**MECHANIC TRACKING SYSTEM**

A PROJECT REPORT

Submitted by

### RAMA CHANDRAN.S

**(Register No: 18CSE34)**

*in partial fulfillment of the requirement for the award of the degree of*

## BACHELOR OF SCIENCE IN COMPUTER SCIENCE

### Under the guidance of

**Dr. M. Venkatesh, MCA.,M.Phil.,Ph.D**

****

### DEPARTMENT OF COMPUTER SCIENCE ST. XAVIER'S COLLEGE (AUTONOMOUS)

(Recognized as ‘College with Potential for Excellence’ by UGC) (Accredited by NAAC at A++ Grade with a CGPA of 3.66 out of 4 in IV Cycle)

### PALAYAMKOTTAI – 627002

**Mar – 2021**

**DEPARTMENT OF COMPUTER SCIENCE ST.XAVIER'S COLLEGE (AUTONOMOUS)**

(Recognized as ‘College with Potential for Excellence’ by UGC) (Accredited by NAAC with A++ Grade with a CGPA of 3.66 out of 4 in IV Cycle)

### PALAYAMKOTTAI -627002

****

**BONAFIDE CERTIFICATE**

This is to certify that the project work entitled **“MECHANIC TRACKING SYSTEM”** is the bonafide work of **RAMA CHANDRAN.S (18CSE34)** who carried out the project under my supervision and submitted during the academic year 2020-2021.

The Viva-Voce held on …………………

## INTERNAL EXAMINER EXTERNALEXAMINER

**CO-ORDINATOR**

**ACKNOWLEDGEMENT**

First I thank the Lord Almighty who blessed me with wonderful faculty members, friends, and family whose love and encouragement have given a significant impact on my every accomplishment.

I express my sincere thanks with love to my parents, who were cheerleaders, supporters throughout the project. Their dedications were shown in many subtle ways and indeed instrumental in achieving the outcome.

I indent to owe my sincere thanks to **REV. DR. S. MARIADOSS, S.J.,** Principal, St. Xavier’s College (Autonomous), Palayamkottai, who always cared to fulfill the needs of the students and be a moral support to me throughout the year.

I express my sincere gratitude to **DR. V. S. JOSEPH ALBERT,** Deputy Principal, Shift-II, St. Xavier’s College (Autonomous), Palayamkottai for his motivation and encouragement.

I wholeheartedly thank **MRS. J. REXY,** Assistant Professor and Co-ordinator, Department of Computer Science for providing me with constructive criticism and suggestions to fulfill my project.

I extend my deep gratitude to my project guide **Dr. M. Venkatesh** for his valuable guidance and encouragement throughout my project.

I thank all my faculty members of the Computer Science Department for their valuable kind advice and pleasing co-ordination throughout my project. I also thank all my friends who supported my aspirations with real encouragement and for their comments and suggestions in each phase of my project.

Once again I praise Lord of Glory for mighty peace.

### RAMA CHANDRAN.S

**18CSE34**

**Abstract**

Consider a situation where your vehicle got punctured suddenly when you are traveling. To repair the vehicle you need to find where the mechanic is and you need to bring him. This project will help you to solve this problem by listing all the mechanics available in the location in addition to other details like contact no, experience, etc. For the Emergency purpose, You can contact the mechanic via mobile no or you can directly bring your vehicle to the mechanic shop. You can also book the mechanic and he will repair your vehicle at your doorstep. In this website there will be two users customer who need to repair their vehicle and the mechanic who need to be registered in this website with all the details about him.

**1. INTRODUCTION**

**1.1 Project Definition:**

When we need a mechanic in a hectic situation like in night time or in an area which is new to us. We can reach out to a mechanic by calling him if we have the phone number or we need to go to the mechanic shop which is nearby. We cannot guarantee that whether the mechanic shop is opened at that time. This project or portal is used to resolve this problem. Through this project, we can see who are all the mechanics available in our current location, whether the mechanic shop is opened, and other details of the mechanic like years of experience, mechanic shop address, contact details of the mechanic, etc. Through this portal, you can also book the mechanic so that the mechanic will reach out to you at the doorstep or at the current location.

The Flow of this portal or project is first of all the users whether it is a mechanic or customer need to register in this portal by providing some basic details like name, email, password, address, etc. Then if the user is a customer he/she can book a mechanic from the list of mechanics while booking the user need to provide some basic details like vehicle name, type(two/four wheeler), model, repair description, etc. Then the mechanic will reach out to the customer. If the user is a mechanic then he can see what are all the bookings available for him what is the repair description etc. The mechanic will also have an option to accept or reject the booking. The status of the booking can also be seen by both the customer and mechanic. For emergency purposes, the customer can directly contact the mechanic via email or mobile number.

**1.2 Project Description:**

This Project contains the following module

1. Customer Registration Module
2. Mechanic Registration Module
3. User Authentication Module
4. Mechanic Listing Module
5. Customer Details Updating Module
6. Mechanic Details Updating Module
7. Booking Module
8. Booking Indication Module
9. Payment Module

**1) Customer Registration module:**   
 Through this module customer will be able create an account in this portal by giving some details like name, age, phone no, email, password, address etc..

**2) Mechanic Registration module:**   
 Through this module mechanic will be able to create an account in this portal by giving some details related to him like name, age, phone no, email, password, mechanic shop address, experience, profile photo etc.

**3) User Authentication Module:**   
 with the help of this module Mechanics/Customers will be able to login into the portal by giving some login credentials like email, password. The user will be redirected to respective pages based on the login credentials given by them.

**4) Mechanic Listing module:**   
 After a customer is registered or logged into the portal. The customer will be able to see all the mechanics who are available in his location. This module also has search functionality that helps the customer to list all the mechanics based on some filters like location, experience, etc.

**5) Customer details updating module:**   
 with the help of this module, customers will be able to change some of their details like name, age, phone no, address, etc.

**6) Mechanic details updating module:**   
 with the help of this module, mechanics will be able to change some of their details like name, age, phone no, mechanic shop address, experience, etc. These changes will be reflected in the mechanic listing module.

**7) Booking module:**   
 This module helps the customer to book a mechanic by giving some details like vehicle name, vehicle model, current location, etc.

**8) Booking indication module:**   
 For the customer, this module contains all the mechanics he booked and the details about the mechanic and for the mechanic, this module contains all the customers who booked him and the details about the customer.

**9) Payment Module:**

After the mechanic has accepted the booking and reaches out to you. He will repair the vehicle and request an amount from the corresponding customer for repairing the vehicle. Then the customer will pay the requested amount to the corresponding mechanic. Both the customer and mechanic will be able to request and pay the amount by giving some credit details like cardholder name, card number, expiry date, etc.

**2. SYSTEM ANALYSIS**

**2.1 Existing System**

In the existing system, we use the traditional approach which is by asking someone the address of the mechanic shop, or if we have the number of the mechanic we can contact him directly. The drawbacks of the existing approach are

* what if the contacted mechanic is unavailable
* we need to take our vehicle to the mechanic shop which is a very tedious work or else
* We need to go to the mechanic shop and ask him to come to our location

**2.2 Proposed System**

The proposed system is designed in such a way that it will resolve all the major problems faced by the existing system. The main functionalities of the proposed system are

* You will have a list of mechanics and their contact details available for you.
* You can book your mechanic with a single click he will be available at your doorstep or in your current location
* You can check the status of your booking

**3. SYSTEM REQUIREMENTS**

**3.1 Hardware Used**

* Processor : Intel i5(3rd gen)
* RAM : 4gb
* Hard Disk : 500gb
* Monitor : 15 inch LCD

**3.2 Software Used**

* Operating System : Windows 10 Pro
* Text Editor : Visual Studio Code
* Front End : HTML,CSS, JavaScript
* Back End : PHP, MySQL
* Browser : Chrome (Version 88.0)
* Local web Server : XAMPP

**3.3 About the Software**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax as C Language.

**3.3.1 Common uses of PHP:**

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
* You add, delete, and modify elements within your database thru PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

**3.3.2 Characteristics of PHP**

Five important characteristics make PHP's practical nature possible:

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions <?php and ?> that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

**3.3.3What can PHP do?**

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

1. **Server-side scripting:** This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming.
2. **Command line scripting:** You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on \*nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.
3. **Writing desktop applications:** PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution.

PHP can be used on all major operating systems, including Linux, many UNIX variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. For the majority of the servers, PHP has a module, for the others supporting the CGI standard, PHP can work as a CGI processor.

One of the strongest and most significant features in PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple. The following databases are currently supported:

* Adabas D
* dBase
* Empress
* FilePro (read-only)
* Hyperwave
* IBM DB2
* Informix
* Ingres
* InterBase
* FrontBase
* mSQL
* Direct MS-SQL
* MySQL

**3.3.4 Why use PHP-Nuke and not static HTML pages?**

* Because managing large sites with only static HTML pages is dangerous for your health.
* Because through the dynamic pages, users can interact (Forum, chat)
* Because through the dynamic pages we can offer value added services (restricted areas, various services based on user classification...)
* Because the information is more easily cataloged.
* Because with a few PHP pages we recall a lot of information.
* Because keeping the contents up-to-date does not demand particular technical expertise and can be managed by anyone (by Davis Batistes).
* It is very intuitive and easy to learn (by Anonymous)
* It is easy to modify by those who intend to personalize the program (By Arus)
* It is easy to use by the lesser experts among us.

**3.4 MySQL**

**3.4.1 What is MySQL?**

MySQL is a fast, easy-to-use RDBMS used being used for many small and big businesses. MySQL is developed, marketed, and supported by **Oracle.** MySQL is becoming so popular because of many good reasons.

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

1. **SYSTEM DESIGN**

**4.1 Architectural Design**

Customer

Mechanic

Mechanic Tracking System

Booking Mechanic

Request for payments

Updating Profile Details

Filtering Mechanics

Payment

Accept/Reject Bookings

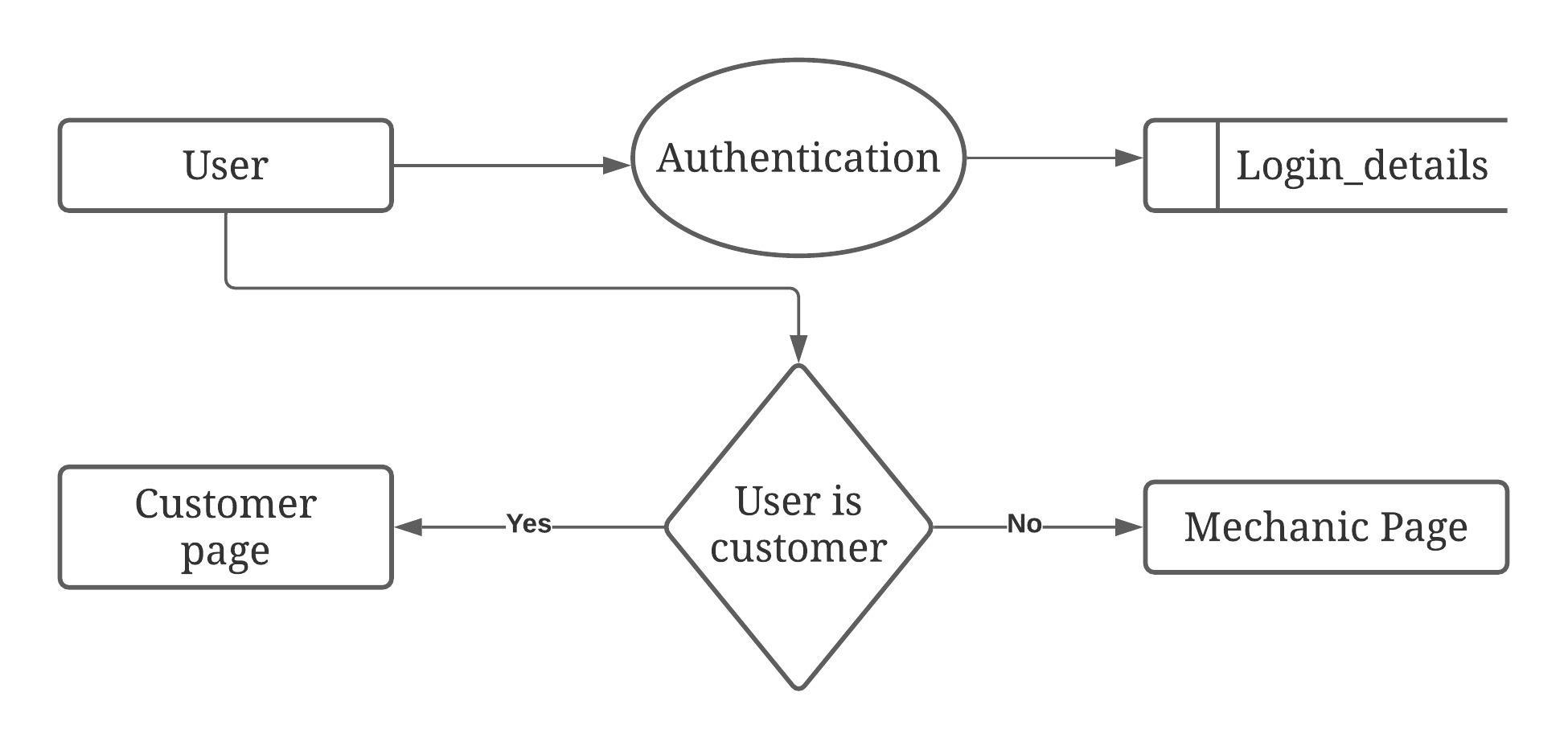
Registration

Registration

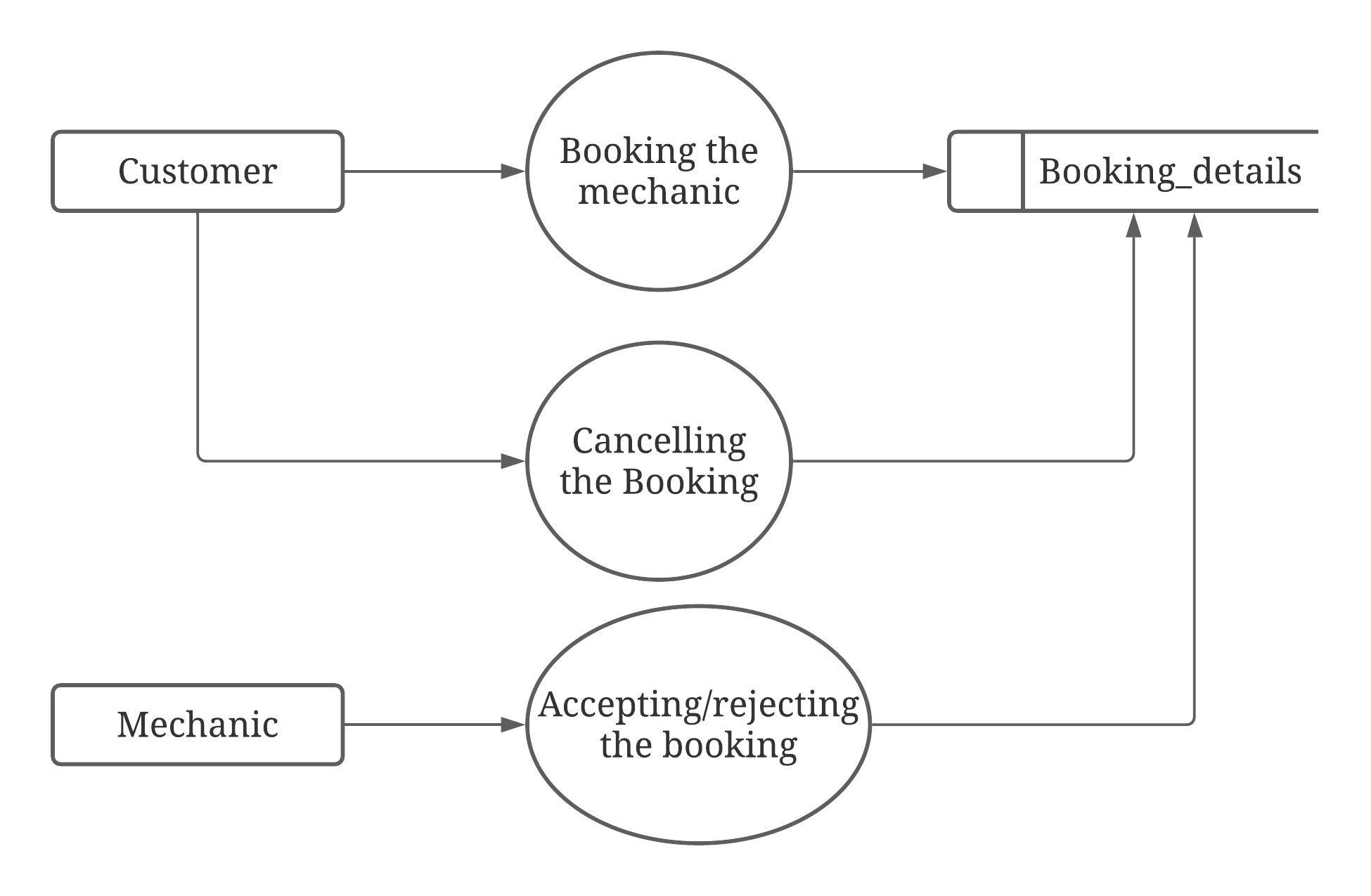
Updating Profile Details

**4.2 Data Flow Diagram**

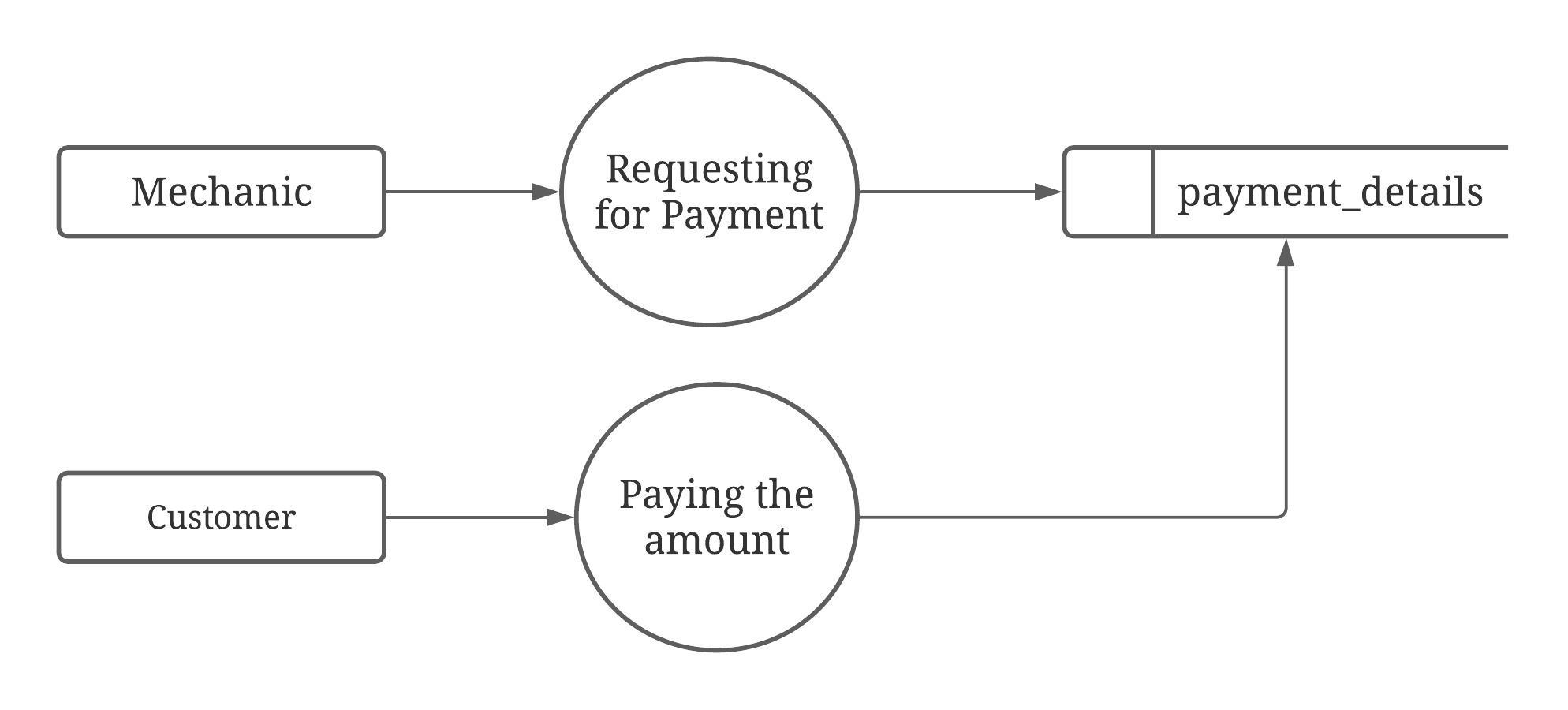
**Login:**

****

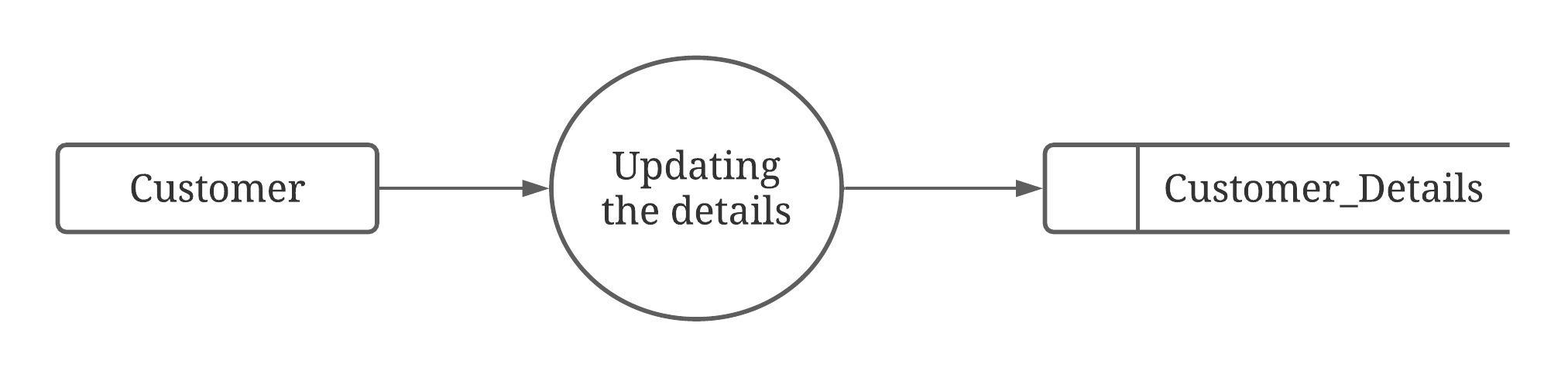
**Booking:**

****

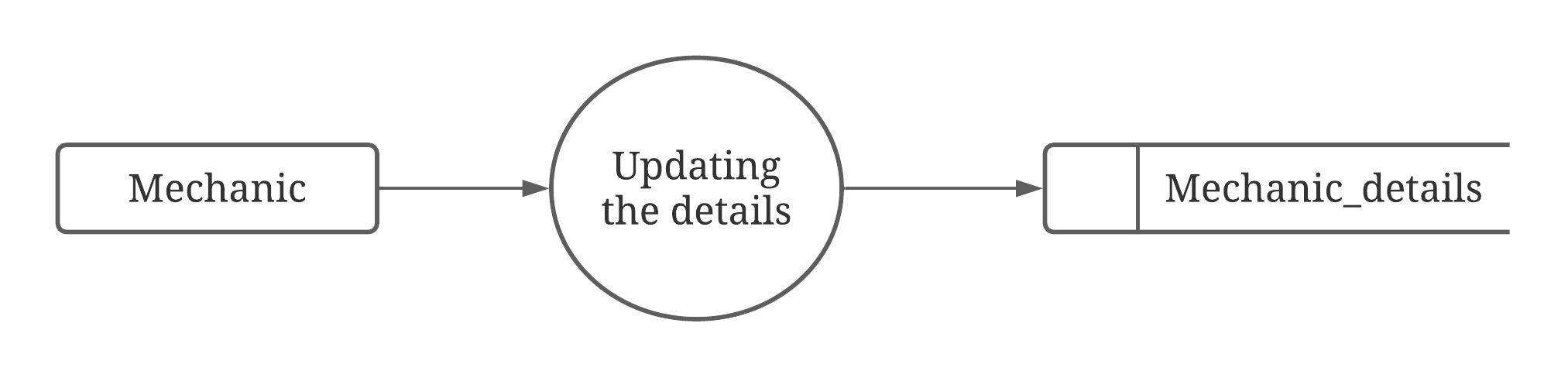
**Payment:**

****

**Customer Account Maintenance:**

****

**Mechanic Account Maintenance:**

****

**4.3 Database Design**

**Database Name:** mechanic\_tracking\_system

**1) Table Name**: customer\_details

**Purpose** : This table is used to store the details o the customer.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COLUMN | TYPE | | SIZE | | CONSTRAINT | DESCRIPTION |
| c\_name | Varchar | | 100 | | Not null | To store the customer name |
| c\_email | Varchar | | 100 | | Primary key | To store the customer email |
| c\_password | Varchar | | 100 | | Not null | To store the customer password |
| c\_phone\_no | Varchar | | 10 | | Not null | To store the phone customer number |
| c\_door\_no | Varchar | | 10 | | Not null | To store the door number of the customer |
| c\_street\_name | Varchar | | 100 | | Not null | To store the street name of the customer |
| c\_city | Varchar | | 100 | | Not null | To store the city of the customer |
| c\_pincode | Int | | 6 | | Not null | To store the pin code of the customer |
| c\_state | Varchar | | 100 | | Not null | To store the state of the customer |
| c\_photo | Text | | - | | - | To store the profile photo of the customer |
| c\_vehicle\_name | Varchar | | 100 | | - | To store the customer’s vehicle name |
| c\_vehicle\_model | Varchar | | 100 | | - | To store the customer’s vehicle model |
| c\_vehicle\_type | | Varchar | | 50 | - | To store the vechicle is four or two wheeler |

**2) Table Name**: mechanic\_details

**Purpose** : This table is used to store the details of the mechanics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN | TYPE | SIZE | CONSTRAINT | DESCRIPTION |
| m\_name | Varchar | 100 | Not null | To store the name of the mechanic |
| m\_email | Varchar | 100 | Primary key | To store the email of the mechanic |
| m\_password | Varchar | 100 | Not null | To store the password of the mechanic |
| m\_phone\_no | Varchar | 10 | Not null | To store the phone number of the mechanic |
| m\_door\_no | Varchar | 10 | Not null | To store the door number of the mechanic |
| m\_street\_name | Varchar | 100 | Not null | To store the street name of the mechanic |
| m\_city | Varchar | 100 | Not null | To store the city of the mechanic |
| m\_pincode | Int | 6 | Not null | To store the pincode of the mechanic |
| m\_area | Varchar | 100 | Not Null | To store the area of the mechanic |
| m\_landmark | Text | - | - | To store the landmark of the machnic |
| m\_state | Varchar | 100 | Not null | To store the state of the mechanic |
| m\_photo | Text | - | - | To store the photo of the mechanic |
| m\_experience | Int | 3 | Not null | To store the experience of the mechanic |
| m\_opening\_time | Varchar | 20 | Not Null | To store the opening time of the mechanic shop |
| m\_closing\_time | Varchar | 20 | Not Null | To store the closing time of the mechanic shop |

**3) Table Name:** login\_details

**Purpose** : To store the login credentials of the users (Customer/mechanic).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN | TYPE | SIZE | CONSTRAINT | DESCRIPTION |
| email | Varchar | 100 | Primary key | To store the users email |
| Password | Varchar | 100 | Not null | To store the passsword of the users |
| Role | Varchar | 50 | Not null | To store the role of the users |

**4) Table Name:** Booking\_details

**Purpose** : To store the booking\_details of the customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN | TYPE | SIZE | CONSTRAINT | DESCRIPTION |
| booking\_id | Int | - | Primary key | To store the id of the booking |
| c\_email | Varchar | 100 | Foreign key | To store the email of the customer |
| m\_email | Varchar | 100 | Foreign key | To store the email of the mechanic |
| booking\_address | Varchar | 100 | Not null | To store the current address of the customer |
| vehicle\_name | Varchar | 100 | Not null | To store the vehicle of the customer |
| vehicle\_model | Varchar | 100 | Not null | To store the model of the vehicle |
| vehilce\_type | Varchar | 100 | Not null | To store the type of the vehicle of the customer |
| repair\_description | Varchar | 100 | Not null | To store the repair description of the vehicle |
| booking\_Time | Timestamp | - | Not null | To store the booking time |
| booking\_status | Varchar | 20 | Not null | To store the status of the booking |

**5) Table Name:** payment\_details

**Purpose :** To store the payment information of both the customer and mechanic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN | TYPE | SIZE | CONSTRAINT | DESCRIPTION |
| payment\_id | Int | **-** | Primary Key | To uniquely identify the payment |
| from\_email | Varchar | 100 | Foreign key | To store the email of amount sender |
| from\_card\_name | Varchar | 100 | **-** | To store the card holder name of amount sender |
| from\_cv | Int | **-** | **-** | To store the card cvv number of amount sender |
| from\_expiry\_year | Varchar | 10 | **-** | To store the card expiry year of the amount |
| to\_email | Varchar | 100 | **-** | To store the email of the amount receiver |
| to\_card\_name | Varchar | 100 | **-** | To store the card holder name of the amount receiver |
| To\_card\_number | Varchar | 20 | **-** | To store the card number of the amount receiver |
| To\_cv | Int | - | **-** | To store the card cvv number of amount receiver |
| To\_expiry\_year | Varchar | 10 | **-** | To store the card expiry year of the amount receiver |
| Amount | Int | - | **-** | To store the amount transferred |
| payment\_status | Varchar | 50 | **-** | To store the status |
| Payment\_time | Timestamp | - | **-** | To store when the payment has done |

**5. SYSTEM TESTING**

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. The tester may analysts, programmer or a specialist trained for software testing, is actually trying to make the program fail. Analysts know that an effective testing program does not guarantee system reliability. Therefore reliability must be designed into the system.

**Unit Testing:**

In unit testing we have to test the programs making up the system. For this reason unit testing is sometimes called as the Program testing. The software units in a system are modules and routines that are assembled and integrated to perform a specific function.

Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within the module alone. Unit testing can be performed from the bottom up, starting with the lowest level modules and proceeding one at a time. Unit testing is done for each module in Mechanic Tracking System. This ensures that the value we enter match with the data type and within the specified limits.

**Integration Testing:**

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing for conducting tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus in the integration testing step, all the errors are corrected for the next testing steps.

In Mechanic Tracking System each module is integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

**Validation Testing:**

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been recovered and corrected and a final series of a software tests-validation tests begin.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

In validation testing if user wants to enter the numeric value he can only enter the numeric value not the text value. For e.g.: in phone number field user can only enter numeric value to it. The system is user friendly with user guide and messages to explain further procedures. An attempt has been made to perfect the process by incorporating validation at each level.

**6. SYSTEM IMPLEMENTATION**

**6.1 Software Installation Procedure:**

To install the software in your system the following steps should be followed

1. Download the Xampp server from the official website (https://www.apachefriends.org/download.html). Based on your system specifications.
2. Install the Xampp server on your system.
3. Copy the project folder and paste it on C:\xampp\htdocs
4. Open the Chrome web browser.
5. Then type "localhost/project\_foder\_name" to run the software

**7. FUTURE ENCHANCEMENT**

Some of the new features that can be added to the software for the enhancement of the software are

1. The customer can give a review about the service of the mechanic via comment.
2. The mechanic can provide more details about the mechanic shop by adding more images of the mechanic shop.
3. Tracking of the mechanic can be enhanced.
4. The customer can also provide a rating to the mechanic.

**8. CONCLUSION**

The Mechanic Tracking System helps the customer to book the mechanic on the go. It also helps the customers to find more details about the mechanic. This website was designed in such a way that it is user-friendly. Therefore any user can make use of the website in an effective manner. This website helps the customer to avoid tedious work like getting the repaired vehicle to the mechanic shop or searching for a mechanic shop. Thus the application is tested and implemented in such a way that it reaches all types of users.

**9. REFRENCES**

**Referred Sites:**

* https://developer.mozilla.org/en-US/docs/Web/HTML
* https://developer.mozilla.org/en-US/docs/Web/css
* https://www.w3schools.com/js/DEFAULT.asp
* https://www.w3schools.com/jquery/
* https://www.w3schools.com/php/
* https://www.w3schools.com/sql/

**Referred books:**

* Thomas A Powell,"HTML & CSS: The Complete Reference",Fifth Edition, 2010.
* Paul Hudson, "PHP in a Nutshell", Reilly Publications.
* Andy Harris, "PHP 5/MYSQL Programming for the absolute beginner", Thomson Course Technology.