**Project Proposal On**

**Online Vehicle Tax Pay System**

****

Submitted By: Submitted To:

Bishesh Man Singh Sudeep Lal Bajimaya

NCC ID: 00175019 Module leader

Softwarica College of IT and E-commerce

Contents

[**Chapter1. Introduction** 5](#_Toc5528855)

[1.1 Project Introduction 5](#_Toc5528856)

[1.2 Background of the project 5](#_Toc5528857)

[1.3 Problem Statement 5](#_Toc5528858)

[1.4 Description of the Project 6](#_Toc5528859)

[1.4.1 Feature of the Project 6](#_Toc5528860)

[1.5 Overview of the Project 7](#_Toc5528861)

[**Chapter2. Scope of the Project** 8](#_Toc5528862)

[2.1 Scope 8](#_Toc5528863)

[2.2 Limitations 8](#_Toc5528864)

[2.3 Aims 8](#_Toc5528865)

[2.4 Objectives 9](#_Toc5528866)

[2.5 Overview of the Scope 9](#_Toc5528867)

[**Chapter3. Development Methodology** 10](#_Toc5528868)

[3.1 Description of the Methodology 10](#_Toc5528869)

[3.2 Design Pattern 11](#_Toc5528870)

[3.3 System Architecture 13](#_Toc5528871)

[**Chapter4. Project Planning** 15](#_Toc5528872)

[4.1 Work Breakdown Structure (WBS) 15](#_Toc5528873)

[4.2 Milestones 16](#_Toc5528874)

[4.3 Gantt Chart 17](#_Toc5528875)

[**Chapter5. Risk Management** 19](#_Toc5528876)

[**Chapter6. Configuration Management** 21](#_Toc5528877)

[**Chapter7. Conclusion** 23](#_Toc5528878)

[**Chapter8. References** 24](#_Toc5528879)

List of Screenshots

[Screenshot 1: Waterfall Model 10](#_Toc5616096)

[Screenshot 2: MVC design pattern 12](file:///C:\Users\Bishesh\Documents\GitHub\Online-Vehicle-Tax-Pay-System\Proposal\00175019_CP_Proposal.docx#_Toc5616097)

[Screenshot 3: Three-tier Architecture 14](#_Toc5616098)

[Screenshot 4: Work Breakdown Structure for Online Vehicle Tax Pay System 15](file:///C:\Users\Bishesh\Documents\GitHub\Online-Vehicle-Tax-Pay-System\Proposal\00175019_CP_Proposal.docx#_Toc5616099)

[Screenshot 5: Schedule of Online Vehicle Tax Pay System 17](#_Toc5616100)

[Screenshot 6: Gantt chart of Online Vehicle Tax Pay System 18](#_Toc5616101)

[Screenshot 7: Github Directory Structure 21](#_Toc5616102)

[Screenshot 8: Tree structure for Online Vehicle Tax Pay System 22](#_Toc5616103)

# **Chapter1. Introduction**

## 1.1 Project Introduction

Online Vehicle Tax Pay System is the system for paying vehicle tax online. This System will let owner pay the tax of their vehicle without getting in queue for hours. Vehicle Insurance is also processed through this system. This project will save valuable time of Vehicle Owner.

## 1.2 Background of the project

People are very busy nowadays. They want their life to be easier and simpler. In this busy life schedule, most of their time passes while standing in queue for many specific purpose. One of them is for paying their vehicle tax. So I intend to decrease this waste time. Here in this system, Owner need to enter their vehicle information and owner’s information like phone number, email id, address. The information will be saved in our system and will get their tax paid.

## 1.3 Problem Statement

Many of the Government System are now online but the process of paying vehicle tax is still in old way i.e standing in queue and wait for the turn. This consumes time of every individual. This is the problem in current system. So this project will focus only on possible way of paying vehicle tax efficient. With this project, Owner can be sure that their vehicle tax will be paid in time without any effort.

## 1.4 Description of the Project

Description of the project include following requirements:

**Front End Language: HTML, CSS, Javascript**

**Programming Language: PHP (Laravel Framework)**

**Tool: Dreamweaver**

**Platform: Web Based**

**Pattern: MVC (Model, View and Controller)**

**Database: MySQL**

For this Project I will use HTML, CSS, and Bootstrap for completing my frontend task while I will be using PHP for backend and MySQL for database. The System will be based on web. Tools like Dreamweaver will be used during this whole project.

## 1.4.1 Feature of the Project

Features of the Online Vehicle Tax Pay System include following things:

* **Register**
* For register user will enter their Name, Mobile Number, Email Address, Last Renewal Date, and Vehicle Type.
* New information regarding clients will sent to Email.
* **User can see the status of their task.**
* User can view if their Bill Book is renewed or not.
* **User can see the Tax Amount of their vehicle.**
* **User will get notification of their next renew time.**
* **Report of the Payment**

## 1.5 Overview of the Project

Online Vehicle Tax Pay System will be useful for the person who are very busy in their work and are unable to get time for paying their vehicle tax. The system will generate the tax amount after the information of the vehicle is provided by owner. It is a simpler yet faster service for people.

The main purpose of this project is to make people life easier by automating one of the must do task of people i.e Vehicle Tax Payment

# **Chapter2. Scope of the Project**

## 2.1 Scope

Online Vehicle Tax Pay System is for supporting people/organization for paying their vehicle tax. It helps owner to pay their tax in easier way. One can even pay their amount directly by bank account or by online payment systems. It doesn’t just get the job done easily but also reduces the queue in the Transport Office as not all owner will be going there to pay tax.

## 2.2 Limitations

Individual may not get their bill book renewed at instance as government does not have the automatic renew system. It may take up to max 24hr for renew depending on the situation. This system doesn’t support android or IOS platform as it is based only on Web.

## 2.3 Aims

* The main aim of this project is to develop web based application where a vehicle owner can enter their vehicle type, last renewal date and get their tax amount and pay the amount without getting in queue for hours.
* Calculates the Tax Amount automatically.

## 2.4 Objectives

* User Friendly Design
* Time Management
* Online Payment System.
* Generate Tax Amount
* Database of Vehicle Information.
* Different diagrams like use case, activity and sequence are designed.
* Can be used for both individual or for organizations.
* Illustrate the flow of application.

## 2.5 Overview of the Scope

The system Online Vehicle Tax Payment System has its own limitations, aims and objectives. Limitation of this system is that payment are not done in real time. It requires time to process payment as the government doesn’t allow the online transaction for the vehicle tax payment. Furthermore the system also run on browser as it is web based only. The aim of this system is to provide hassle free environment for paying their vehicle tax. The Objective of this system is to provide bug free system for user and well documented for future use.

# **Chapter3. Development Methodology**

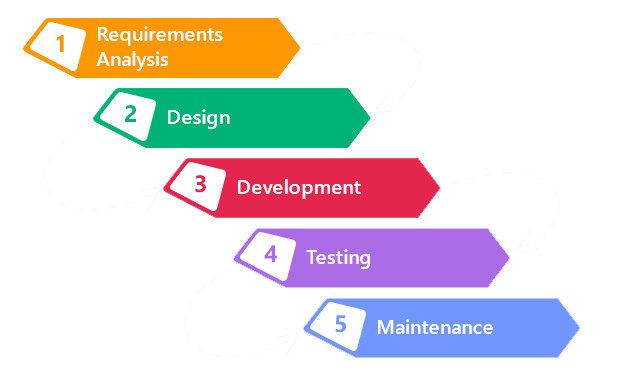
## 3.1 Description of the Methodology

**Waterfall Model**

The waterfall model is a sequential model divided mainly into five phases. It is a very simple and easy to understand. Each phase must be completed before moving into another phase.

Phases of Waterfall model are as follows:

* Requirements analysis
* Design
* Implementation
* Verification
* Maintenance



Screenshot 1: Waterfall Model

I have chosen Waterfall Methodology due to following reason:

* It’s easier to use and understand.
* Since the project is small, waterfall method is suitable for it.
* Minimal Management
* No overlapping

Advantage of using Waterfall Model:

* Earlier phases must be completed before continuing to next phase. So overlapping doesn’t occur.
* Developer and Customer agree upon delivery of project at before starting the project.
* Scope of the project is known in advance. So Progress is measured easily.
* Customer Presence is not required after requirement phase.

Disadvantage of using Waterfall Model:

* All the deliverables are based upon documented requirement. So the customer may not see the product until it is fully done. If changes occurred by that time it is difficult to implement.
* Moving backward to phases is difficult and costlier.
* Testing time is at last phase which is against the rule of software development.

## 3.2 Design Pattern

Here in this project I will use **Model View Controller (MVC)** design pattern.

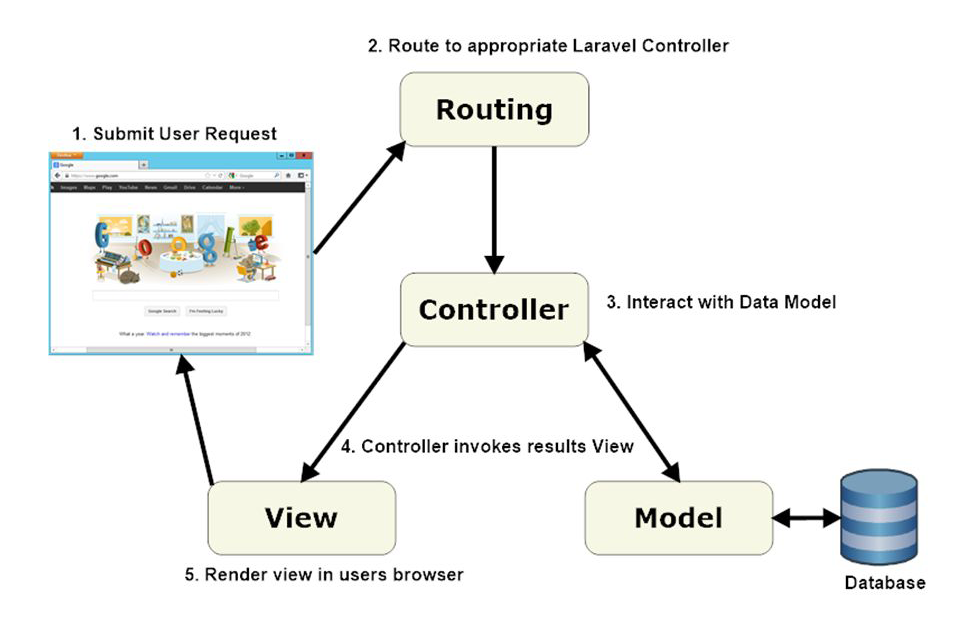
**Model** – this part is concerned with the business logic and the application data. It can be used to perform data validations, process data and store it. The data can come from;

* flat file
* database
* XML document
* Other valid data sources.

**Views** – this part deals with presenting the data to the user. This is usually in form of HTML pages.

**Controller** – this is the part deals with the users’ requests for resources from the server.

In a nutshell, the controller links the models and views together depending on the requested resources.



Screenshot 2: MVC design pattern

I will use MVC Design Pattern in this project because of the following reasons:

* It’s faster for developing Web Application.
* MVC Model Returns The Data Without The Need of Formatting.
* The Modification Never Affects The Entire Model.

**Advantages of using MVC design pattern:**

* It hides all complex details of implementation.
* It provides standard methods to build applications.
* With MVC architecture, you have full control over how your application appears to the world by choosing the application routes. This comes in handy when you are trying to improve your application for SEO purposes
* MVC can force you to split your files into logical directories which makes it easier to find files when working on large projects.

**Disadvantages of using MVC design pattern:**

* Modification option for core behavior is lacking.
* The view and the controller are closely coupled which makes modification to one affect the other.
* Changes to the model interface will necessitate changes to the controller and the view.
* When the model is active frequent changes to model can result in excessive updates of the corresponding views.

## 3.3 System Architecture

**Three Tier Architecture**

As it’s name, this tier consists of three tier i.e Presentation Tier, Application Tier and Data Tier.

**Presentation Tier:**

It contains UI part of the application. Here data is presented to the user or input is taken from the user. This tier communicates with other tiers by sending results to the browser and other tiers in the network.

**Application Tier:**

All the logic written like validation, calculations, insertions are written in this layer. It is the intermediary layer which helps to make communication faster between presentation and data layer.

**Data Tier:**

Database resides into this Tier. Data Access layer contains methods to connect with database and to perform CRUD from database based on input data.



Screenshot 3: Three-tier Architecture

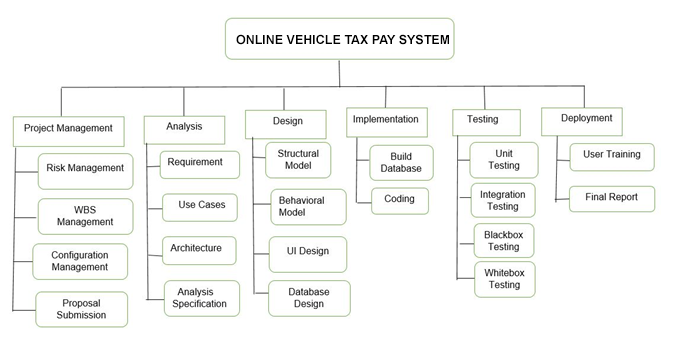
The reason I used 3 tier architecture are as follows:

* Each tier are independent of the other tiers, so updates or changes can be carried out without affecting the whole application.
* Each tier can exploit the modular architecture of enabling systems using easily scalable components, which increases availability.
* Tiers are based on the deployment of layers, scaling out an application is reasonably straightforward.

# **Chapter4. Project Planning**

## 4.1 Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) is a breakdown of project into smaller components to remove complexity and for manageable purpose. It decomposes the project into phases, deliverables and work packages. It breakdown the entire project into meaningful components. The Work Breakdown Structure for Online Vehicle Tax Pay System are as given below:

* Project Management
* Analysis
* Design
* Implementation
* Deployment

Screenshot 4: Work Breakdown Structure for Online Vehicle Tax Pay System

## 4.2 Milestones

**Milestones Table:**

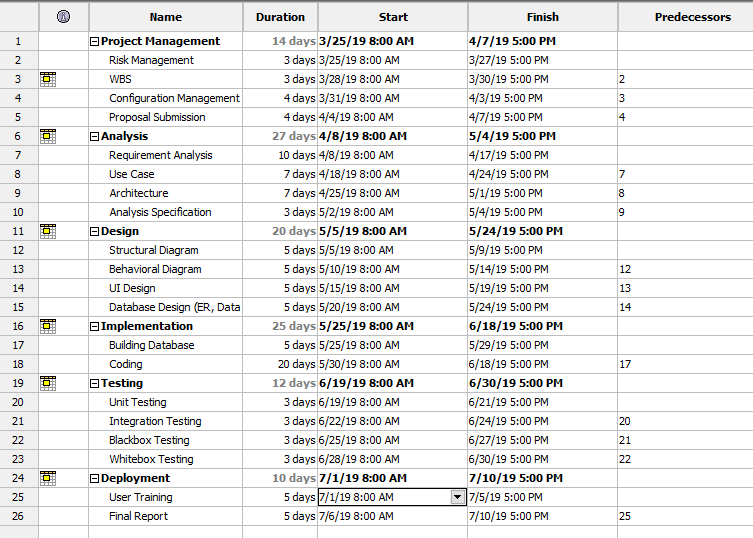
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.N** | **Milestones** | **Start Date** | **End Date** | **Total Days (108)** |
| 1. | **Project Management**  Risk Management  WBS  Configuration Management  Proposal Submission | **03/25/2019**  03/25/2019  03/28/2019  03/31/2019  04/04/2019 | **04/07/2019**  03/27/2019  03/30/2019  04/03/2019  04/07/2019 | **14**  3  3  4  4 |
| 2 | **Analysis**  Requirement analysis  Use Case  Architecture  Analysis specification | **04/08/2019**  04/08/2019  04/18/2019  04/25/2019  05/02/2019 | **05/04/2019**  04/17/2019  04/24/2019  05/01/2019  05/04/2019 | **27**  10  7  7  3 |
| 3 | **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER, Data Dictionary) | **05/05/2019**  05/05/2019  05/10/2019  05/15/2019  05/20/2019 | **05/24/2019**  05/09/2019  05/14/2019  05/19/2019  05/24/2019 | **20**  5  5  5  5 |
| 4 | **Implementation**  Building Database  Coding | **05/25/2019**  05/25/2019  05/30/2019 | **06/18/2019**  05/29/2019  06/18/2019 | **25**  5  20 |
| 5 | **Testing**  Unit Testing  Integration Testing  Black box Testing  White box Testing | **06/19/2019**  06/19/2019  06/22/2019  06/25/2019  06/28/2019 | **06/30/2019**  06/21/2019  06/24/2019  06/27/2019  06/30/2019 | **12**  3  3  3  3 |
| 6 | **Deployment**  User Training  Final Report | **07/01/2019**  07/01/2019  07/06/2019 | **07/10/2019**  07/05/2019  07/10/2019 | **10**  5  5 |

Here, total of **108 days** starting from **March 25th 2019** to **July 10th 2019** will be total time span for development of this project from project management to its deployment.

## 4.3 Gantt Chart

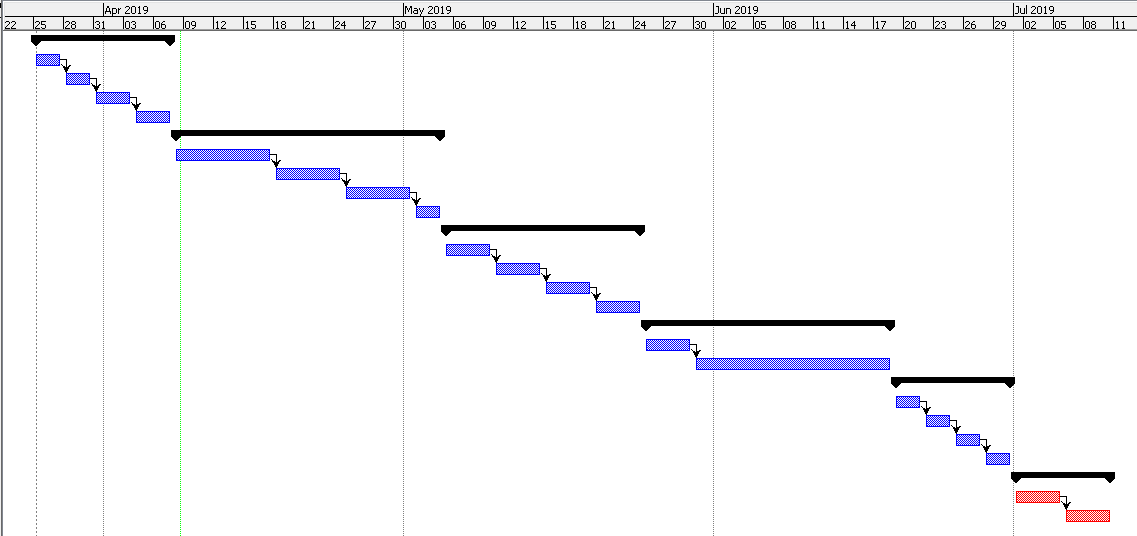
Below are the schedule and the Gantt chart of project

Schedule:



Screenshot 5: Schedule of Online Vehicle Tax Pay System

Gantt chart:



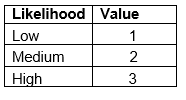
Screenshot 6: Gantt chart of Online Vehicle Tax Pay System

# **Chapter5. Risk Management**

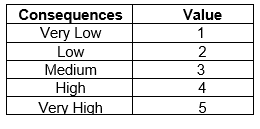
The process of identification, analysis and acceptance or mitigation of uncertainty in investment decisions is known as risk management. It controls or minimize the flow of unexpected risk, time and money. We have mathematical Calculation to get the impact of the risk.

Impact = Likelihood \* Consequences

Likelihood values of risk:



Consequences values of risk:



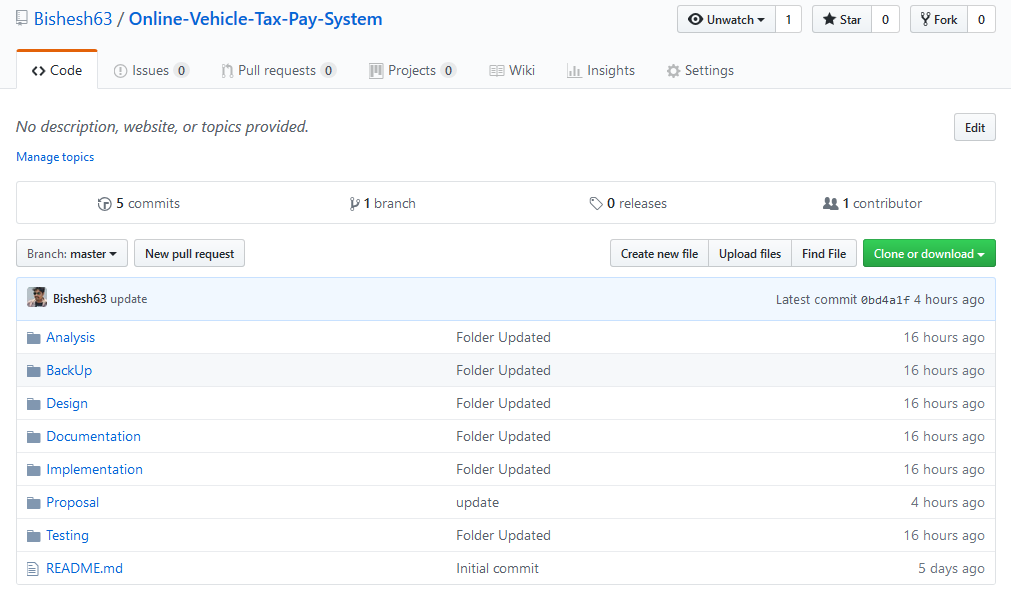
**Risk Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N** | **Risks** | **Likelihood** | **Consequences** | **Impact** | **Solution** |
| 1 | Lack of resources | 2 | 3 | 6 | Resources should be provided before starting project. |
| 2 | Operating System failure | 2 | 2 | 4 | Backup File |
| 3 | User Experience | 2 | 3 | 6 | Provide System Manual to user. |
| 4 | Virus, Threats | 2 | 4 | 8 | Daily Based Scanning and updated antivirus should be installed. |
| 5 | Natural Disaster | 1 | 5 | 5 | Implementation of cloud back up system. |
| 6 | Hard drive failure | 1 | 4 | 4 | To prevent from failure reduce data load and provide back-up system. |

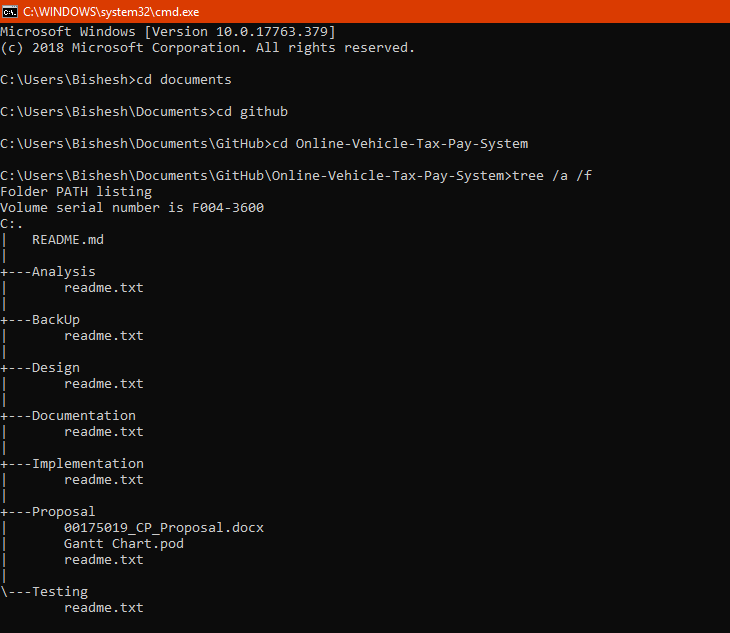
# **Chapter6. Configuration Management**

Configuration Management is the governance and system engineering process which identifies and track individual configuration items, documenting functional capabilities and interdependencies. It ensures consistency among physical assets in an operational environment.

Below information shows the directory structure of project:



Screenshot 7: Github Directory Structure



Screenshot 8: Tree structure for Online Vehicle Tax Pay System

# **Chapter7. Conclusion**

There is always rush in Transport Department Office for paying taxes of vehicles. People have to wait in lines for hours leaving their jobs behind. The main purpose of my project is to make rush in Transport Department lower. Here my project will provide platform to pay vehicle taxes in easy way for such people who are very busy in their daily work. This will save people’s time and money.

# **Chapter8. References**