CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

“IDEAL BAJAJ” is a web application which is meant for users who are looking for online bike booking and those who want to manage bike services. The proposed system includes three users they are administrator, customer and the manager. Registered customers can login to the site and place their orders and write complaints. Manager will get up to date notifications of each order and they can contact with the customers and get orders from them. The administrator has the central control over the whole system. Administrators add managers, bikes and user management.

1.2 PROJECT SPECIFICATION

The proposed system is a web application in which user can Book bike. It will also provide bike service details, bike details etc.

The system includes 3 modules. They are:

1.Admin module

Admin must have a login into this system. He has the overall control of the system.
Admin can add bikes, managers and gallery. Admin can view customers and feedback.

2.Manager module

Manager can approve or reject bookings and he can view the customer complaints and respond to them.

3.Customer module

Customer can book bike and do secure online payment and write their complaints as well as feedback.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

Systems development can generally be thought of as having two major components: Systems analysis and Systems design. System design is the process of planning a new system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective. System analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system. This is the job of the systems analyst.

Analysis specifies what the system should do. Design states how to accomplish the objective. Notice that each of the processes mentioned involves people. Managers and employees have good ideas about what works and what does not, about what flows smoothly and what causes problems, about where change is needed and where it is not, and especially about where change will be accepted and where it will not. Despite technology, people are still the keys that make the organizations work. Thus, communicating and dealing with people are very important parts of the systems analyst’s job.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

In the existing system all bike showroom activities done manually, It is a time consuming process. The main demerit of existing system is that we have to travel all available showroom for its enquiry.

2.3 DRAWBACKS OF EXISTING SYSTEM

1. It is time consuming process.
2. More manual hours need to generate required reports.
3. There is no possibility to track the approvals of requests.
4. No co-ordination between various departments.

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the products and users. It allows customers to make their orders and do their transactions by using online payment method Users of this proposed system are admin, customer and manager. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Online shopping advancements have been so drastic that it has evolved to be a part of our life. Today customer doesn’t drive down to some shop for buying a product but preferably check over the internet for price, offers, reviews and order online. Instead of having so many section and so many people for each & every section, it is enough to have only one person to do all works in a computer. So the concern save space, time, money and the like. In the long run, this system minimizes the expenses of the concern to the maximum extents; each and every work can be done perfectly and neatly with the proposed system. Any information can be obtained at any time easily, accurately and when necessary. It can be changed, modified or updated. So many facilities like the above are available in the proposed system. All the works can be done within a fraction of seconds. Thus it fulfills the objectives of the system.

The current systems working procedure consist of the user visiting the IDEAL BAJAJ web application by himself and visiting the advertisements and offers then book the desired Bajaj bike.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- **Entire working of the existing system is automated**

The proposed system is a fully automated system allowing the user to book bajaj through the website and write feedback online and get alerts about booking status. The user doesn't need to do any manual work.

- **Ensure data accuracy**

The proposed system eliminates manual errors while documenting the past bookings and delivery details.

- **Data security**

The product will avoid the burden of hard copy storage since documents are accepted as digital copies and are stored in database. Data security is ensured and each data can be stored and retrieved uniquely.

- **Better service**

Since each user requests are processed uniquely the overall service delivery of the system is improved. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, IDEAL BAJAJ was divided according to the system used, its development cost and cost for hosting the project According to all the calculations the project was developed in a low cost. As it is completely developed using open source software the only cost was spent for hosting the project which is affordable.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB.

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

IDEAL BAJAJ, GUI is simple so that users can easily use it. IDEAL BAJAJ is simple enough so that no training is needed.

3.2 SYTEM SPECIFICATIONS

3.2.1 Hardware Specification

Processor	-	Intel Core i3
RAM	-	4GB
Hard Disk	-	1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 8 and above
Technologies Used	-	HTML5,CSS,PHP,JQuery,JS,AJAX

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language.PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledford in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, "SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- The MySQL Database Server is very fast, reliable, scalable, and easy to use.

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

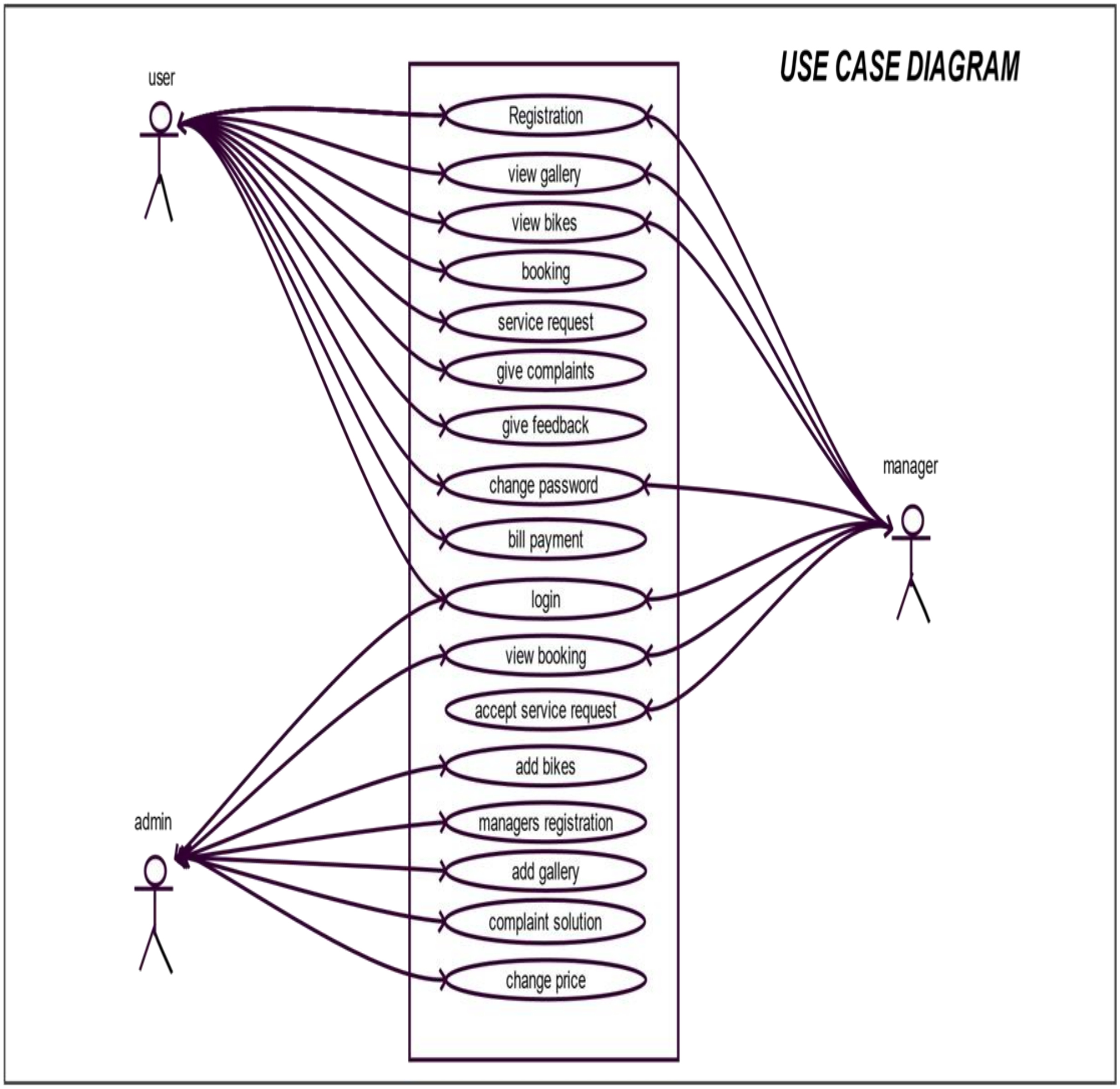
4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems system objectives can include planning overall requirements validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.


USE CASE DIAGRAM
user
manager
admin
Registration
view gallery
view bikes
booking
service request
give complaints
give feedback
change password
bill payment
login
view booking
accept service request
add bikes
managers registration
add gallery
complaint solution
change price

4.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

• A sequence diagram simply depicts interaction between

Objects in a sequential order i.e. the order in which these

Interactions take place.

• We can also use the terms event diagrams or event

Scenarios to refer to a sequence diagram.

• Sequence diagrams describe how and in what order the

Objects in a system function.

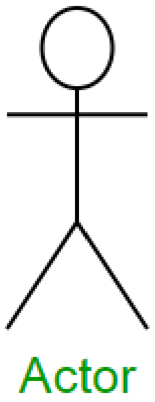
• These diagrams are widely used by businessmen and

Software developers to document and understand

Requirements for new and existing systems.

Sequence Diagram Notations

1. Actors – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.



2. Lifelines

A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format – Instance Name: Class Name We display a lifeline in a rectangle called head with its name and type. The head is located on top of a vertical dashed line (referred to as the stem) as shown above. If we want to model an unnamed instance, we follow the same pattern except now the portion of lifeline's name is left blank.

3.Messages

Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines an messages form the core of a sequence diagram.

4.Synchronous messages

A synchronous message waits for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message. The caller continues only when it knows that the receiver has processed the previous message i.e. it receives a reply message. A large number of calls in object-oriented programming are synchronous. We use a solid arrow head to represent a synchronous message.

5.Asynchronous Messages

An asynchronous message doesn’t waits for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not. We used a lined arrow head to represent an asynchronous message.

Create Message Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines an messages form the core of a sequence diagram.

A synchronous message waits for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message. The caller continues only when it knows that the receiver has processed the previous message i.e. it receives a reply message. A large number of calls in object-oriented programming are synchronous. We use a solid arrow head to represent a synchronous message

An asynchronous message doesn’t wait for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not. We used a lined arrow head to represent an asynchronous message.

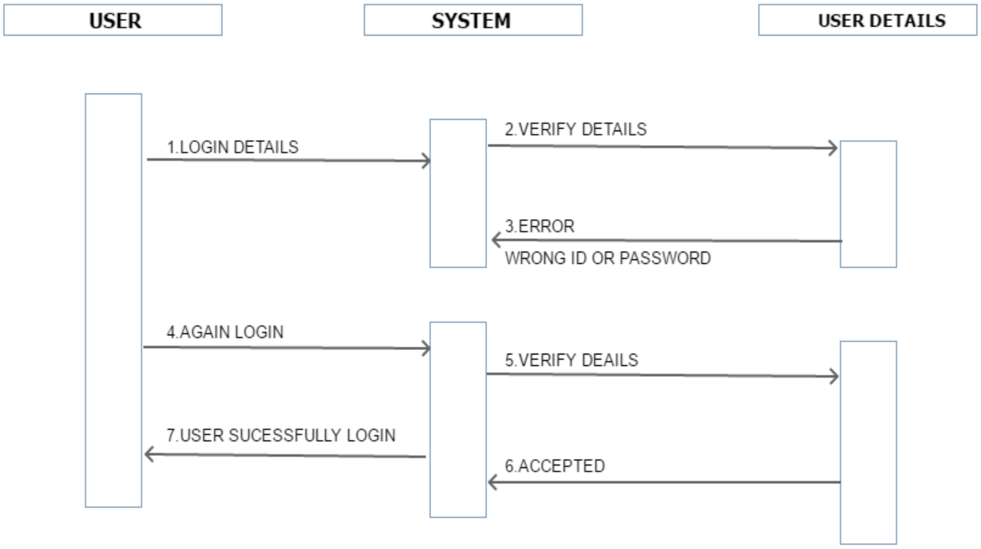
6.Create Message

We use a create message to instantiate a new object in diagram. There are situations when a particular message call requires creation of an object. It is represented with a dotted arrow and create word labelled onto it to specify that it is the create message symbol.We use a create message to instantiate a new object in diagram. There are situations when a particular message call requires creation of an object. It is represented with a dotted arrow and create word labelled onto it to specify that it is the create message symbol.

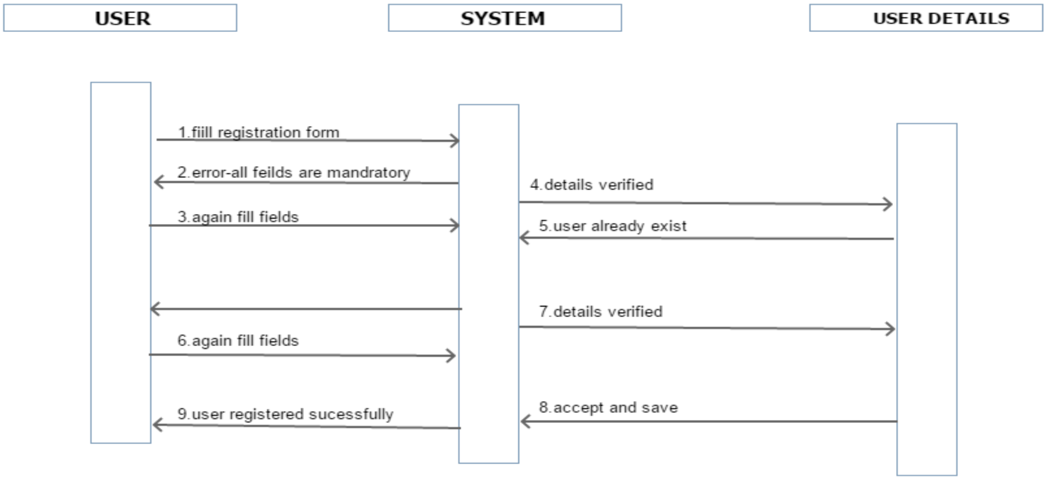
7.Delete message

We use a Delete Message to delete an object. When an object is deallocated memory or is Destroyed within the system we use the Delete Message symbol. It destroys the occurrence of the object in the system. It is represented by an arrow terminating with a x.

Sequence diagram for login



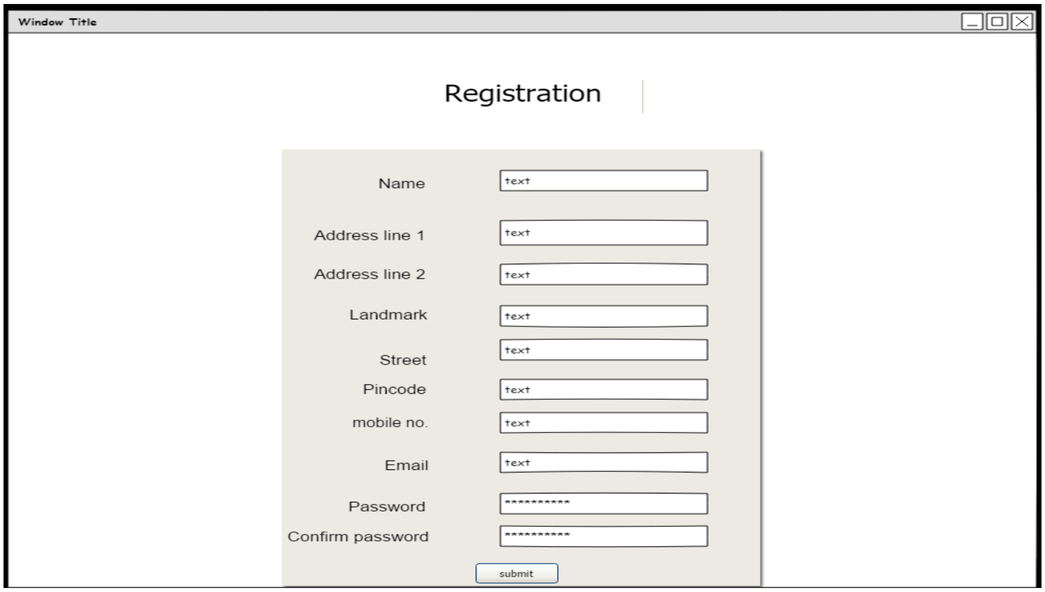
Sequence diagram for Registration



4.4 USER INTERFACE DESIGN

4.4.1 INPUT DESIGN

USER REGISTRATION



Window Title

Registration

Name text

Address line 1 text

Address line 2 text

Landmark text

Street text

Pincode text

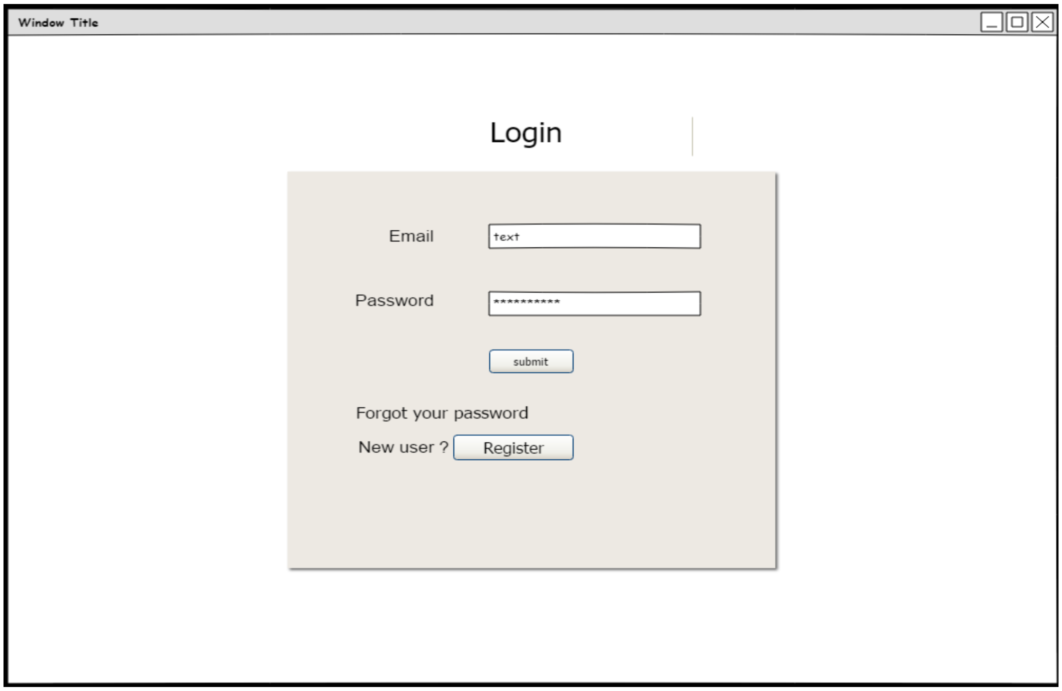
mobile no. text

Email text

Password **********

Confirm password **********

USER LOGIN



Window Title

Login

Email text

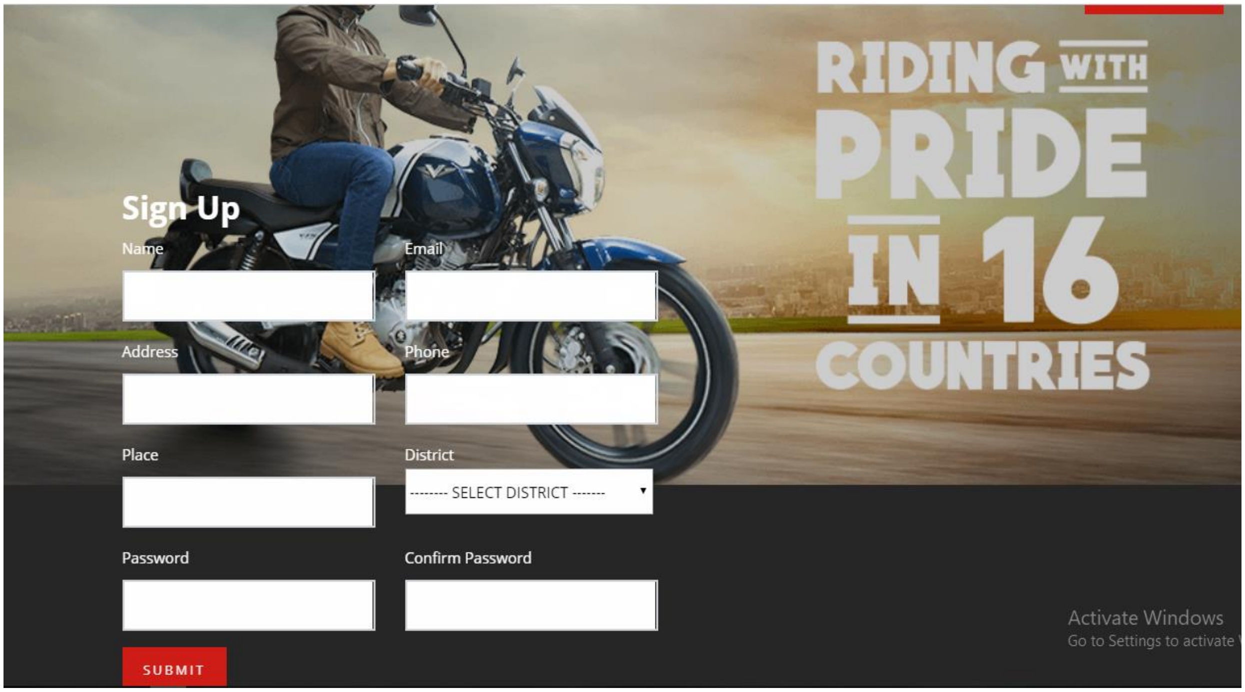
Password **********

Forgot your password

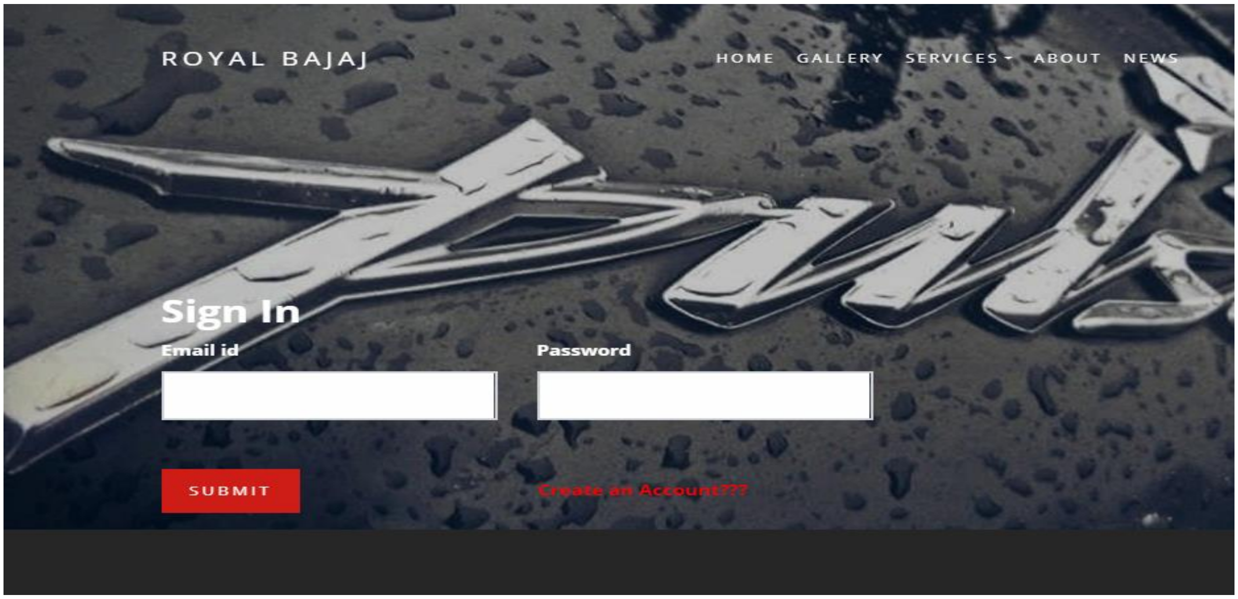
New user ? Register

4.4.2 OUTPUT DESIGN

User Registration



User Login



4.5 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.5.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.5.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modeling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows "relations within relations" or "relations as attribute values within tuples". The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN

TABLE NO: 1
TABLE:tbl_login
PRIMARY KEY: login_id
TABLE DESCRIPTION: To Store All User Login Details

s.no	Field name	Datatype	Size	Constraint	Description
1	login_id	varchar	10	p.k	userid
2	username	varchar	25	Not null	username
3	password	varchar	25	Not null	password
4	usertype	varchar	20	Not null	usertype

TABLE NO: 2
TABLE:tbl_reg
PRIMARY KEY: Userid
TABLE DESCRIPTION: Registration of users and view their details

s.no	Field name	Datatype	Size	Constraint	Description
1	Userid	Varchar	25	p.k	Userid
2	Name	Varchar	25	Not null	Name
3	Address	Varchar	100	Not null	Address
4	Email	Varchar	50	Not null	Email
5	Phone	Int	12	Not null	Phone
6	District_id	Int	10	Not null	District id
7	Place	Varchar	25	Not null	Place
8	Username	Varchar	25	Not null	Username
9	Password	Varchar	25	Not null	Password

TABLE NO: 3
TABLE:tbl_bikes
PRIMARY KEY: Chassis_no
TABLE DESCRIPTION:Bike Details

s.no	Field name	Datatype	Size	Constraint	Description
1	Bike id	varchar	10	Not null	Bike id
2	Name	varchar	20	Not null	Name
3	Power	varchar	10	Not null	Power
4	Prize	varchar	20	Not null	Prize
5	Color	varchar	10	Not null	Color
6	Type	varchar	10	Not null	Type
7	Chassis_no	varchar	25	p.k	Chassis no

TABLE NO: 4
TABLE:tbl_showroom
PRIMARY KEY: Showroom_id
TABLE DESCRIPTION:Showrorm Details

s.no	Field name	Datatype	Size	Constraint	Description
1	Showroom_id	Varchar	10	p.k	Showroom id
2	Branch	Varchar	20	Not null	Branch name
3	Address	Varchar	25	Not null	Address
4	Manager	Varchar	20	Not null	Manager
5	Email	Varchar	20	Not null	Email
6	Phone	Int	12	Not null	Phonenumber

TABLE NO: 5
TABLE:tbl_service
PRIMARY KEY: Service_id
TABLE DESCRIPTION:Service Details

s.no	Field name	Datatype	Size	Constraint	Description
1	Date	Date		Not null	Date
2	Userid	Varchar	25	f.k	User id
3	Service_id	Int	20	p.k	Service no
4	Chassis_no	Varchar	25	f.k	Chassis no

TABLE NO: 6
TABLE:tbl_feedback
PRIMARY KEY: Feedback_id
TABLE DESCRIPTION:Feedback Details

s.no	Field name	Datatype	Size	Constraint	Description
1	Feedback_id	Varchar	20	p.k	Number of feedback
2	Feedback	Varchar	100	Not null	Feedback
3	Userid	Varchar	20	f.k	User id

TABLE NO: 7
TABLE:tbl_district
PRIMARY KEY: District_id
TABLE DESCRIPTION:District Details

s.no	Field name	Datatype	Size	Constraint	Description
1	District_id	Int	10	p.k	District id
2	Name	Varchar	10	Not null	District name

CHAPTER 5

TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. More over differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and change over,the implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customers to place bike booking orders and track orders. Provides lots of advantages like request service, feedback options, enhanced user interface, view latest bikes options, delivery options, order process estimate and may more. The proposed system completely eliminates the manual process of ordering Bajaj bikes and introduces a more efficient completely automated system and thus increases sales.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment of booking should be done in online mode.
- Customers can register online and book and pay for their booking.
- The online system allows users to write complaints about the website as well as showrooms.

BIBILIOGRAPHY

BIBILIOGRAPHY

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html
- www.stackoverflow.com

APPENDIX

SAMPLE CODE

Signup.php

```
<?php
include 'co.php';
if(isset($_POST['submit']))
{
  $a=$_POST['name'];
  $b=$_POST['email'];
  $c=$_POST['address'];
  $d=$_POST['phone'];
  $h=$_POST['place'];
  $e=$_POST['dis'];
  $g=$_POST['pass'];
  $p=md5($g);
   //echo "<script>alert('$g');</script>";
  //$p=md5($g);
  $sql="select * from login where emailid='$b'";
  $result=mysqli_query($con,$sql);
  $rowcount=mysqli_num_rows($result);
  if($rowcount==0)
  {
$sq="insert into login(emailid,password,usertype)values('$b','$p',1)";
if(mysqli_query($con,$sq))
{
            $s=mysqli_query($con,"select userid from login where emailid='$b'");
            $r=mysqli_fetch_array($s,MYSQLI_ASSOC);
            $lid=$r['userid'];
            //echo "<script>alert('$lid');</script>";
$sql="insert into
creg(userid,name,emailid,address,phone,place,district,status)values('$lid','$a','$b','$c','$d','$h','$e',
1)";
 $ch=mysqli_query($con,$sql);
if($ch)
{?>
            <script>
 alert("Registration Successfull");
  window.location='signinn.php';
</script>
           <?php
}
else
{
  echo"error:".$sql."<br>".mysqli_error($con);
}
}
}
else
{
  ?>
  <script>alert("user already exist!!")
  window.location='signupp.php';
</script>
  <?php
}
}
mysqli_close($con);
?>
```

Signin.php

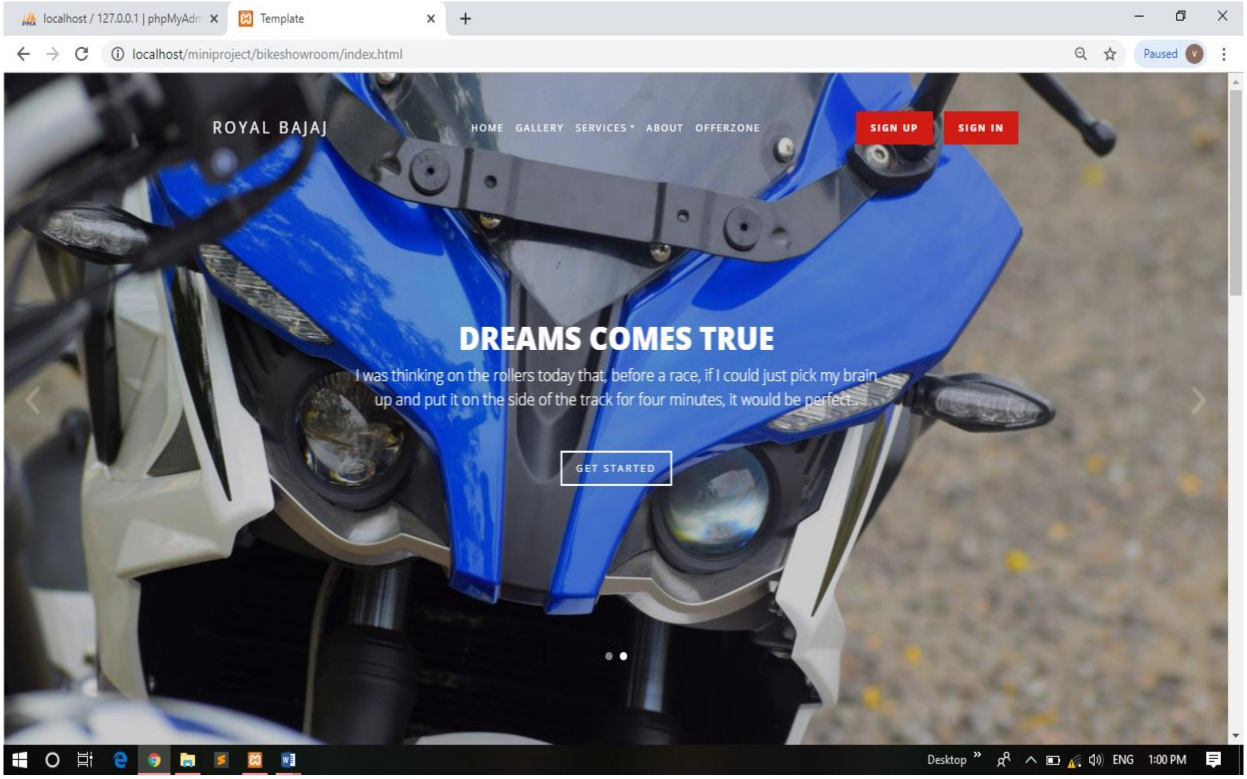
```
<?php
session_start();
//echo "popo";
include("co.php");
if(isset($_POST['submit']))
{
   $email =$_POST['email'];
  $pass =$_POST['pass'];
  $pas=md5($pass);
//echo $u_pass;
$sql="select * from login where emailid='$email'";
//echo $sql;
$result=mysqli_query($con,$sql);
$rowcount=mysqli_num_rows($result);
if($rowcount!=0)
{
  while($row=mysqli_fetch_array($result))
  {
    $dbu_name=$row['emailid'];
    $dbu_pass=$row['password'];
    $dbu_type=$row['usertype'];
    $id=$row['userid'];
    if($dbu_name==$email && $dbu_pass==$pas)
    {
      $_SESSION['emailid']=$dbu_name;
    $_SESSION['password']=$dbu_pass;
    $_SESSION['login']="1";
    $_SESSION['userid']=$id;
    $id1=$_SESSION['userid'];
      //$_SESSION['uname']=$dbu_name;
```

```
        //$_SESSION['pass']=$dbu_pass;
     //echo $dbu_type;
  if($dbu_type==0)
  {
    $_SESSION['log_id']=$row['userid'];
    //$_SESSION['usertype']="admin";
            header('Location:/miniproject/bikeshowroom/admin/index.php');
  }
  else if($dbu_type==1)
  {
    //$_SESSION['usertype']="Seller";
            // $_SESSION['userid']=$row[userid];

              $_SESSION['type']="Staff";
             $sql1="select * from creg where userid ='$id'";
             $result1=mysqli_query($con,$sql1);
             if($row1=mysqli_fetch_array($result1)){
                  $usr_name=$row1['name'];
                  $_SESSION['name']=$usr_name;


              header('Location: /miniproject/bikeshowroom/user/userpanel.php');
  }
}
  else if($dbu_type==2)
  {
     $_SESSION['log_id']=$row['userid'];
    //$_SESSION['usertype']="User";
      header('location: /miniproject/bikeshowroom/manager/managerpanel.php');
  }
}
else
```

```
        { ?>
  <script type="text/javascript">;
  alert("invalid login");
   window.location='signinn.php';
  </script>
  <?php
 }
  }
}
else
{ ?>
  <script type="text/javascript">;
  alert("invalid login");
   window.location='signinn.php';
  </script>
  <?php
 }
}
?>
```

SCREENSHOTS

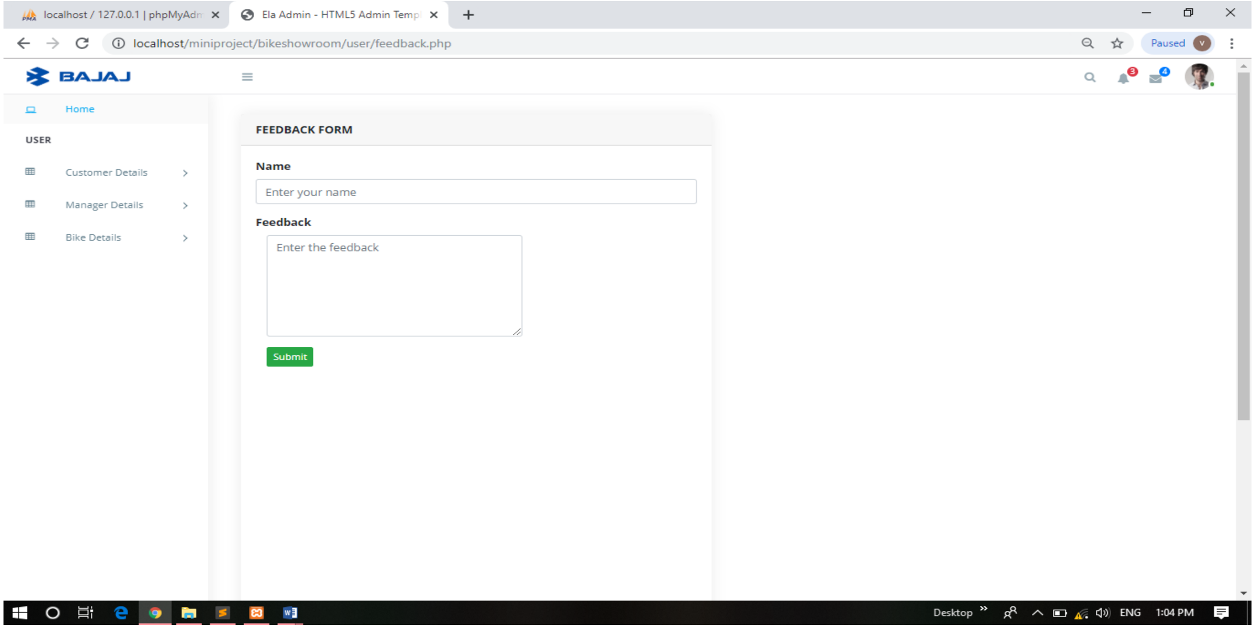
USER HOME



DREAMS COMES TRUE

I was thinking on the rollers today that, before a race, if I could just pick my brain up and put it on the side of the track for four minutes, it would be perfect.

USER FEEDBACK



localhost / 127.0.0.1 | phpMyAdm
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/user/feedback.php
Paused
BAJAJ
Home
USER
Customer Details
Manager Details
Bike Details
Desktop
ENG
1:04 PM

FEEDBACK FORM

Name

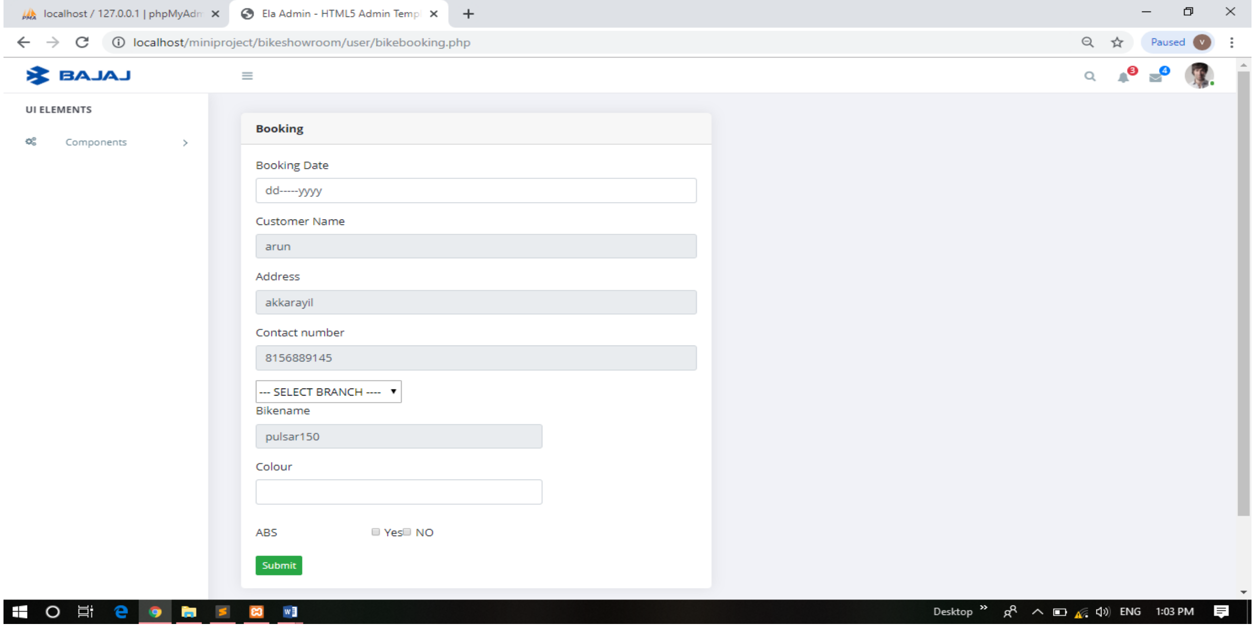
Enter your name

Feedback

Enter the feedback

Submit

USER BIKE BOOKING


localhost / 127.0.0.1 | phpMyAdm
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/user/bikebooking.php
Paused
BAJAJ
UI ELEMENTS
Components
Desktop
ENG
1:03 PM

Booking

Booking Date

dd-----yyyy

Customer Name

arun

Address

akkarayil

Contact number

8156889145

--- SELECT BRANCH ---- ▾

Bikename

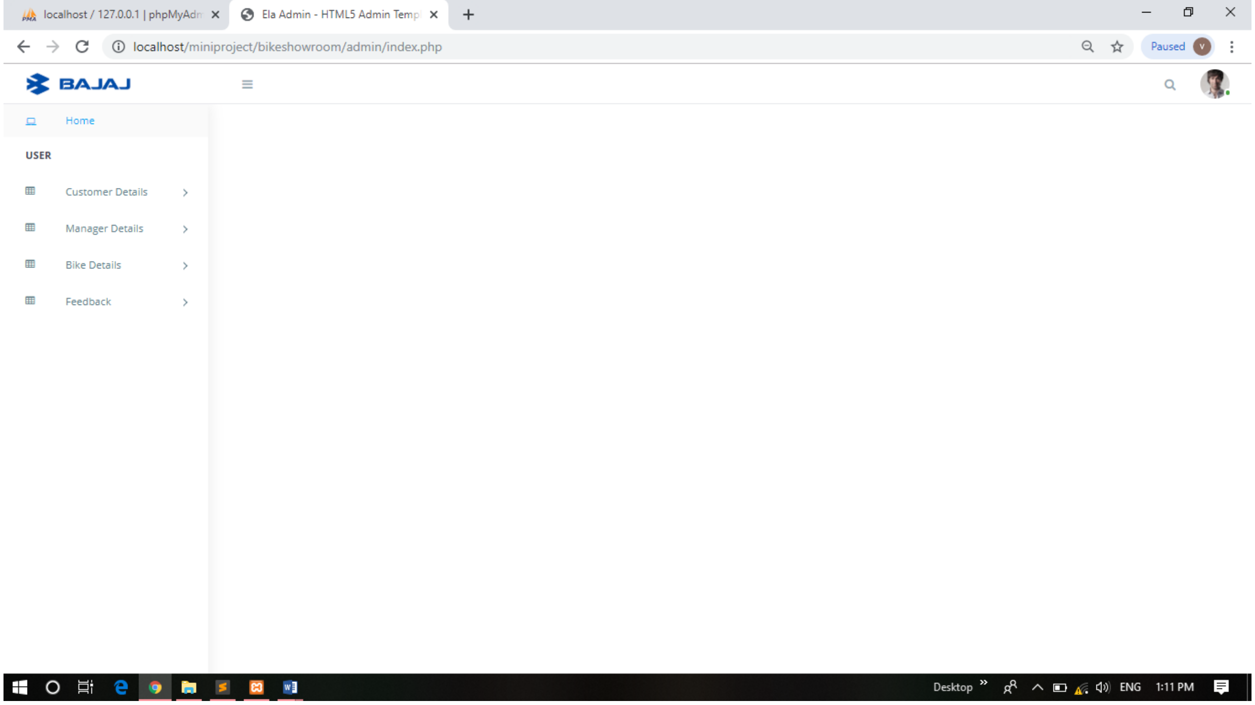
pulsar150

Colour

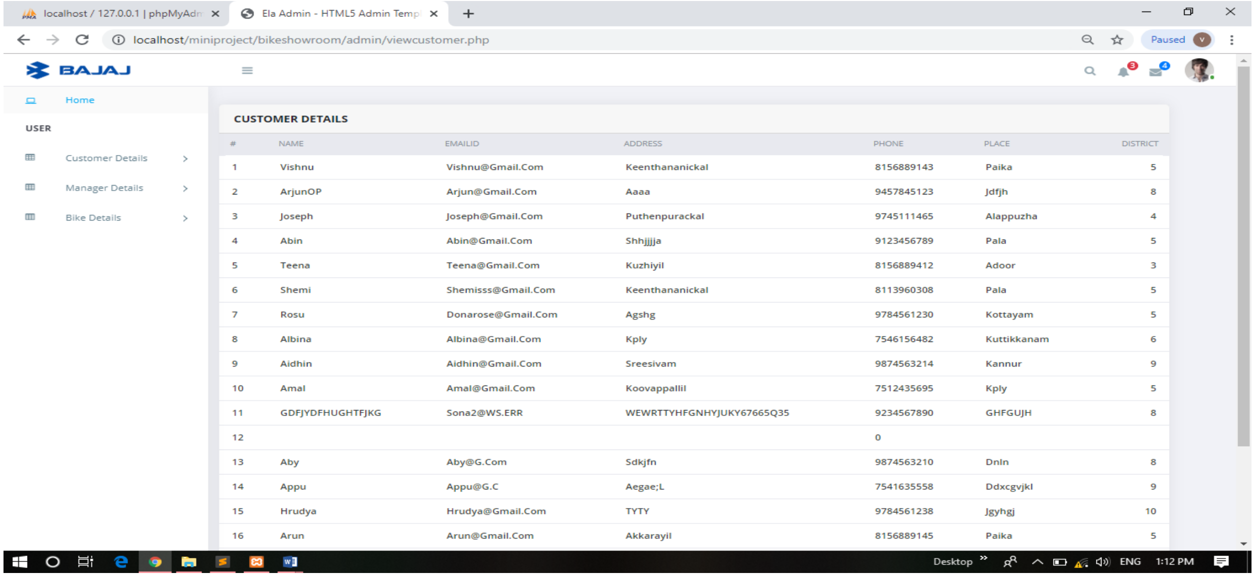
ABS □ Yes □ NO

Submit

ADMIN HOME


localhost / 127.0.0.1 | phpMyAdm
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/admin/index.php
Paused
BAJAJ
Home
USER
Customer Details
Manager Details
Bike Details
Feedback
Desktop
ENG
1:11 PM

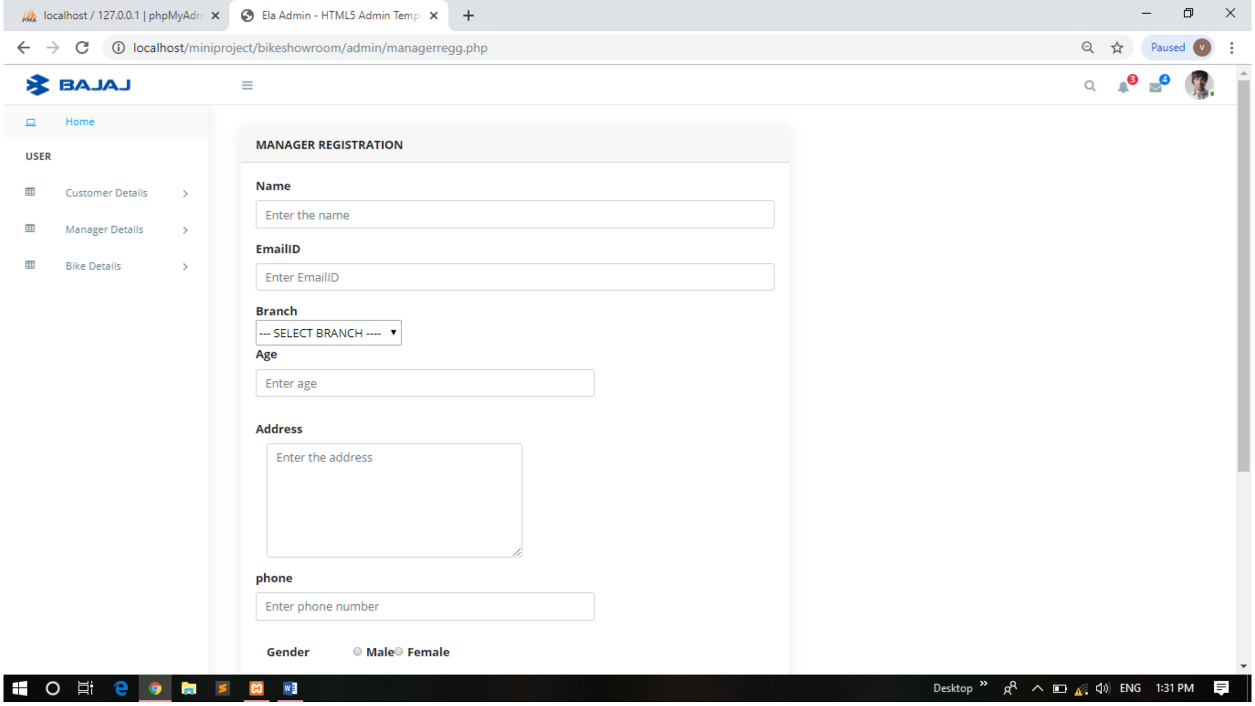
ADMIN VIEW CUSTOMERS


localhost / 127.0.0.1 | phpMyAdm
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/admin/viewcustomer.php
Paused
BAJAJ
Home
USER
Customer Details
Manager Details
Bike Details
Desktop
ENG
1:12 PM

CUSTOMER DETAILS

#	NAME	EMAILID	ADDRESS	PHONE	PLACE	DISTRICT
1	Vishnu	Vishnu@Gmail.Com	Keenthananickal	8156889143	Paika	5
2	ArjunOP	Arjun@Gmail.Com	Aaaa	9457845123	Jdfjh	8
3	Joseph	Joseph@Gmail.Com	Puthenpurackal	9745111465	Alappuzha	4
4	Abin	Abin@Gmail.Com	Shhjjjja	9123456789	Pala	5
5	Teena	Teena@Gmail.Com	Kuzhiyil	8156889412	Adoor	3
6	Shemi	Shemisss@Gmail.Com	Keenthananickal	8113960308	Pala	5
7	Rosu	Donarose@Gmail.Com	Agshg	9784561230	Kottayam	5
8	Albina	Albina@Gmail.Com	Kply	7546156482	Kuttikkanam	6
9	Aidhin	Aidhin@Gmail.Com	Sreesivam	9874563214	Kannur	9
10	Amal	Amal@Gmail.Com	Koovappallil	7512435695	Kply	5
11	GDFJYDFHUGHTFJKG	Sona2@WS.ERR	WEWRTTYHFGNHYJUKY67665Q35	9234567890	GHFGUJH	8
12				0		
13	Aby	Aby@G.Com	Sdkjfn	9874563210	Dnln	8
14	Appu	Appu@G.C	Aegae;L	7541635558	Ddxcgvjkl	9
15	Hrudya	Hrudya@Gmail.Com	TYTY	9784561238	Jgyhgj	10
16	Arun	Arun@Gmail.Com	Akkarayil	8156889145	Paika	5

ADMIN ADD MANAGER


localhost / 127.0.0.1 | phpMyAdm
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/admin/managerregg.php
Paused
BAJAJ
Home
USER
Customer Details
Manager Details
Bike Details
Desktop
ENG
1:31 PM

MANAGER REGISTRATION

Name
Enter the name

EmailID
Enter EmailID

Branch
--- SELECT BRANCH ----

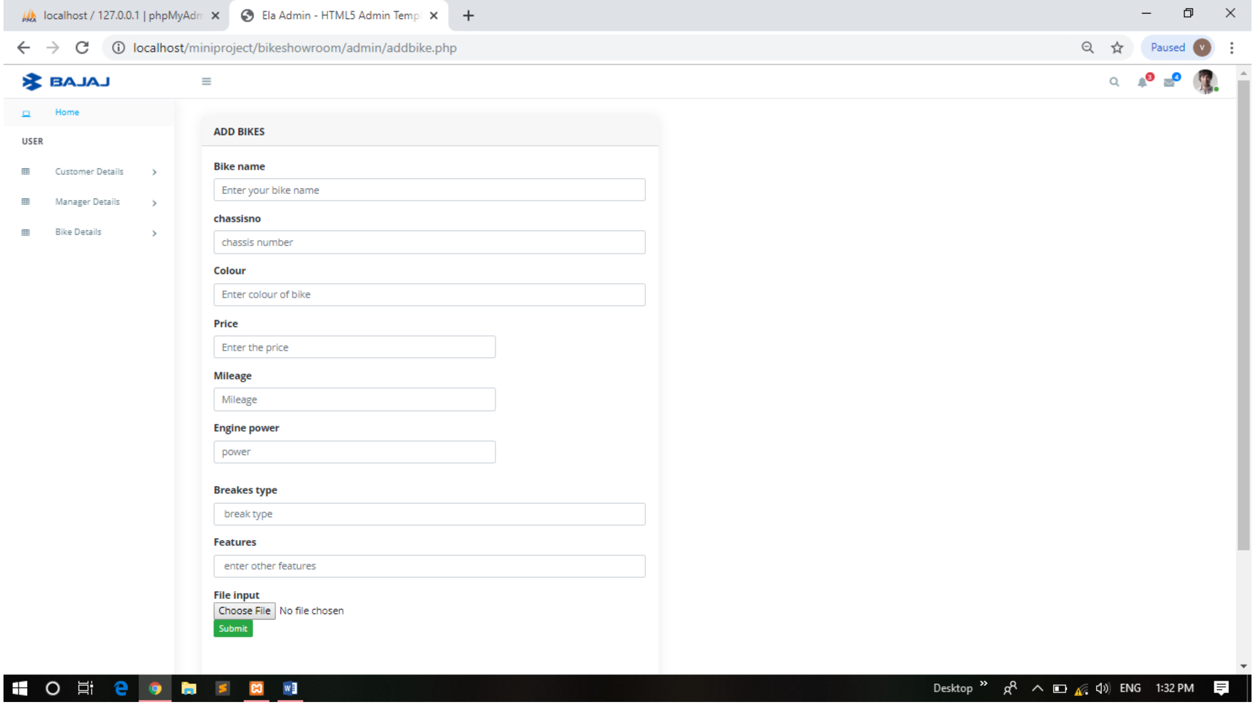
Age
Enter age

Address
Enter the address

phone
Enter phone number

Gender ○ Male ○ Female

ADMIN ADD BIKES


localhost / 127.0.0.1 | phpMyAdm
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/admin/addbike.php
Paused
BAJAJ
Home
USER
Customer Details
Manager Details
Bike Details
Desktop
ENG
1:32 PM

ADD BIKES

Bike name

Enter your bike name

chassisno

chassis number

Colour

Enter colour of bike

Price

Enter the price

Mileage

Mileage

Engine power

power

Breakes type

break type

Features

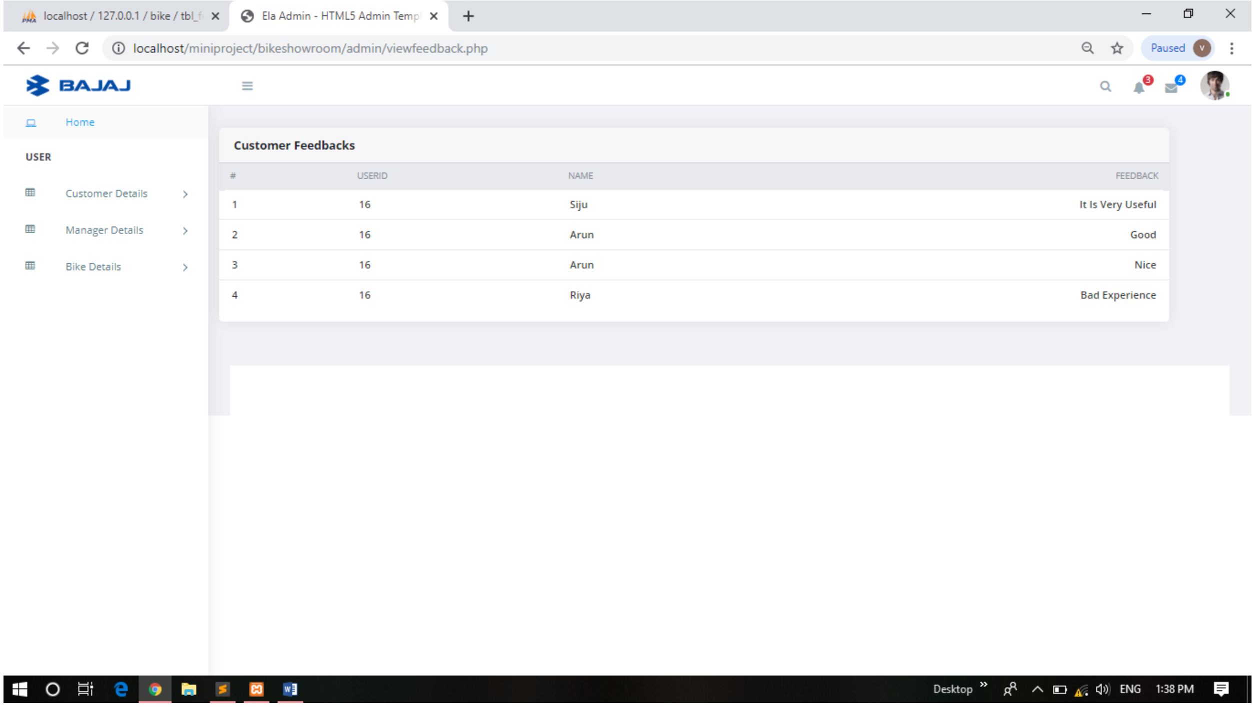
enter other features

File input

Choose File No file chosen

Submit

ADMIN VIEW FEEDBACKS


localhost / 127.0.0.1 / bike / tbl_f
Ela Admin - HTML5 Admin Temp
localhost/miniproject/bikeshowroom/admin/viewfeedback.php
Paused
BAJAJ
Home
USER
Customer Details
Manager Details
Bike Details
Desktop
ENG
1:38 PM

Customer Feedbacks

#	USERID	NAME	FEEDBACK
1	16	Siju	It Is Very Useful
2	16	Arun	Good
3	16	Arun	Nice
4	16	Riya	Bad Experience