MOBILE REPAIR MANAGEMENT SYSTEM

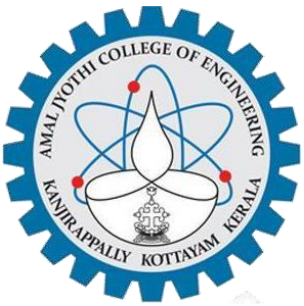
Project Report Submitted By

KEVIN LIZA ALEX

Reg. No: AJC17MCA-I027

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

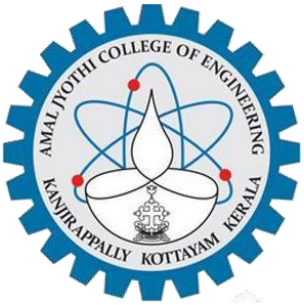
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**MOBILE REPAIR MANAGEMENT SYSTEM"** is the bonafide work of **KEVIN LIZA ALEX (Reg. No: AJC17MCA-I027)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Gloriya Mathew
Internal Guide

Rev. Fr. Dr. Rubin Thottupurathu Jose
Coordinator

Rev. Dr. Fr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"MOBILE REPAIR MANAGEMENT SYSTEM"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **KEVIN LIZA ALEX**

KANJIRAPPALLY **Reg. No: AJC17MCA-I027**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department and project coordinator **Rev. Fr. Dr. Rubin Thottupurathu Jose** for hi valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere to my guide, **Ms. Gloriya Mathew** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

KEVIN LIZA ALEX

ABSTRACT

Mobile Repair Management Website is web application with aim of management and monitoring of mobile repairs at the repair shop using the website from the initial stage of requesting submitting a repair request of mobile phone to repair from customer through the stages of repair and finally the delivery of the repaired mobile phone. The main goal is to ensure a fewer contact repairs and effortless management and monitoring of all the repairs and to ensure more transparency to the existing repair of mobile phones. The proposed extension is the addition of a new module, the inventory manager and admin. The inventory manager module manages the status of spare parts inventory and the admin module manages the modules interacting with the application. Currently the system has no provision to manage stock and to control them and the lack of user control is solved by the additions on admin module. Technician assigns the spare to repair the mobile phone from a list of items. The proposed extension adds the provision to manage the quantity of stock and user control. The technicians and inventory manager together manage the spare part inventory.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

"MOBILE REPAIR MANAGEMENT SYSTEM" is an automation and implementation of a new system where users who submit their mobile phones to repair can submit their mobile phones to the shop by requesting a pickup which is the first step in interacting with the web application. After successful pickup user can track the repair states and carry out payment after successful repair of their products. Currently each shop has their own database of application that works as a database to store daily repair information. What my project implements is an interface to the outside people, called users of this project to carry out the above-mentioned functionality provided by this project. The main users of this project are people who makes repair request, and admin also known as technicians. Users initiate their interaction with the web application by submitting a repair request for repairing their mobile phone display, battery or phone after successful login to the application. Each request is identified using mobile details and IMEI number. After successful completion of submitting a repair request, this request will be sent to the admin also known as technicians' side. They can acknowledge request from user and updates its status (pending as default, received, progressing, completed, out for delivery, waiting for pickup) from the admin side and users can see the updates on their repair request. Payment options is enabled for users after completion of repairs.

1.2 PROJECT SPECIFICATION

The proposed system is made to provide a hassle-free access to check the status of their repairs at the repairs shop. After successful completion of the repair the web application provides the provision complete payment through the application itself. User can get this status of their repairs from anywhere at any time. User has their own page to track their status and manage their profile. Also, the tasks are handled by the technicians who are called as admins here. They manage backend activities updates and management of the activities of this application. Repair request, pickup, repair status updates and user management are the main tasks of the technicians or admins.

The system includes 4 modules. They are:

1. Technician Module

Technicians are the ones who carry out the mobile phone repairs submitting at the shop. They must have a login into this system. He has the overall control over the repair management of the system. Technicians can manage users registered to the system. Technicians has full control over each repair and its status update. The systems list all the repairs unassigned to any technicians as a pool of jobs and each logged in technicians can collect any job and view and manage it in their My jobs tab. Technician has options to add new mobile brand, services. They manage the users. They can control users to the application. And also, they can request for updating of spare part stock in case of out-of-stock scenario. Each technician manages the status of the repairs they get. Each one is entrusted with the task to manage their repairs and update their corresponding users.

2. Customer Module

Customers are those who make interactions to the application. Here customers are those who submits their mobile phone to repair. Their interaction with the application starts by requesting for a pickup of the mobile phones to repair. They can make repair request, track its status through an individual interface for each user. At each stage the admins update the status. So that each user can check the stats of their repairs under their own pages after successful loin. Each repair starts when user makes a repair request an opt for pickup or drop at shop options to service their phones. Each user has their own profiles and provision to update passwords. After successful completion of repairs, user can complete payment. Payment option is only available after successful repair from admin side. And they can request for pickup of repaired product or opt for delivery to home.

3. Admin Module

A dedicated admin module enables an added administrative control on interaction of users to the proposed system. Admin has specific controls over user management, new user registration such as titles like technician and inventory manager. Login credentials for the

technicians and inventory manager are created and allotted by the admin. There by restricts the unauthorized creation of login credentials and only authorized credentials are created and given access to the system.

4. Inventory Manager Module

An inventory manager module is that manages the stock inventory of the spare parts to replace instead of the faulty ones. The inventory manager has the provision to add new spare parts, their quantity and rate. The stock requirement submitted by the technician is managed by the inventory manager. The inventory manager can see the stock requirement and the updates the quantity based on the requirement and stock arrival. The stock os managed dynamically such that after each repairs the stock of each spare part in the inventory is decreased and managed properly.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. Thissystem is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

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

There exists no exact system that exists. The prevailed system is a database system for the technician aka admin to store the history all the repairs at their show till the date. The customer has no options to interact with the shop online. They can't track their repair status or make payment for the repairs. Presently the method followed is the physical interaction by the user with the technicians at the shop.

It is necessary to implement a new system for the hassle-free interaction of customer with the repair shop. The proposed system aims at reducing the efforts of each user and providing them with a lot more features online than accessing the shop physically. Thereby provides an interface for users to interact with their repairs to the technicians and track the status and complete payments.

2.3 DRAWBACKS OF EXISTING SYSTEM

- No provision for user interaction: Exists only a database to record all the repairs as archival system.
- No live status update of repair for users: User has to frequently depends on contacting the shop physically or via call which require the participation of the technician every time.
- Only direct payment to shop is available: All the repair payments have to be made through physical contact with repair shop. No online modes are available other than cards.
- Has to wait to get response if there is rush in the shop: In case of high repair traffics at the shop each user has to face inconvenience to get response for the repairs and queries from the technicians.

2.4 PROPOSED SYSTEM

The proposed system is defined to meet 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 as well as user convenience. The system is proposed in such consideration. In the

proposed system there is technician who can manage users, manage repair request and its status. It allows to get live updates on their repair request and make payment once the repair is complete. Modules of this proposed system are technician, customers, admin, inventory manager. The aim of proposed system is to develop a system that gives user a hassle-free interaction to track their repair and its related services.

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:

- **Realtime service status update:**

Our customers will get real-time status update and support services for each of their service requests. Could manage each service request individually.

- **Easy pickup and drop:**

Customers have the provision to request for pickup from and drop at the customers location to repair mobile phones and for those which has been repaired. Also, we provide a provision to drop at and pickup from the shop those phones which has completed repair.

- **Online payment:**

Enables users to make payment online for successful completion of repair service.

- **Inventory management:**

A dedicated provision to manage the stock of the spare parts required to complete the repair is implemented. The inventory manager manages the stock of spare parts required and maintained based on the request by the technician.

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 feasibilitystudy 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 anew 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 Economic 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 gives an indication of the system is economically possible for development.

The cost of project, Mobile Repair Management System 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.

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 for developing the system, and 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. 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 inserver 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.

Mobile Repair Management System, GUI is simple so that users can easily use it. Mobile Repair Management System is simple enough so that no training is needed for interacting with the system.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 GB

Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML, CSS

Back End - MySQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, JQuery, PHP, CSS

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 Ledorf 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 onmost 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 orsystem. 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 itsphysical 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

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After

some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE 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.

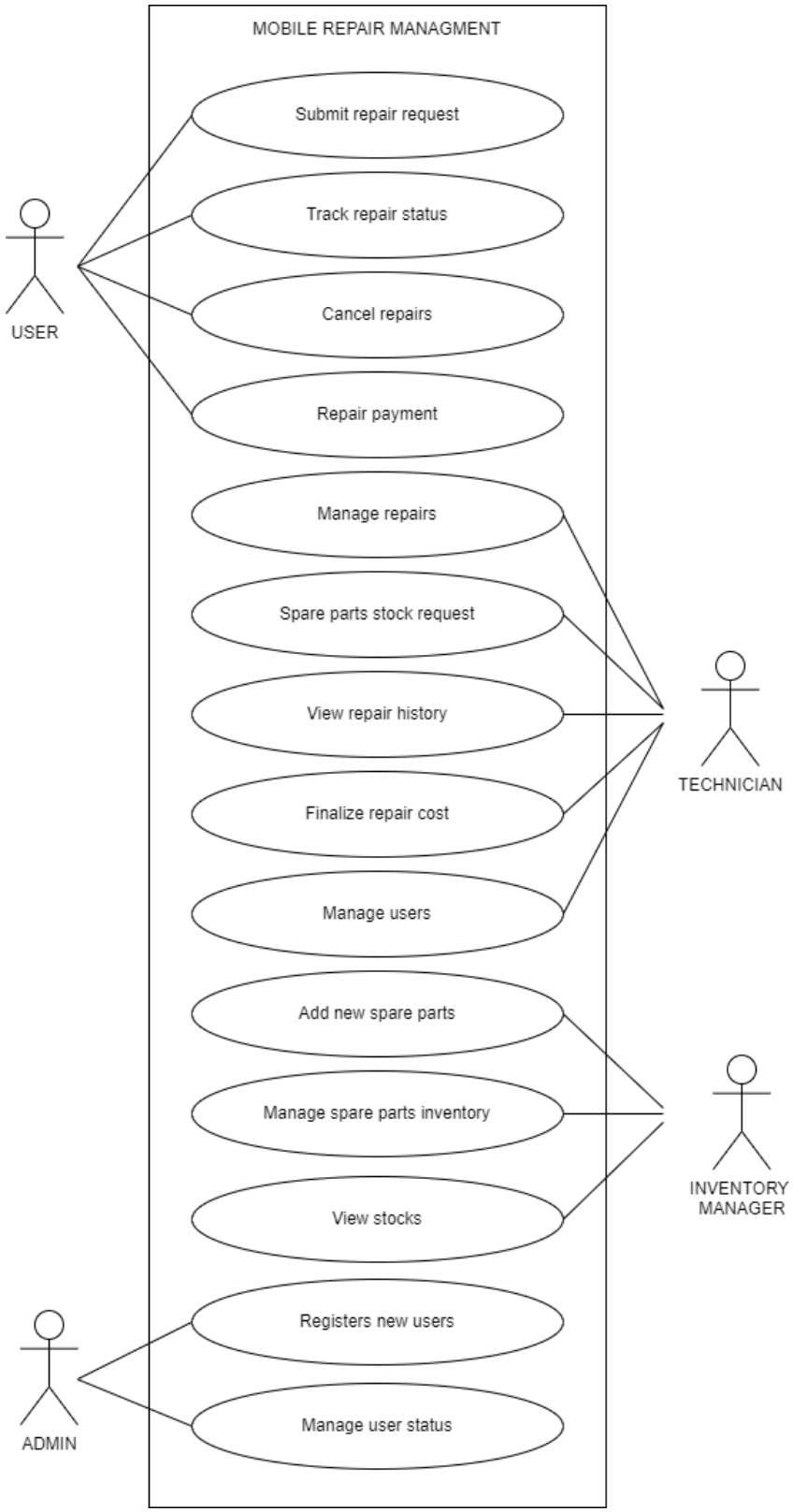
- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles

- The use cases, which are the specific roles are played by the actors within andaround the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficientuse case diagram

- The name of a use case is very important. The name should be chosen in such a wayso that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagramis to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig. 1 - Use case diagram for Mobile Repair Management System:



4.2.2 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. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **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.

ii. **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.

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

Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message

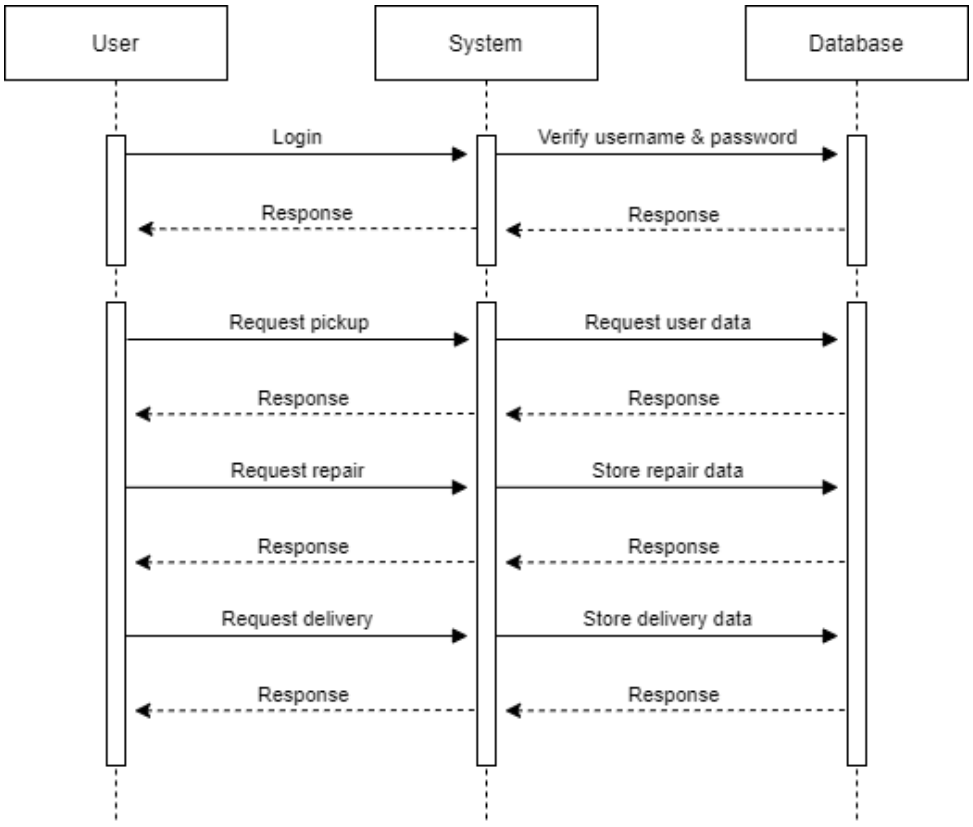
- Found Message
- Lost Message

iv. Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of Sequence Diagram –

- Used to model and visualize the logic behind a sophisticated function, operation orprocedure.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Fig. 2 - Sequence diagram for Mobile Repair Management System:



4.2.3 CLASS DIAGRAM

Purpose of Class Diagrams

- Shows static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribed by UML
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business perspective

A UML class diagram is made up of:

- A set of classes and
- A set of relationships between classes

What is a Class

A description of a group of objects all with similar roles in the system, which consists of:

- Structural features (attributes) define what objects of the class "know"
- Represent the state of an object of the class
- Are descriptions of the structural or static features of a class
- Behavioral features (operations) define what objects of the class "can do"
- Define the way in which objects may interact
- Operations are descriptions of behavioral or dynamic features of a class

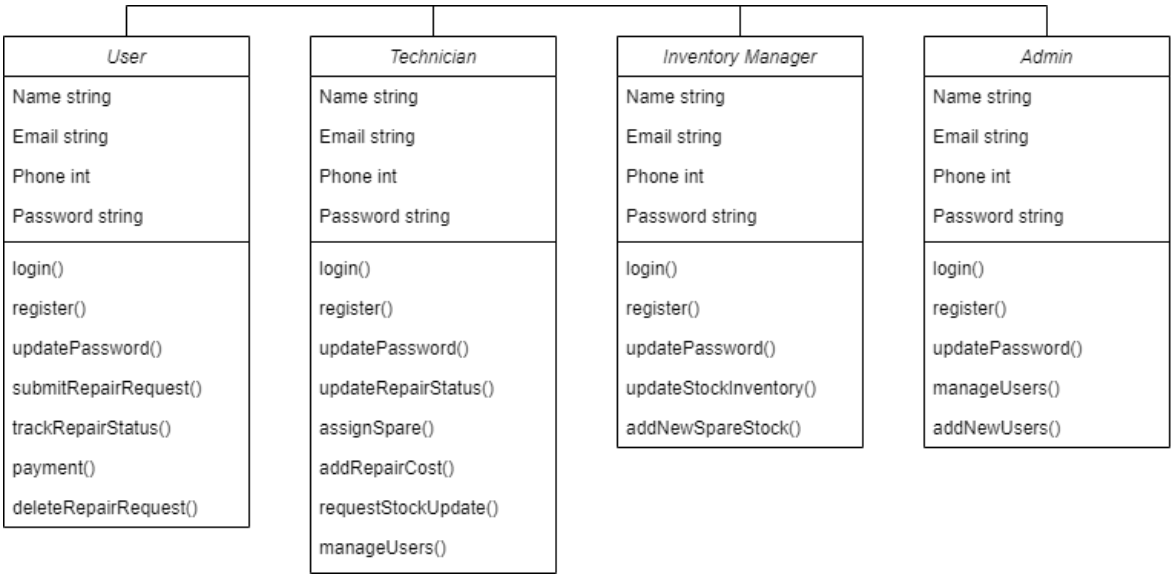
Class Notation

A class notation consists of three parts:

- Class Name: The name of the class appears in the first partition.
- Class Attributes: Attributes are shown in the second partition. The attribute type is shown after the colon. Attributes map onto member variables (data members) in code.

- Class Operations (Methods): Operations are shown in the third partition. They are services the class provides. The return type of a method is shown after the colon at the end of the method signature. The return type of method parameters is shown after the colon following the parameter name. Operations map onto class methods in code

Fig. 3 - Class Diagram for Mobile Repair Management System:



4.2.4 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

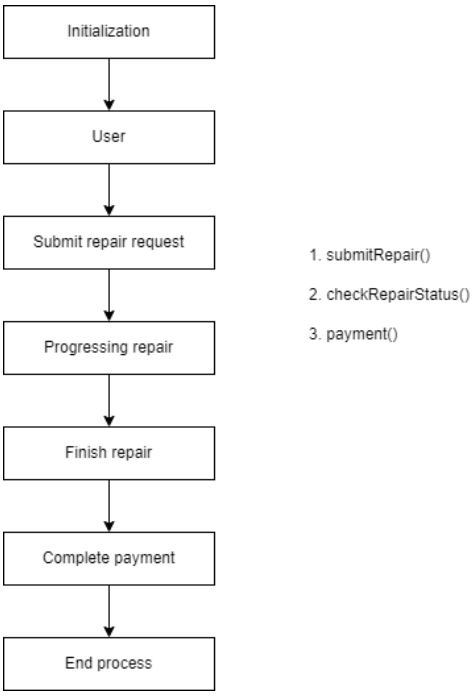
Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

Notations of a collaboration diagram

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Thefour major components of a collaboration diagram are:

- Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.
- Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.
- Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.
- Messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

Fig. 4 - Collaboration Diagram for Mobile Repair Management System:



4.2.5 ACTIVITY DIAGRAM

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements −

- Activities
- Association
- Conditions
- Constraints

Fig. 5 - Activity Diagram for user:

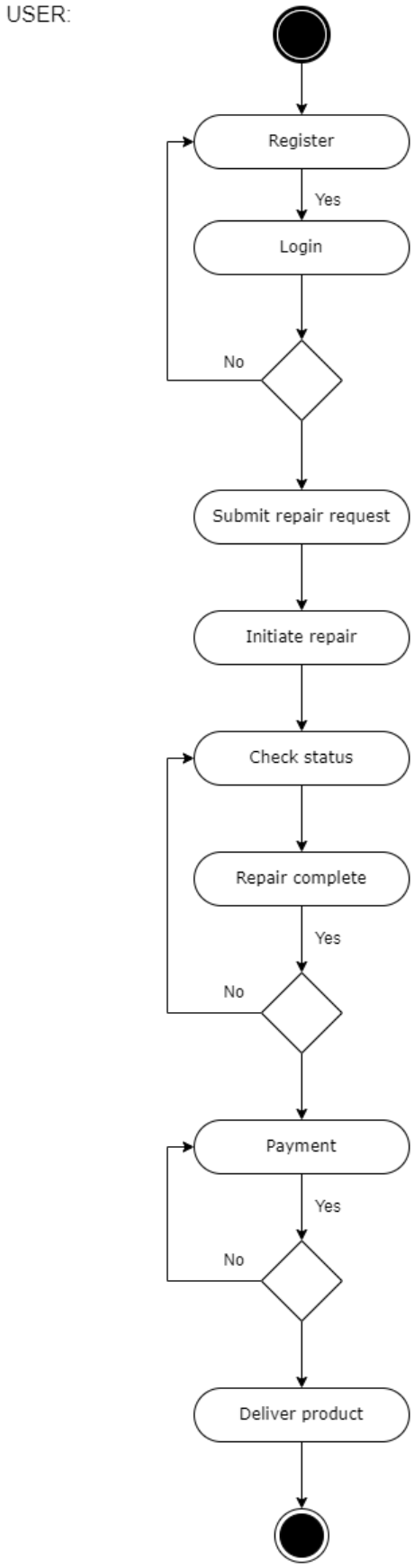


Fig. 6 - Activity Diagram for technician:

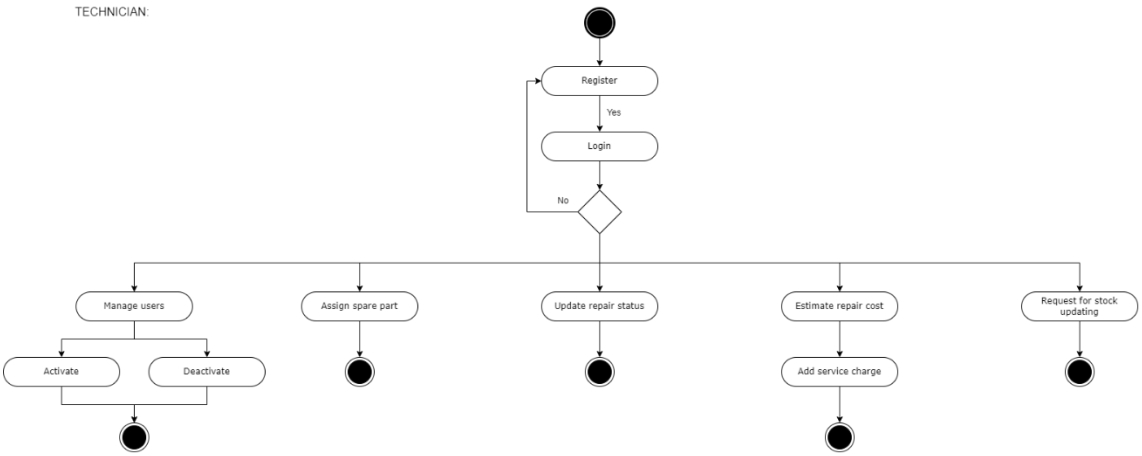
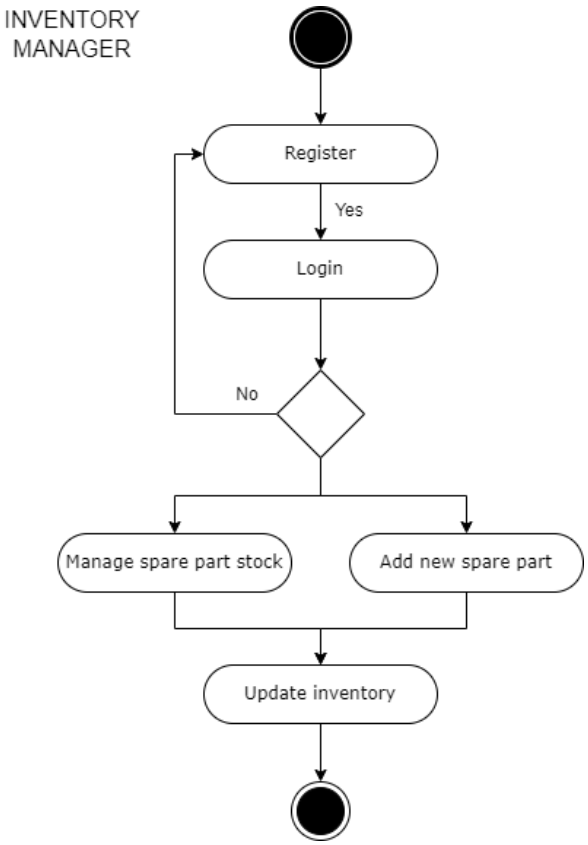


Fig. 7 - Activity Diagram for inventory manager:



4.2.6 STATE CHART DIAGRAM

State chart diagram describes different states of a component in a system. The states are specific to a component/object of a system.

A state chart diagram describes a state machine. State machine can be defined as a machine

which defines different states of an object and these states are controlled by external or internal events.

As State chart diagram defines the states, it is used to model the lifetime of an object.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

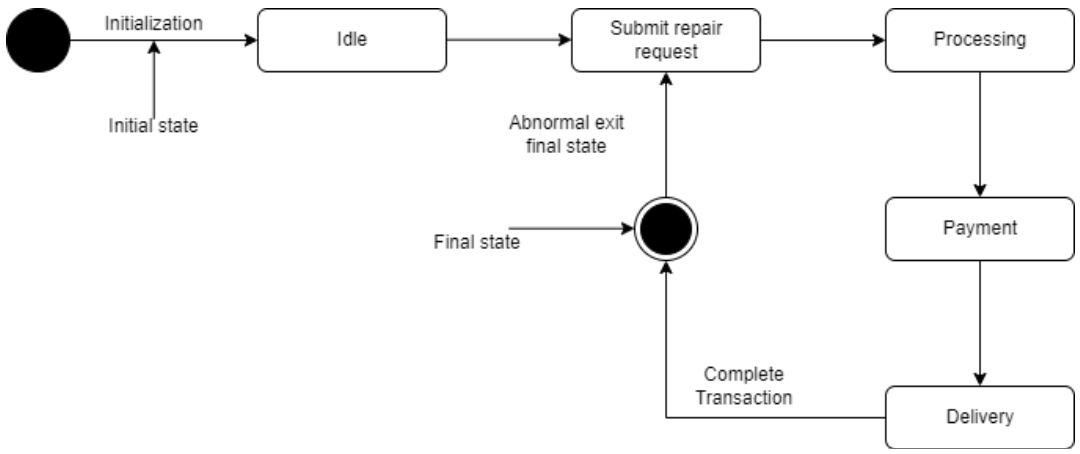
State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

How to Draw a State chart Diagram?

State chart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.

State chart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.

Fig. 8 – State Chart Diagram for Mobile Repair Management System:



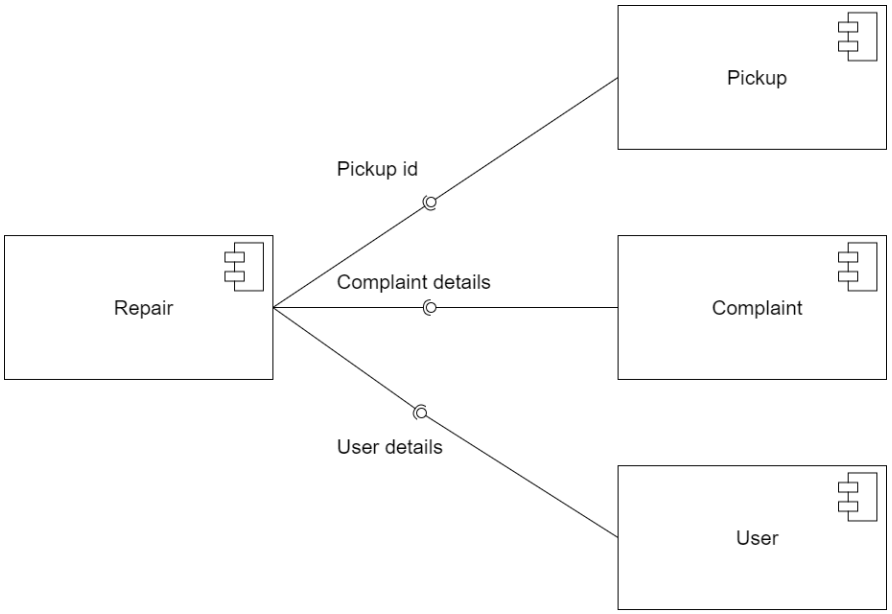
4.2.7 COMPONENT DIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces. The component diagrams have remarkable importance. It is used to depict the functionality and behavior of all the components present in the system, unlike other diagrams that are used to represent the architecture of the system, working of a system, or simply the system itself.

In UML, the component diagram portrays the behavior and organization of components at any instant of time. The system cannot be visualized by any individual component, but it can be by the collection of components.

Fig. 9 - Component Diagram for Mobile Repair Management System:



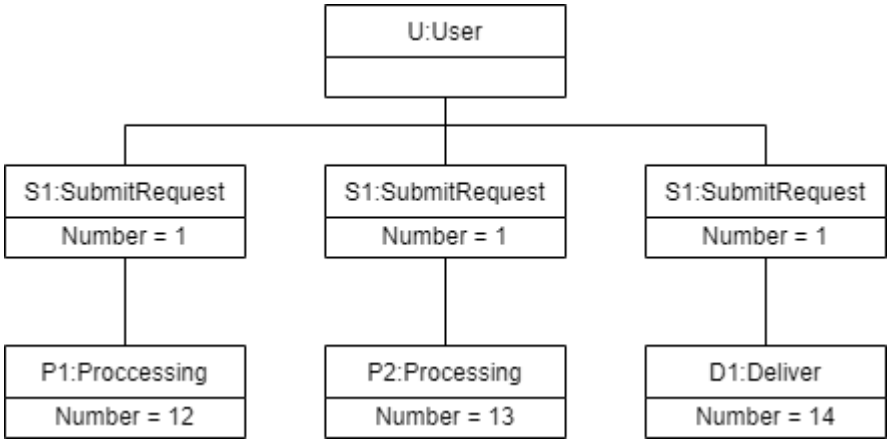
4.2.8 OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

Notations Used in Object Diagram–

- Objects or Instance specifications – When we instantiate a classifier in a system, the object we create represents an entity which exists in the system. We can represent the changes in object over time by creating multiple instance specifications. We use a rectangle to represent an object in an Object Diagram. An object is generally linked to other objects in an object diagram.
- Links – We use a link to represent a relationship between two objects.
- Dependency Relationships – We use a dependency relationship to show when one element depends on another element.
- Association – Association is a reference relationship between two objects (or classes).
- Aggregation – Aggregation represents a “has a” relationship.
- Composition – Composition is a type of association where the child cannot exist independent of the other.

Fig. 10 - Object Diagram for Mobile Repair Management System:



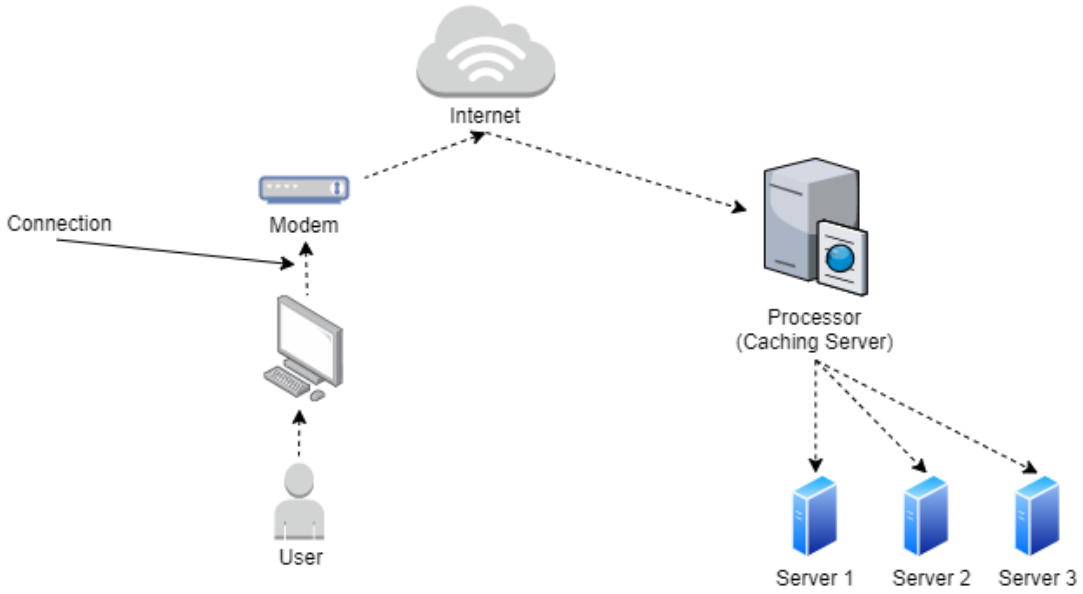
4.2.9 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagram represents the deployment view of a system. It is related to the component diagram because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

Notation of Deployment diagram –

- The deployment diagram consists of the following notations:
- Component: A rectangle with two tabs that indicates a software element.
- Artifact: A product developed by the software, symbolized by a rectangle with the name and the word “artifact” enclosed by double arrows.
- Interface: A circle that indicates a contractual relationship. Those objects that realize the interface must complete some sort of obligation.
- Node: A hardware or software object, shown by a three-dimensional box.

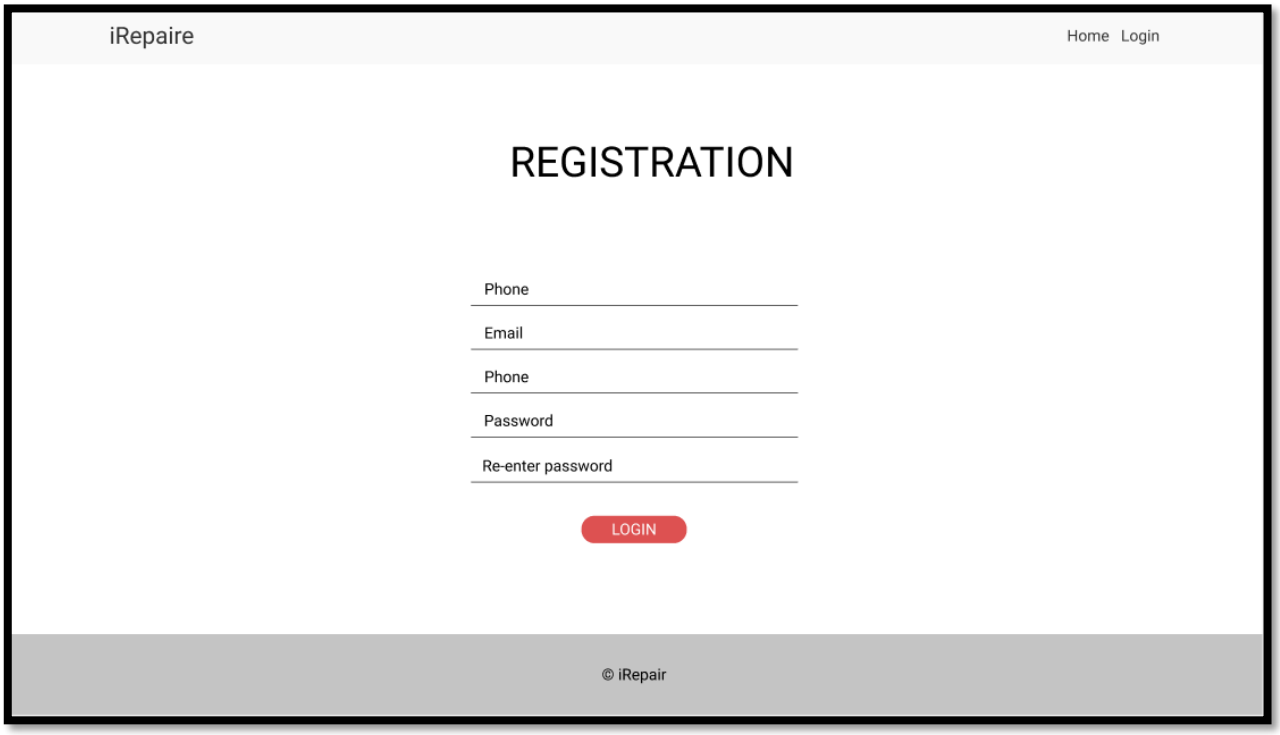
Fig. 11 - Deployment diagram for Mobile Repair Management System:



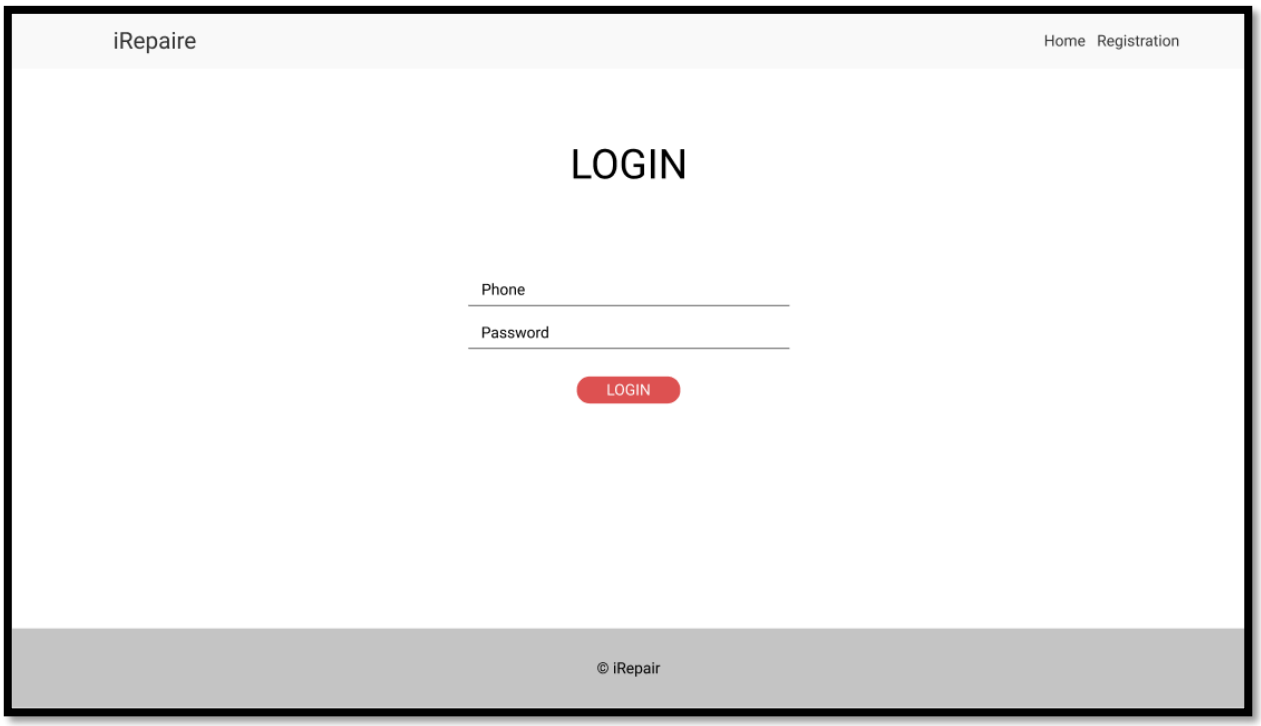
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1-INPUT DESIGN

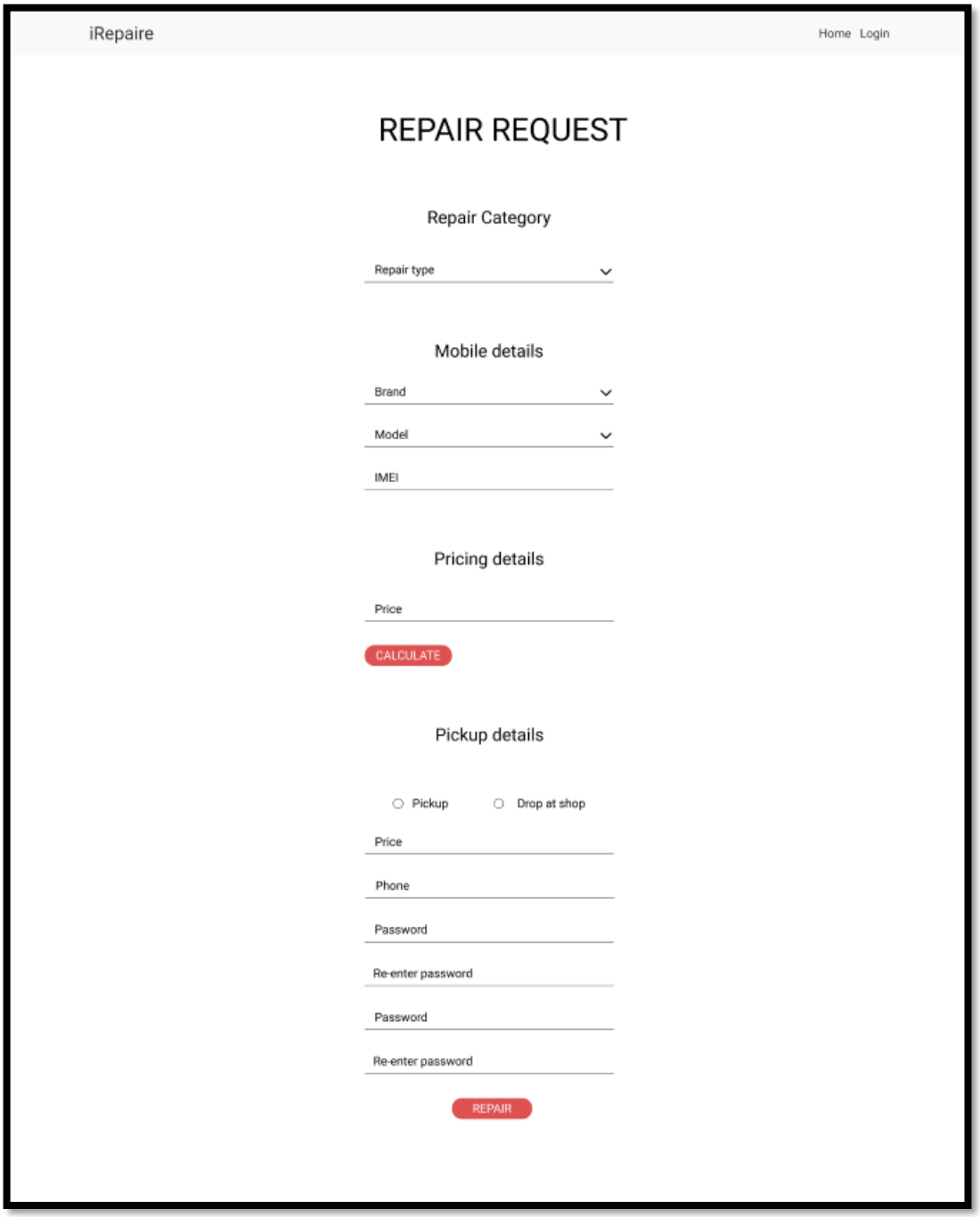
Form Name : Customer Registration



Form Name : User Login

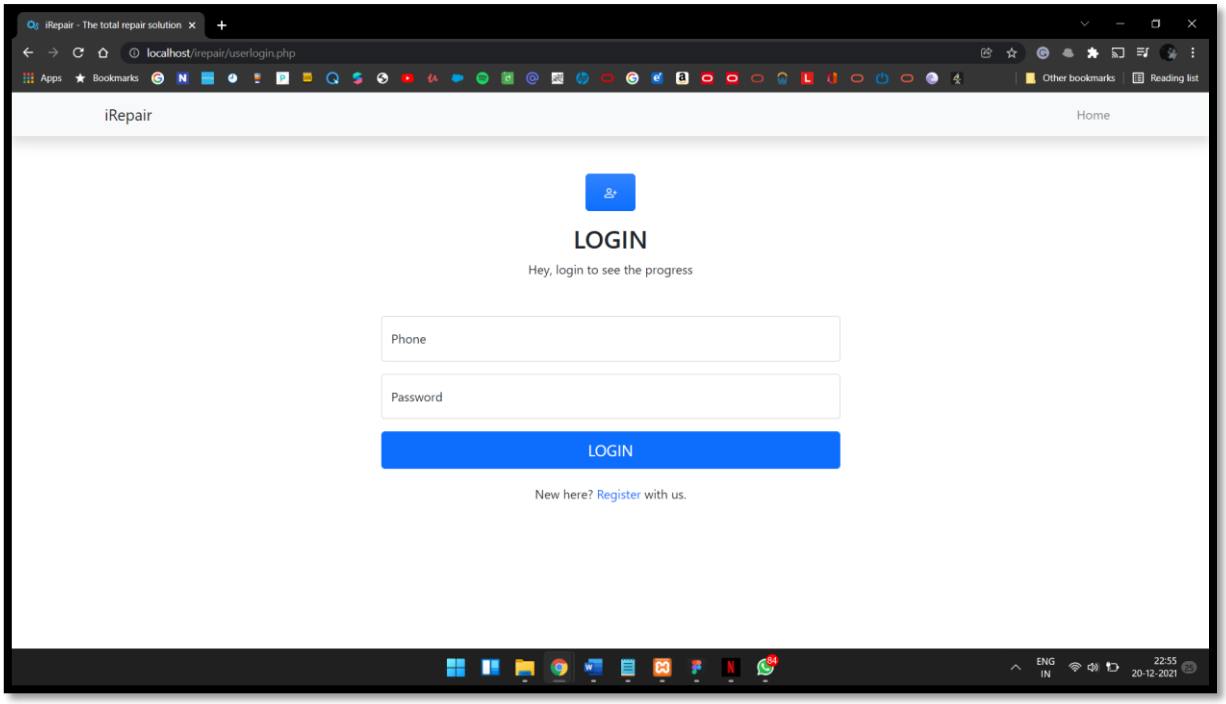


Form Name : Repair request

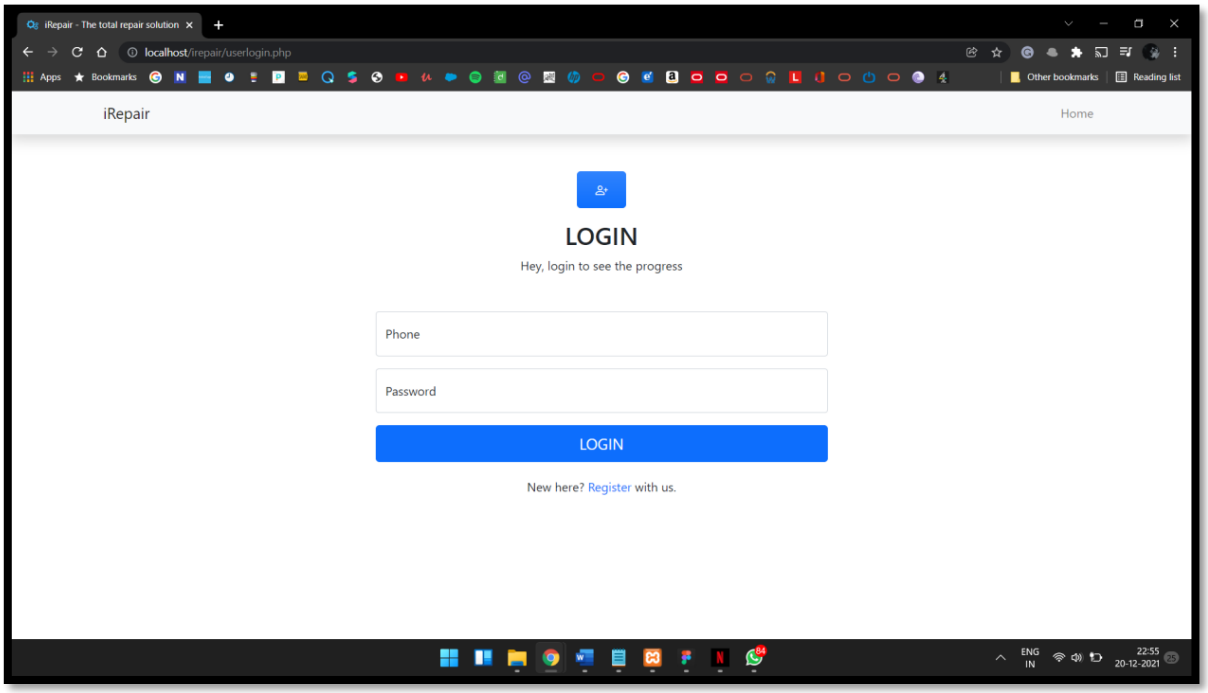


4.3.2 OUTPUT DESIGN

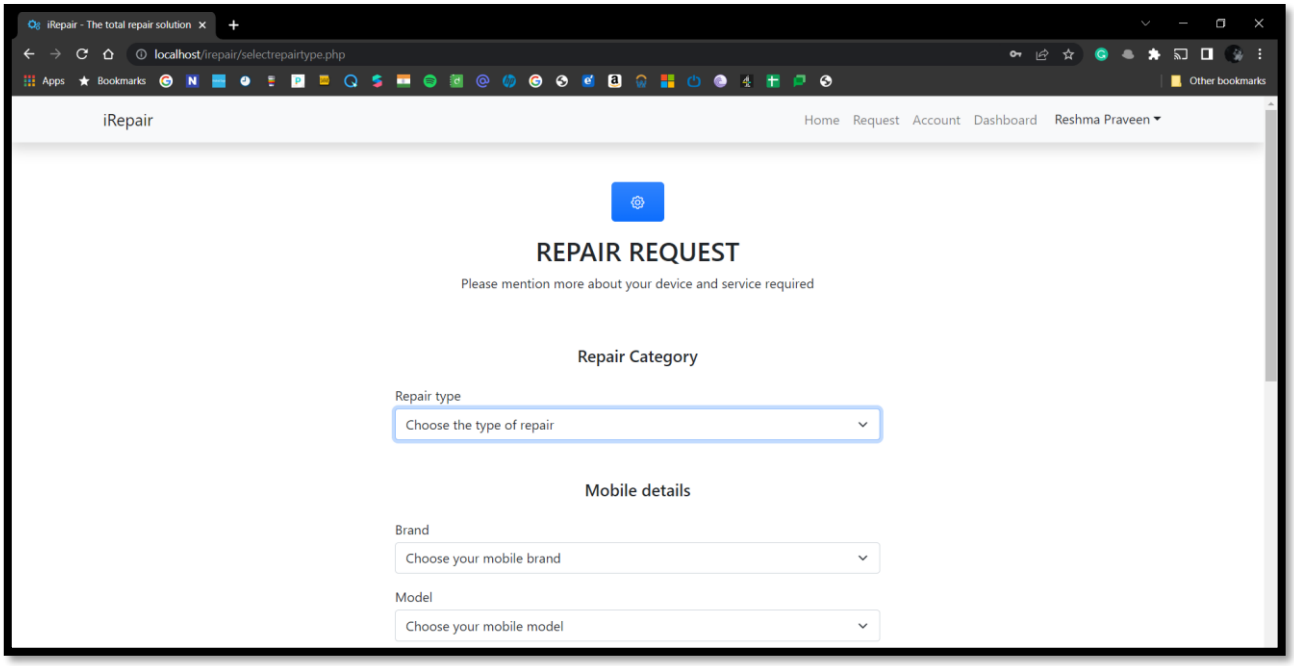
User Login

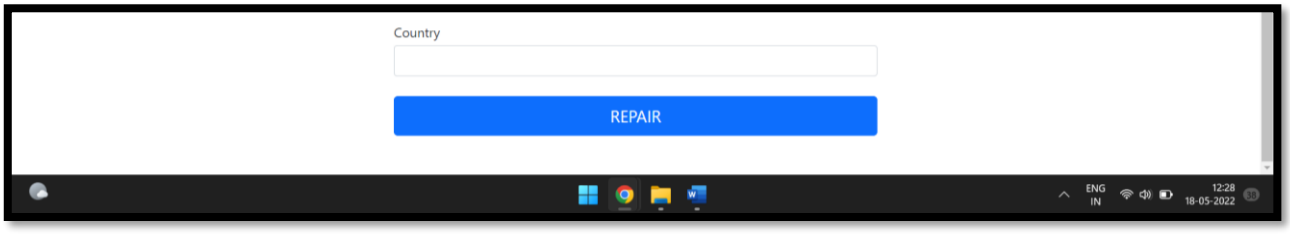


User Registration



Repair request





4.4 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.

4.4.1 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 assigneda 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.

4.4.2 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 modelling use two concepts, keys and relationships. A key uniquely identifies a row in atable. 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 doesnot 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 : 01

Table Name : tbl_user

Primary Key : id

Table Description: To store user registration information

Fieldname	Datatype	Size	Description
id	integer	11	Primary key
name	varchar	20	Name of user
email	varchar	20	Email of user
phone	bigint	10	Phone number of user
password	varchar	1000	Password
status	integer	1	Status

Table No : 02

Table Name : tbl_spares

Primary Key : id

Table Description: To store spare parts details

Fieldname	Datatype	Size	Description
id	integer	11	Primary key
category	varchar	20	Name of repair category
brand	varchar	30	Mobile brand name
model	varchar	30	Mobile model name
price	integer	11	Price of the spare
quantity	integer	11	Stocked spare count
requirement	integer	11	Required spare count
status	integer	1	State

Table No : 03

Table Name : tbl_repairdetails

Primary Key : id

Table Description: To store user login information

Fieldname	Datatype	Size	Description
id	integer	11	Primary key
username	varchar	20	Name of user
useremail	varchar	20	Email of user
userphone	bigint	10	Phone number of user
repairtype	varchar	1000	Password
brand	integer	1	Mobile phone brand
model	varchar	20	Mobile phone mobile
spare	varchar	200	ID of spare part used
imei	bigint	20	Mobile IMEI number
rate	integer	10	Repair cost
payment_status	integer	11	Status of payment
addressline1	varchar	50	Address line 1
addressline2	varchar	50	Address line 2
city	varchar	20	Name of city
state	varchar	20	Name of State
zip	integer	10	Zip code
country	varchar	20	Name of country
date	varchar	50	Drop date
rstate	integer	1	Repair state
state	integer	1	State

CHAPTER 5

SYSTEM 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 term's 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 appears to be working according to the specification, that performance requirement appears to havebeen 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 beanticipated and error handling paths set up to reroute or cleanly terminate processing whenan 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 andrun individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.1.1 Test Case 1

Test Case 1

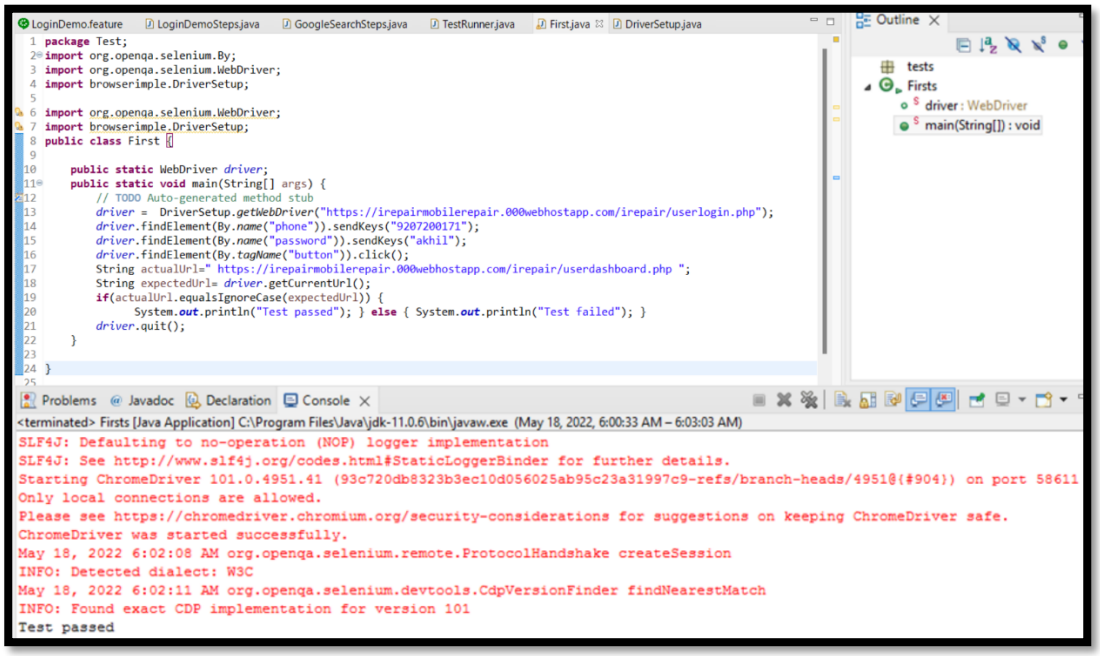
Project Name: Mobile Repair Management Website					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By: Kevin Liza Alex**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 17-05-2022		
Module Name: Login Screen			**Test Executed By: Ms. Gloriya Mathew**		
Test Title: Verify login with validphone number and password			**Test Execution Date:** 18-05-2022		
Description: Test the Login Page					
Pre-Condition: User should have valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass

2	Provide Valid Phone Number	Phone number: 9207458668	User should beable to Login	User Logged inand navigated to user Dashboard with records	Pass
3	Provide Valid Password	Password: reshma			
4	Click on Sign In button				
5	Provide Invalid Phone number or password	Phone number: 9447279392 Password: User1234	User shouldnot be able to Login	Message for enter valid phone number or password displayed	Pass
6	Provide Null Phone number or Password	Phone number: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

Code:

```
package tests;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimple.DriverSetup;
public class First {
	public static WebDriver driver;
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		driver = DriverSetup.getWebDriver("https://irepairmobilerepair.000webhostapp.com/irepair/userlogin.php");
		driver.findElement(By.name("phone")).sendKeys("9207200171");
		driver.findElement(By.name("password")).sendKeys("akhil");
		driver.findElement(By.tagName("button")).click();
		String actualUrl="https://irepairmobilerepair.000webhostapp.com/irepair/userdashboard.php ";
		String expectedUrl= driver.getCurrentUrl();
		if(actualUrl.equalsIgnoreCase(expectedUrl)) {
```

```
                    System.out.println("Test passed"); } else { System.out.println("Test failed"); }
                driver.quit();
            }

}
```



5.2.1.2 Test Case 2

Test Case 2

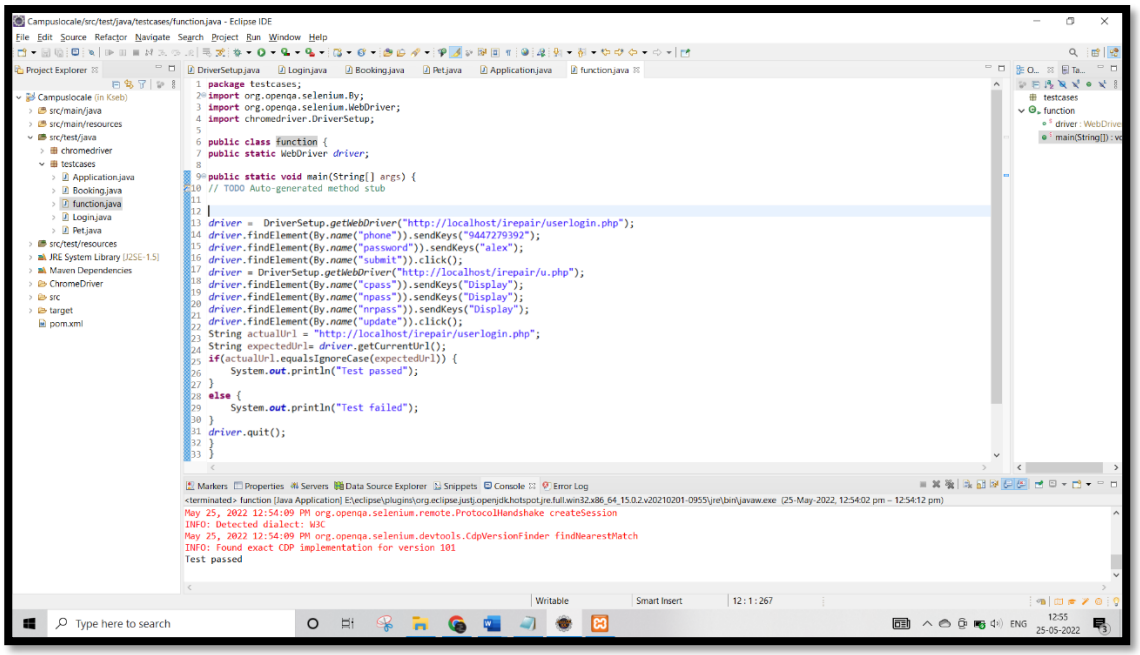
Project Name: Mobile Repair Management Website					
Login Test Case					
Test Case ID: Fun_2			**Test Designed By: Kevin Liza Alex**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 25-05-2022		
Module Name: User			**Test Executed By: Ms. Gloriya Mathew**		
Test Title: Submit repair request			**Test Execution Date:** 25-05-2022		
Description: To submit a repair request to process a repair job					
Pre-Condition: User should be logged in to submit a repair request					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation toLogin Page		Login Page should be	Login page displayed	Pass

			displayed		
2	Provide Valid Phone Number	Phone number: 9447279392	User should beable to Login	User Logged inand navigated to user Dashboard with records	Pass
3	Provide Valid Password	Password: alex			
4	Click on Sign In button				
5	Navigate to User account page		User page should be displayed	User page is displayed	Pass
6	Provide a current user password	Password: alex	User should be able to input value	User should be able to input values and redirect to user page.	Pass
7	Provide new password	New password: alex12			
8	Retype new password	Reenter new password: alex12			
9	Click on Update button				
Post-Condition: User password is validated with database and successfully redirect into account page.The Account session details are logged in database.					

Code:

```
package testcases;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import chromedriver.DriverSetup;


public class Function {

public static WebDriver driver;

public static void main(String[] args) {
```

```
// TODO Auto-generated method stub
driver =  DriverSetup.getWebDriver("http://localhost/irepair/userlogin.php");
driver.findElement(By.name("phone")).sendKeys("9447279392");
driver.findElement(By.name("password")).sendKeys("alex");
driver.findElement(By.name("submit")).click();
driver = DriverSetup.getWebDriver("http://localhost/irepair/u.php");
driver.findElement(By.name("cpass")).sendKeys("Display");
driver.findElement(By.name("npass")).sendKeys("Display");
driver.findElement(By.name("nrpass")).sendKeys("Display");
driver.findElement(By.name("update")).click();
String actualUrl = "http://localhost/irepair/userlogin.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
}
else {
System.out.println("Test failed");
}
driver.quit();
}
}
```



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 objectiveis 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. Moreover 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 BlackBox 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 thatwillfully 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 changeover.

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 is equipped with all the basic functionality for users such as submitting their mobile phones to repair, to track its repair status to make payment for successful repairing. User can either drop their phone at shop or the technician will contact once the user submits the repair request. User can use the application to satisfy their basic requirement of checking the status of their repairs and completion of payment. Technician makes this function possible by proper updating of the repair status of each repair from their side. So that the user can check the status from their own individual page after successful login. The user has individual dashboard with each repair listed with repair details updated from the technician side. Each technician has the provision to pick from a pool of repair requests where all the repair jobs unassigned to any technician are list. From there each technician after successful login can fetch any job and manage individually. Also, the technician has provision to request for stock updating in case of out-of-stock scenario and the inventory manager has to update the stock requirement submitted by the technician. And the admin has the provision to manage the over all interaction of each user to the system.

7.2 FUTURE SCOPE

The system has more future scope to enhance the functionality of the existing proposed system current. The following are the major updates user consideration to be implemented to enhance the user interaction of each user a further more. Even though basic functionalities are implemented addition of the following updates gives the users the edge to prefer the web application over physical interaction with the shop for mobile repairs.

- Enhanced pickup and drop: Currently the system lack the provision to manage the pickup and delivery with ease. The proposed scope enables the interaction of the user with system and the technician in such a manner that the technician will pick up the phones to repair from the user and the rest of the processes progress as earlier.

- Dedicated warranty control: Currently there is no provision to manage the warranty of the spare part replaced. So, it is necessary to elevate a closed repair job and provide additional support. Sometimes the arise a situation where additional repair or replacement of a particular spare is required.
- Option to control previous repairs: For the case of any additional clarification for the recently done repair. If a user is not satisfied with the repair of if they need additional clarification, they should be able to process the same repair or elevate that particular repair.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- 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.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Customer

selectrepairtype.php

```
<?php
include("include/header.php");
include("include/dbconnect.php");
if (isset($_SESSION["id"]) != session_id()) {

        echo ("<script LANGUAGE='JavaScript'>
      window.alert('Please login to continue!');
      window.location.href='userlogin.php';
    </script>");
} else {
?>
        <style>
                @media all and (min-width: 992px) {

                        .navbar .nav-item .dropdown-menu {
                                display: none;
                        }

                        /* .navbar .nav-item:hover .nav-link {} */

                        .navbar .nav-item:hover .dropdown-menu {
                                display: block;
                        }

                        .navbar .nav-item .dropdown-menu {
                                margin-top: 0;
                        }
                }
        </style>

        <!-- <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
                <ul class="navbar-nav">
                        <li class="nav-item dropdown">
                                <a class="nav-link dropdown-toggle" href="#" data-bs-
toggle="dropdown"> Hover me </a>
                                <ul class="dropdown-menu">
                                        <li><a class="dropdown-item" href="#"> Submenu
item 1</a></li>
                                        <li><a class="dropdown-item" href="#"> Submenu
item 2 </a></li>
                                        <li><a class="dropdown-item" href="#"> Submenu
item 3 </a></li>
```

```
        </ul>
                        </li>
                </ul>
        </nav> -->

        <?php
        if (isset($_POST["repair"])) {
                include "include/dbconnect.php";

                $sid = $_SESSION["s_id"];
                $sname = $_SESSION["s_name"];
                $semail = $_SESSION["s_email"];
                $sphone = $_SESSION["s_phone"];

                $r_repairtype = $_POST["repairtype"];
                $r_brand = $_POST["brand"];
                $r_model = $_POST["model"];
                $r_imei = $_POST["imei"];
                $r_ad1 = $_POST["ad1"];
                $r_ad2 = $_POST["ad2"];
                $r_city = $_POST["city"];
                $r_state = $_POST["state"];
                $r_zip = $_POST["zip"];
                $r_country = $_POST["country"];
                $sql = "INSERT INTO tbl_repairdetails (username, userphone, useremail,
repairtype, brand, model, imei, addressline1, addressline2, city, state, zip, country, date)
VALUES ('$sname', '$sphone', '$semail', '$r_repairtype', '$r_brand', '$r_model', '$r_imei',
'$r_ad1', '$r_ad2', '$r_city', '$r_state', '$r_zip', '$r_country', now())";
                // echo $sql;
                $res = mysqli_query($conn, $sql);
                echo ("<script LANGUAGE='JavaScript'>
                window.alert('Pickup registered succesfully.');
                window.location.href='userdashboard.php';
                </script>");
        }
        ?>
        <script type="text/javascript">
                function valRegister() {
                        if (document.getElementById('repairtype').value.length == 0 ||
document.getElementById('brand').value.length == 0 ||
document.getElementById('model').value.length == 0 ||
document.getElementById('imei').value.length == 0 ||
document.getElementById('price').value.length == 0 ||
document.getElementById('pickup').value.length == 0) {
                                alert('Fill all the fields');
                                return false;
                        } else if (document.getElementById('imei').value.length > 15 ||
document.getElementById('imei').value.length < 15 ||
isNaN(document.getElementById('imei').value)) {
```

```
alert("IMEI number should be a number of length 15!");
                        return false
                    } else {
                        return true;
                    }
                }
            </script>

            <div class="my-5">
                <div class="container my-5 px-5">
                    <div class="text-center mb-5">
                        <div class="bg-primary bg-gradient text-white rounded-3 mb-
3" style="width: 4rem; height: 3rem; display: inline-flex; align-items: center; justify-content:
center;"><i class="bi bi-gear"></i>
                        </div>
                        <h2 class="">REPAIR REQUEST</h2>
                        <p class="sectionheaddescription">Please mention more about
your device and service required</p>
                    </div>
                    <div class="row gx-5 justify-content-center">
                        <div class="col-lg-6">
                            <form id="contactForm" method="post"
onsubmit="return valRegister()">

                                <h5 class="text-center mb-4 mt-3">Repair
Category</h5>

                                <div class="form-group mb-3">
                                    <label class="mb-1"
for="exampleFormControlInput1">Repair type</label>
                                    <select class="form-select" aria-
label="Default select example" name="repairtype" id="repairtype" required />
                                    <option selected>Choose the type of
repair</option>
                                    <option
value="Display">Display</option>
                                    <option
value="Battery">Battery</option>
                                    <option value="Board">Board</option>
                                    </select>
                                </div>

                                <h5 class=" text-center mb-4 mt-5">Mobile
details</h5>

                                <div class="form-group mb-3">
                                    <label class="mb-1"
for="exampleFormControlInput1">Brand</label>
                                    <select class="form-select" aria-
```

```
label="Default select example" name="brand" id="brand"
onchange="selectmodel(this.id,'model')">
                                        <option selected>Choose your
mobile brand</option>
                                        <option
value="Apple">Apple</option>
                                        <option
value="Pixel">Pixel</option>
                                        <option
value="Samsung">Samsung</option>
                                        <option
value="Xiaomi">Xiaomi</option>
                                        <option
value="Realme">Realme</option>
                                        <option
value="Vivo">Vivo</option>
                                        <option
value="Oppo">Oppo</option>
                                        <option
value="Honor">Honor</option>
                                    </select>
                                </div>

                                <div class="form-group mb-3">
                                    <label class="mb-1"
for="exampleFormControlInput1">Model</label>
                                    <select class="form-select" aria-
label="Default select example" name="model" id="model">
                                        <option selected>Choose your
mobile model</option>
                                        <!-- <option value="iPhone 13
Pro">iPhone 13 Pro</option>
                                        <option value="Pixel 6">Pixel
6</option>
                                        <option value="Samsung
S21">Samsung S21</option> -->
                                    </select>
                                </div>

                                <div class="form-group mb-3">
                                    <div class="form-group mb-4">
                                        <label for="usr" class="mb-
1">IMEI</label>
                                        <input type="number"
class="form-control" id="imei" name="imei" />
                                    </div>
                                </div>

                                <!-- <div class="d-grid text-center">
```

```
<input type="button" name="checkimei" class="btn btn-primary w-25 btn-sm"
id="checkimei" value="CHECK">
                                        </div> -->

                                        <?php
                                        if (array_key_exists('checkimei', $_POST)) {
                                                $r_imei = $_POST["imei"];
                                                $sql2 = "SELECT * FROM tbl_user
WHERE  imei ='$r_imei'";

                                                $result = mysqli_query($conn, $sql2);
                                                if (mysqli_num_rows($result) > 0) {
                                                        echo ("<script
LANGUAGE='JavaScript'>
                                                window.alert('A repair already exist for this
phone. Can not continue with request.');

                                                </script>");
                                                } else {
                                                        echo ("<script
LANGUAGE='JavaScript'>
                                                window.alert('Incorrect current password!');
                                                window.location.href='userpage.php';
                                                </script>");
                                                }
                                        }

                                        ?>


                                        <h5 class="text-center mb-4 mt-5">Pickup
details</h5>



                                        <div class="form-group mb-4" id="mm">
                                                <div class="form-group mb-4">
                                                        <label for="usr" class="mb-
1">Address Line 1</label>
                                                        <input type="text" class="form-
control" id="ad1" name="ad1">
                                                </div>
                                                <div class="form-group mb-4">
                                                        <label for="usr" class="mb-
1">Address Line 2</label>
                                                        <input type="text" class="form-
control" id="ad2" name="ad2">
                                                </div>
                                                <div class="form-group mb-4">
                                                        <label for="usr" class="mb-
```

```
1">City / Town</label>
                                        <input type="text" class="form-
control" id="city" name="city">
                                    </div>
                                    <div class="form-group mb-4">
                                        <label for="usr" class="mb-
1">State / Province / Region</label>
                                        <input type="text" class="form-
control" id="state" name="state">
                                    </div>
                                    <div class="form-group mb-4">
                                        <label for="usr" class="mb-
1">Zip / Postal Code</label>
                                        <input type="number"
class="form-control" id="zip" name="zip">
                                    </div>
                                    <div class="form-group mb-4">
                                        <label for="usr" class="mb-
1">Country</label>
                                        <input type="text" class="form-
control" id="country" name="country">
                                    </div>

                                </div>

                                <div class="form-group mb-4" id="nn"
style="display:none">
                                    <div class="form-group mb-4">
                                        <label for="usr" class="mb-
1">Drop date</label>
                                        <input type="date" name="bod"
id="cmbQua" min="<?php date_default_timezone_set('Asia/Kolkata');

                                        $rrdate = date('dd-mm-yy');

                                        echo $rrdate; ?>" max="2022-
01-01" />
                                    </div>

                                </div>

                                <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
                                <script>
                                    function selectmodel(s1, s2) {
                                        var s1 =
document.getElementById(s1);
                                        var s2 =
document.getElementById(s2);
```

```
s2.innerHTML = "";
                                                    if (s1.value == "Apple") {
                                                        var optionArray =
['iPhone 13 Pro Max|iPhone 13 Pro Max', 'iPhone 13 Pro|iPhone 13 Pro', 'iPhone 13|iPhone
13'];
                                                        for (var option in
optionArray) {
                                                            var pair =
optionArray[option].split("|");
                                                            var newoption =
document.createElement("option");
                                                            newoption.value
= pair[0];
        newoption.innerHTML = pair[1];
        s2.options.add(newoption);
                                                        }
                                                    } else if (s1.value == "Pixel") {
                                                        var optionArray = ['Pixel
6 Pro|Pixel 6 Pro', 'Pixel 6|Pixel 6'];
                                                        for (var option in
optionArray) {
                                                            var pair =
optionArray[option].split("|");
                                                            var newoption =
document.createElement("option");
                                                            newoption.value
= pair[0];
        newoption.innerHTML = pair[1];
        s2.options.add(newoption);
                                                        }
                                                    } else if (s1.value ==
"Samsung") {
                                                        var optionArray =
['Samsung S20 Ultra|Samsung S20 Ultra', 'Samsung S20+|Samsung S20+', 'Samsung
S20|Samsung S20'];
                                                        for (var option in
optionArray) {
                                                            var pair =
optionArray[option].split("|");
                                                            var newoption =
document.createElement("option");
                                                            newoption.value
= pair[0];
        newoption.innerHTML = pair[1];
```

```
        s2.options.add(newoption);
                                                        }
                                                    }

                                                }

                                                function calPrice() {
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Apple") {

        document.getElementById("price").value = "9000";
                                                    }
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Pixel") {

        document.getElementById("price").value = "8000";
                                                    }
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Samsung") {

        document.getElementById("price").value = "7000";
                                                    }
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Xiaomi") {

        document.getElementById("price").value = "7500";
                                                    }
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Realme") {

        document.getElementById("price").value = "6000";
                                                    }
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Vivo") {

        document.getElementById("price").value = "6500";
                                                    }
                                                    if
(document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Oppo") {

        document.getElementById("price").value = "6500";
                                                    }
```

```
                    if (document.getElementById("repairtype").value == "Display" &&
document.getElementById("brand").value == "Honor") {

        document.getElementById("price").value = "5500";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Apple") {

        document.getElementById("price").value = "9000";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Pixel") {

        document.getElementById("price").value = "8000";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Samsung") {

        document.getElementById("price").value = "7000";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Xiaomi") {

        document.getElementById("price").value = "7500";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Realme") {

        document.getElementById("price").value = "6000";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Vivo") {

        document.getElementById("price").value = "6500";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Oppo") {

        document.getElementById("price").value = "6500";
                                                                }
                                                                if
(document.getElementById("repairtype").value == "Battery" &&
document.getElementById("brand").value == "Honor") {
```

```
        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Apple") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Pixel") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Samsung") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Xiaomi") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Realme") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Vivo") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Oppo") {

        document.getElementById("price").value = "5500";
                                                        }
                                                        if
(document.getElementById("repairtype").value == "Board" &&
document.getElementById("brand").value == "Honor") {

        document.getElementById("price").value = "5500";
```

```
                }
                                                        // $aa =
document.getElementById("brand").value;
                                                        // alert($aa);
                                                }
                                        </script>

                                        <!-- Submit success message-->
                                        <!---->
                                        <!-- This is what your users will see when the form-->
                                        <!-- has successfully submitted-->
                                        <div class="d-none" id="submitSuccessMessage">
                                                <div class="text-center mb-3">
                                                        <div class="fw-bolder">Form
submission successful!</div>
                                                        To activate this form, sign up at
                                                        <br />
                                                        <a
href="https://startbootstrap.com/solution/contact-
forms">https://startbootstrap.com/solution/contact-forms</a>
                                                </div>
                                        </div>
                                        <!-- Submit error message-->
                                        <!---->
                                        <!-- This is what your users will see when there is-->
                                        <!-- an error submitting the form-->
                                        <div class="d-none" id="submitErrorMessage">
                                                <div class="text-center text-danger mb-3">Error
sending message!</div>
                                        </div>
                                        <!-- Submit Button-->
                                        <div class="d-grid">
                                                <input type="submit" name="repair" class="btn
btn-primary btn-lg" value="REPAIR">
                                        </div>
                                </form>
                        </div>
                </div>
        </div>

        </body>

        </html>
        <script>
                $("#repairtype").focus();
        </script>
<?php
}
?>
```

adminrepairstatus.php

```
<?php
include("include/header.php");
include("include/dbconnect.php");
if (isset($_SESSION["id"]) != session_id()) {

  echo ("<script LANGUAGE='JavaScript'>
        window.alert('Login first');
        window.location.href='userlogin.php';
      </script>");
} else {
?>
  <style>
    .css-serial {
      counter-reset: serial-number;
      /* Set the serial number counter to 0 */
    }

    .css-serial td:first-child:before {
      counter-increment: serial-number;
      /* Increment the serial number counter */
      content: counter(serial-number);
      /* Display the counter */
    }
  </style>
  <h5 class="text-center mb-4 mt-5">MANAGE REPAIRS</h5>

  <div class="container mb-5">
    <table class="table table-bordered css-serial">
      <thead class="">
        <tr>
          <th scope="col">#</th>
          <th scope="col">Date</th>
          <th scope="col">Repair</th>
          <th scope="col">Mobile</th>
          <th scope="col">Spare</th>
          <th scope="col">Spare update</th>
          <th scope="col">IMEI</th>
          <th scope="col">Rate</th>
          <th scope="col">Status</th>
          <th scope="col">Status update</th>
          <th scope="col">Details</th>
        </tr>
      </thead>
      <tbody>

        <?php
        include("include/dbconnect.php");
        $sql = "SELECT * FROM tbl_repairdetails order by date desc";
```

```
        $result = mysqli_query($conn, $sql);
        $rows = mysqli_num_rows($result);
        if (mysqli_num_rows($result) > 0) {
          while ($row = mysqli_fetch_assoc($result)) {
            echo "
              <tr>
              <td scope='row'></th>
              <td scope='row'>" . $row['date'] . "</th>
              <td scope='row'>" . $row['repairtype'] . "</th>
              <td scope='row'>" . $row['brand'] . " " . $row['model'] . "</th>
              <td scope='row'>";
            if ($row['spare'] != 'Empty') {
              $fecthSpareName = "SELECT model FROM `tbl_spares` WHERE `id`=" . $row['spare'] .
"";
              $fecthSpareNameResult = mysqli_query($conn, $fecthSpareName);
              if (mysqli_num_rows($fecthSpareNameResult) > 0) {
                while ($spareName = mysqli_fetch_assoc($fecthSpareNameResult)) {
                  echo $spareName['model'];
                }
              }
            }

            echo " </th>
              <td scope='row'>
              <form method='post' action='updatespare.php'>
                        <select class='custom-select' id='inputGroupSelect01' name='spareselect'>
                           <option selected>Choose...</option>";
            $repairType = $row['repairtype'];
            $fecthGradesql = "SELECT * FROM `tbl_spares` WHERE `category`='$repairType'";
            $fecthGradesqlResult = mysqli_query($conn, $fecthGradesql);
            if (mysqli_num_rows($fecthGradesqlResult) > 0) {
              while ($Typerow = mysqli_fetch_assoc($fecthGradesqlResult)) {

                echo "<option value='" . $Typerow['id'] . "'>" . $Typerow['model'] . "</option>";
              }
            }
            echo "       </select>
                         <br><button class='btn btn-secondary' value='" . $row['id'] . "'
name='updateSpareStatus'>Update</button>
                        </form>
              </th>
              <td scope='row'>" . $row['imei'] . "</th>
              <td scope='row'><form method=post><input class='w-50 form-control' type='text'
value='" . $row['rate'] . "' name='rateval'></input> <button class='btn btn-secondary' value='" .
$row['id'] . "' name=ratebtn>Update</button></form></th>
              <td scope='row'>";
            if ($row['rstate'] == 0) {
              echo "<p class='card-text'>PENDING</p>";
            } else if ($row['rstate'] == 1) {
              echo "<p class='card-text text-secondary'>RECIEVED</p>";
            } else if ($row['rstate'] == 2) {
```

```
            echo "<p class='card-text text-warning'>PROGRESSING</p>";
          } else if ($row['rstate'] == 3) {
            echo "<p class='card-text text-success'>COMPLETED</p>";
          } else if ($row['rstate'] == 4) {
            echo "<p class='card-text text-success'>OUT FOR DELIVERY</p>";
          } else if ($row['rstate'] == 5) {
            echo "<p class='card-text text-success'>WAITING FOR PICKUP</p>";
          } else if ($row['rstate'] == 6) {
            echo "<p class='card-text text-success'>CLOSED</p>";
          } else {
          }
          echo "</th>";
          echo "<td>

                    <form method=post>
                    <select class='custom-select' id='inputGroupSelect01' name='status'>
                      <option selected>Choose...</option>
                      <option value='0'>Pending</option>
                      <option value='1'>Recieved</option>
                      <option value='2'>Progress</option>
                      <option value='3'>Completed</option>
                      <option value='4'>Out for delivery</option>
                      <option value='5'>Waiting for pickup</option>
                      <option value='6'>Closed</option>
                    </select>
                    <button class='btn btn-secondary' value='" . $row['id'] . "'
name=rbtn>Update</button>
                    </form>
                    </td>";



          echo "<td scope='row'><a class='btn btn-light' name='mbtn' value='hai'
href='adminuserdetail.php?id=" . $row['id'] . "'>Show</a></th>
          </tr>";
        }
      }
      ?>

    </tbody>
  </table>

 </div>

 <?php
 if (isset($_POST['rbtn'])) {
  include "include/dbconnect.php";
  $id = $_POST['rbtn'];
  $sval = $_POST['status'];
  $sql = "UPDATE `tbl_repairdetails` SET `rstate`= '$sval' WHERE id = $id";
  //echo $sql;
```

```
    $result = mysqli_query($conn, $sql);
    //echo "hai";
   }

   if (isset($_POST['ratebtn'])) {
    include "include/dbconnect.php";
    $id = $_POST['ratebtn'];
    $sval = $_POST['rateval'];
    $sql = "UPDATE `tbl_repairdetails` SET `rate`= '$sval' WHERE id = $id";
    //echo $sql;

    $result = mysqli_query($conn, $sql);
    //echo "hai";
   }
   ?>

   </body>

   </html>
<?php
}
?>
```

inventorymanage.php

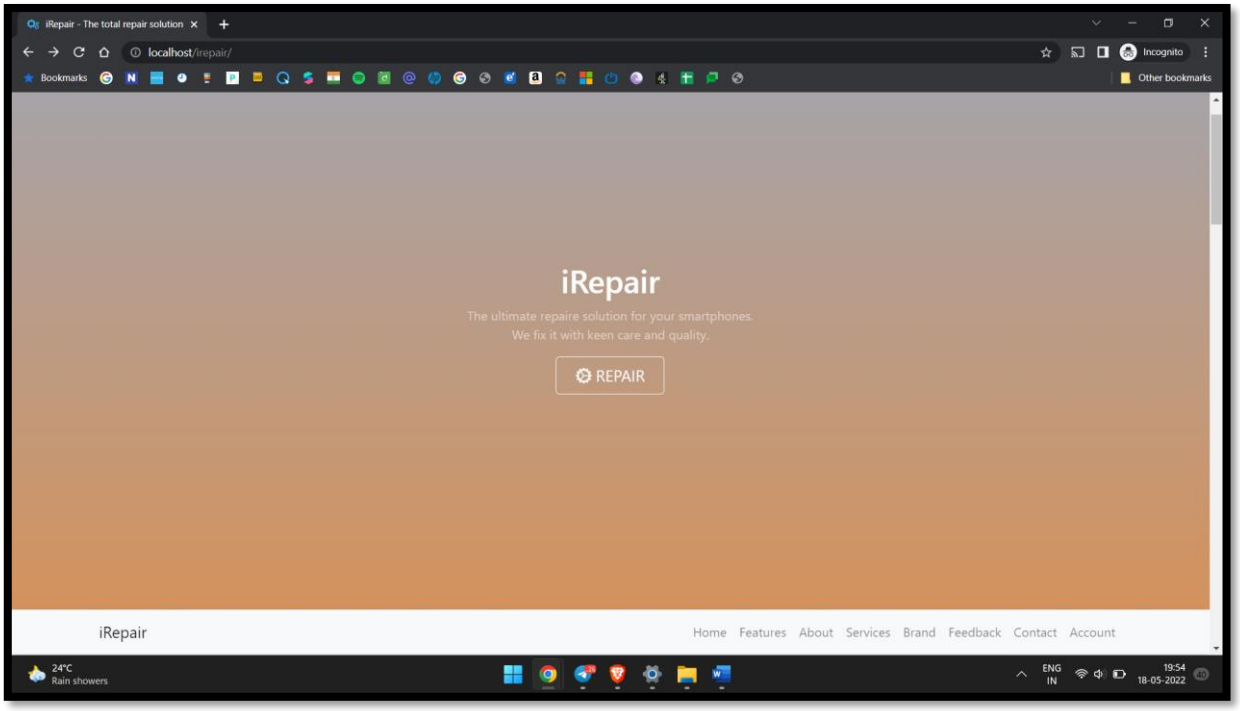
```
<?php
include("include/header.php");
include("include/dbconnect.php");
if (isset($_SESSION["id"]) != session_id()) {

    echo ("<script LANGUAGE='JavaScript'>
        window.alert('Login first');
        window.location.href='userlogin.php';
      </script>");
} else {
?>
    <style>
        .css-serial {
            counter-reset: serial-number;
            /* Set the serial number counter to 0 */
        }

        .css-serial td:first-child:before {
            counter-increment: serial-number;
            /* Increment the serial number counter */
            content: counter(serial-number);
            /* Display the counter */
        }
    </style>
```

```
<h5 class="text-center mb-4 mt-5">MANAGE STOCK REQUESTS</h5>

<div class="container mb-5">
   <table class="table table-bordered css-serial">
      <thead class="">
         <tr>
            <th scope="col">#</th>
            <th scope="col">Brand</th>
            <th scope="col">Model</th>
            <th scope="col">Price</th>
            <th scope="col">Quantity</th>
            <th scope="col">Requirement</th>
            <th scope="col">Status</th>
            <th scope="col">Stock update</th>
         </tr>
      </thead>
      <tbody>

         <?php
         include("include/dbconnect.php");
         $sql = "SELECT * FROM tbl_spares";
         $result = mysqli_query($conn, $sql);
         $rows = mysqli_num_rows($result);
         if (mysqli_num_rows($result) > 0) {
            while ($row = mysqli_fetch_assoc($result)) {
               echo "
        <tr>
        <td scope='row'></th>
        <td scope='row'>" . $row['brand'] . "</th>
        <td scope='row'>" . $row['model'] . "</th>
        <td scope='row'>" . $row['price'] . "</th>
        <td scope='row'>" . $row['quantity'] . "</th>
        <td scope='row'>" . $row['requirement'] . "</th>
        <td scope='row'>";
               if ($row['requirement'] == 0 and $row['quantity'] > 0) {
                  echo "<p class='card-text text-success'>STOCKED WITHOUT
REQUIREMENT</p>";
               } else if ($row['requirement'] > 0 and $row['quantity'] != 0) {
                  echo "<p class='card-text text-warning'>STOCKED WITH
REQUIREMENT</p>";
               } else if ($row['requirement'] == 0 and $row['quantity'] == 0) {
                  echo "<p class='card-text text-danger'>URGENT STOCK REQUIRED</p>";
               } else {
               }
               echo "</th>";
               echo "<td>

                  <form method=post>
```

```
 <input name='status' id='inputGroupSelect01' type='number' required />

                        <button class='btn btn-secondary' value="' . $row['id'] . "'
name=rbtn>Update</button>
                    </form>
                    </td>";

                echo "<td scope='row'><a class='btn btn-light' name='mbtn' value='hai'
href='adminuserdetail.php?id=" . $row['id'] . "'>Show</a></th>
        </tr>";
            }
        }
        ?>

        </tbody>
    </table>

</div>

<?php
if (isset($_POST['rbtn'])) {
    include "include/dbconnect.php";
    $id = $_POST['rbtn'];
    $sval = $_POST['status'];
    $sql = "UPDATE `tbl_spares` SET `requirement`= `requirement`-'$sval' , `quantity` =
`quantity` + '$sval' WHERE id = $id";
    //echo $sql;

    $result = mysqli_query($conn, $sql);
    //echo "hai";
    if ($result) {
        echo ("<script LANGUAGE='JavaScript'>
        window.location.href='adminspares.php';
  </script>");
    }
}
?>

</body>

</html>
<?php
}
?>
```

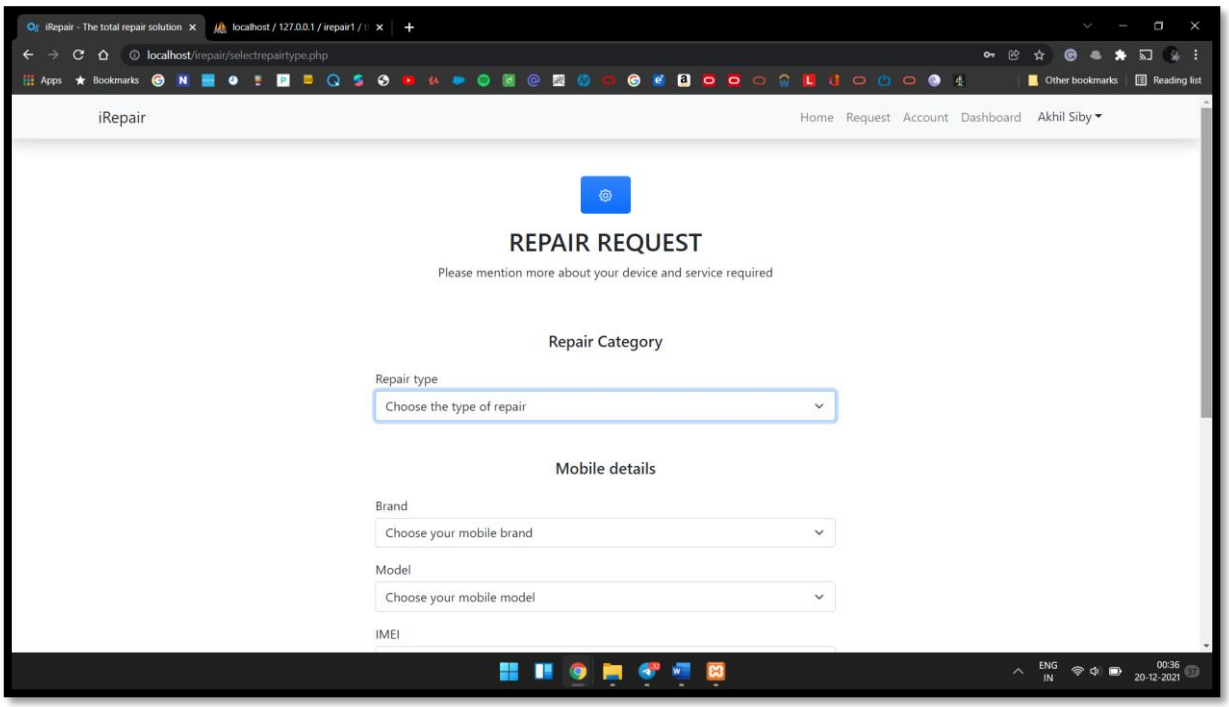
9.2 SCREENSHOTS

Landing Page

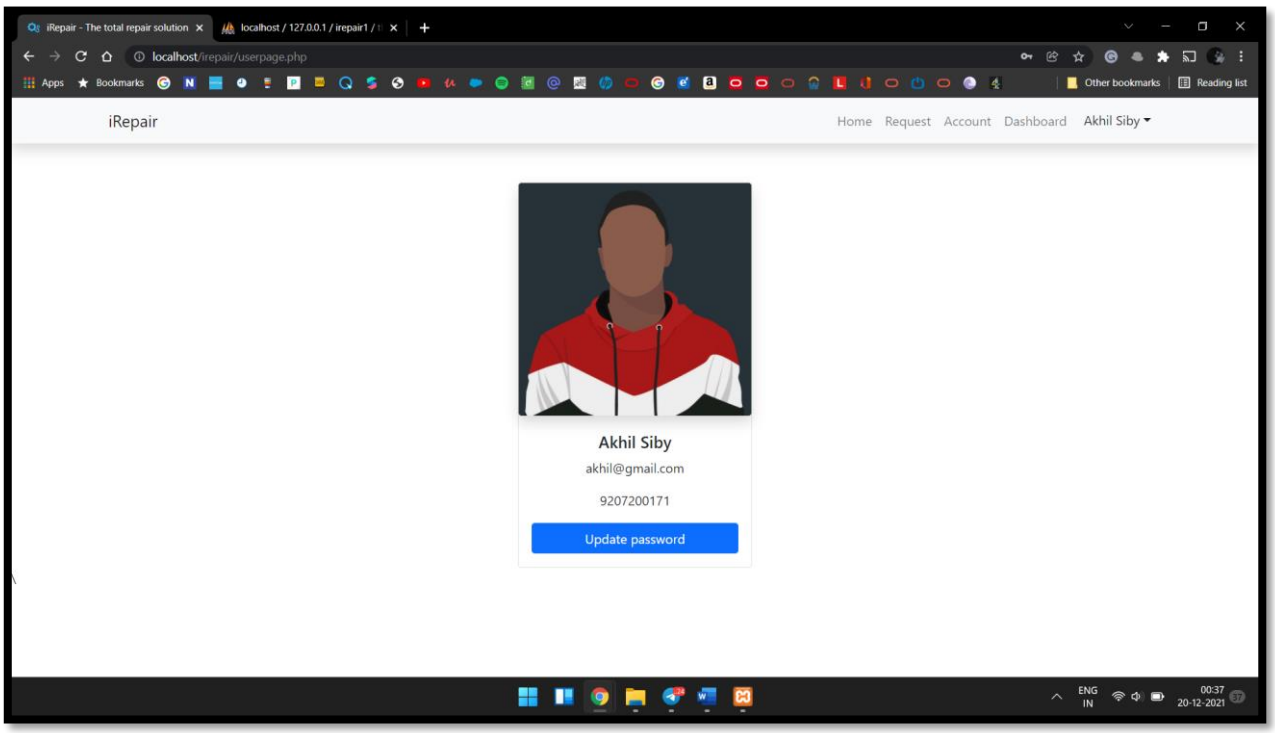


USER PAGES

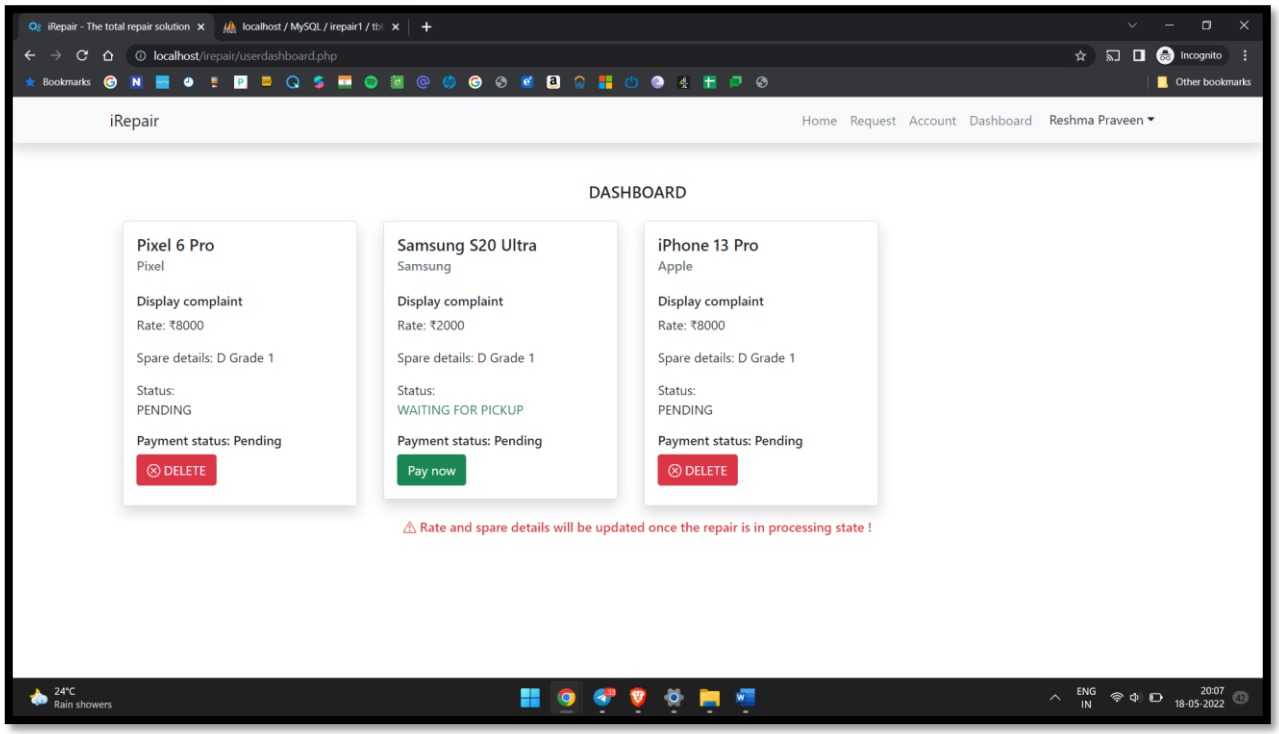
Repair Request Page



Account Page

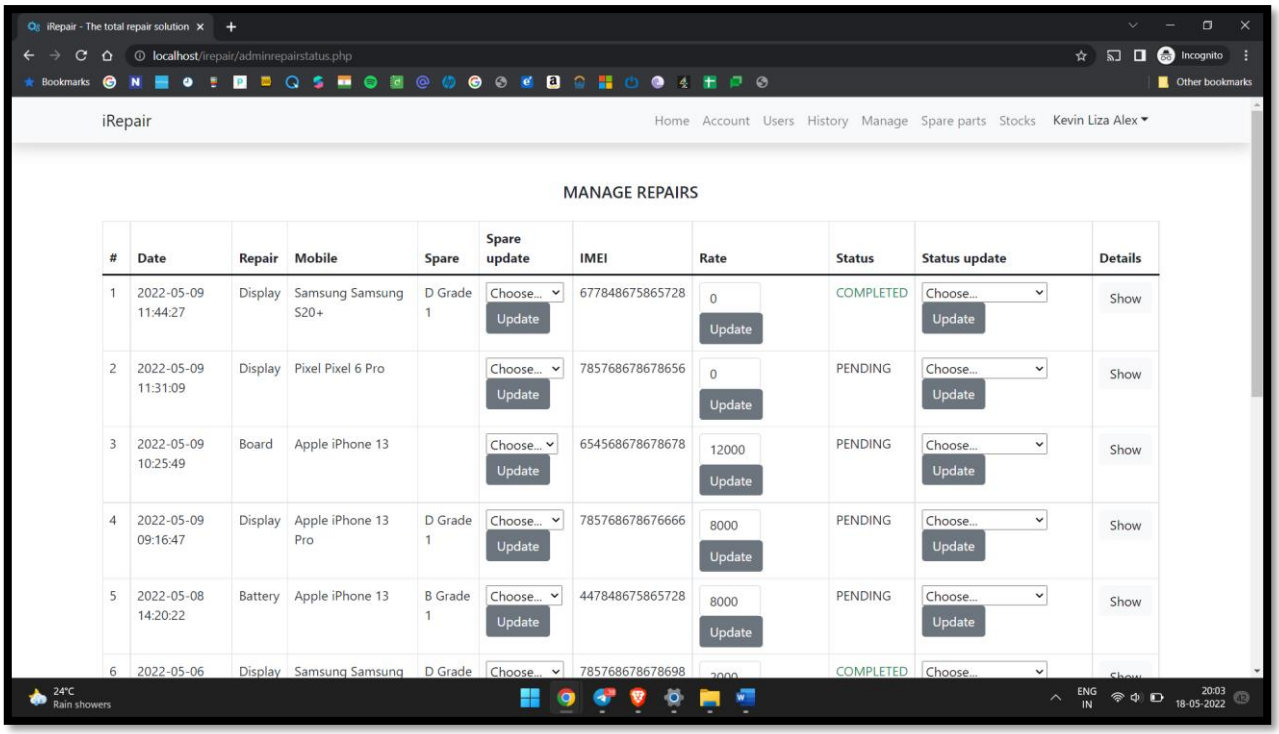


Dashboard



TECHNICIAN PAGES

Manage Repairs Page



User Management Page