CHAPTER 1

INTRODUCTION

1.1 Introduction To Project

Parking management system is designed for managing the records of the incoming and outgoing vehicles in a parking area.

In now days many public places such as malls, multiplex system, hospitals, offices, market areas there is a crucial problem of vehicle parking. The workers of parking area have to look for vacant slots available for parking they also have to generate bill manually as it consumes much time and wastes a lot of energy.

So this parking system will reduce wastage of time and energy by helping workers of parking area by maintain record of vacant slots in parking area and generating bill digitally workers just have to operate this parking system correctly which consumes less time.

This parking system is capable of generating fair without any mistake it is capable for maintaing record of parked vehicles and will show the details about vacant slot

Parking managers manually count automobiles, which takes time. In this case, a physical examination of the vehicle's condition and handwritten tickets are necessary.

In this instance, a manual technique produces 50% entry errors, which have a major negative impact on the bottom line. For record keeping, manual systems need a lot of time and paper, and the records are also not kept in good condition.

The demand for parking in Indian metros is seeing no signs of stopping. Because land resources are limited, it isn’t possible to conveniently plan parking spaces according to demand. Instead of increasing available parking spaces, an effective technology-based solution must be employed to optimize the use of available spaces.

1.2 Objectives

The major goal of the parking system is to manage the information regarding time, vehicles, parking spaces, patrons, and parking payments. It manages all information regarding time, kind, and parking costs. Since the project was entirely developed on the administrative side, only the administrative is assured access.

The project's goal is to create an application software that will lessen the amount of human effort required to manage parking spaces, vehicle types, and length. It keeps track of all the information regarding parking spaces, clients, and parking payments. Principal goals are:

- Maintain records in short time of period.
- Determines the parking area is full or not.
- Manage the information of duration.
- Calculates parking fare.

1.3 Problem Foundation

Parking managers manually count automobiles, which takes time. In this case, a physical examination of the vehicle's condition and handwritten tickets are necessary.

In this instance, a manual technique produces 50% entry errors, which have a major negative impact on the bottom line. For record keeping, manual systems need a lot of time and paper, and the records are also not kept in good condition.

The demand for parking in Indian metros is seeing no signs of stopping. Because land resources are limited, it isn't possible to conveniently plan parking spaces according to demand. Instead of increasing available parking spaces, an effective technology-based solution must be employed to optimize the use of available spaces.

1.4 Existing System

Purchasing an automobile is no longer regarded as a luxury. Having a car is more of a necessity than a luxury. Individuals are growing their automobile ownership as their financial situation improves. The complexities and tensions of parking grow as a result of this. For parking lot spaces to run well, cooperation and collaboration along with efficient parking management software are essential.
From a simple coffee machine to a massive tractor, everything in today's society is automated. Modern machines have made our lives so much easier that we can't understand how we would function without them. And, as if simplifying work wasn't enough, more

and more attempts are being made toward complete automation of common tasks, which will benefit us on a whole new level.

The parking sector was also introduced with a similar change through automatic parking systems and Airbnb for parking. The automatic parking system has proven to be 100 times more efficient than the traditional parking system because of robust parking management software. The reason for the integration of automatic parking systems through software for parking management is because typical and conventional parking systems, which are currently in use, have a number of flaws.

The traditional parking system's functionality may have been satisfactory in the past, but modern buildings face new challenges that necessitate newer solutions such as Airbnb for parking. Rent Park, a leading parking management software provider, has listed out some of the most common problems with manual parking systems.

- Paper-Based Record System:
- Increased Labor Cost:
- Manual Checks:
- High Dependence on Guard

1.5 Unique Features Of System

• Creating & Changing Issues at ease

• Query Issue List to any depth

• Reporting & Charting in more comprehensive way

• User Accounts to control the access and maintain security

• Multi-level Priorities & Severities.

• Targets & Milestones for guiding the programmers

• Attachments & Additional Comments for more information

• Robust database back-end

• It contain better storage capacity.

• Accuracy in work.

1.6 Project Category

This is an RDBMS based project which is currently using MySQL for all the transaction statements. MySQL is an opensource RDBMS System.

Brief Introduction about RDBSM :

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMS have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s.

Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use.

However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational

Fig 1.1:Rdbms

It have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s.

CHAPTER 2

REQUIRMENT ANALYSIS AND SYSTEM SPECIFICATIONS

2.1 Feasibility Study

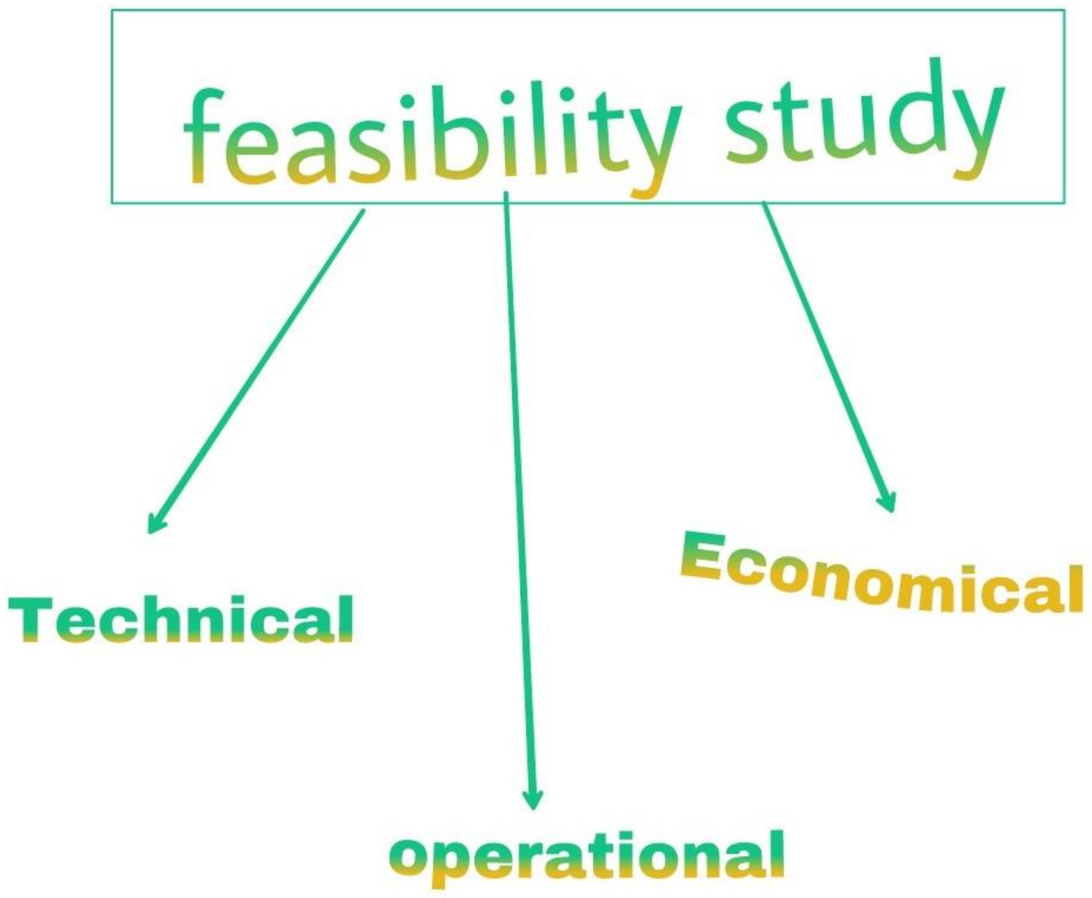


Fig 2.1: Feasibility Study

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

4. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

5. Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including

2.2 SOFTWARE REQUIRMENTS

2.2.1 Hardware Requirements

- RAM – 2/4GB -DDR3/DDR4.
- System – Can run on any system (operating system) windows 7, 8, 8.1,10 & 11 single Language (32/64 BIT)
- I3 / I5 Processor– 2.4GHz CPU.
- Hard drive space – Minimum required- 10 GB internal.

2.2.2 Software Requirements

I. Idle python 3.10.5

II. Pyqt 5

III. Python database connectivity

2.3 Validation

• All the fields such as Car, Parking Space, Parking Area are validated and does not take invalid values

• Each form for Car, Parking, Parking Fees can not accept blank value fields

• Avoiding errors in data

• Controlling amount of input

• Integration of all the modules/forms in the system

• Preparation of the possible test data with all the validation checks.

• Actual testing done manually.

• Recording of all the reproduced errors.

• Prepared the test result scripts after rectification of the errors.

• Functionality of the entire module/forms.

• Validations for user input.

• Checking of the Coding standards to be maintained during coding.

• Testing the module with all the possible test data.

• Testing of the functionality involving all type of calculations etc

2.4 Sdlc Model

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application. SDLC can apply to technical and non-technical systems.
SDLC Methodologies are processes and practices used by software development teams in order to successfully navigate the Software Development Life Cycle (SDLC).

We're not just here to provide you with an exhaustive list of obscure SDLC methodologies. Instead, we're going to set the record straight on SDLC Methodologies. On the web, you'll find articles that will define and explain a long list of SDLC Methodologies and give a brief summary of each so you can "choose" which is best for your project. It seems simple and harmless enough, but this is not how SDLC methodologies are used in the professional software development world.

Most legacy SDLC methodologies aren't even taught in University or bootcamp classrooms. Instead, today's classes teach Agile frameworks like Scrum and Kanban. Despite this, SDLC methodologies have indeed evolved greatly over time, to the point where once-ubiquitous methodologies like Waterfall have become obsolete and irrelevant other than serving as the history that helps us understand the birth of Agile.

Today, the dominant SDLC methodology used by professional software organizations is Agile along with the many Agile frameworks like Scrum and Kanban that extend its principl. To understand the story of SDLC Methodologies, it is best to look at them chronologically. And while there are a number of methodologies that have been tried, all of them except the Agile family has fallen out of use today. You could even say we live in a Post-Agile world.

The First SDLC Methodology - The Waterfall Method - 1970s to 90s

The Waterfall Method - 1970s to 90s

The first SDLC methodology to take hold in software development was the Waterfall method. Associated with Winston W. Royce, It was first introduced in a paper he wrote and used it as an example of what a *bad* methodology looks like: "I believe in this concept, but the implementation described above is risky and invites failure." Despite his warnings and guidance, the Waterfall methodology quickly became the standard and stayed that way for over 20 years.

Waterfall is broken down into phases, and other modern methodologies can even pull from these phases and utilize them, these phases are:

- Requirement Analysis
- Planning
- Architectural Design
- Software Development
- Testing
- Deployment
- Maintenance

According to the Waterfall method, the software development process goes through all the SDLC phases with no overlapping and consists of a single development cycle. According to the fact that it is a linear-sequential life cycle model, any phase in the development process can begin only if the previous one is complete. Teams are large and everyone on the team (business analysts, architects, developers, tests, operations, etc.) all work within their own silos.

After the entire architecture, data structures, and functional designs are ready, the development team starts coding the software. Only after all code is written can integration and validation start. This means that the code is not tested before the Testing phase and only unit tests are executed during development.

Finally, the software finishes testing and is deployed to production and for the first time, where users are able to take it for a test drive. The Waterfall method can take several months or even years to complete, which means that if it doesn’t meet user expectations, changes are extremely slow and expensive. In many cases, defects never get fixed at all.

Likewise, due to the lack of feedback from customers or other stakeholders during the design and development process, it was quite common for Waterfall teams to build unnecessary or under-used features, leading to wasted time, effort, and money.

As technology leaders of the 1990s began realizing that the Waterfall method had a tendency to produce lengthy and costly business outcomes, they started seeking more flexible alternatives.

Alternative Methodologies Come and Golean

Waterfall was showing its age, and it never really worked well, to begin with. As a result, pioneers in software developed novel methodologies aiming to either improve or replace Waterfall.

Methodologies like Prototyping, Iterative, Spiral, V-Shape, came and went, and more modern frameworks like Scrum, XP (Extreme Programming), and Kanban were developed around the same time as the standard we use today, Agile. In fact, a lot of folks that signed the Agile Manifesto were XP creators and users.

Understanding some of these now-outdated models helps us better understand how Waterfall transitioned into Agile:

The Prototyping Model

An outdated methodology that is no longer in active use, it served its purpose as one of the earliest alternatives to Waterfall, dating back to the mid 1970s. The Prototype method revolves around the creation of a low fidelity prototype for the purposes of collecting early feedback from prospective users. From there, prototypes are evolved into final software requirements.

The Iterative Model

The Iterative methodology was an early precursor to Agile. It emphasized iterative and incrementa*l* action. Its earliest reported use was as part of NASA's Project Mercury in the early 1960s.

With the Iterative Model, only the major requirements are known from the beginning. Based on these, the development team creates a quick and cheap first version of the software. Then, as additional requirements are identified, additional iterations of the software are designed and built. Each iteration goes through all the phases of the SDLC and these cycles are repeated until completion. It was common for the team to work on several SDLC phases at the same time.

2.4.1 Why Do We Use It?

If you are thinking why you should use this approach for the project success, then read the below points and know how SDLC could contribute to your project's success:

1. A better plan of SDLC will give a whole picture of the work that is needed to be done throughout the
   Project. This will help in having control over the various phases and activities. The entire project can be managed very efficiently with SDLC project planning.

2. This will add more transparency and visibility to the tasks at hand and will help in the timely completion of the project.

3. This plan when well versed with the team will help in making sure that risk management is done effectively. The project will be kept on track and the timely delivery along with the cost as well as the
   resources management can be done in a precise way.
4. This will help in having all the requirements in the open and will keep everyone in the team on the same page throughout the project.
5. With the help of the SDLC plan, the idea of the project can be brought into action and the functions can be started. It will help in delivering a better user experience, along with better change management
   as well as process development.
6. It is nonetheless to say that it is going to increase the quality of code written and deliver better efficiency in the project. This can be a way to reduce cost as it can have a better approach to solve problems and gather requirements

2.2.2 Requirements

When the proposal is given by the customer and the idea behind is discussed thoroughly, the time to know about the resources, time, and other things needed to deliver that proposal comes into the picture. This phase is very much important as here all the details are discussed about the time and resources and all the resources are accumulated. In this phase, various documents are written which can be referred back whenever needed in the future. This could be considered as the planning stage before the team is put into any action for the work needed to be done.

2.2.3 Analysis

During this phase, the specific details are being gathered and there are various details to the existing or the new prototypes being discussed. Doing this, the developers create SRS with all the details of the software, the hardware needed, and network requirements for the project to be completed. During this, the analysis is one on what kind of system will be used, along with the costs of solutions for the project proposal. You can learn about management and complete this phase by going for PMP online training.

2.2.4 Design

With this stage, we can say that visualization plays a major role. In the design stage, which is a precursor to the actual development stage, the developers foresee the overall application and gather various things including the system interface, different network requirements, user interfaces, and databases required. This is an important fact as with this stage the theoretical phases are going to have tangible phases and the work is going to happen in the project.

2.2.5 Development

The part where the developers are responsible for writing the code as a part of the software development comes under this. The code is based on all the discussions that were part of the previous stages and based on those outlines and specifications, the code is built by the team. There is a team of front-end as well as back-end developers in the team which will work for the better writing of the code and deliver it to the next stage of the cycle. In SDLC project planning, all these things are taken care of with proper planning.

2.2.6 Testing

We all know that writing code is not going to help to develop software. There is rigorous testing involved whenever there is software development. Quality assurance is very much needed whenever we are going to make the development in the project. The bugs along with defects are being raised, fixed, and then retested again. This process continues till the time we have fine working software in the end. The time needed for this phase depends on the developer's efficiency, the complex nature of the project, as well as the end-users requirements. This can be a very short period or could go for longer periods as well.

2.2.7 Deployment

This is one of the best stages as most of the work is completed. The code is written; the software is developed and tested successfully. Now it is time to roll over the new development work to the production. Here the software or the new code is ready for the end-users to use. The released product can be used by the users in the business environment or can be downloaded by them for use.

2.2.8 Maintenance

Even after the software is being rolled to the production environment, it needs to be maintained and effective to be used for a better user experience. The changes should be relevant to the business requirements and this may need time to time maintenance. So this phase will make sure that all the things are working in a fine manner.

2.2.9 Final Words

The management of the projects is one of the crucial tasks and going for SDLC can be very beneficial. Now we have seen that SDLC holds the best outcome for the projects and also SDLC makes sure that the projects are completed on time with maximum efficiency. This is the reason that more and more companies are going for this approach. There are varieties of models which you can choose as per your project requirements. You should make sure that each phase of SDLC is being implemented with its proper function. This will make sure that

CHAPTER 3

SYSTEM DESIGN

3.1 System Design

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Requirement Analysis, Design, Coding, Testing, Deployment, and Maintenance .

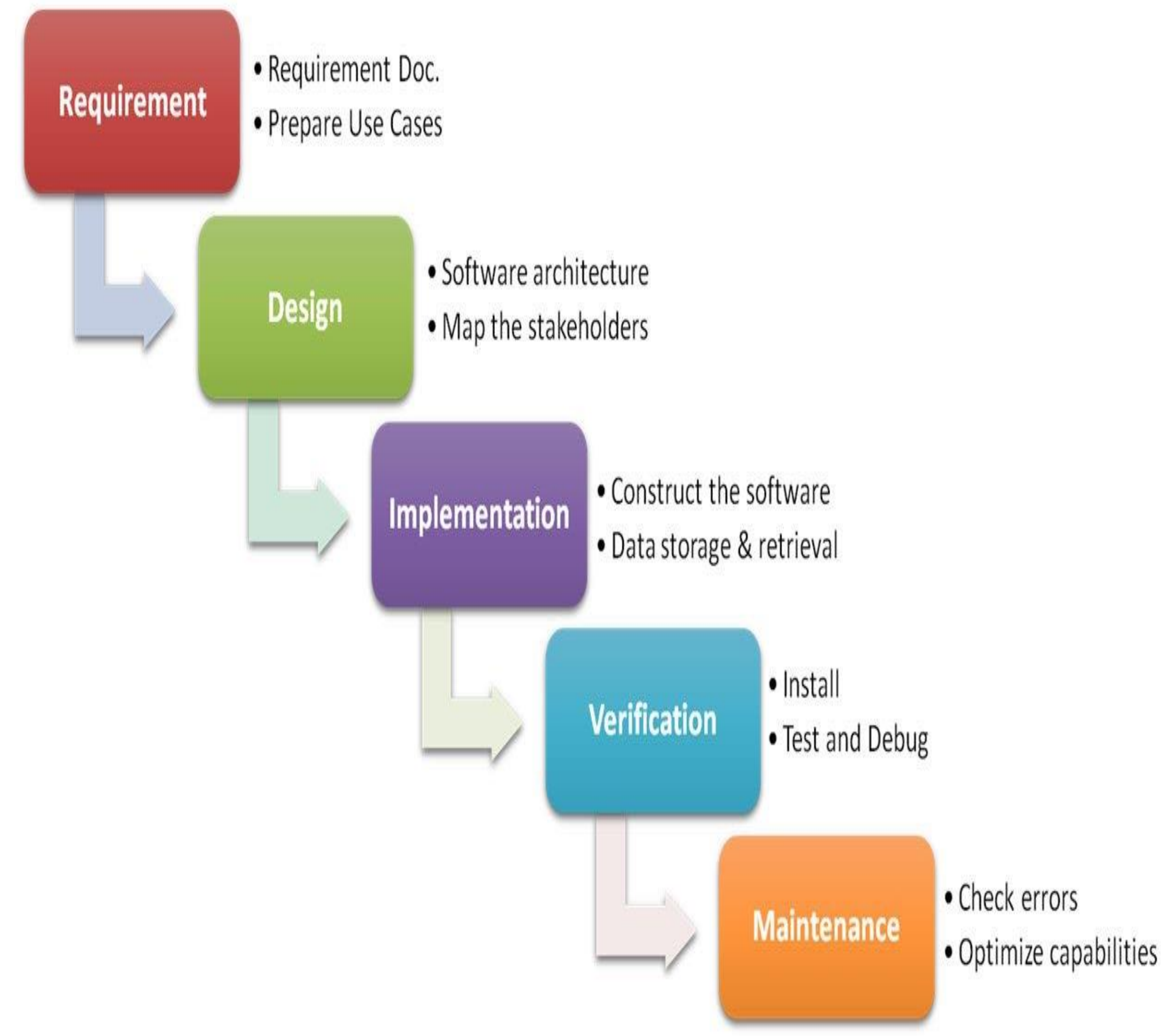


Fig 3.1:System Design

The sequential phases in Waterfall model are:

- **Requirement Gathering And Analysis**

  All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally all requirements documented in a requirement specification doc.

- **System Design**

  Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

- **Implementation**

  With inputs from system design, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

- **Integration and Testing**

  All the units developed in the implementation phase are integrated into a system after testing of each unit. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. Post integration the entire system is tested for any faults and failures.

- **Maintenance**

This phase of "The Waterfall Model" is virtually never ending phase. There are some issues which come up in the client environment. Not all the problems come in picture directly but they arise time to time and needs to be solved. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap. Waterfall model is the earliest SDLC approach that was used for software development.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.

Waterfall Model Pros & Cons:

Advantage

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.

3.2 DFD: Data Flow Diagram

Data Flow Diagrams were first developed by Larry Constantine as a way of expressing system requirements in a graphical form. DFD is also known as bubble chart and has a purpose of clarifying system requirements and identifying major transformations and will become the program in the system design.

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destinations.

Purpose:

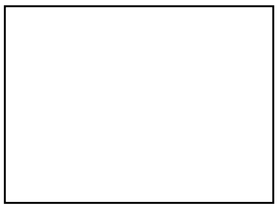
The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers.

The diagrams are:

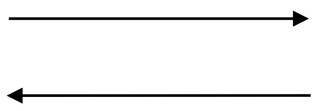
- Graphical, eliminating thousands of words.
- Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it.
- hierarchical, showing systems at any level of detail and
- Allowing user understanding and reviewing.

<u>DFD Symbols Are As Follows:</u>

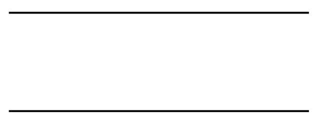
- The External Entity symbol represents sources of data to the system or destinations of data from the system.



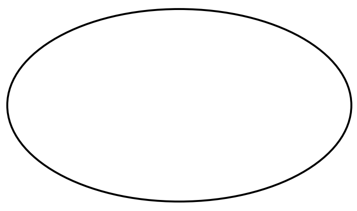
- The Data Flow symbol represents movement of data.



- The Data Store symbol represents data that is not moving (delayed data at rest).



- The Process symbol represents an activity that transforms or manipulates the data.



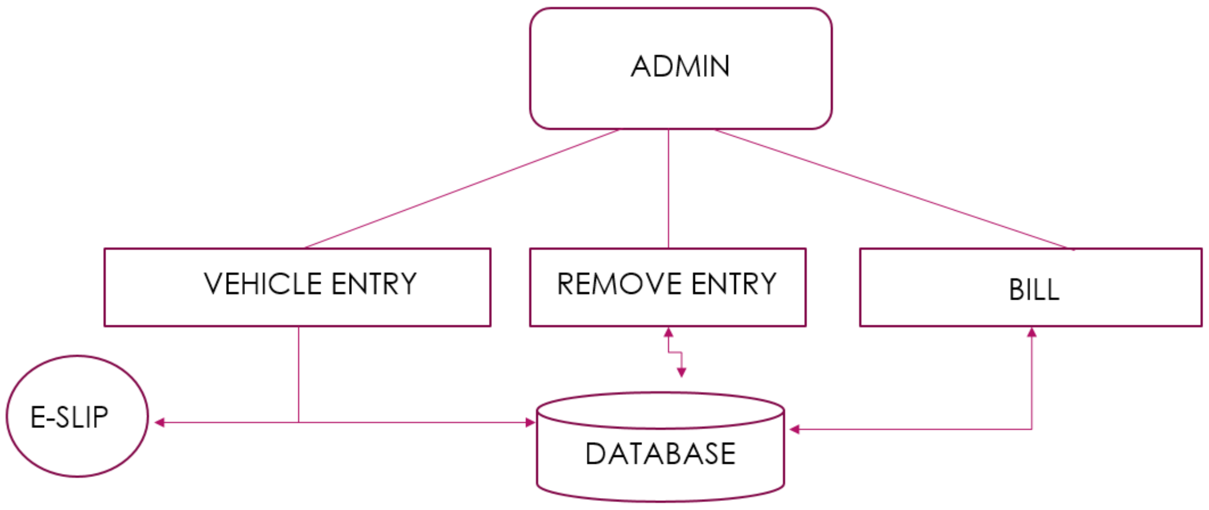


Fig 3.2:Dfd

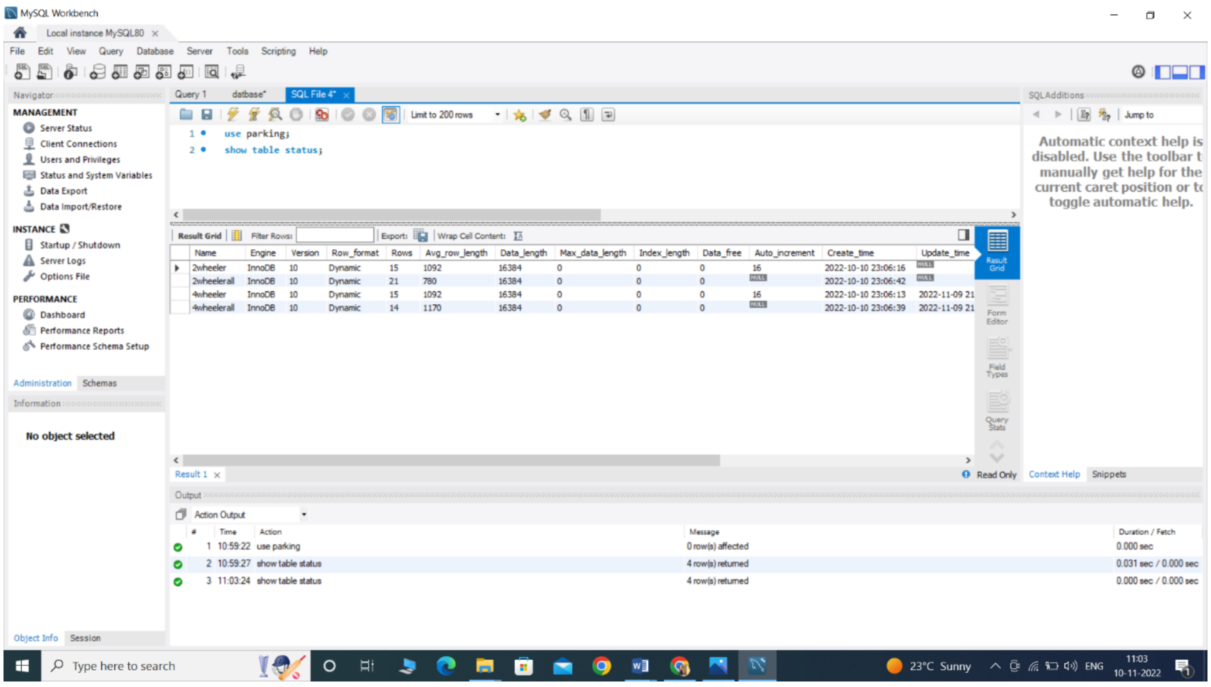
3.3 User Interface Design

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly

3.4 Database Design

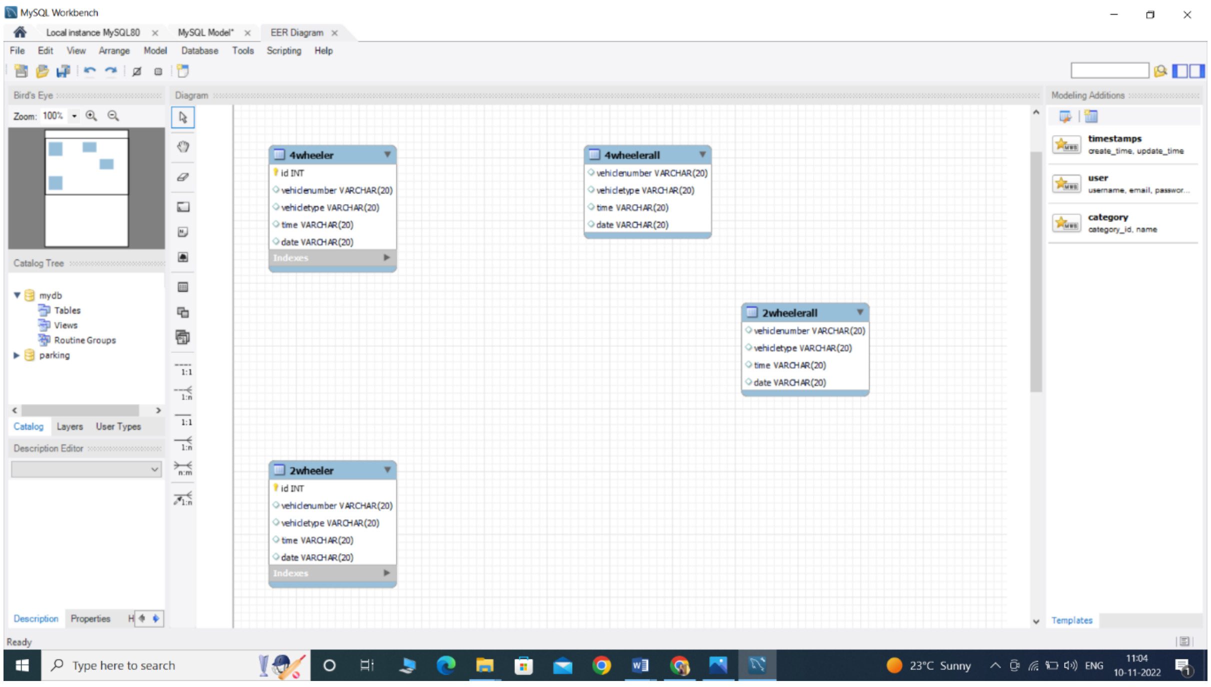


```
use parking;
show table status;
```

	Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length	Data_free	Auto_increment	Create_time	Update_time
▶	2wheeler	InnoDB	10	Dynamic	15	1092	16384	0	0	0	16	2022-10-10 23:06:16	NULL
	2wheelerall	InnoDB	10	Dynamic	21	780	16384	0	0	0	NULL	2022-10-10 23:06:42	NULL
	4wheeler	InnoDB	10	Dynamic	15	1092	16384	0	0	0	16	2022-10-10 23:06:13	2022-11-09 21
	4wheelerall	InnoDB	10	Dynamic	14	1170	16384	0	0	0	NULL	2022-10-10 23:06:39	2022-11-09 21

```
#  Time      Action             Message             Duration / Fetch
1  10:59:22  use parking        0 row(s) affected   0.000 sec
2  10:59:27  show table status  4 row(s) returned   0.031 sec / 0.000 sec
3  11:03:24  show table status  4 row(s) returned   0.000 sec / 0.000 sec
```

Fig 3.3:Database Table



```
4wheeler
id INT
vehiclenumber VARCHAR(20)
vehicletype VARCHAR(20)
time VARCHAR(20)
date VARCHAR(20)

4wheelerall
vehiclenumber VARCHAR(20)
vehicletype VARCHAR(20)
time VARCHAR(20)
date VARCHAR(20)

2wheelerall
vehiclenumber VARCHAR(20)
vehicletype VARCHAR(20)
time VARCHAR(20)
date VARCHAR(20)

2wheeler
id INT
vehiclenumber VARCHAR(20)
vehicletype VARCHAR(20)
time VARCHAR(20)
date VARCHAR(20)
```

Fig 3.4:Er Diagram

CHAPTER 4

IMPLEMENTATION, TESTING & MAINTENANCE

4.1 Introduction To Languages

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed

Fig. 4.1:Python Symbol

History Of Python

➢ Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

➢ Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages.

➢ At the time when he began implementing Python, Guido van Rossum was also reading the published scripts from "Monty Python's Flying Circus" (a BBC comedy series from the seventies, in the unlikely case you didn't know). It occurred to him that he needed a name that was short, unique, and slightly mysterious, so he decided to call the language Python.

➢ Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

➢ Python 1.0 was released on 20 February, 1991. ➢ Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle detecting garbage collector and support for Unicode. With this release the development process was changed and became more transparent and communitybacked.

➢ Python 3.0 (which early in its development was commonly referred to as Python 3000 or py3k), a major, backwards-incompatible release, was released on 3 December 2008 after a long period of testing. Many of its major features have been back ported to the backwards-compatible Python 2.6.x and 2.7.x version series.

➢ In January 2017 Google announced work on a Python 2.7 to go transcompiler, which The Register speculated was in response to Python 2.7's planned end

Fig. 4.2: Guido van Rossum

Scope Of Python :-

1 - Science – Bioinformatics

2 - System Administration - Unix - Web logic - Web sphere

3 - Web Application Development

What Can We Do With Python?

1 - System programming

2 - Graphical User Interface Programming

3 - Internet Scripting

4 - Component Integration

5 - Database Programming

6 - Gaming, Images, XML , Robot and more

Who Uses Python Today?

• Python is being applied in real revenue-generating products by real companies

.• Google makes extensive use of Python in its web search system, and employs Python's creator.

• Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.

• ESRI uses Python as an end-user customization tool for its popular GIS mapping products.

WHY DO PEOPLE USE PYTHON ?

- The YouTube video sharing service is largely written in Python.
- Python is object-oriented o Structure supports such concepts as polymorphism, operation overloading, and multiple inheritance.
- Indentation o Indentation is one of the greatest future in Python.
- It's free (open source) o Downloading and installing Python is free and easy o Source code is easily accessible

- It's powerful o Dynamic typing o Built-in types and tools o Library utilities

  Third party utilities (e.g. Numeric, NumPy, SciPy) o Automatic memory management

- It's portable o Python runs virtually every major platform used today o As long as you have a compatible Python interpreter installed, Python programs will run in exactly the same manner, irrespective of platform.

4.2 TECHNOLOGIES USED FOR IMPLEMENTATION

4.2.1 Python Version 3.10.5

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed

4.2.2 Pyqt 5

Fig. 4.3:Pyqt Symbol

Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development.

PyQt is a Python binding for Qt, which is a set of C++ libraries and development tools providing platform-independent abstractions for graphical user interfaces (GUIs). Qt also provides tools for networking, threads, regular expressions, SQL databases, SVG, OpenGL, XML, and many other powerful features.

Developed by RiverBank Computing Ltd, PyQt’s latest editions are:

1. PyQt5: An edition that’s built against Qt 5.x only
2. PyQt6: An edition that’s built against Qt 6.x only

In this tutorial, you’ll use PyQt5, as this version is the future of the library. From now on, be sure to consider any mention of PyQt as a reference to PyQt5.

Note: If you want to dive deeper into the differences between these two versions of the library, then check out the PyQt5 documentation on the topic.
PyQt5 is based on qt5 . Therefore, it provides classes and tools for GUI creation, XML handling, network communication, regular expressions, threads, SQL databases, web browsing, and other technologies available in Qt. PyQt5 implements binding for many Qt classes in a set of Python modules, which are organized in a top-level Python package called PyQt6. For PyQt5 to work, you need Python 3.6.1 or later.

PyQt6 is compatible with Windows, Unix, Linux, macOS, iOS, and Android. This is an attractive feature if you’re looking for a GUI framework to develop multiplatform applications that have a native look and feel on each platform.

PyQt6 is available under two licenses:

1. The Riverbank Commercial License
2. The General Public License (GPL), version 3

Your PyQt6 license must be compatible with your Qt license. If you use the GPL license, then your code must also use a GPL-compatible license. If you want to use PyQt6 to create commercial applications, then you need a commercial license for your installation.

Note: The Qt Company has developed and currently maintains its own Python binding for the Qt library. The Python library is called Qt for Python and is the official Qt for Python. Its Python package is called PySide.

PyQt and PySide are both built on top of Qt. Their APIs are quite similar because they reflect the Qt API. That's why porting PyQt code to PySide can be as simple as updating some imports. If you learn one of them, then you'll be able to work with the other with minimal effort. If you want to dive deeper into the differences between these two libraries, then you can check out 5. If you need more information about PyQt5 licensing, then check out the license FAQs page on the project's official documentation.

Installing PyQt

You have several options for installing PyQt on your system or development environment. The recommended option is to use to use binary wheels. Wheels are the standard way to install Python packages from the Python package index, PyPI.

In any case, you need to consider that wheels for PyQt6 are only available for Python 3.6.1 and later. There are wheels for Linux, macOS, and Windows (64-bit).

All of these wheels include copies of the corresponding Qt libraries, so you won't need to install them separately.

Another installation option is to build PyQt from source. This can be a bit complicated, so you might want to avoid it if possible. If you really need to build from source, then check out what the library's documentation recommends in those cases.

Alternatively, you have the option of using package managers, such as APT on Linux or Homebrew on macOS, to install PyQt6. In the next few sections, you'll go through some of the options for installing PyQt6 from different sources and on different platforms.

Virtual Environment Installation With Pip

Most of the time, you should create a Python virtual environment to install PyQt6 in an isolated way. To create a virtual environment and install PyQt6 in it, run the following on your command line:

- Windows
- Linux + macOS

```
PS> python -m venv venv
PS> venv\Scripts\activate
```

```
(venv) PS> python -m pip install pyqt6
```

Here, you first create a virtual environment using the venv module from the standard library. Then you activate it, and finally you install PyQt6 in it using pip. Note that you must have Python 3.6.1 or later for the install command to work correctly.

System-Wide Installation With pip

You'll rarely need to install PyQt directly on your system Python environment. If you ever need to do this kind of installation, then run the following command on your command line or in your terminal window without activating any virtual environment:

```
$ python -m pip install pyqt6
```

With this command, you'll install PyQt6 in your system Python environment directly. You can start using the library immediately after the installation finishes. Depending on your operating system, you may need root or administrator privileges for this installation to work.

Even though this is a fast way to install PyQt6 and start using it right away, it's not the recommended approach. The recommended approach is to use a Python virtual environment, as you learned in the previous section.

Platform-Specific Installation

Several Linux distributions include binary packages for PyQt6 in their repositories. If this your case, then you can install the library using the distribution's package manager. On Ubuntu, for example, you can use the following command:

```
$ sudo apt install python3-pyqt5
```

With this command, you'll install PyQt5 and all of its dependencies in your base system, so you can use the library in any of your GUI projects. Note that root privileges are needed, which you invoke here with the sudo command.

If you're a macOS user, then you can install PyQt5 using the Homebrew package manager. To do this, open a terminal and run the following command:

```
$ brew install pyqt5
```

After running this command, you'll have PyQt5 installed on your Homebrew Python environment, and it'll be ready for you to use.

If you use a package manager on Linux or macOS, then there's a chance you won't get the latest version of PyQt6. A pip installation will be better if you want to ensure that you have the latest release.

Creating Your First PyQt Application

Now that you have a working PyQt installation, you're ready to create your first GUI app. You'll create a Hello, World! application with Python and PyQt. Here are the steps that you'll follow:

1. Import QApplication and all the required widgets from PyQt6.QtWidgets.
2. Create an instance of QApplication.
3. Create your application's GUI.
4. Show your application's GUI.
5. Run your application's event loop, or main loop.

You can download the source code for the examples that you'll code in this section by clicking the link below:

To kick things off, start by creating a new file called hello.py in your current working directory:

```
# hello.py

"""Simple Hello, World example with PyQt5."""

import sys

# 1. Import QApplication and all the required widgets
from PyQt5.QtWidgets import QApplication, QLabel, QWidget
```

First, you import sys, which will allow you to handle the application's termination and exit status through the exit() function. Then you import QApplication, QLabel, and QWidget from QtWidgets, which is part of the PyQt5 package. With these imports, you're done with step one.

To complete step two, you just need to create an instance of QApplication. Do this as you would create an instance of any Python class:

```
# hello.py
# ...

# 2. Create an instance of QApplication
app = QApplication([])
```

In this line of code, you create the instance of QApplication. You should create your app instance before you create any GUI object in PyQt.

Internally, the QApplication class deals with command-line arguments. That's why you need to pass in a list of command-line arguments to the class constructor. In this example, you use an empty list because your app won't be handling any command-line arguments.

Note: You'll often find that developers pass sys.argv to the constructor of QApplication. This object contains the list of command-line arguments passed into a Python script. If your application needs to accept command-line arguments, then you should use sys.argv to handle them. Otherwise, you can just use an empty list, like you did in the above example.
Step three involves creating the application's GUI. In this example, your GUI will be based on the QWidget class, which is the base class of all user interface objects in PyQt.

Here's how you can create the app's GUI:

```
# hello.py
# ...

window = QWidget()
window.setWindowTitle("PyQt App")
window.setGeometry(100, 100, 280, 80)
helloMsg = QLabel("<h1>Hello, World!</h1>", parent=window)
helloMsg.move(60, 15)
```

In this code, window is an instance of QWidget, which provides all the features that you'll need to create the application's window, or form. As its names sugges

Note: If you develop internationalized applications, then you've probably seen translated text get cut off mid-sentence. This is likely to happen when the target natural language is more verbose than the original one. Layout managers can help you prevent this common issue by

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.

4.2.3 Tikenter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit,[1] and is Python's de facto standard GUI.[2] Tkinter is included with standard Linux, Microsoft Windows and macOS installs of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Steen Lumholt and Guido van Rossum,[3] then later revised by Fredrik Lundh.

4.2.4 My Sql

Fig. 4.4:My Sql

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

4.3 Testing Techniques And Test Plan

4.3.1 Testing

testing is the process of evaluation a software item to detect differences between given input and expected output. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Also to assess the feature of a system item. Testing assesses the quality of the product. testing is a process that should be done during the development process. In other words testing is a verification and validation process.

Verification

Verification is the process of reviewing the intermediate work products of a software development lifecycle to ensure that we are on track to complete the final result.

Validation

Validation is the process of assessing the completed product to see whether it fits the business requirements.

Testing Techniques

Blackbox Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

It explains the process of giving the input to the system and checking the output, without considering how the system generates the output. It is also known as Behavioral Testing.

Whitebox Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. It is the process of giving the input to the system and checking, how the system processes the input to generate the output. It is mandatory for a tester to have the knowledge of the source code.

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input. Unit testing is done at the developer's site to check whether a particular piece / unit of code is working fine. It tests the unit of the program as a whole.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing. Integration testing is performed when various modules are integrated with each other to form a sub-system or a system. This mostly focuses in the design and construction of the software architecture. This is further classified into Bottom-Up Integration and Top-Down Integration testing.

Bottom-Up Integration Testing: Here the lowest level components are tested first and then the testing of higher level components is done using 'Drivers'. The entire process is repeated till the time all the higher level components are tested.

Top-Down Integration Testing: This is totally opposite to bottom-up approach, as it tests the top level modules and the branch of the modules are tested step by step using 'Stubs', until the related module comes to an end.

System Testing: This testing conducted on a complete, integrated system, to evaluate the system's compliance with the specified requirements. This is done to check if the system meets its functional and non-functional requirements and is also intended to test beyond the bounds defined in the software / hardware requirement specifications. This testing conducted on a complete, integrated system, to evaluate the system's compliance with the specified requirements. This is done to check if the system meets its functional and non-functional requirements and is also intended to test beyond the bounds defined in the software / hardware requirement specifications.

User Acceptance Testing: Acceptance testing is performed to verify that the product is acceptable to the customer and if it's fulfilling the specified requirements of that customer. This testing includes Alpha and Beta testing. Acceptance testing is performed to verify that the product is acceptable to the customer and if it's fulfilling the specified requirements of that customer. This testing includes Alpha and Beta testing.

Alpha Testing: Alpha testing is performed at the developer's site by the customer in a closed environment. This is done after the system testing.

Beta Testing: This is done at the customer's site by the customer in the open environment. The presence of the developer, while performing these tests, is not mandatory. This is considered to be the last step in the software development life cycle as the product is almost ready.

Regression Testing: Regression testing is one of the most important types of testing, which checks whether a small change in any component of the application affects the unchanged components or not. This is done by re-executing the previous versions of the application

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

4.3.2 Test Plan

Making a test plan is the most crucial task of the test management process. According to IEEE 829, follow the following seven steps to prepare a test plan.

- First, analyze product structure and architecture.
- Now design the test strategy.
- Define all the test objectives.

- Define the testing area.
- Define all the useable resources.

Schedule all activities in an

A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.

The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.

The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager. The test plan is prepared by the Test Lead (60%), Test Manager(20%), and by the test engineer(20%).

Types Of Test Plan

There are three types of the test plan

- Master Test Plan
- Phase Test Plan
- Testing Type Specific Test Plans

Master Test Plan

Master Test Plan is a type of test plan that has multiple levels of testing. It includes a complete test strategy.

Phase Test Plan

A phase test plan is a type of test plan that addresses any one phase of the testing strategy. For example, a list of tools, a list of test cases, etc.

Specific Test Plans

Specific test plan designed for major types of testing like security testing, load testing, performance testing, etc. In other words, a specific test plan designed for non-functional testing.

- appropriate manner.
- Determine all the Test Deliverables.

Test Plan Guidelines

- Avoid Overlapping and repetition.
- Avoid Lengthy Paragraph.
- Use lists and tables.
- Update plan.
- Don't use outdated documents.

4. Type of Test Plan

The following are the three types of test plans:

- **Master Test Plan-** In this type of test plan, includes multiple test strategies and has multiple levels of testing.
- **Phase Test Plan-** In this type of test plan, emphasis on any one phase of testing.
- **Specific Test Plan-** In this type of test plan, it is designed for specific types of testing especially non-functional testing.

5. Test Plan Attributes

There is no hard and fast rule of preparing a test plan but it has some standard 15 attributes that companies follow:

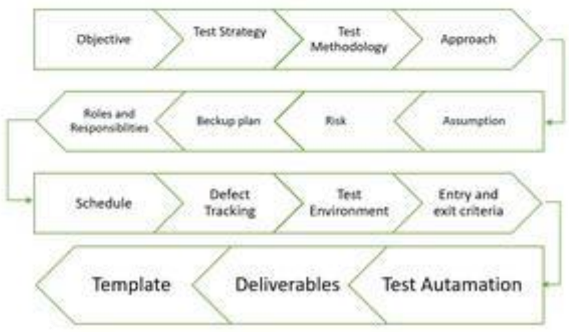


Fig. 4.5:Test Plan

A. Objective: It describes the aim of the test plan, whatever the good process and procedure they are going to follow in order to give quality software to customers. The overall objective of the test is to find as many defects as possible and to make software bug free. The test objective must be broken into components and sub-components. In every component following activities should be performed.

- List all the functionality, performance to be tested.
- Make goals and targets based on the application feature.

B. Test Strategy: It is a crucial document that is to be performed and usually designed by the Test Manager. It helps to determine Test Effort and Test cost. Test strategy helps to determine the features that are going to be tested and the features that will not be tested. The scope can be divided into two parts:

- **In-Scope:** The modules that are to be tested rigorously.
- **Out Scope:** The modules that are not to be tested rigorously.

Example: In an application A, B, C, D features have to be developed, but the B feature has already been designed by other companies. So the development team will purchase B from that company and perform only integrated testing with A, B, C.

C. Testing Methodology: The methods that are going to be used for testing depend on application to application. The testing methodology is decided based on the feature and application requirements.

Since the testing terms are not standard, one should define what kind of testing will be used in the testing methodology. So that everyone can understand it.

D. Approach: The approach of testing different software is different. It deals with the flow of applications for future references. It has two aspects:

- **High-Level Scenarios:** For testing critical features high-level scenarios are written. For Example, login to a website, booking from a website.
- **The Flow Graph:** It is used when one wants to make benefits such as converging and merging easy.

CHAPTER 5

RESULT AND DISCUSSIONS

5.1 User Interface Representation

User Interface Design is concerned with the dialogue between a user and thecomputer. It is concerned with everything from starting the system or logging into thesystem to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various quidelines for User Interface Design:

1. The system user should always be aware of what to do next.

2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.

3. Message, instructions or information should be displayed long enough to allowthe system user to read them.

4. Use display attributes sparingly.

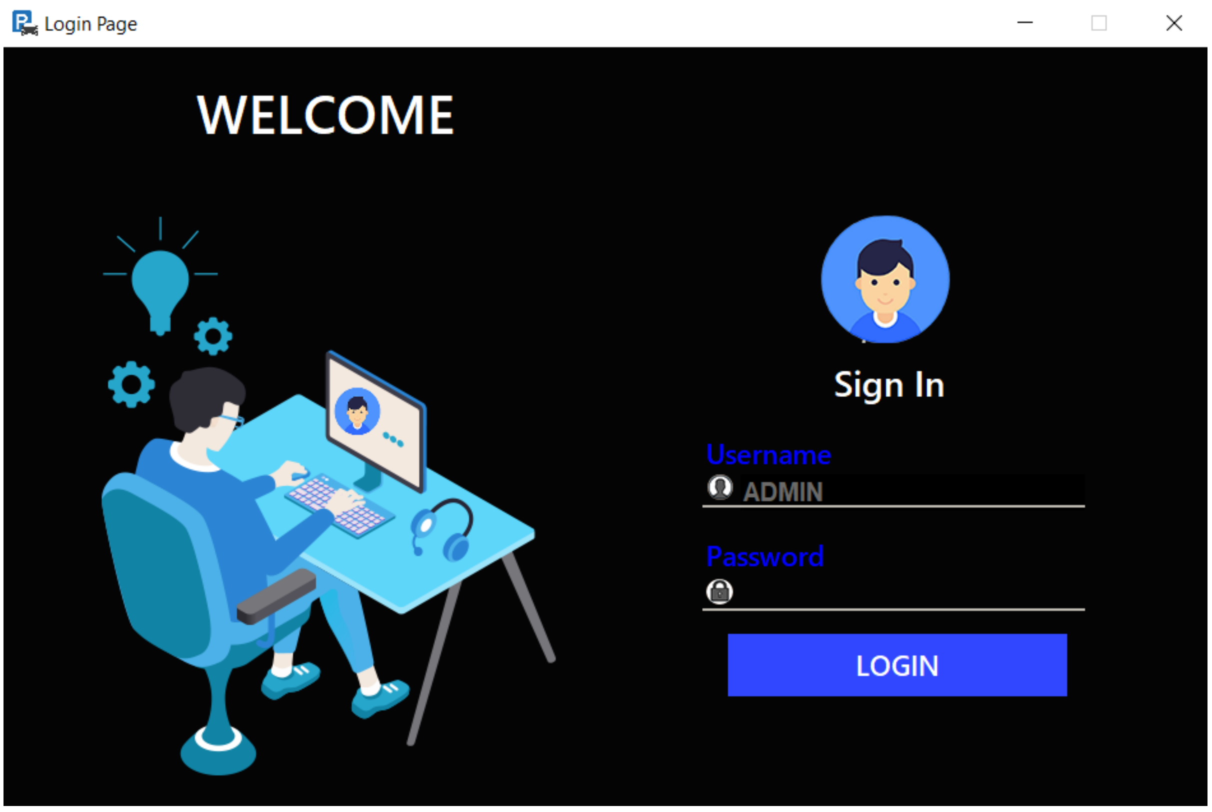


Fig 5.1:Snapshot of Project

5.2 Snapshots of System

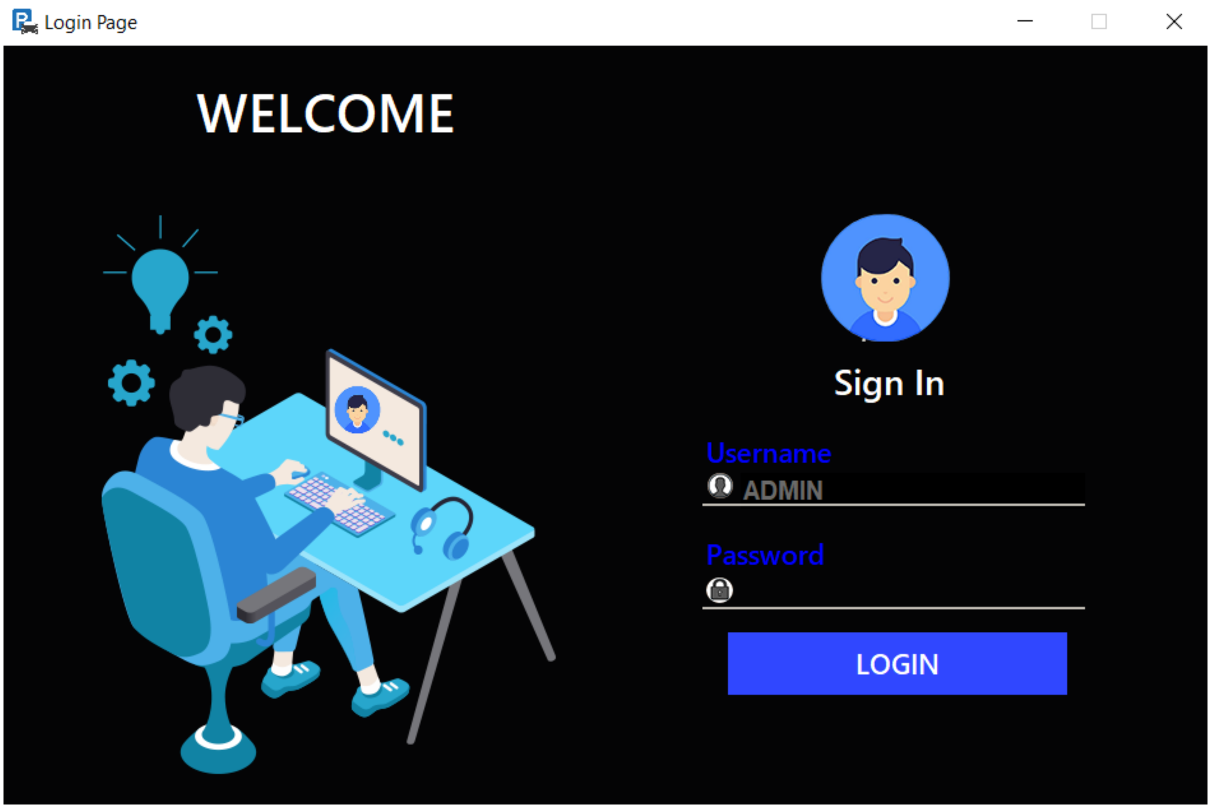


Fig 5.2:Snapshot of Project

Logging in is usually used to enter a specific page, website or application, which trespassers cannot see. Once the user is logged in, the login token may be used to track what actions the user has taken while connected to the site. Logging out may be performed explicitly by the user taking some actions, such as entering the appropriate command or clicking a website link label as such. It can also be done implicitly, such as by the user powering off his or her workstation, closing a web browser window, leaving a website, or not refreshing a website within a defined period.

A login page may have a return URL parameter, which specifies where to redirect back after logging in or out. For example, it is returnto= on this site.

In the case of websites that use cookies to track sessions, when the user logs out, session-only cookies from that site will usually be deleted from the user's computer. In addition, the server invalidates any associations with the session, thereby making any session-handle in the user's cookie store useless. This feature comes in handy if the user is using a public computer or a computer that is using a public wireless connection. As a security precaution, one should not rely on implicit means of logging out of a system, especially not on a public computer; instead, one should explicitly log out and wait for the confirmation that this request has taken place.

Login page is important in this system as no one else should have permission to updtae or remove vehicle entry from databse except admin. That ‘s why first window of this system is login page after entering correct password user will enter to main screen.

5.2.1 Main Menu

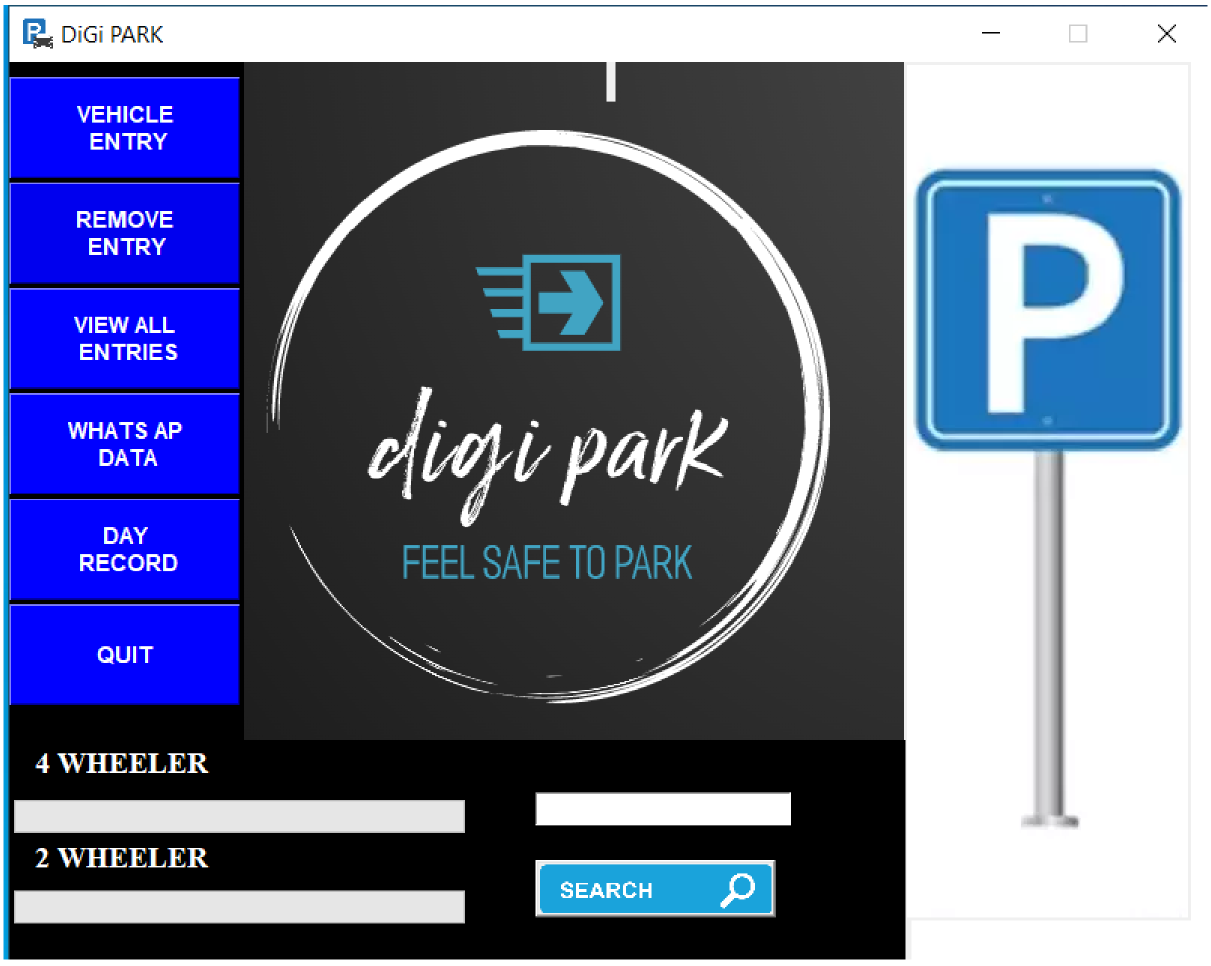


Fig 5.3:Snapshot of Project

A user chooses an option from a menu by using an input device. Some input methods require linear navigation: the user must move a cursor or otherwise pass from one menu item to another until reaching the selection. On a computer terminal, a reverse video bar may serve as the cursor.

Some of the input devices used in menu interfaces are touchscreens, keyboards, mice, remote controls, and microphones. In a voice-activated system, such as interactive voice response, a microphone sends a recording of the user's voice to a speech recognition system, which translates it to a command.In this system main menu is the main window as it navigates user to other windows.

5.2.2 Vehicle Entry

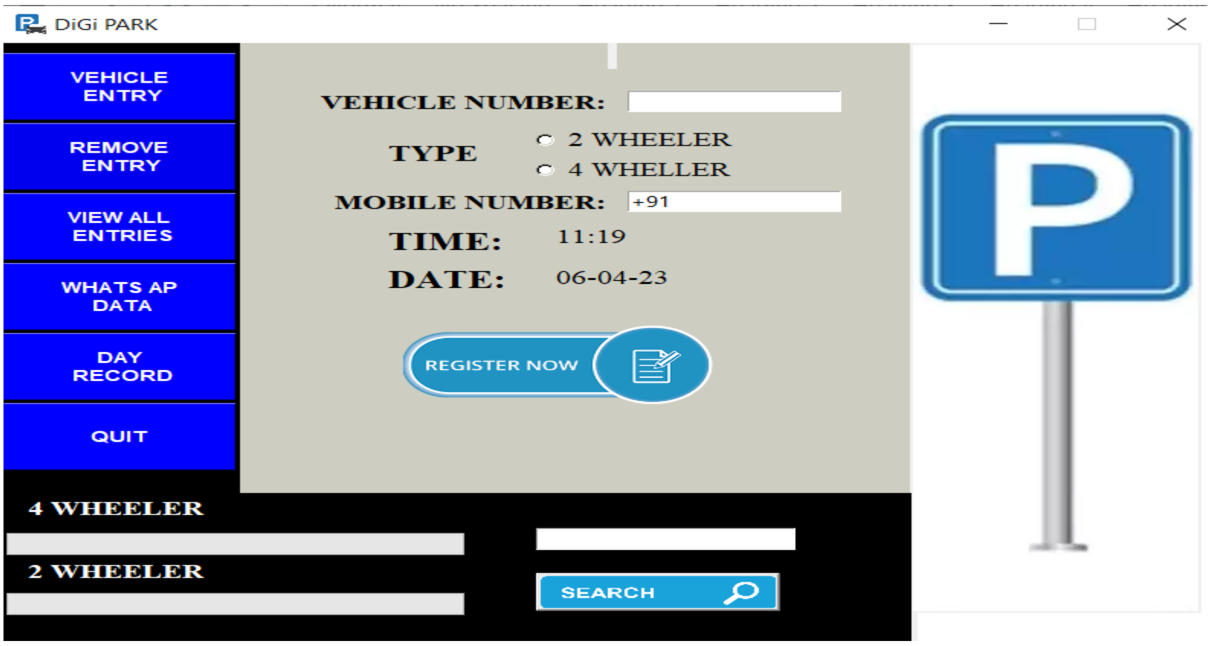


Fig 5.4:Snapshot of Project

5.2.3 Generate Bil

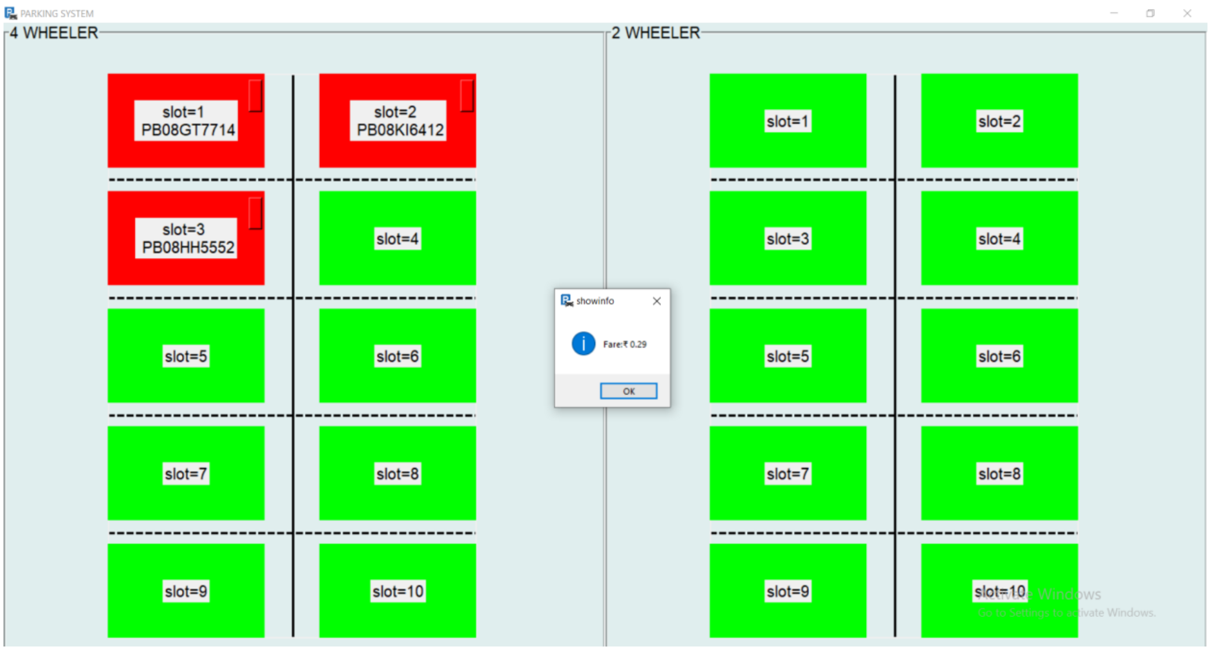


Fig 5.5:Snapshot of Project

5.2.4 Remove Entry

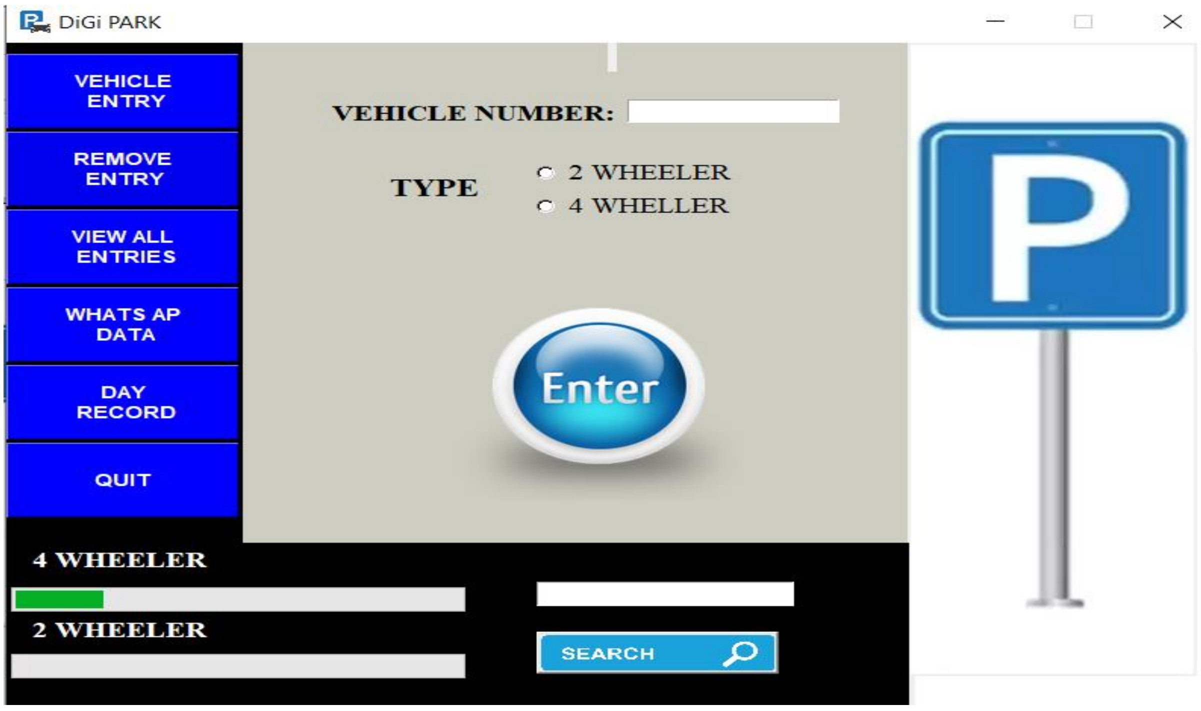


Fig 5.6:Snapshot of Project

5.2.4 Slots

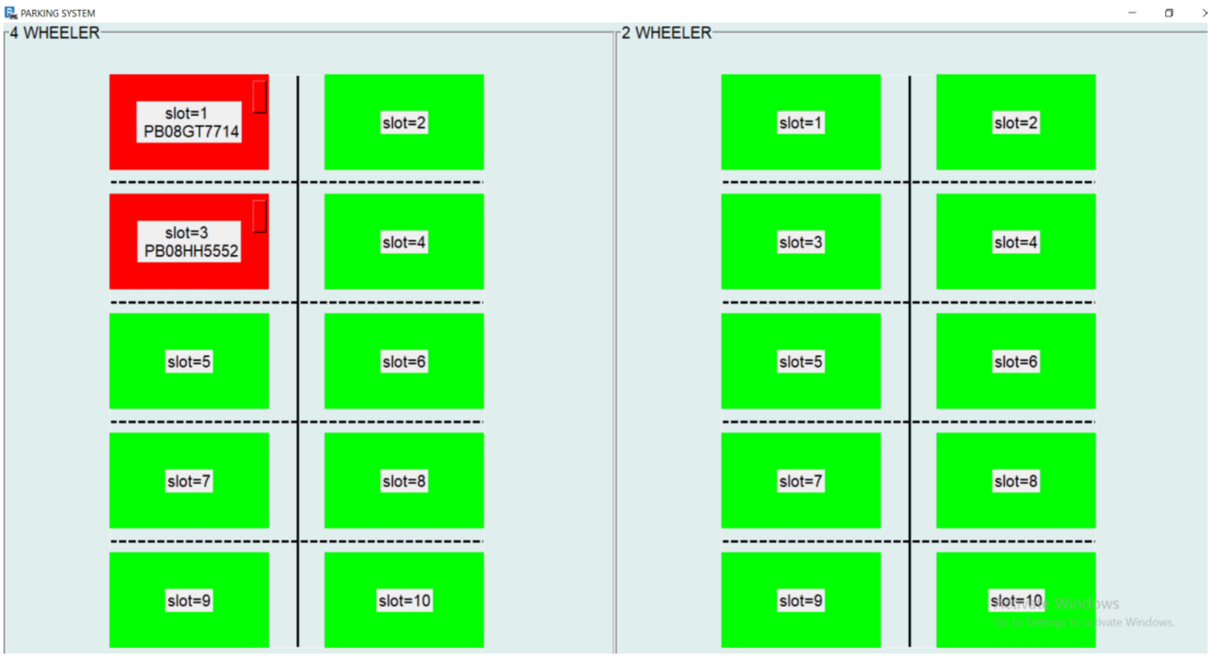
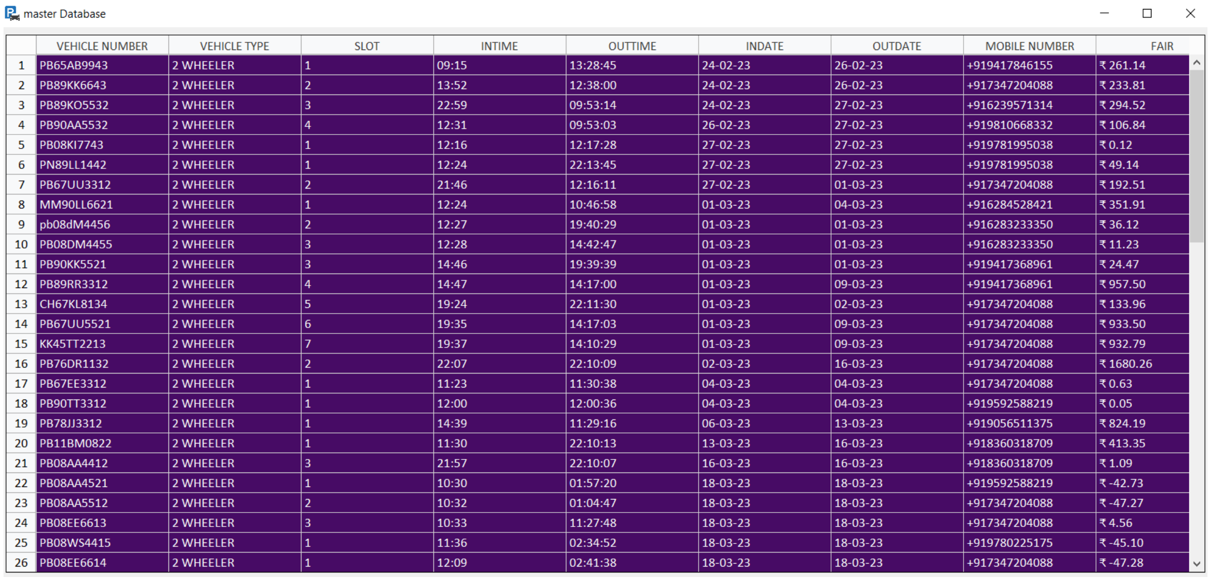


Fig 5.7:Snapshot of Project

5.2.5 View All Entries

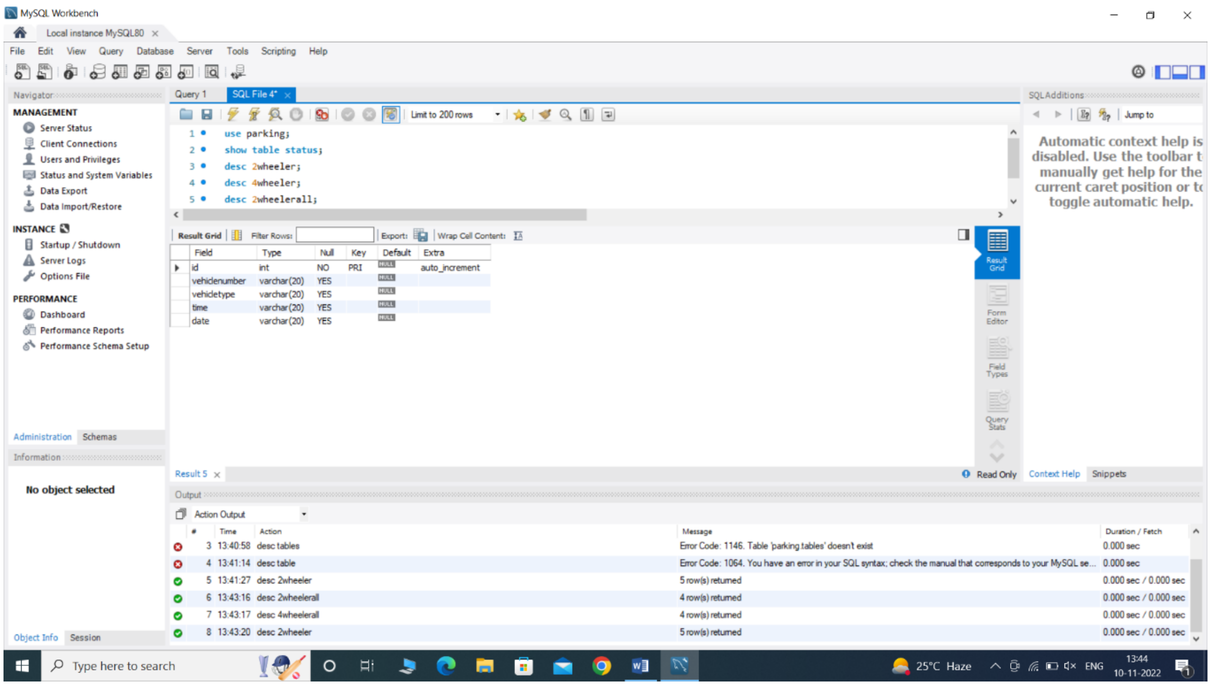


master Database

	VEHICLE NUMBER	VEHICLE TYPE	SLOT	INTIME	OUTTIME	INDATE	OUTDATE	MOBILE NUMBER	FAIR
1	PB65AB9943	2 WHEELER	1	09:15	13:28:45	24-02-23	26-02-23	+919417846155	₹ 261.14
2	PB89KK6643	2 WHEELER	2	13:52	12:38:00	24-02-23	26-02-23	+917347204088	₹ 233.81
3	PB89KO5532	2 WHEELER	3	22:59	09:53:14	24-02-23	27-02-23	+916239571314	₹ 294.52
4	PB90AA5532	2 WHEELER	4	12:31	09:53:03	26-02-23	27-02-23	+919810668332	₹ 106.84
5	PB08KI7743	2 WHEELER	1	12:16	12:17:28	27-02-23	27-02-23	+919781995038	₹ 0.12
6	PN89LL1442	2 WHEELER	1	12:24	22:13:45	27-02-23	27-02-23	+919781995038	₹ 49.14
7	PB67UU3312	2 WHEELER	2	21:46	12:16:11	27-02-23	01-03-23	+917347204088	₹ 192.51
8	MM90LL6621	2 WHEELER	1	12:24	10:46:58	01-03-23	04-03-23	+916284528421	₹ 351.91
9	pb08dM4456	2 WHEELER	2	12:27	19:40:29	01-03-23	01-03-23	+916283233350	₹ 36.12
10	PB08DM4455	2 WHEELER	3	12:28	14:42:47	01-03-23	01-03-23	+916283233350	₹ 11.23
11	PB90KK5521	2 WHEELER	3	14:46	19:39:39	01-03-23	01-03-23	+919417368961	₹ 24.47
12	PB89RR3312	2 WHEELER	4	14:47	14:17:00	01-03-23	09-03-23	+919417368961	₹ 957.50
13	CH67KL8134	2 WHEELER	5	19:24	22:11:30	01-03-23	02-03-23	+917347204088	₹ 133.96
14	PB67UU5521	2 WHEELER	6	19:35	14:17:03	01-03-23	09-03-23	+917347204088	₹ 933.50
15	KK45TT2213	2 WHEELER	7	19:37	14:10:29	01-03-23	09-03-23	+917347204088	₹ 932.79
16	PB76DR1132	2 WHEELER	2	22:07	22:10:09	02-03-23	16-03-23	+917347204088	₹ 1680.26
17	PB67EE3312	2 WHEELER	1	11:23	11:30:38	04-03-23	04-03-23	+917347204088	₹ 0.63
18	PB90TT3312	2 WHEELER	1	12:00	12:00:36	04-03-23	04-03-23	+919592588219	₹ 0.05
19	PB78JJ3312	2 WHEELER	1	14:39	11:29:16	06-03-23	13-03-23	+919056511375	₹ 824.19
20	PB11BM0822	2 WHEELER	1	11:30	22:10:13	13-03-23	16-03-23	+918360318709	₹ 413.35
21	PB08AA4412	2 WHEELER	3	21:57	22:10:07	16-03-23	16-03-23	+918360318709	₹ 1.09
22	PB08AA4521	2 WHEELER	1	10:30	01:57:20	18-03-23	18-03-23	+919592588219	₹ -42.73
23	PB08AA5512	2 WHEELER	2	10:32	01:04:47	18-03-23	18-03-23	+917347204088	₹ -47.27
24	PB08EE6613	2 WHEELER	3	10:33	11:27:48	18-03-23	18-03-23	+917347204088	₹ 4.56
25	PB08WS4415	2 WHEELER	1	11:36	02:34:52	18-03-23	18-03-23	+919780225175	₹ -45.10
26	PB08EE6614	2 WHEELER	1	12:09	02:41:38	18-03-23	18-03-23	+917347204088	₹ -47.28

Fig 5.8:Snapshot of Project

5.3 Snapshots Of Database Table



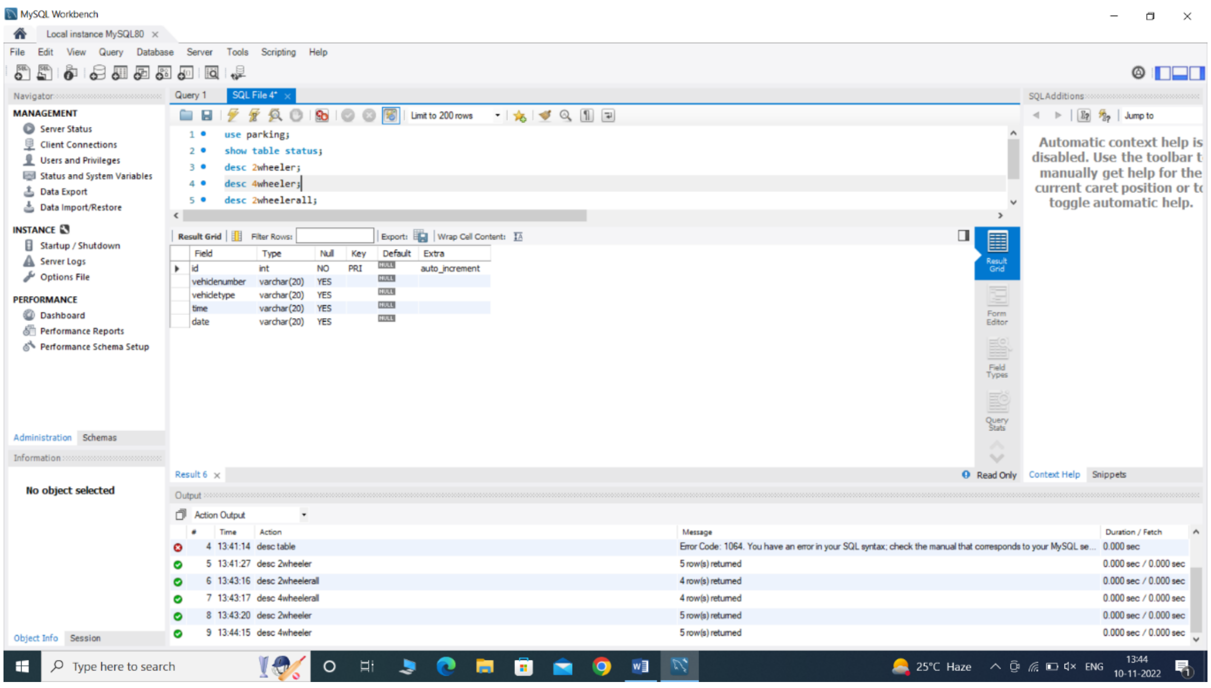
```
use parking;
show table status;
desc 2wheeler;
desc 4wheeler;
desc 2wheelerall;
```

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	vehiclenumber	varchar(20)	YES		NULL	
	vehicletype	varchar(20)	YES		NULL	
	time	varchar(20)	YES		NULL	
	date	varchar(20)	YES		NULL	

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

	#	Time	Action	Message	Duration / Fetch
✗	3	13:40:58	desc tables	Error Code: 1146. Table 'parking.tables' doesn't exist	0.000 sec
✗	4	13:41:14	desc table	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
✓	5	13:41:27	desc 2wheeler	5 row(s) returned	0.000 sec / 0.000 sec
✓	6	13:43:16	desc 2wheelerall	4 row(s) returned	0.000 sec / 0.000 sec
✓	7	13:43:17	desc 4wheelerall	4 row(s) returned	0.000 sec / 0.000 sec
✓	8	13:43:20	desc 2wheeler	5 row(s) returned	0.000 sec / 0.000 sec

Fig 5.9:Snapshot of Database Table



```
use parking;
show table status;
desc 2wheeler;
desc 4wheeler;
desc 2wheelerall;
```

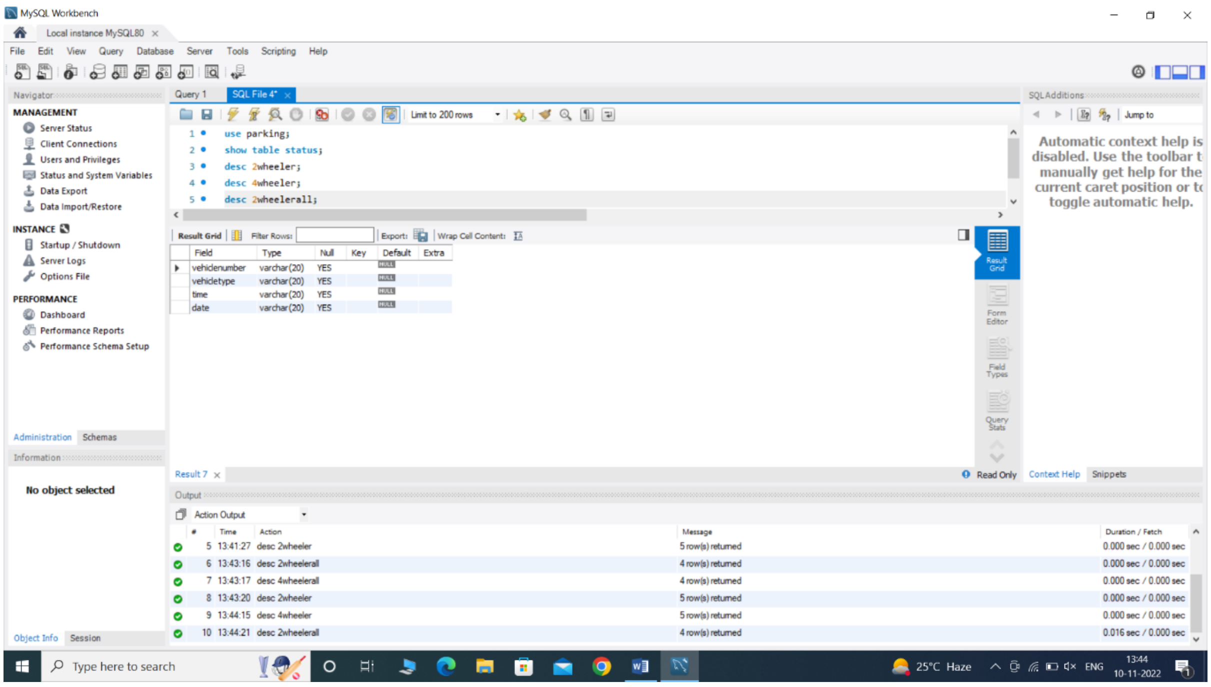
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
vehiclenumber	varchar(20)	YES		NULL	
vehicletype	varchar(20)	YES		NULL	
time	varchar(20)	YES		NULL	
date	varchar(20)	YES		NULL	

Result 6

Output — Action Output

#	Time	Action	Message	Duration / Fetch
4	13:41:14	desc table	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
5	13:41:27	desc 2wheeler	5 row(s) returned	0.000 sec / 0.000 sec
6	13:43:16	desc 2wheelerall	4 row(s) returned	0.000 sec / 0.000 sec
7	13:43:17	desc 4wheelerall	4 row(s) returned	0.000 sec / 0.000 sec
8	13:43:20	desc 2wheeler	5 row(s) returned	0.000 sec / 0.000 sec
9	13:44:15	desc 4wheeler	5 row(s) returned	0.000 sec / 0.000 sec

Fig 5.11:Snapshot of Database Table



```
use parking;
show table status;
desc 2wheeler;
desc 4wheeler;
desc 2wheelerall;
```

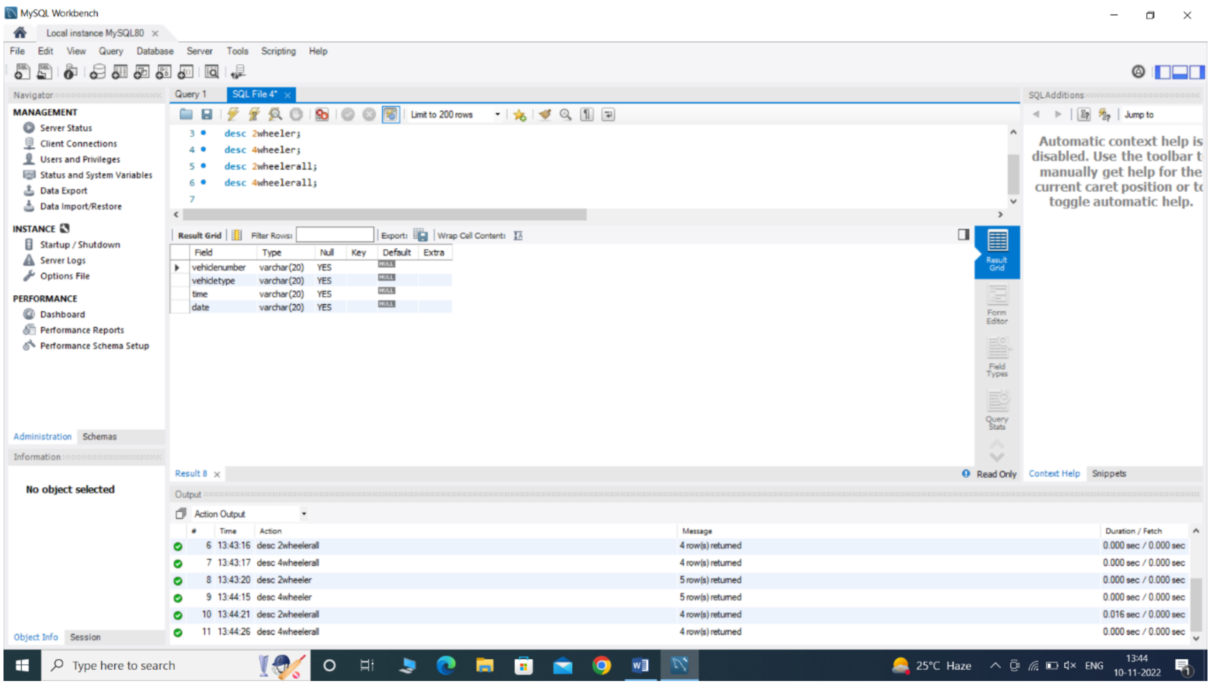
Field	Type	Null	Key	Default	Extra
vehiclenumber	varchar(20)	YES		NULL	
vehicletype	varchar(20)	YES		NULL	
time	varchar(20)	YES		NULL	
date	varchar(20)	YES		NULL	

Result 7

Output — Action Output

#	Time	Action	Message	Duration / Fetch
5	13:41:27	desc 2wheeler	5 row(s) returned	0.000 sec / 0.000 sec
6	13:43:16	desc 2wheelerall	4 row(s) returned	0.000 sec / 0.000 sec
7	13:43:17	desc 4wheelerall	4 row(s) returned	0.000 sec / 0.000 sec
8	13:43:20	desc 2wheeler	5 row(s) returned	0.000 sec / 0.000 sec
9	13:44:15	desc 4wheeler	5 row(s) returned	0.000 sec / 0.000 sec
10	13:44:21	desc 2wheelerall	4 row(s) returned	0.016 sec / 0.000 sec

Fig 5.12:Snapshot of Database Table



```
desc 2wheeler;
desc 4wheeler;
desc 2wheelerall;
desc 4wheelerall;
```

Field	Type	Null	Key	Default	Extra
vehiclenumber	varchar(20)	YES		NULL	
vehicletype	varchar(20)	YES		NULL	
time	varchar(20)	YES		NULL	
date	varchar(20)	YES		NULL	

```
#   Time      Action             Message             Duration / Fetch
6   13:43:16  desc 2wheelerall   4 row(s) returned   0.000 sec / 0.000 sec
7   13:43:17  desc 4wheelerall   4 row(s) returned   0.000 sec / 0.000 sec
8   13:43:20  desc 2wheeler      5 row(s) returned   0.000 sec / 0.000 sec
9   13:44:15  desc 4wheeler      5 row(s) returned   0.000 sec / 0.000 sec
10  13:44:21  desc 2wheelerall   4 row(s) returned   0.016 sec / 0.000 sec
11  13:44:26  desc 4wheelerall   4 row(s) returned   0.000 sec / 0.000 sec
```

Fig 5.13:Snapshot of Database Table

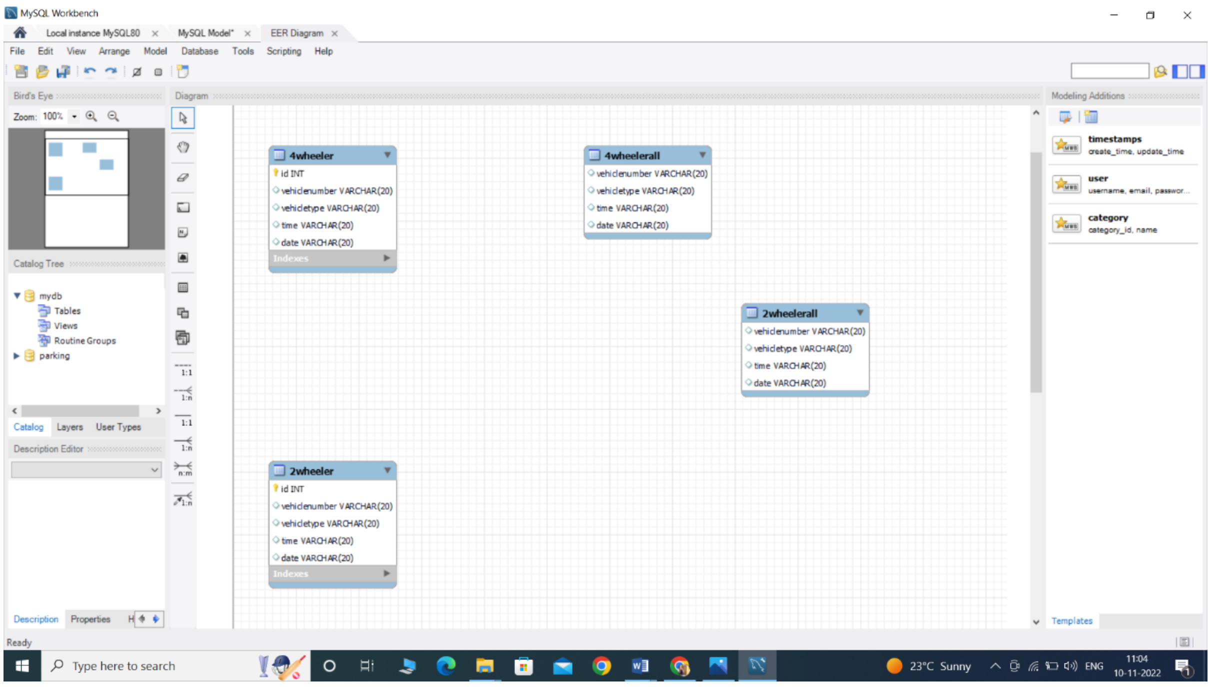


Fig 5.13: Er Diagram

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

My idea incorporated all the parts of the software system. When integrated properly, my initial software design is a viable way to implement this type of system. The floor interface guiding multiple cars appropriately. The other interface effectively tracked parked vehicles and moving traffic within the garage.

There were a number of challenges I had overcame in the design and testing of the Development of Smart Parking System. The first was the actual setup connection between VB.NET interfaces and Microsoft SQL Server database of the model parking garage. The challenge I faced involved connection between Microsoft SQL Server database and the VB.NET program. My original plan was to send the parking spot information in different 57 interfaces from Microsoft SQL Server database to the VB.NET interfaces. I wanted to send the status of each parking spot (available or unavailable) as a separate interface. I was able to establish consistent and reliable networking results. The next step for the Development of Smart Parking System is to sell the concept to a leisure mall that would be willing to try it out on a small scale area. A leisure mall like Suria KLCC would be a perfect test leisure mall for the Development of Smart Parking System. My database can be easily modified for Suria KLCC. I would then have to get a navigation system manufacturer to incorporate the software and interface into their existing system. The, the Development of Smart Parking System will be on its way to becoming a solution to the parking schema

Proper planning needs to be done carefully and suspiciously in order to make sure the business that has been planned will be a profitable and long lived business. The opportunity to develop and implement a well-defined business strategy is very valuable to the business owner, their customer and as well as the community.

6.2 Future Scope

There are a number of improvements and modifications that can be made to my design to increase real world practicality and functionality. The symbols used need to have the ability to differentiate between pedestrians and cars. My Development of Smart Parking System garage is so simple that adding more interfaces was unnecessary. With only twelve parking spots in my system, more than five cars moving through the garage at one time would simply produce congestion. Other improvements would be additional guidance devices such as a return interface with audio or light guidance. Upon the return, either audio playback or a series of LEDs would guide the driver to their vehicle. This was excluded from my original design system due to time constraints. Such an interface could also be integrated with the elevator so that the driver might be taken directly to the appropriate floor. 58 Any types of sensors that could sense the car from about six feet away could be implemented. Six feet is a more realistic distance between a parked car and its parking meter. Also, more powerful transmitters and receivers to increase the range of operation could also be implemented. Ideally, the parking attendance at his or her station should be able to receive data up to several miles away from the transmitter at the parking space. Finally, the concept of the open space locator could be expanded to include multiple parking meters from multiple parking lots. This system can clearly be applied to a number of different parking environments: large parking lots, underground parking facilities, stadium parking, airport parking, etc. The system can be easily networked to monitor and control any number of different parking facilities. Furthermore, this information can be made available online for convenient access. Not only is this system useful and applicable to parking facilities, it can be easily modified to accommodate the needs of hospitals, resorts or conference centers. A similar guidance system could be used to direct patients and doctors to different wings, hotel guests to various attractions, or clients to the proper meeting room. The Development of Smart Parking system is beneficial wherever frequent searching processes are involved. For example, it can be used anywhere from locating specific products within large industrial warehouses to finding a book at your local library.

REFERENCES

- https://www.ibm.com/in-en/topics/software-testing
- https://www.guru99.com/what-everybody-ought-to-know-aboutplaning.html
- https://en.wikipedia.org/wiki/Python_(programming_language)
- https://www.javatpoint.com/pyqt-library-in-python
- https://www.w3schools.com/MySQL/default.asp
- https://www.geeksforgeeks.org/system-design-tutorial/
- https://www.investopedia.com/terms/f/feasibility-study.asp
- https://www.merriam-webster.com/dictionary/analysis
- https://www.lucidchart.com/pages/data-flow-diagram