayList<String> lv_arr = new ArrayList<~>();


            try
            {
                JSONArray jsonArray = new JSONArray(response);
```

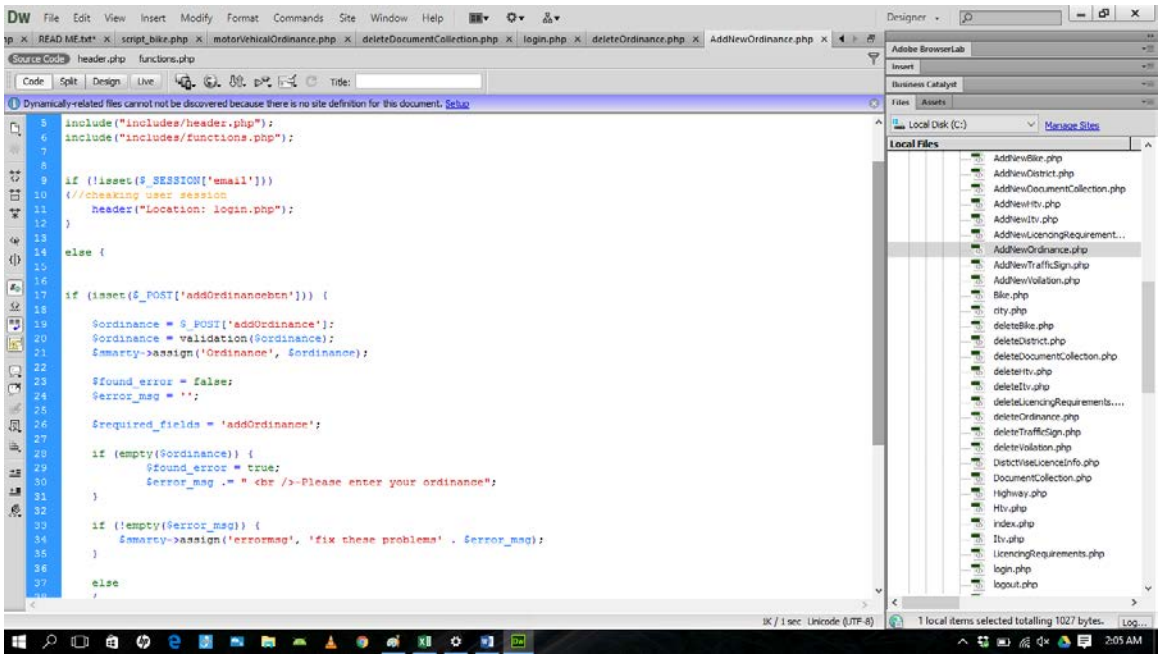
```
                for(int x=0;x<jsonArray.length();x++)
                {
                    JSONObject object = new JSONObject(jsonArray.getString(x));
                    String str = object.getString("district_name");
                    String str1 = object.getString("place");
                    String str2 = object.getString("detail");


                    lv_arr.add("district_name :  "+str+"          place : "+str1+"          detail: "+str2);

                    Log.d("", "");
                }


            }
            catch (JSONException e) {
                e.printStackTrace();
            }


            lv1.setAdapter(new ArrayAdapter<String>(context,R.layout.textview , lv_arr));

        }
    }


    }
}
```

10. Admin Panel code screen shorts

Login

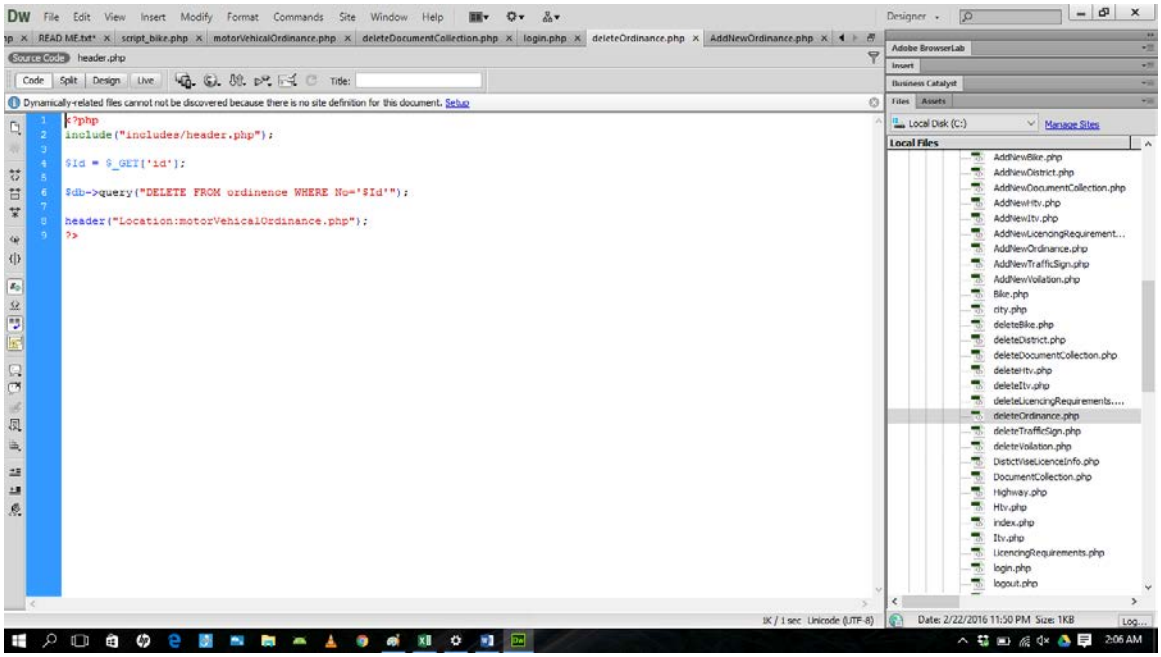


Add ordinance



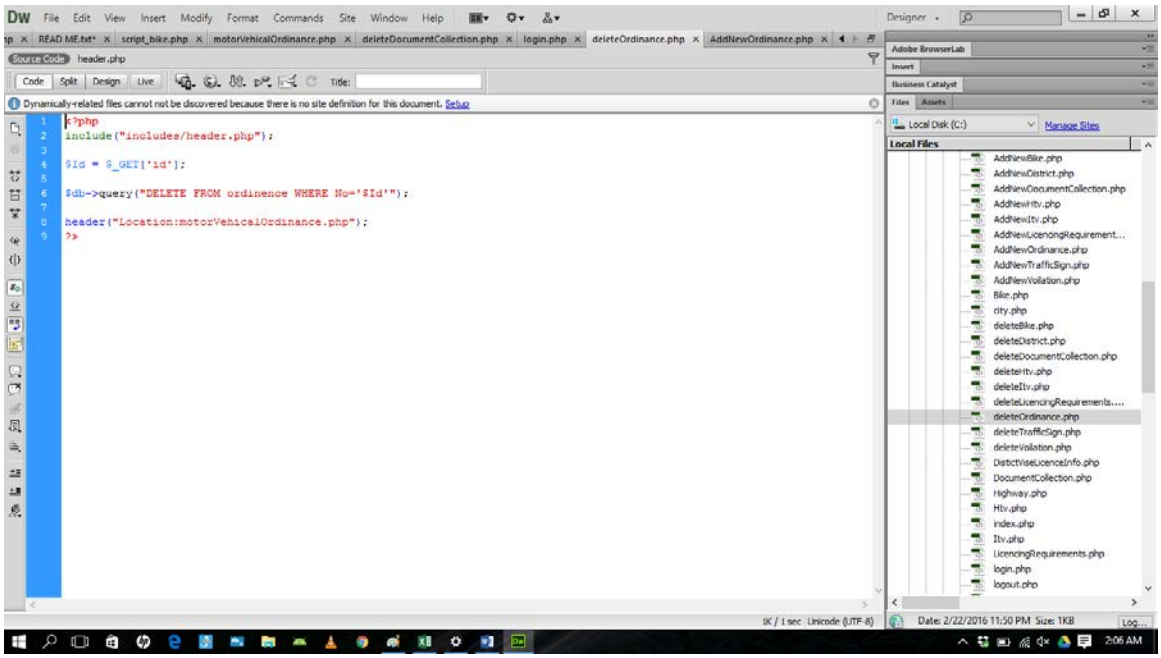
```
include("includes/header.php");
include("includes/functions.php");


if (!isset($_SESSION['email']))
//cheaking user session
    header("Location: login.php");
}

else {


if (isset($_POST['addOrdinancebtn'])) {

    $ordinance = $_POST['addOrdinance'];
    $ordinance = validation($ordinance);
    $smarty->assign('Ordinance', $ordinance);

    $found_error = false;
    $error_msg = '';

    $required_fields = 'addOrdinance';

    if (empty($ordinance)) {
            $found_error = true;
            $error_msg .= " <br />-Please enter your ordinance";
    }

    if (!empty($error_msg)) {
        $smarty->assign('errormsg', 'fix these problems' . $error_msg);
    }

    else
```

Delete Ordinance

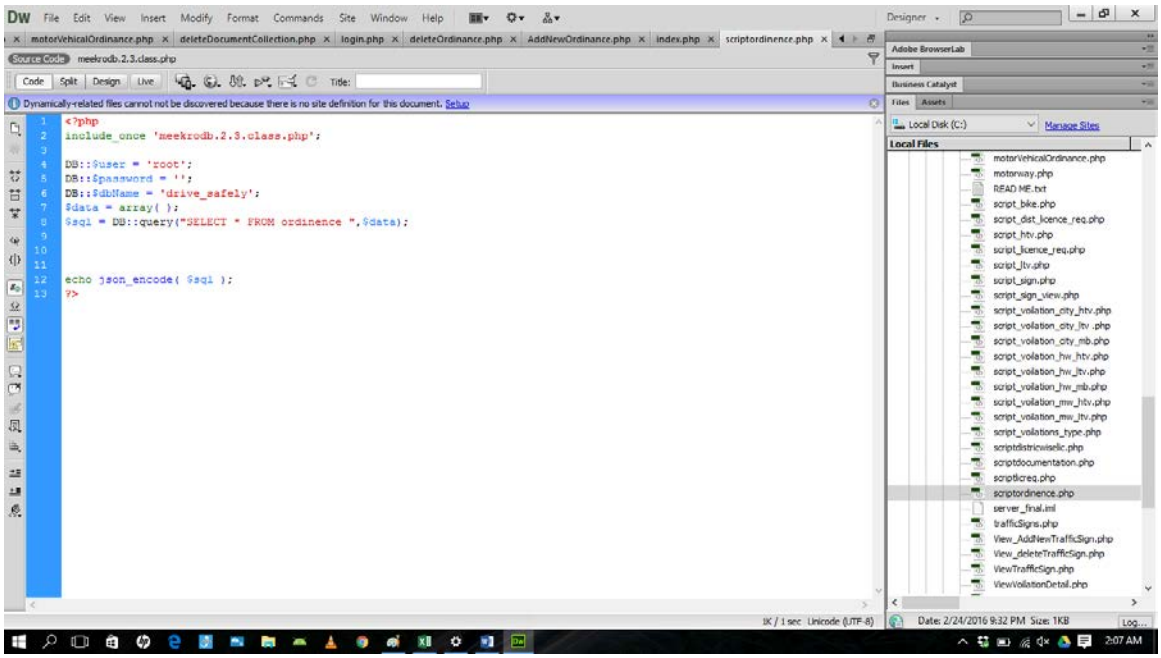


```
<?php
include("includes/header.php");

$Id = $_GET['id'];

$db->query("DELETE FROM ordinence WHERE No='$Id'");

header("Location:motorVehicalOrdinance.php");
?>
```

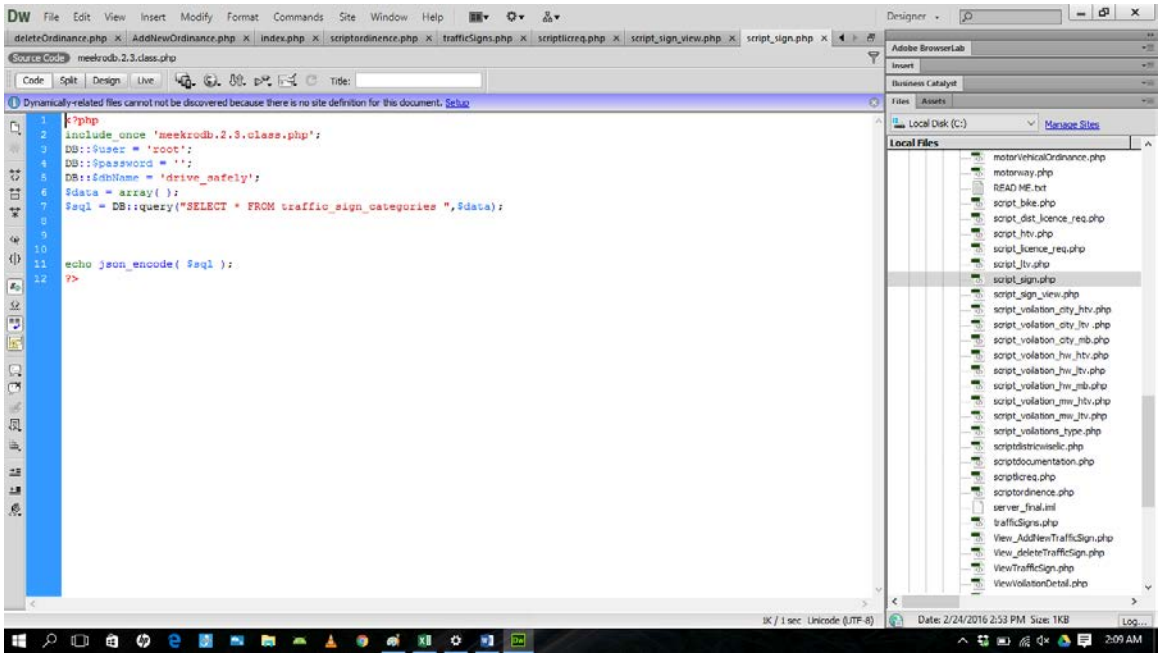
11. Index



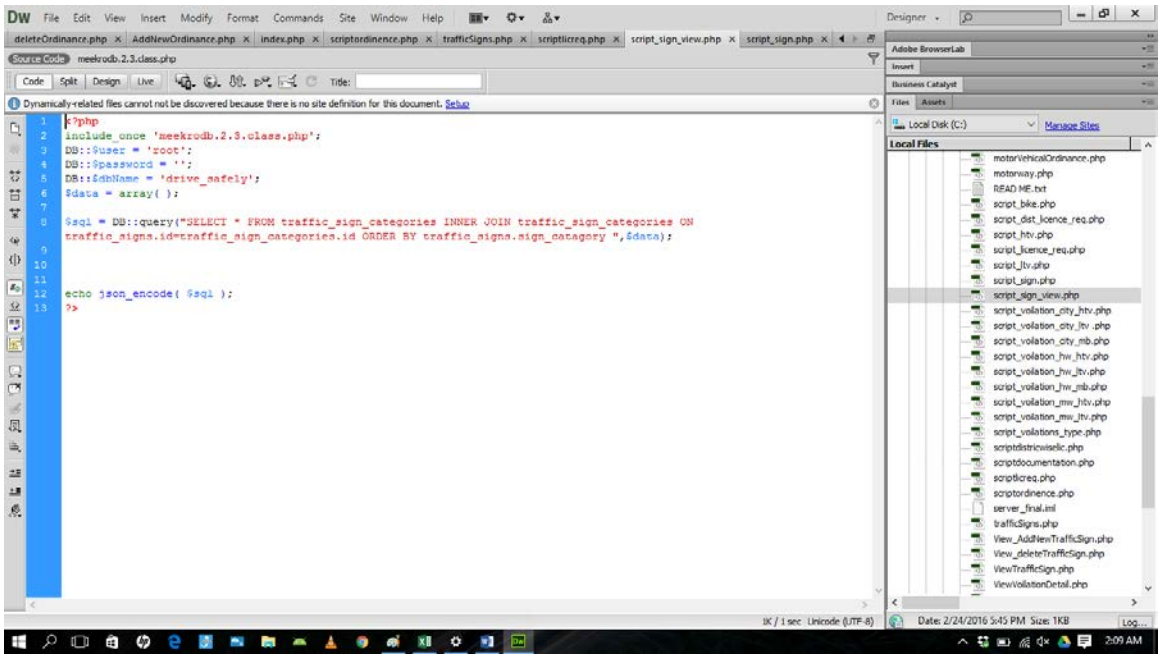
Script Ordinance



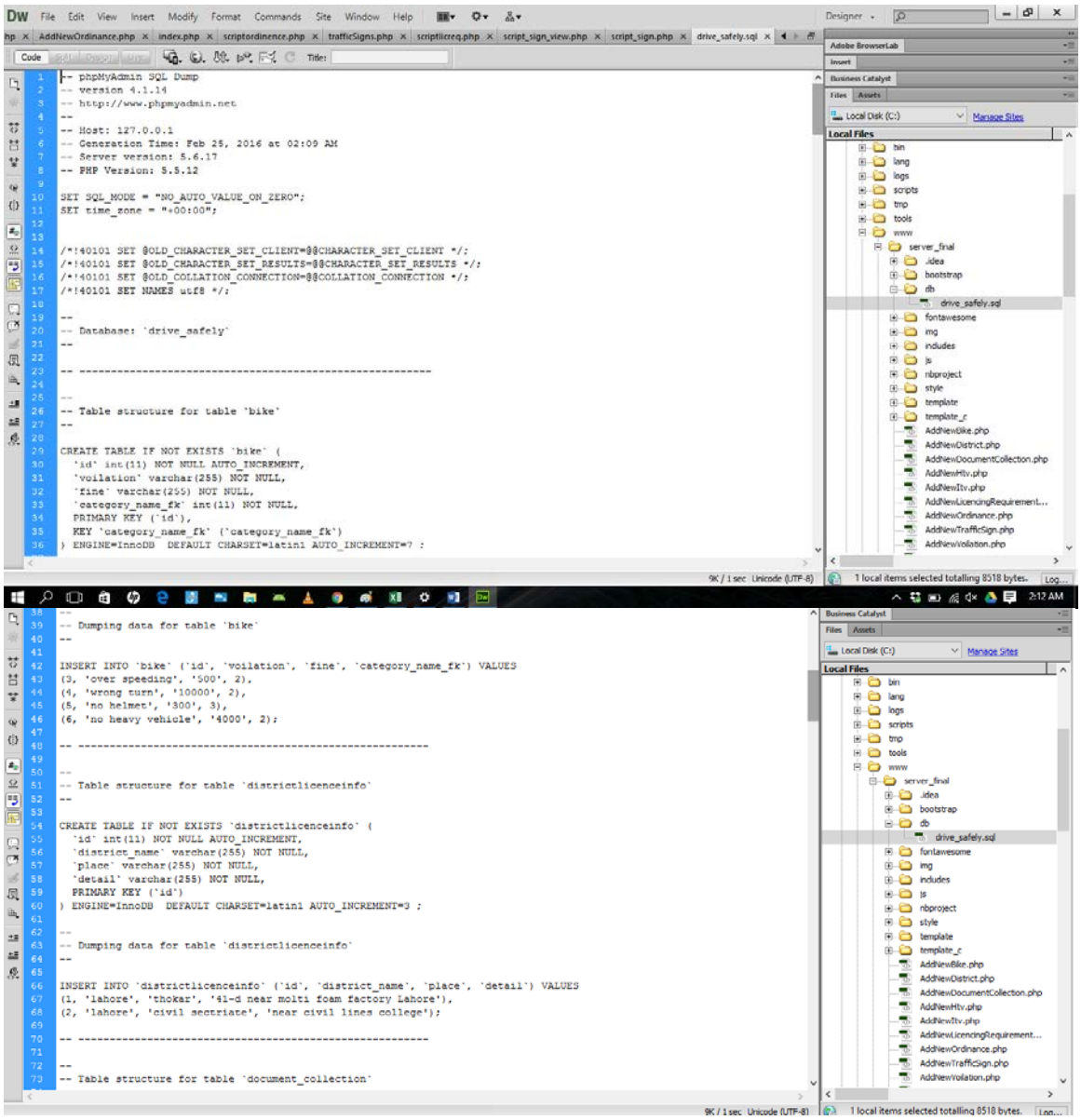
Script Traffic signs

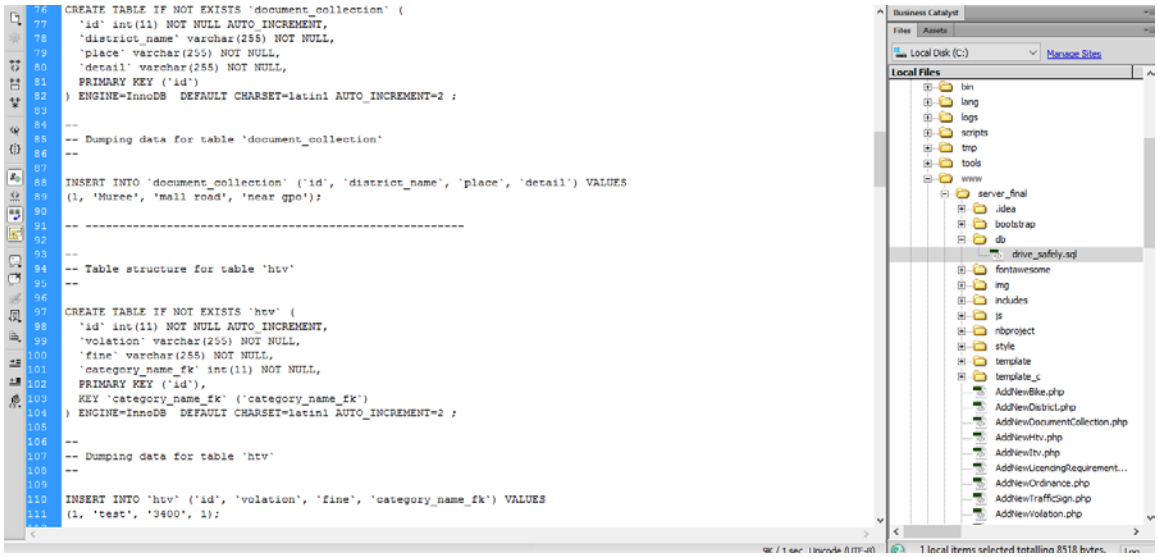


Script Traffic signs View



SQL Dump file


-- phpMyAdmin SQL Dump
-- version 4.1.14
-- http://www.phpmyadmin.net
-- Host: 127.0.0.1
-- Generation Time: Feb 25, 2016 at 02:09 AM
-- Server version: 5.6.17
-- PHP Version: 5.5.12
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
-- Database: `drive_safely`
-- Table structure for table `bike`
CREATE TABLE IF NOT EXISTS `bike` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`voilation` varchar(255) NOT NULL,
`fine` varchar(255) NOT NULL,
`category_name_fk` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `category_name_fk` (`category_name_fk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
-- Dumping data for table `bike`
INSERT INTO `bike` (`id`, `voilation`, `fine`, `category_name_fk`) VALUES
(3, 'over speeding', '500', 2),
(4, 'wrong turn', '10000', 2),
(5, 'no helmet', '300', 3),
(6, 'no heavy vehicle', '4000', 2);
-- Table structure for table `districtlicenceinfo`
CREATE TABLE IF NOT EXISTS `districtlicenceinfo` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`district_name` varchar(255) NOT NULL,
`place` varchar(255) NOT NULL,
`detail` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
-- Dumping data for table `districtlicenceinfo`
INSERT INTO `districtlicenceinfo` (`id`, `district_name`, `place`, `detail`) VALUES
(1, 'lahore', 'thokar', '41-d near molti foam factory Lahore'),
(2, 'lahore', 'civil secriate', 'near civil lines college');
-- Table structure for table `document_collection`



```
CREATE TABLE IF NOT EXISTS `document_collection` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `district_name` varchar(255) NOT NULL,
  `place` varchar(255) NOT NULL,
  `detail` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

--
-- Dumping data for table `document_collection`
--

INSERT INTO `document_collection` (`id`, `district_name`, `place`, `detail`) VALUES
(1, 'Muree', 'mall road', 'near gpo');

-- --------------------------------------------------------

--
-- Table structure for table `htv`
--

CREATE TABLE IF NOT EXISTS `htv` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `volation` varchar(255) NOT NULL,
  `fine` varchar(255) NOT NULL,
  `category_name_fk` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `category_name_fk` (`category_name_fk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

--
-- Dumping data for table `htv`
--

INSERT INTO `htv` (`id`, `volation`, `fine`, `category_name_fk`) VALUES
(1, 'test', '3400', 1);
```

12. Testing

12.1 Test case Admin Login

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin Login

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Open Admin panel
2. Provide valid admin user name and password
3. Click login

Expected Results

Successfully Login

Actual Result

Successfully Login

Result

Test case passed

12.2 Test case Admin Stay Login

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin stays login on panel

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Open Admin panel
2. Provide valid admin user name and password
3. Clicks on remember me button
4. Click login

Expected Results

Successfully Login

Actual Result

Successfully Login

Results

Test Case Passed

12.3 Test case Admin Home Page

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin home page

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Open Admin panel
2. Provide valid admin user name and password
3. Click login
4. View's the home page

Expected Results

Successfully Login and view home page

Actual Result

Successfully Login and view home page

Results

Test Case Passed

12.4 Test case Traffic Signs

Severity

Major

Test Engineer

Waleed Mustafa

Description

User wants to view Traffic Signs

Probability

Always Reproduced

Status

New

Steps to Procedure

1. User opens the application and from 1st menu enters traffic signs section
2. Different types of traffic signs sections like mandatory or informative are shown in next menu
3. Selects any of them which he likes

4. Traffic signs with name and picture in a table view are shown
5. Selects the sign which he finally wants to view and can see the traffic sign fully in a single view.

Expected Results

Successfully viewed required information.

Actual Result

Successfully viewed required information.

Results

Test Case Passed

12.5 Test case Violations

Severity

Major

Test Engineer

Waleed Mustafa

Description

User wants to view Violations

Probability

Always Reproduced

Status

New

Steps to Procedure

1. User enters the violation and fine menu from home screen.
2. User selects the violations.
3. Selects the area for which he is looking up.
4. Selects the type of vehicle
5. List related to violations of that specific type of vehicle and area is shown

Expected Results

Successfully viewed required information.

Actual Result

Successfully viewed required information.

Results

Test Case Passed

12.6 Test case Motor Vehicle Ordinance

Severity

Major

Test Engineer

Waleed Mustafa

Description

User wants to view Motor Vehicle Ordinance

Probability

Always Reproduced

Status

New

Steps to Procedure

1. User enters the violation and fine menu from home screen.
2. User selects the Ordinances from Motor Vehicle Ordinance.
3. Ordinances are viewed and can be read.

Expected Results

Successfully viewed required information.

Actual Result

Successfully viewed required information.

Results

Test Case Passed

12.7 Test case Fine Payment and Document Collection

Severity

Major

Test Engineer

Waleed Mustafa

Description

User wants to know where to pay fine and collect documents

Probability

Always Reproduced

Status

New

Steps to Procedure

1. User selects fine payment from home screen.
2. Looks for the city
3. Looks up the area from the city where he was fined
4. Than in description views the bank where fine is to be paid and address of document collection center.

Expected Results

Successfully viewed required information.

Actual Result

Successfully viewed required information.

Results

Test Case Passed

12.8 Test case Licensing Information

Severity

Major

Test Engineer

Waleed Mustafa

Description

User wants to know how to apply and submit documents for driving license.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. User enters the licensing information menu from home screen.
2. Selects the license requirements.
3. Information related to the process for driving license will be shown

Expected Results

Successfully viewed required information.

Actual Result

Successfully viewed required information.

Results

Test Case Passed

12.9 Test case District vise Licensing Information

Severity

Major

Test Engineer

Waleed Mustafa

Description

User wants to know where to apply and submit documents for driving license according to his district.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. User enters the licensing information menu from home screen.
2. Selects the District vise license requirements.
3. Information related to the process for driving license will be shown

Expected Results

Successfully viewed required information.

Actual Result

Successfully viewed required information.

Results

Test Case Passed

12.10 Test case Delete Traffic Sign Category

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a traffic sign category.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Traffic signs option.
2. Ensures the category has no data in it.
3. Clicks on delete button in front specific category.
4. Category Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.11 Test case Delete Traffic Sign

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a traffic sign.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Traffic signs option.
2. Opens the category which contains the sign to be deleted.
3. Clicks delete button in front of the sign.
4. Traffic Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.12 Test case Delete Motor Vehicle Ordinance

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a motor vehicle ordinance.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Motor vehicle ordinance.
2. Clicks on delete button in front of specific ordinance.
3. Ordinance Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.13 Test case Delete Area of Drive

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete area of drive.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the violations.
2. Clicks on delete button in front of specific area of drive.
3. Area of drive Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.14 Test case Delete Violation

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a Violation.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Violations.
2. Clicks on Area of drive.
3. Clicks on type of vehicle.
4. Clicks on delete button in front of specific Violation.
5. Violation Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.15 Test case Delete Fine Payment and Doc Collection Center

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a fine payment center and document collection center.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the fine payment and document collection.
2. Clicks on delete button in front of specific fine payment and document collection center.
3. Fine payment and document collection center Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.16 Test case Delete License Requirement Information

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a License requirement information.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the get a license.
2. Enters license requirement
3. Clicks on delete button in front of specific license type.
4. License Requirement Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.17 Test case Delete District from Get a License

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to delete a District from get a license district vise licensing information.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters Get a License
2. Clicks on District vise licensing Information.
3. Clicks on delete button in front of specific District.
4. District Deleted.

Expected Results

Successfully deleted information.

Actual Result

Successfully deleted information.

Results

Test Case Passed

12.18 Test case Add Traffic Sign Category

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a traffic sign category.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Traffic signs option.
2. Clicks on Add button.
3. New page opens up.
4. Name of Category provided.
5. Clicks submit button
6. Category Created.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.19 Test case Add Traffic Sign

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a traffic sign.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Traffic signs.
2. Opens the category in which the sign is to be added.
3. Clicks add button.
4. New Page opens up.
5. Provide Name.
6. Upload Picture.
7. Clicks Submit Button.
8. Traffic Sign Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.20 Test case Add Motor Vehicle Ordinance

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a motor vehicle ordinance.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Motor vehicle ordinance.
2. Clicks on add button.
3. New Page opens up.
4. Provides complete ordinance.
5. Clicks Submit Button.
6. Ordinance Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.21 Test case Add Area of Drive

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add area of drive.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the violations.
2. Clicks on add button.
3. New Page opens up.
4. Provides Area name.
5. Clicks submit button
6. Area of drive Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.22 Test case Add Violation

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a Violation.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the Violations.
2. Clicks on Area of drive.
3. Clicks on type of vehicle.
4. Clicks on add button.
5. Provides details for violation.
6. Clicks on submit button
7. Violation Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.23 Test case Add Fine Payment and Doc Collection Center

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a fine payment center and document collection center.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the fine payment and document collection.
2. Clicks on add button.
3. Provides Details.
4. Clicks on submit button.
5. Fine payment and document collection center Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.24 Test case Add License Requirement Information

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a License requirement information.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters the get a license.
2. Enters license requirement
3. Clicks on add.
4. Fills the required fields.
5. Clicks on submission button.
6. License Requirement Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12.25 Test case Add District from Get a License

Severity

Major

Test Engineer

Waleed Mustafa

Description

Admin wants to add a District to, get a license district vise licensing information.

Probability

Always Reproduced

Status

New

Steps to Procedure

1. Admin enters Get a License
2. Clicks on District vise licensing Information.
3. Clicks on add button.
4. Fills the required Field of information.
5. Clicks on submit Button.
6. District Added.

Expected Results

Successfully added information.

Actual Result

Successfully added information.

Results

Test Case Passed

12. Results

100%	Completion
100%	Accuracy
100%	Correctness

13. Coclusion

Drive legally is a system which will help people learning about traffic Laws, signs. It will encourage people to get driving license. When people will come to know how easy and simple it is to have license they will happily go for it.

14. Future Work

Drive legally is a system which is available to be used by only Android users in future it will be available for other platforms like IOS, web and Windows operating system

13. References:

1.
http://projects.pucit.edu.pk/downloads/deliverable%20%201%20%20planning%20&%20requirements.doc
2.
Student paper submitted to Higher Education Commission Pakistan on 2015-05-21
3.
http://ckverma.com/Book/SwEnggFundamentals.doc
4.
http://ckverma.com/Book/SwEnggFundamentals.doc
5.
http://www.siudmysore.gov.in/PMdetails/RM.pdf