SMART REPAIR

Project Report Submitted By

BENSY BENNY

Reg. No:AJC20MCA-2033

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

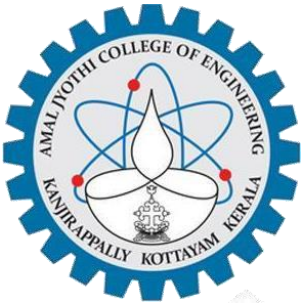
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**SMART REPAIR"** is the bonafide work of **BENSY BENNY (Reg.No:AJC20MCA-2033)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Ms. Sona Maria Sebastian
Internal Guide

Ms. Nimmy Francis
Coordinator

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

External Examiner

DECLARATION

I hereby declare that the project report **"SMART REPAIR"** 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 Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date: 21/07/2022 **BENSY BENNY**

KANJIRAPPALLY **Reg. No: AJC20MCA-2033**

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. Lilly Kutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev.Fr. Dr. Rubin Thottupurathu Jose** and **Ms**. **Nimmy Francis** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Sona Maria Sebastian** 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.

BENSY BENNY

ABSTRACT

The SMART REPAIR is a Online Complaint Service system has been developed to anyone can post their complaints by searching for the mechanics that are located nearby their places and solving their complaints at the door step just by one click. This Project is very useful for everyone who wants any complaint services because now a day's everyone wants to save time and shot out their problems within time, therefore, the services are very useful for people. In this Project, public can register complaints for Electrical Works, Plumbing, Carpentry Services, Water purifier, Washing machine, Refrigerator, Car Repair etc.

Customers can search the nearby mechanic shop and can register the complaint online. In this project identification and solution for the complaints given by the people, deal with them within the specified time limit is the main concept of the project. The complaints can be assigned to different mechanics and will be tracked. The Administrator can control all the activities in the system, for creating issues using registration, assign to a mechanic or a service engineer, check the service performance and examines whether the problem is Correct or not within a period. In registration, it should be open and assigned to a mechanic and he can update the status of the complaint. So this helps the peoples to find the solution for their problem in an easily manner.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
2.5	ADVANTAGES OF PROPOSED SYSTEM	7
3	REQUIREMENT ANALYSIS	9
3.1	FEASIBILITY STUDY	10
3.1.1	ECONOMICAL FEASIBILITY	10
3.1.2	TECHNICAL FEASIBILITY	11
3.1.3	BEHAVIORAL FEASIBILITY	11
3.2	SYSTEM SPECIFICATION	12
3.2.1	HARDWARE SPECIFICATION	12
3.2.2	SOFTWARE SPECIFICATION	12
3.3	SOFTWARE DESCRIPTION	12
3.3.1	PHP	12
3.3.2	MYSQL	13
4	SYSTEM DESIGN	15
4.1	INTRODUCTION	16
4.2	UML DIAGRAM	16
4.2.1	USE CASE DIAGRAM	17
4.2.2	SEQUENCE DIAGRAM	20
4.2.3	STATE CHART DIAGRAM	22
4.2.4	ACTIVITY DIAGRAM	22
4.2.5	CLASS DIAGRAM	24
4.2.6	OBJECT DIAGRAM	25
4.2.7	COMPONENT DIAGRAM	26
4.2.8	DEPLOYMENT DIAGRAM	27

4.3	**USER INTERFACE DESIGN**	**28**
4.3.1	**INPUT DESIGN**	**28**
4.3.2	**OUTPUT DESIGN**	**30**
4.4	**DATA BASE DESIGN**	**32**
5	**SYSTEM TESTING**	**39**
5.1	**INTRODUCTION**	**40**
5.2	**TEST PLAN**	**41**
5.2.1	**UNIT TESTING**	**41**
5.2.2	**INTEGRATION TESTING**	**42**
5.2.3	**VALIDATION TESTING**	**42**
5.2.4	**USER ACCEPTANCE TESTING**	**43**
5.2.5	**SELENIUM TESTING**	**44**
6	**IMPLEMENTATION**	**49**
6.1	**INTRODUCTION**	**50**
6.2	**IMPLEMENTATION PROCEDURE**	**50**
6.2.1	**USER TRAINING**	**51**
6.2.2	**TRAINING ON APPLICATION SOFTWARE**	**51**
6.2.3	**SYSTEM MAINTENANCE**	**51**
6.2.4	**HOSTING**	**52**
7	**CONCLUSION & FUTURE SCOPE**	**53**
7.1	**CONCLUSION**	**54**
7.2	**FUTURE SCOPE**	**54**
8	**BIBLIOGRAPHY**	**55**
9	**APPENDIX**	**57**
9.1	**SAMPLE CODE**	**58**
9.2	**SCREEN SHOTS**	**67**
9.3	**PLAGIARISM REPORT**	**81**

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

The Smart Repair System has been developed to anyone can post their complaints by searching for the mechanics that are located nearby their places and solving their complaints at the door step just by one click. The Smart Repair System has been developed to anyone can post their complaints by searching for the mechanics that are located nearby their places and solving their complaints at the door step just by one click. This Project is very useful for everyone who wants any complaint services because now a day's everyone wants to save time and shot out their problems within time, therefore, the services are very useful for people. In this Project, public can register complaints for Electrical Works, Plumbing, Carpentry Services, Water purifier, Washing machine, Refrigerator, Car Repair etc..Customers can search the nearby mechanic shop and can register the complaint online.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which provides complaint services available at anywhere with a fast manner. Its more time consume for all process

The system mainly includes 4 modules. They are:

1. User Module

- Login
- Register
- Post Complaint
- View complaint status
- Get Admin Contact details
- Manage Profile.
- Search area, city, state and country wise population.
- Search Plumber, Carpenter, Electrician, Ac Maintenance, Sweeper, Gardener , etc
- Feedback

2. Admin Module

- Generate id & password for officer
- Can create/delete an account.
- Can view the accounts.
- Add Customers
- Add business partners
- Update Employees
- Make Payment

3. Buisness Partner Module

- Manage Profile.
- Hire Employees
- Manage Employee.
- Assign Duty to Employee.
- View Job Requests History.
- Approve and Reject Job Requests.
- Update Profile

4. Employee Module

- Login
- Register
- Update Profile
- View Booking Service History
- Attend and Update Services

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the process of gathering and analysing data, identifying problems, and using the data to recommend system modifications. During this problem-solving process, there must be considerable communication between the system users and the system developers. A system analysis or research should be the first step in any system development process. The system is carefully inspected and evaluated. As an interrogator, the system analyst delves deeply into the operation of the present system. The system's input is recognised, and the system is viewed as a whole. The various processes can be connected to the organisational outputs. Finding the problem, identifying the relevant and crucial variables, weighing and merging the many components, and coming up with the best or, at the very least.

A complete study of the process which is necessary to thoroughly examine the process utilising a variety of techniques, including surveys and interview sessions. The data acquired by various sources needs to be thoroughly evaluated in order to draw a conclusion. The conclusion is knowing how the system functions. This system is known as the current one. Now, problem areas have been identified after a detailed examination of the current system. Now the designer takes on the role of a problem-solver and attempts to fix the problems the company is experiencing. The solutions are replaced with proposals. Therefore, the suggestion and the current system are compared analytically, and the best one is picked. The proposal is presented to the user with the option to accept or reject it. The proposal is presented to the user with the option to accept or reject it.

The process of acquiring and analysing data in order to use it for future system studies is known as preliminary study. Initial research is a problem-solving activity that necessitates close coordination between system users and developers. It conducts a number of feasibility studies. These investigations provide an approximate estimate of the system activities, which can be used to determine the tactics to be used for an efficient system research and analysis.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. User can register without validation, Each customer can create their own profile, user needs to locate the mechanic from a nearby place by himself. It takes long time and difficulty to find mechanics and also very Expensive .The proposed system rectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system users can book complaints and get services available at anywhere with a fast manner.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Human effort is needed.
- It is difficult to find mechanics.
- More manual hours need to process application.
- It is very Expensive

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed the issues of existing system can be overcome by the proposed system. It makes the process is more easier and It provides complaint services available at anywhere with a fast manner. The User can post a feedback and admin can view and send solution to the user and the Customer and Online Payment Service is also provided. Then we provides:

- It makes the process is more easier.
- It provides complaint services available at anywhere with a fast manner
- The User can post and admin can view and send solution to the user
- Its more time consume for all process.

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:

- **You are able to get creative and innovative project plans:-**

Our users will get easy contact with repair services via online and they have the option to view booking progress and can send feedbacks.

- **Better security: -**

To keep data secure, precautions must be taken to prevent unwanted access. Security entails protecting the data from many types od destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. As we use secured databases to maintain the papers, it will also ensure data security.

- **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- **Better service: -**

The system saves time and money of customers by quickly providing all the repair services they required. It provides the information about the available services and user friendly. The Long-term storage of data is possible without any information being lost.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organisation in relation to the work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. A system proposal's workability, which includes the influence on the organisation, capacity to satisfy user needs, and efficient use of resources, is the basis for a feasibility study. As a result, before a proposed application is accepted for development, it often undergoes a feasibility assessment.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics: -

3.1.1 Economical Feasibility

The cost and benefit analyses are required to support the emerging system. criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables.

Some significant financial queries raised during the initial probe include the following:

- The costs of doing a thorough system examination.
- The price of the hardware and software.
- The advantages in terms of lower costs or fewer expensive mistakes.

The proposed system was created as part of a project, hence there are no manual expenses associated with it. Additionally, the fact that all of the resources are already at hand indicates that the system may be developed affordably.

The cost of project, SMART REPAIR 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 needs to be assessed first from a technical standpoint. An overview design of the system's requirements in terms of input, output, programmes, and procedures must serve as the foundation for the assessment of this viability. The inquiry must next advise the kind of equipment, necessary procedure for constructing the system, and means of operating the system once it has been designed after having identified an outline system.

The following technical difficulties came up throughout the investigation:

- Does the suggested technology work with the current technology?
- Can the system grow if it is improved?

The project should be created in such a way that the required performance and functionality are met within the limitations. The system may still be used even through the technology may become outdated after a while because a newer version of the same software still works with an earlier version. Therefore, this project only has a few limitations. The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project. The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project. The system used had a strong processor, an Intel i3 core, 4GB of RAM, and a 1TB hard drive.

3.1.3 Behavioral Feasibility

The following inquiries are part of the suggested system:

- Is there enough assistance for the users?
- Will the suggested system harm anyone?

The project would be advantageous because, when created and implemented, it would achieve the goals. The project is deemed to be behaviorally feasible after carefully weighing all behavioural factors.

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

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

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language used for general-purpose programming as well as web development. P Most web servers support the free deployment of PHP, which is also available as a solitary shell on practically all platforms and operating systems. PHP is now used by 2.1 million web servers and more than 244 million webpages. The reference version of PHP, which Rasmus Ledorf created in 1995, is now created by the PHP group. PHP: HypertextPreprocessor is the current meaning of the recursive acronym PHP, which previously stood for personal home page. The PHP processor module on a web server translates PHP code to create the finished web page. Instead of referring to an external file, PHP commands can be simply put into an HTML source file to handle data. The GNU General Public License is in conflict with PHP because it has evolved to incorporate a command-line api capability and can be used independently due to restrictions on the usage of the term PHP(GPL). Most web servers support the free deployment of PHP, which is also available as a solitary shell on pratically all platforms and operating systems.

3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

- **MySQL is a database management system.**

  A systematic collection of data is called a database. It might be anything, such as a straightforward grocery list, a photo gallery, or the enormous amount of data in a business network. A database management system like MySQL Server is required in order to add, access, and process data that is stored in a computer database. Database management systems, whether used as stand-alone programmes or as a component of other applications, are essential to computing because computers are excellent at processing vast volumes of data.

- **MySQL databases are relational.**

  Instead of placing all the facts in one huge warehouse, a relational database keeps the data in individual tables. Physical files that are designed for speed contain the database structures. The logical model provides a flexible programming environment with objects like databases, tables, views, rows, and columns. One-to-one, one-to-many, unique, compulsory or optional, and "pointers" between distinct tables are a few examples of the rules you might build up to regulate the relationships between various data fields. With a well-designed database, your application won't ever encounter inconsistent, duplicate, orphan, out-of-date, or missing data since the database enforces these rules. MySQL stands for "Structured Query Language" with the SQL prefix. The most popular standard language for accessing databases is SQL. Relying on your development platform, you might explicitly enter SQL (for example, to produce reports), incorporate SQL statements into other languages' code, using a language-specific API which obscures the SQL syntax. By way of the ANSI/ISO SQL Standard, SQL is defined. Since its inception in 1986, the SQL standard must have gone through various iterations. In this document, "SQL92" refers to the 1992 standard, "SQL: 1999" to the 1999 standard, and "SQL: 2003" to the most recent version of the program. The SQL Standard as it exists at any one time is referred to as "the SQL standard."

- **MySQL software is Open Source.**

  Anybody can use and alter software that is open source, according to the term. The MySQL software is available for free download and usage online by anyone. You are free to examine the program code and modify it as necessary. The GPL (GNU General Public License) is used by the MySQL software to specify what you are allowed to do and are not allowed to do with the software in certain circumstances. You can purchase a commercially licenced version from us if the GPL makes you uncomfortable or if you need to integrate MySQL code into a for-profit application. For further details, see the MySQL Ownership Overview.

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

  You ought to give it a shot because that is what you're after. In addition to your other apps, web servers, and other software, MySQL Server can function smoothly on a laptop or desktop while requiring little to no maintenance. You can modify the settings to utilise all the RAM, processing capacity, and I/O capacity if you concentrate an entire unit to MySQL.

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

  The MySQL Database Software is a client/server system made up of a multi-threaded SQL server that supports several backends, a number of distinct client programmes and libraries, administrative tools, and a broad variety of application programming interfaces (APIs). Additionally, we offer MySQL Server as a multi-threaded embedded library that you might integrate into your programme to create a standalone solution that is more manageable, quicker, and smaller.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any engineered system or product's development process begins with design. A creative process is design. The secret to an efficient system is a decent design. The process of using different methodologies and concepts to specify a process or even a system in enough detail to allow for its physical implementation is referred to as "design." One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical reality. Irrespective of the development approach that is employed, software design forms the entity of the engineering phase. The architectural detail needed to construct a system or product is developed through the system design. This programme has also through the highest suitable design phase, touching up all efficiency, performance, and accuracy levels, like in the instance of any systematic approach. A user-oriented document is converted into a document for programmers or database staff throughout the design phase. The two stages of system design development are logical design and physical design.

4.2 UML DIAGRAM

A common language known as UML is used to specify, visualise, build, and document the software system artefacts. The Object Management Group (OMG) was responsible for developing UML, and a draught of the UML 1.0 definition was presented to the OMG in January 1997.Compared to other popular programming languages like C++, Java, COBOL, etc., UML is unique. A visual language called UML is used to create software blueprints. A general-purpose visual modelling language for software system visualisation, specification, construction, and documentation is what UML is known as. UML is not just used to represent software systems, despite the fact that this is its most common application. It is often used to model systems that are not software-based. For instance, the manufacturing facility's process flow, etc. Although UML is not a programming language, tools can be used to ability to gain access using UML diagrams in a variety of languages. The analysis and design of objects-oriented systems are directly related to UML. UML has been standardised to the point where it is now an OMG standard. A comprehensive UML diagram that depicts a system is made up of all the

elements and relationships. The most crucial aspect of the entire procedure is the UML diagram's aesthetic impact. It is completed by using all the additional components. The following nine diagrams are part of UML.

- 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 visual representation of the interactions between system components. A approach for identifying, outlining, and organising system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order goods sales and services. UML (Unified Modeling Language), a standardized language for the modelling of real-world objects and systems, uses use case diagrams.

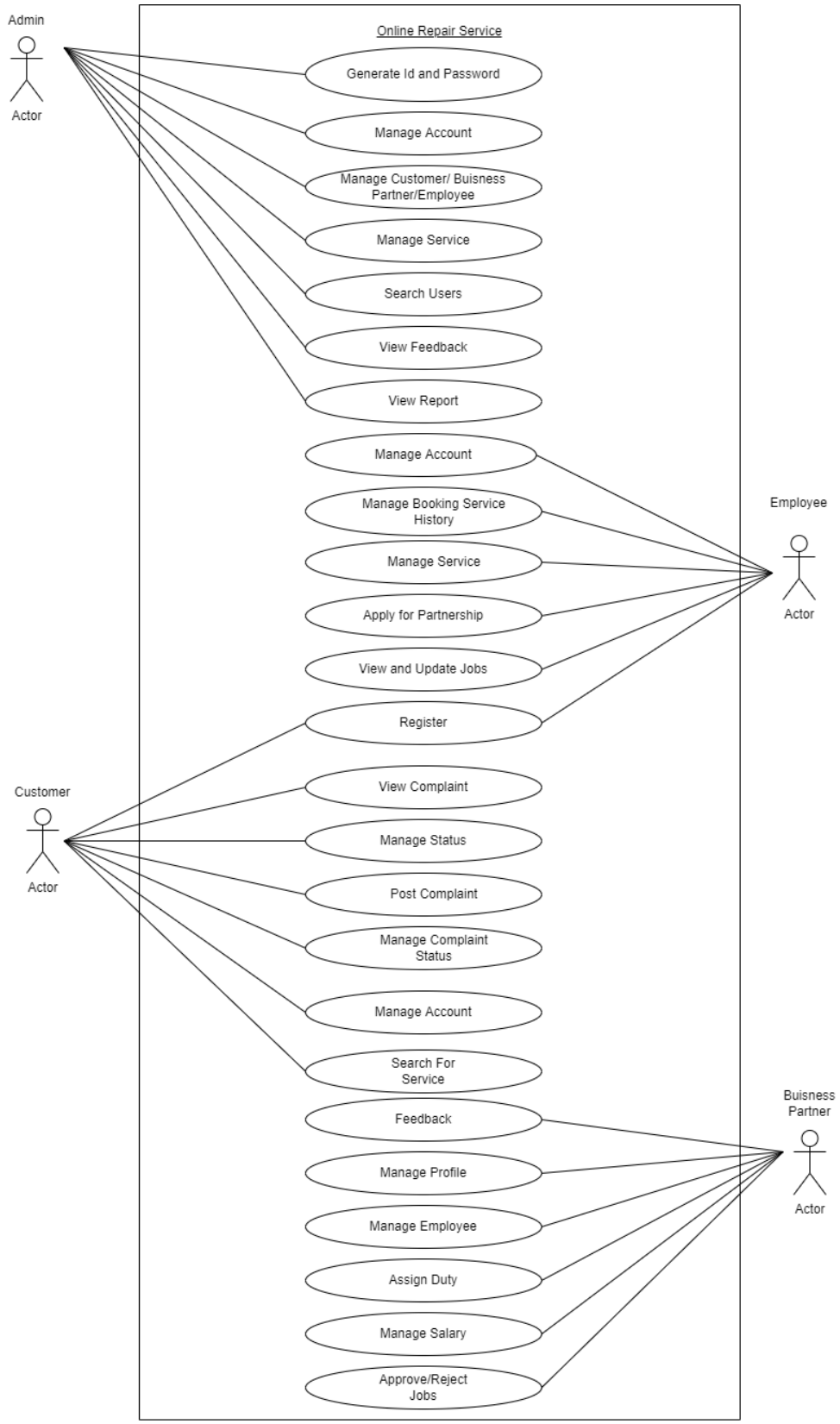
The planning of overall requirements, testing a hardware architecture, testing and debugging a software application in development, producing an online help reference, or carrying out a task focused on customer support are just a few examples of system objectives. For instance, use cases in a sales revenue context can involve customer service, item ordering, catalogue updating, and payment processing. There are four elements in a use case diagram.

- The boundary, which isolates the subject of interest from its surroundings.
- The performers, who are typically system participants identified by the roles they play.
- The actors among and across the system play the roles specified by the use cases.
- The connections and interactions between the actors and use case.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- A use case's naming is very significant. The name should indeed be selected in a way that makes it clear what functions are being performed.
- Give the actors names that fit them.
- Clearly depict links and dependencies in the diagram.
- Keep in mind that the diagram's primary function is to indicate the needs; do not attempt to include all possible relationships.
- Use notes whenever required to clarify some important points.

Fig 1 : Use case diagram for SMART REPAIR



4.2.2 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order that these components interact. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence diagrams show the actions taken by the components of a system in chronological order. Businesspeople and software engineers frequently use these diagrams to record and comprehend the specifications for new and current systems.

Sequence Diagram Notations –

i. **Actors –** In a UML diagram, an actor represents a particular kind of role in which it communicates with the system's objects. An actor has always been beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and perhaps other subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.

ii. **Lifelines –** A named piece that shows a specific member in a sequence diagram is called a lifeline. In essence, a lifeline represents each specific case in a sequence diagram. The top of a sequence diagram is where the lifeline pieces are placed.

iii. **Messages –** Using messages, communication among objects is demonstrated. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

The following categories serve as general classifications for messages:

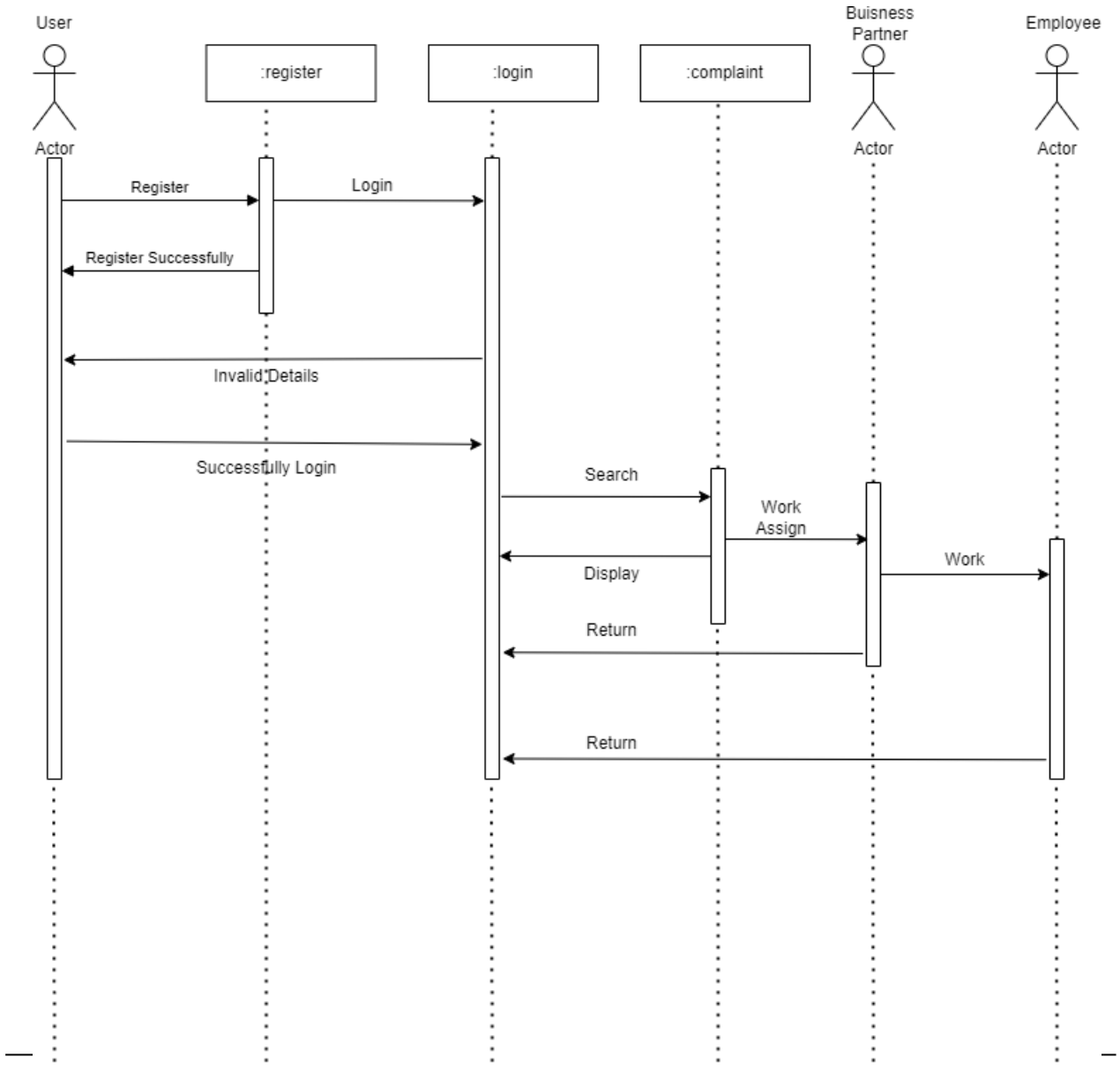
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. **Guards –** To In the UML, we utilise guards to model circumstances. When we really need to limit the transmission of communication under the guise of a condition having met, we use them. Software engineers rely on guards to inform them of the limitations imposed by a system or specific process.

Uses of sequence diagrams –

- Used to simulate and visualise the reasoning behind a complex function, process, or method.
- The intricacies of UML use case diagrams are also displayed using them.
- Used to comprehend the precise operation of present or upcoming systems.
- Visualize the flow of information between different system elements or objects.

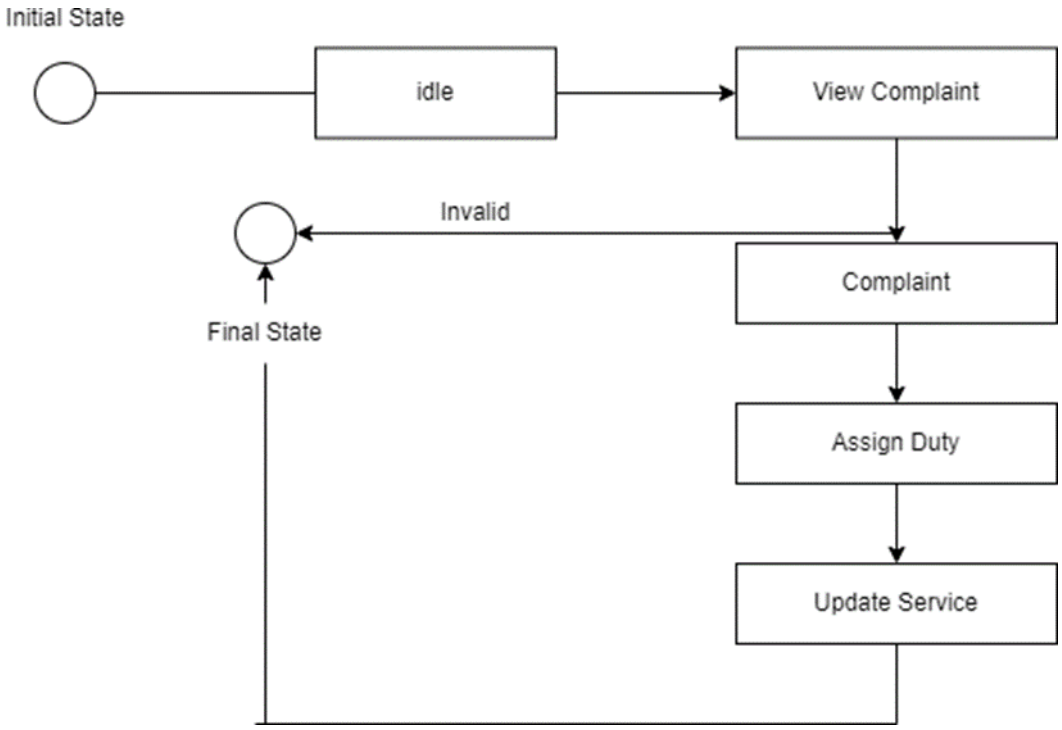
Fig 1 : Sequence diagram for SMART REPAIR



4.2.3 STATECHART DIAGRAM

Statechart Diagrams are used to depict how a software system behaves. A class, a subsystem, a package, or even a complete system's behaviour can be represented by a state machine diagram in a UML model. A Statechart or State Transition diagram is another name for it. We may efficiently model the interactions or communication that take place between external entities and a system using statechart diagrams. The event-based system is modelled using these diagrams. With the aid of an event, a state of an object can be managed. Statechart diagrams are employed in the application system to depict the many states of an entity.

Fig 1 : State chart diagram for SMART REPAIR

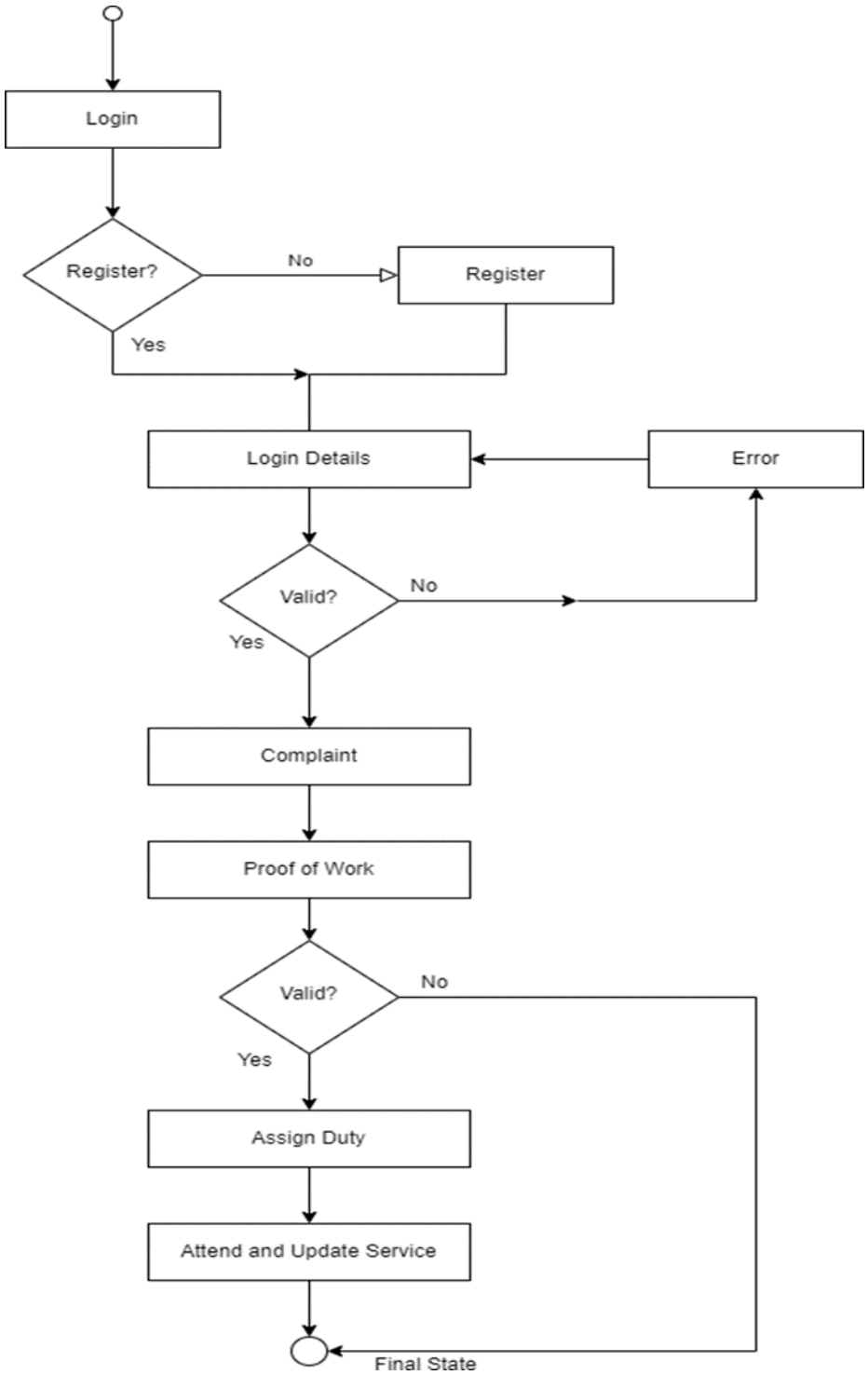


4.2.4 ACTIVITY DIAGRAM

Activity diagrams show how multiple levels of abstraction of activities are organized to produce a service. Typically, an event must be accomplished by some operations, especially when the operation is meant to accomplish several different things that call

for coordination. Another common requirement is how the occurrences in a single use case correspond to one another, especially in cases where operations may coincide and require collaboration. It is also appropriate for simulating how a set of related use cases operate together to reflect business workflows.

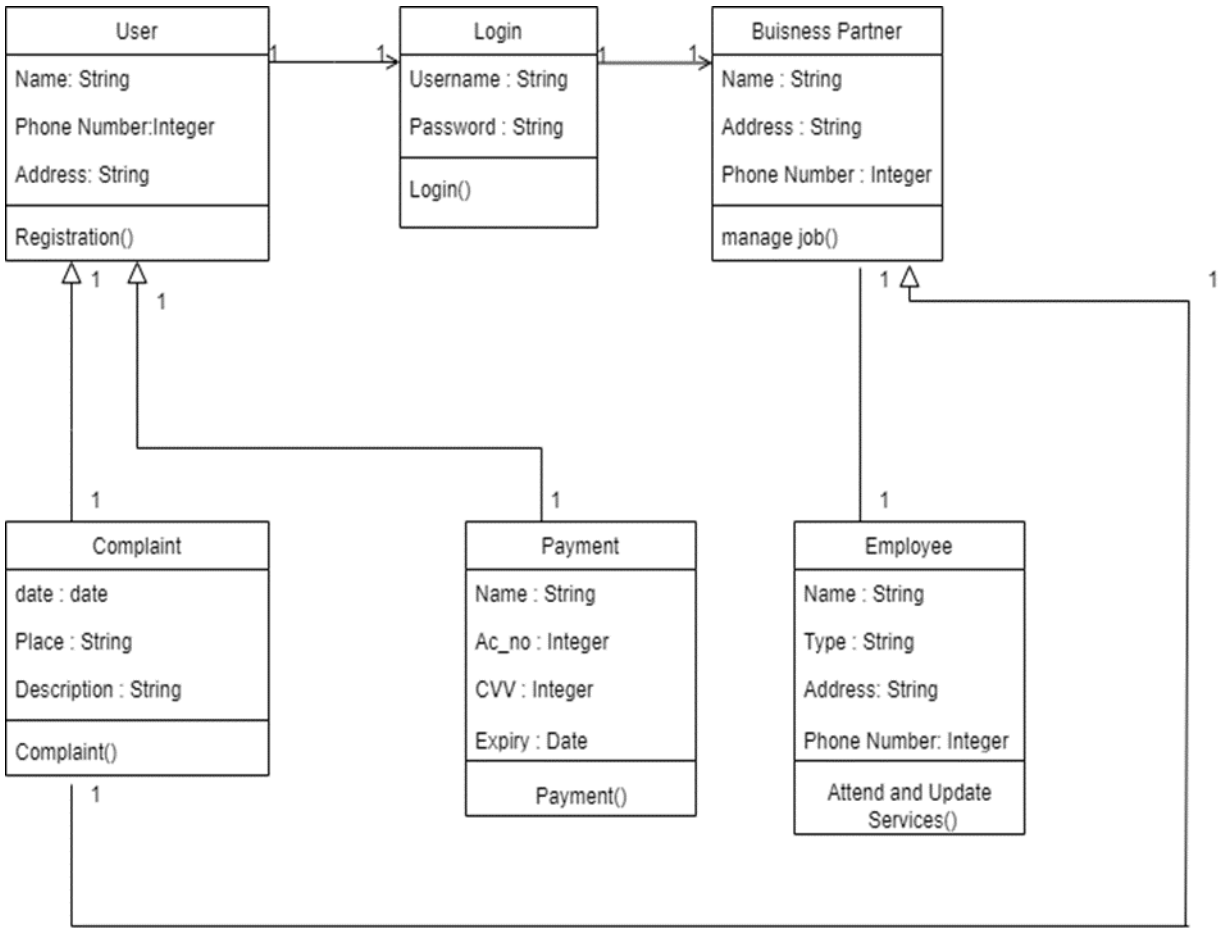
Fig 1 : Activity diagram for SMART REPAIR



4.2.5 CLASS DIAGRAM

A static diagram is a class diagram. It symbolises an application's static view. Class diagrams are employed not only for representing, describing, and documenting various parts of a system but also for creating the executable code for a software programme. A class diagram explains a class's properties and functions as well as the limitations placed on the system. Due to the fact that class diagrams are the only UML diagrams that can be directly transferred to object-oriented languages, they are extensively utilised in the modelling of object-oriented systems. A class diagram displays several classes, interfaces, relationships, collaborations, and constraints. A structural diagram is another name for it.

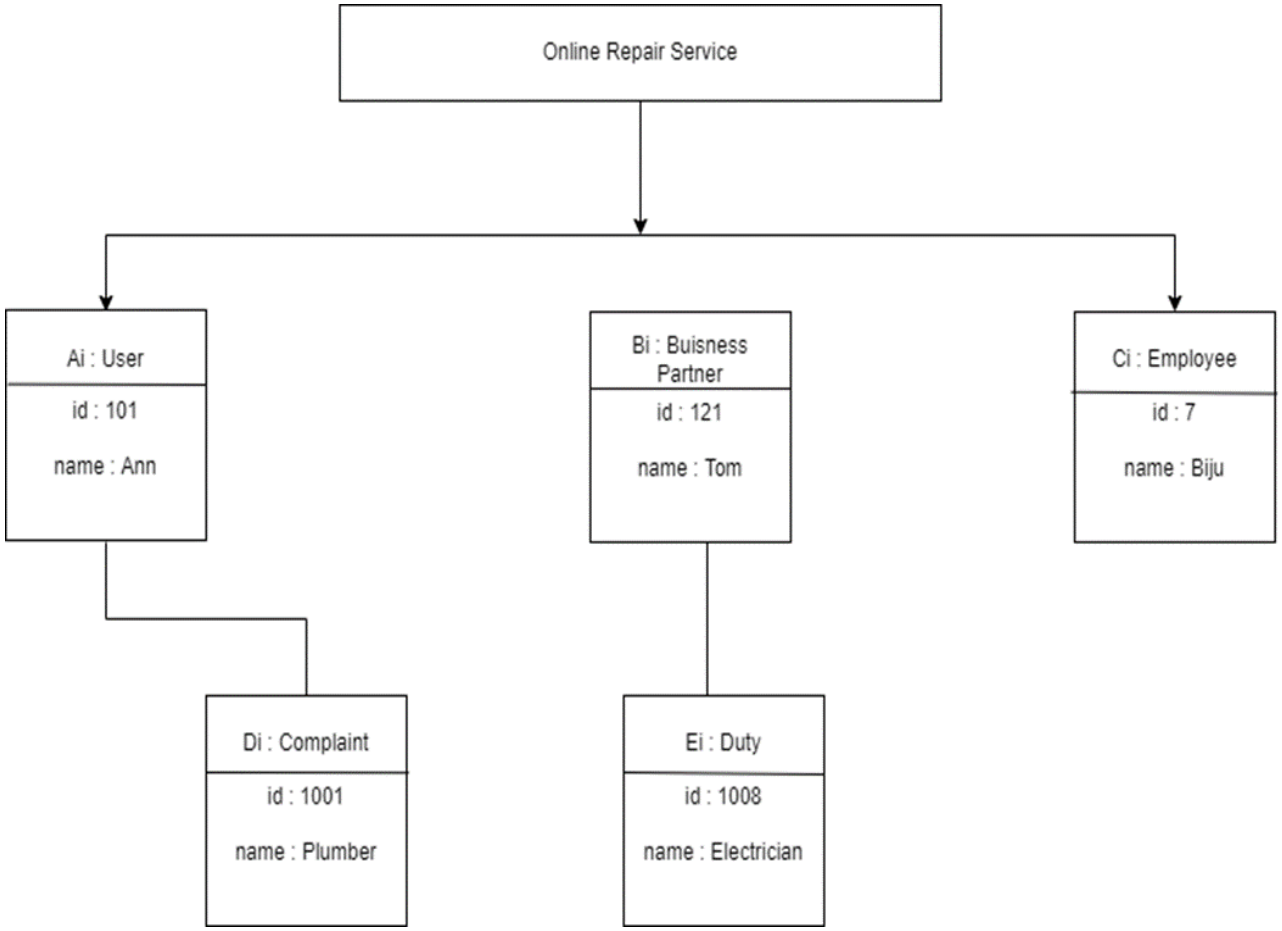
Fig 1 : Class diagram for SMART REPAIR



4.2.6 OBJECT DIAGRAM

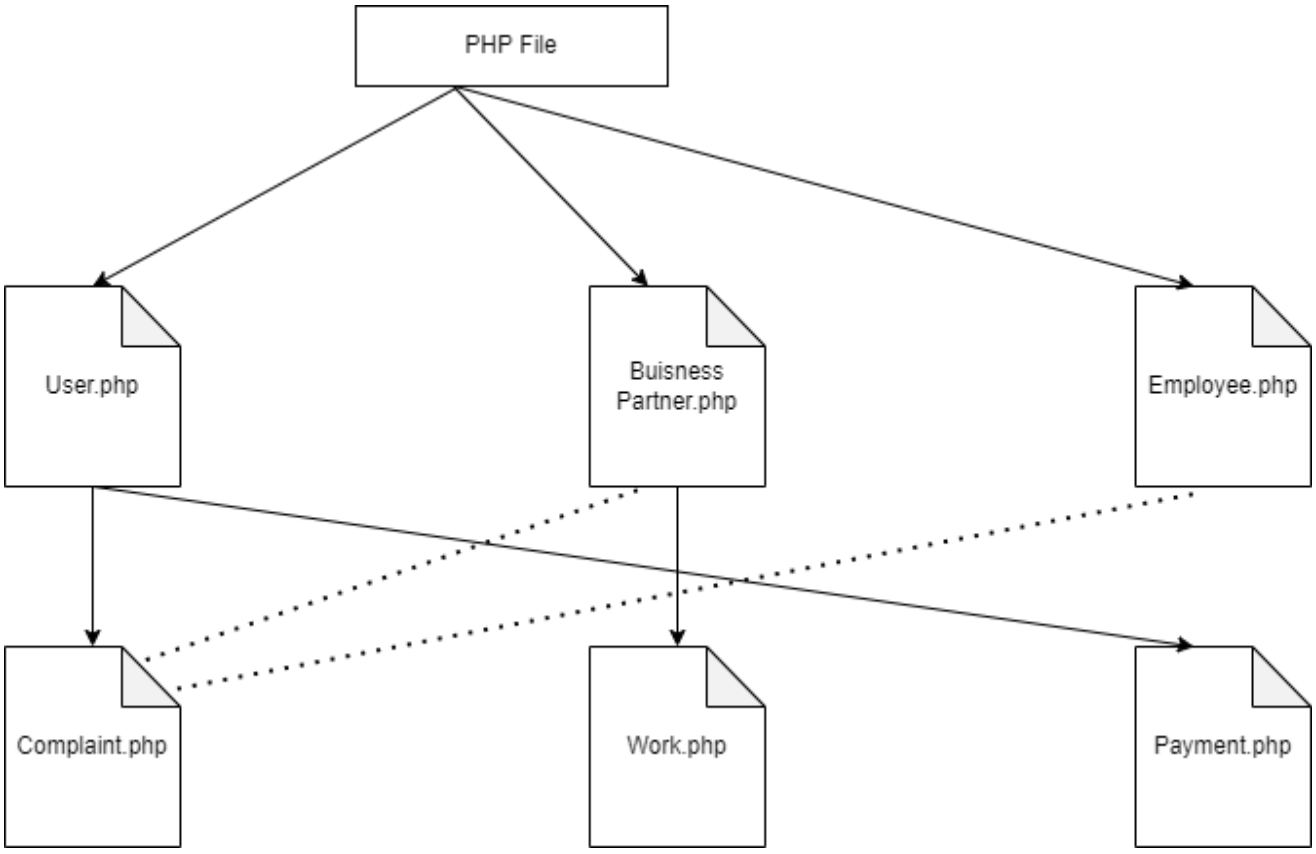
Class diagrams are the source from which object diagrams are produced, making class diagrams necessary for object diagrams. A class diagram is represented by an instance in object diagrams. Both class diagrams and object diagrams use similar fundamental ideas. Object diagrams can also show a system's static view, although this static view only shows the system as it was at a specific time. A group of items and their connections are represented as an instance using object diagrams.

Fig 1 : Object diagram for SMART REPAIR



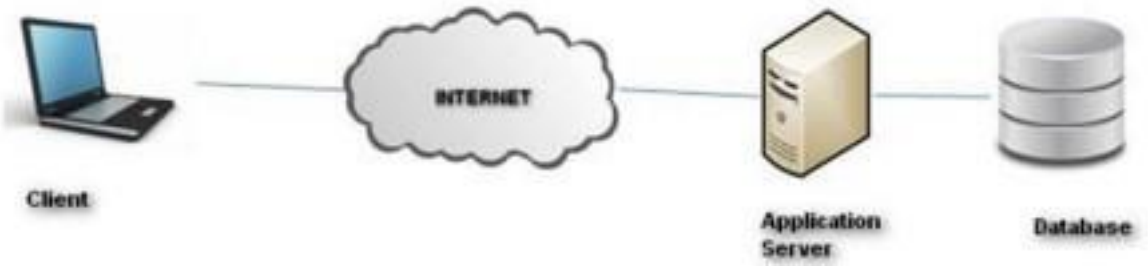
4.2.7 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behaviour. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.



4.2.8 DEPLOYMENT DIAGRAM

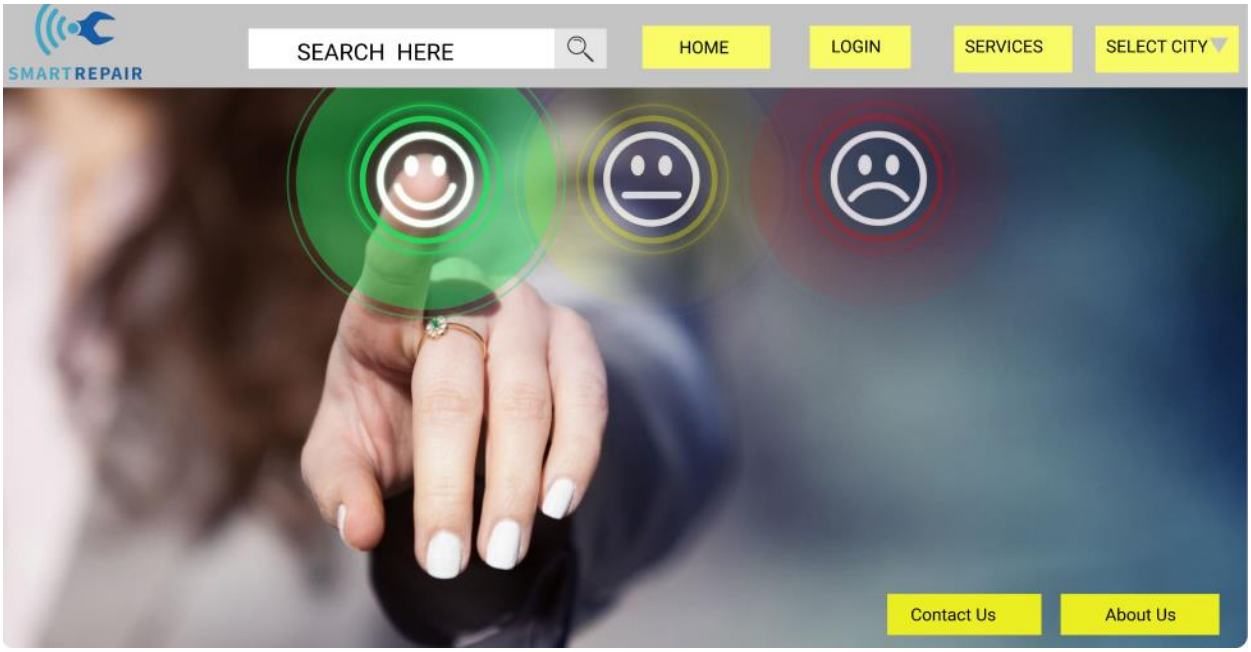
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



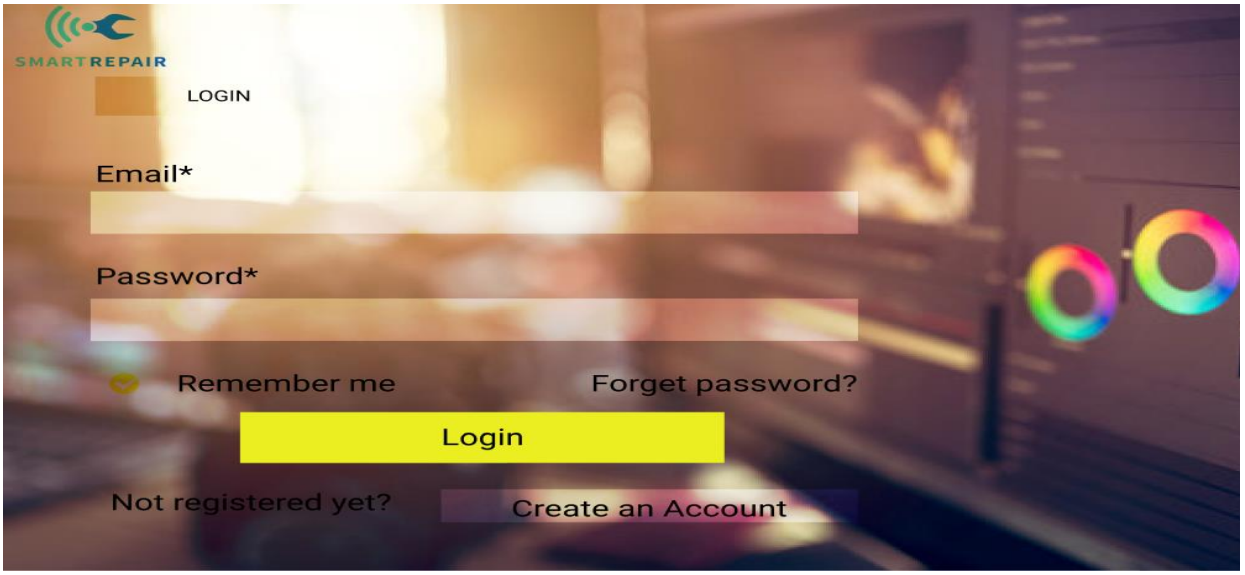
4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

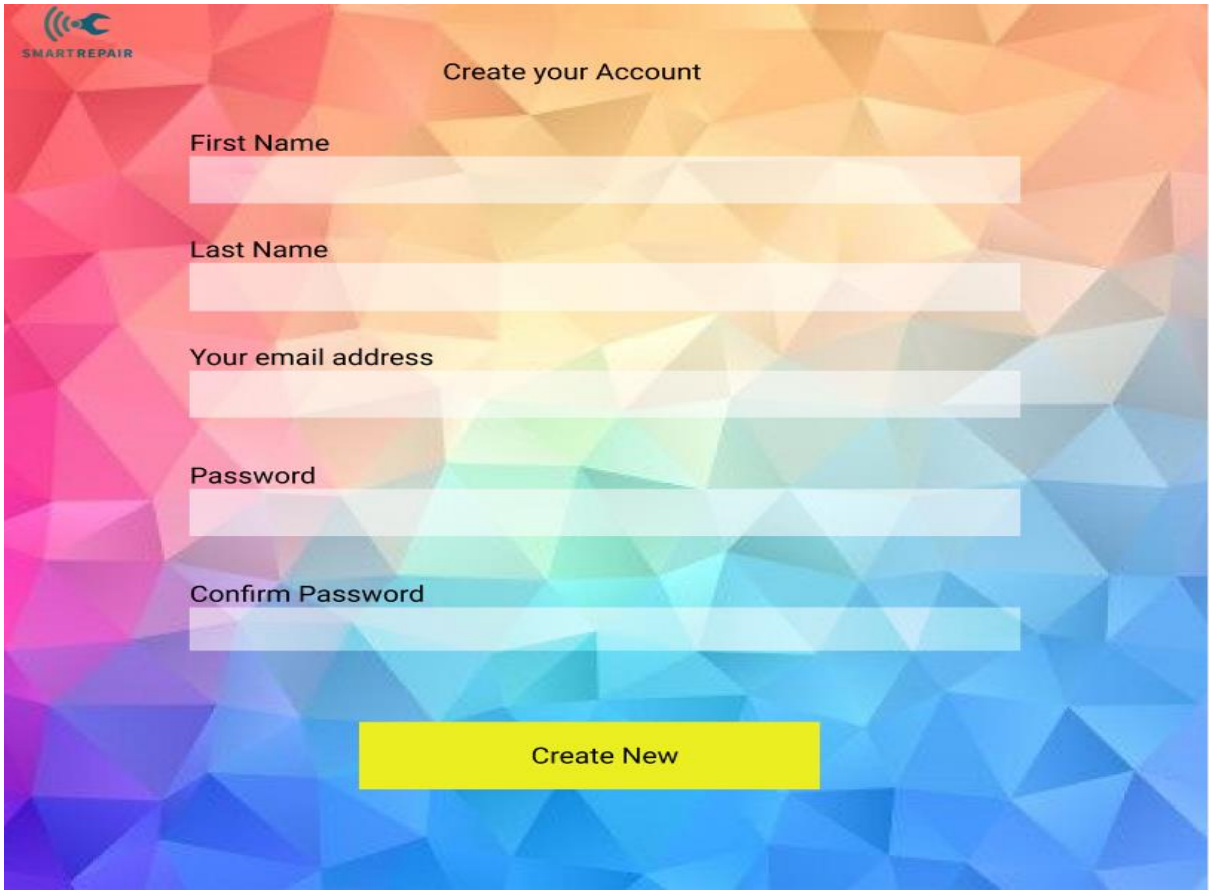
Form Name : Home



Form Name : User Login

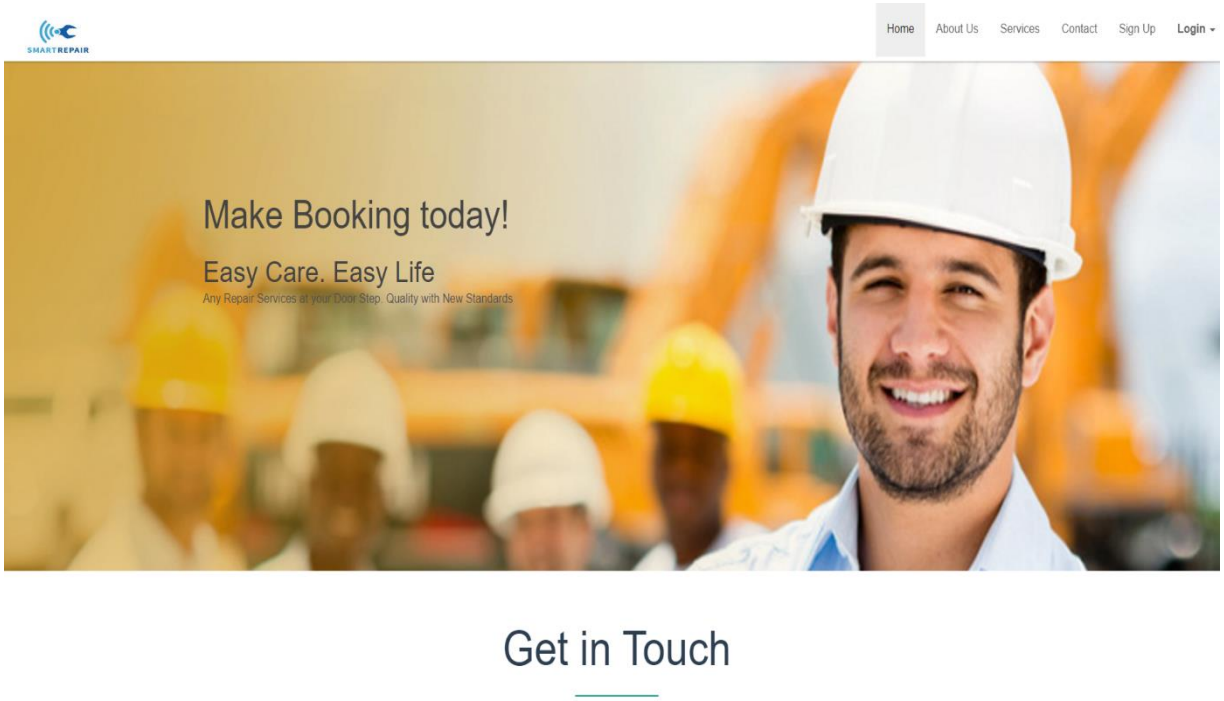


Form Name : User Registration Form

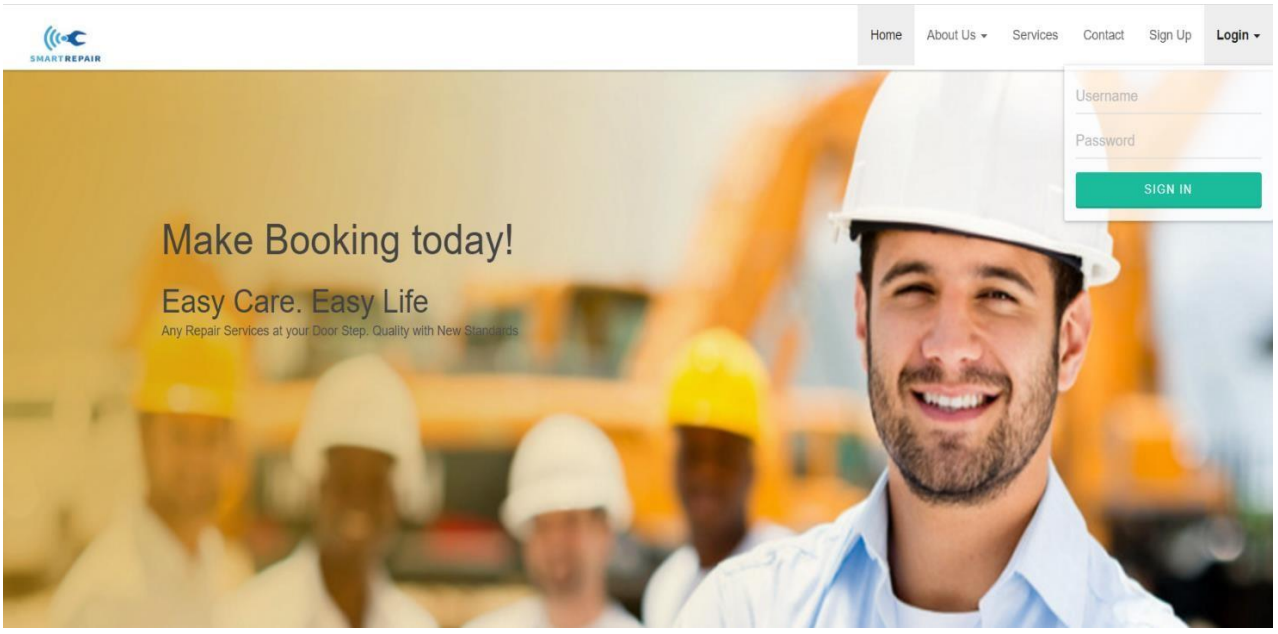


4.3.2 OUTPUT DESIGN

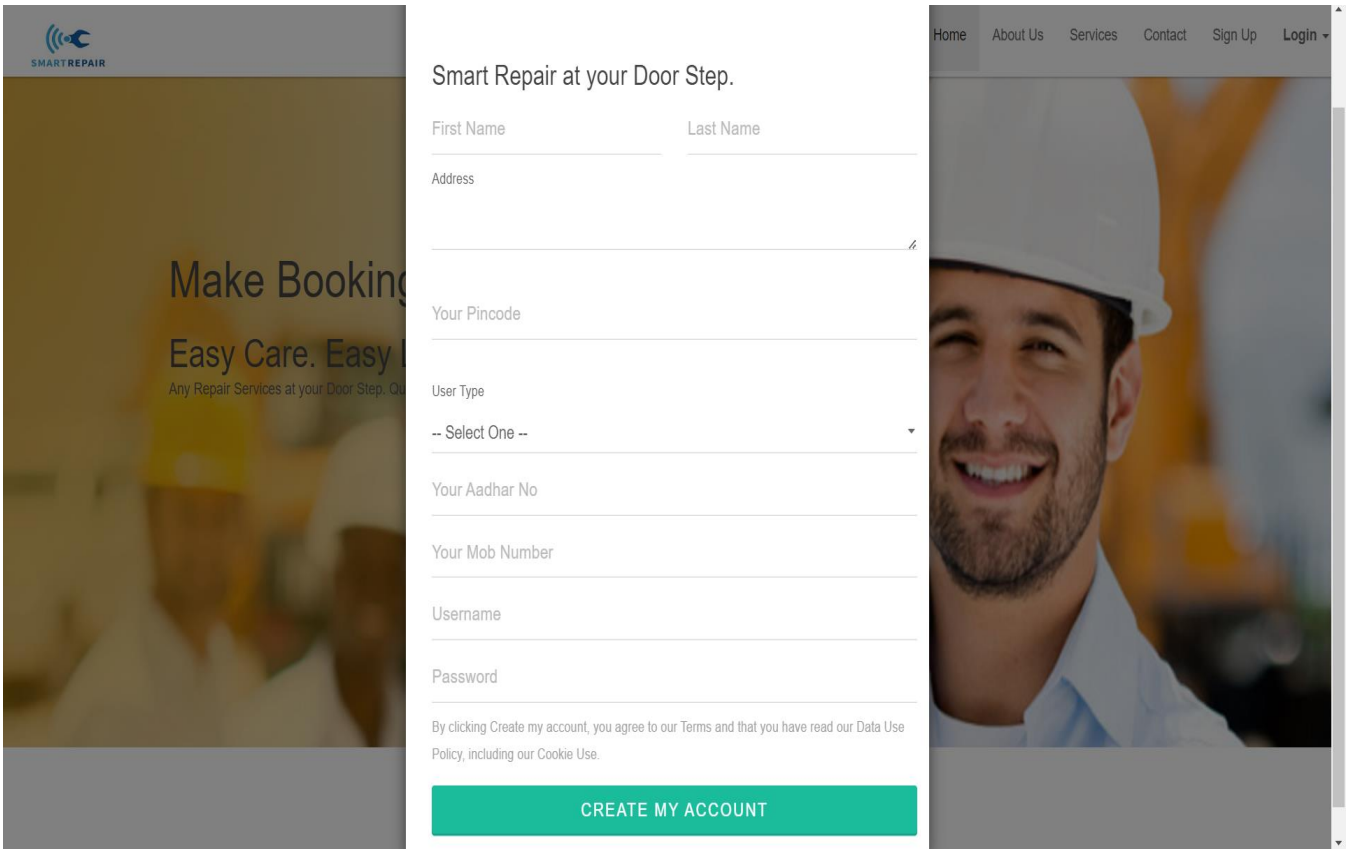
Index



User Login



User Registration



4.4 DATABASE DESIGN

A database A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly and accurately meet these criteria. This process, known as information level design, is carried out independently of all DBMSs.

The architecture for the unique DBMS which will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the features of the particular DBMS which will be used are discussed. Parallel to the system design is a database design. The database's data arrangement aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. Each relationship resembles a file or table of records with values. A row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation in technical relational model language. A relational database is made up of a number of tables, each with its own name. In a story, each row represents a group of associated values.

Relations, Domains & Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are known to as columns. Each table in the database has relationships already established between them. Referential and entity relationship integrity are both guaranteed in this way. A group of atomic values make up a domain D. Choosing a data type wherein the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name.

A relation's values are all atomic, which means they cannot be broken down.

Relationships

- Key is used to create table relationships. Primary Key and Foreign Key are the two principal keys that are most crucial. With the use of these keys, relationships for entity integrity and referential integrity can be created.
- Entity Integrity forbids the use of null values for any Primary Key.
- No Primary Key may contain null values, according to Referential Integrity.
- Referential Integrity: A Primary Key value in the same domain must correspond to each unique Foreign Key value. Super Key and Candidate Keys are additional keys.

4.6.2 Normalization

The simplest possible grouping of data is used to put them together so that future adjustments can be implemented with little influence on the data structures. The formal process of normalising data structures in a way that reduces duplication and fosters integrity. Using the normalisation technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the very same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalised.

It means placing things in their natural form, as the name suggests. By using normalisation, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly linked to the data by the customer. By removing recurring groups from the data, normalisation prevents data redundancy, which puts a heavy strain on the computer's resources. These consist of:

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

First Normal Form

According to the First Normal Form, each attribute's domain must only include atomic values, and each attribute's value in a tuple would have to be a single value from that domain. In other words, 1NF forbids using relationships as attribute values within tuples or relations within relations. Single atomic or indivisible values are the only attribute values that are permitted under 1NF. The data must first be entered into First Normal Form. This can be accomplished by separating data into tables of a similar type for each table. Depending on the needs of the project, a Primary Key or Foreign Key is assigned to each table. For each nested relation or non-atomic attribute, new relations are formed in this process. This got rid of data groups that were repeated. If a relation meets the requirements that include only the primary key, it is said to be in first normal form.

Second Normal Form

According to Second Normal Form, no non-key attribute should be functionally dependent on a portion of the main key for relations where the primary key has several attributes. Every partial key with its dependent characteristics is broken down in this process, and a new relation is established up. As well as any properties that are completely functionally dependent on the original primary key, make sure to maintain a relationship with them. By removing the data that depends only on a portion of the key, this phase aids in data cleansing. When the first normal form requirements for the primary key are met and all of the relation's non-primary key properties are completely dependent on the primary key alone, a relation is said to be in second normal form.

Third Normal Form

Relation should not have a non-key attribute that is functionally determined by some other non-key attribute or by a collection of non-key attributes, according to the Third Normal Form. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This procedure is used to eliminate anything not wholly dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third normal form.

TABLE DESIGN

Table No : 01

Table Name : tbl_UserType

Primary Key : UserType_id

Table Description: To store Types of Users

Filedname	Datatype	Size	Description
UserType_id	int	10	Usertype ID
Type_name	varchar	20	User Type
Status	Varchar	50	Status

Table No : 02

Table Name : tbl_login

Primary Key : Login_id

Foreign Key : UserType_id

Table Description : To store user Login information

Filedname	Datatype	Size	Description
Login_id	int	10	Login ID
UserType_id	int	10	Usertype ID
Username	Varchar	25	Username
Password	Varchar	12	**P**assword
Status	Varchar	50	Status

Table No : 03

Table Name : tbl_ registration

Primary Key : Reg_id

Foreign Key : Login_id

Table Description: To store user registration information

Fieldname	Datatype	Size	Description
Reg_id	int	15	Reg ID
Fname	Varchar	25	First name
Lname	Varchar	25	Last name
Address	Varchar	50	Address
Pincode	int	20	Pincode
Mobile_no	bigint	20	Moblie Number
Aadhaar_No	bigint	20	Aadhaar Number
Status	Varchar	10	Status
log_id	int	10	Login ID

Table No : 04

Table Name : tbl_Category_Complaint

Primary Key : Category_id

Table Description: To store Category Types

Filedname	Datatype	Size	Description
Category_id	int	20	Category ID
Category_Type	varchar	30	Category Type
Status	Varchar	50	Status

Table No 05

Table Name : tbl_Complaint

Primary Key : Complaint_id

Foreign Key : Reg_id

Foreign Key of Tbl_Category_Complaint : Category_id

Table Description: To store User Bookings

Fieldname	Datatype	size	Description
Complaint_id	int	20	Complaint ID
Reg_id	int	20	Reg ID
Category_id	Int	20	Category ID
Date	date		Date
Place	Varchar	30	Place
Description	varchar	30	Description
Status	Varchar	12	Status
Progress	text		Progress
employ_id	int	20	Employee ID

Table No : 06

Table Name : tbl_payment

Primary Key : Pay_id

Foreign Key : Reg_id

Table Description: To store Payment Details

Fieldname	Datatype	Size	Description
Pay_id	int	15	Pay ID
Reg_id	int	20	Reg ID
Amount	int	12	Amount
Date	date		Date
Ac_no	int	15	Account Number
Ac_Holder_name	Varchar	20	Account Holder Name
CVV	int	10	Card Verification Value
Expiry Date	date		Expiry Date

Table No : 07

Table Name : tbl_feedback

Primary Key : Feedback_id

Foreign Key : Reg_id

Table Description: To store feedback information from user

Fieldname	Datatype	Size	Description
Feedback_id	int	12	Feedback ID
Reg_id	int	12	Reg ID
Feedback	Varchar	50	Feedback
Date	Date		Date
Status	Varchar	20	Status

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation.

The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details.

Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. They are:

A Program is tested by being run with the goal of identifying any errors.

- A successful test is one that finds an undetected error.
- A good test case is one that has a high likelihood of doing so.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied.

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.

5.1 TEST PLAN

The procedures that must be followed in order to fulfil various testing methodologies are suggested in a test plan. The test plan specifies the activities that must be completed. Software developers are responsible for creating a computer programme, as well as any related documentation and data structures. It is always the software developers' job to test each of the program's individual parts to ensure that it serves the intended function. The objective of the independent test group (ITG) is to address the issues that result from letting the person who created something test it. The exact objectives of the testing should be stated in terms of numbers. Information on the mean time to failure, the cost to find and fix issues, the remaining defect density or frequency of occurrence, and the number of test work hours required for each regression test should all be included in the test plan.

The levels of testing include:

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

5.2.1 Unit Testing

The smallest unit of software design, the software component or module, is the focus of unit testing, which focuses verification efforts on it. Testing critical control routes to identify flaws inside the module's boundary is done using the component level design description as a reference. the specified untested area for unit testing and the test complexity level. Unit testing focuses on the white box, and multiple components may be tested at once. The modular interface is tested to ensure that data enters and exits the software unit under test correctly. The local data structure is reviewed to ensure that data temporarily stored maintains its integrity during each step of an algorithm's execution. Boundary conditions are assessed to verify that each statement in a module has been run at least once. Finally, each path for managing errors is examined.

Testing of data flow through a module interface are important before beginning any additional tests. If data cannot correctly enter and exit the system, all other tests are useless. The unit test's selective analysis of execution routes is a crucial task. In order to cleanly reroute or stop work when an error does occur, error handling channels must be set up and error scenarios must be anticipated in excellent design. Boundary testing is the last stage of unit testing. At its boundaries, software frequently fails..

When unit testing the Sell-Soft System, each module was treated as a separate entity and put through a range of test inputs. The modules' internal logic had some flaws, which were addressed. After coding, each module is examined and run independently. All unused code was removed to ensure that every module operates correctly and achieves the desired result.

5.2.2 Integration Testing

The meticulous process of building the program's structure and running tests to detect interface problems is known as integration testing. The objective is to build a programme structure that was chosen by design and tested in units. The entire programme is put to the test. Since it is difficult to pinpoint the causes due to the scale of the whole programme, correction is difficult. As soon as these errors are corrected, new ones appear, and the process keeps repeating itself. After unit testing was finished in the system, all of the modules were combined to check for interface inconsistencies. When differences in programme structures were eliminated, a distinctive programme structure also emerged.

5.2.3 Validation Testing or System Testing

This marks the conclusion of the testing procedure. This required comprehensive testing of the system, which covered all forms, codes, modules, and class modules. System tests and BlackBox tests are common names for this kind of testing. The primary focus of the black box testing approach is on the software's functional requirements.

Using Black Box testing, the software engineer can design sets of input circumstances to fully exercise all functional requirements for a programme. Black box testing seeks to solve the following types of issues: incorrect or missing functions, interface issues, issues with data structures or accessing external data, and performance issues.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration is tested for user acceptance; in this case, it must satisfy the business' requirements. The programme should consult the user and the perspective system as it is being developed in order to make any necessary adjustments. Input screen designs, output screen designs, and other factors were taken into consideration.

A variety of test data are used to conduct the aforementioned testing. The process of system testing requires the preparation of test data. The system under examination is then tested using the sample data that has been prepared. The system's flaws are once more discovered during testing, repaired with the help of the aforementioned techniques, and recorded for future use.

Automation Testing

Automatic tests are performed on software and other computer products to ensure they adhere to strict standards. It basically serves as a test to make sure the hardware or software works exactly as intended. It looks for mistakes, faults, and any other issues that might arise during the course of the product's construction. Automation testing can be done at any time of day. By means of predefined sequences, the software is examined. The information is then summarised, and this data can be compared to the outcomes of earlier test runs.

Benefits of Automation Testing

Detailed reporting capabilities - Test cases for different scenarios are carefully built for automation testing. These planned sequences can cover a lot of ground and produce in-depth reports that are simply impossible for a human to produce.

Improved bug detection - Finding bugs and other flaws in a product is one of the key reasons to test it. This procedure can be made simpler with automation testing. Additionally, it has a greater test coverage analysis capability than people might have.

• Simplifies testing - Most SaaS and IT organisations routinely include testing in their daily operations. The key is to keep things as basic as you can. Automation has a lot of advantages. The test scripts can be reused when automating test tools.

• Quickens the testing procedure - Machines and automated technology operate more quickly than people. This is why we employ them, along with increased accuracy. Your software development cycles are subsequently shortened by this.

Reduces human intervention - Tests can be run at any time of day, even overnight, without needing humans to oversee it. Plus, when it's conducted automatically, this can also reduce the risk of human error

5.2.5 Selenium Testing

Selenium is an open-source application that automates web browsers. You can write test scripts using a single interface in a variety of computer languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. The Selenium testing tool automates the cross-browser compatibility testing of web applications. It is used to ensure that web apps are of a high level, whether they are responsive, progressive, or standard. Selenium is a free piece of software**.**

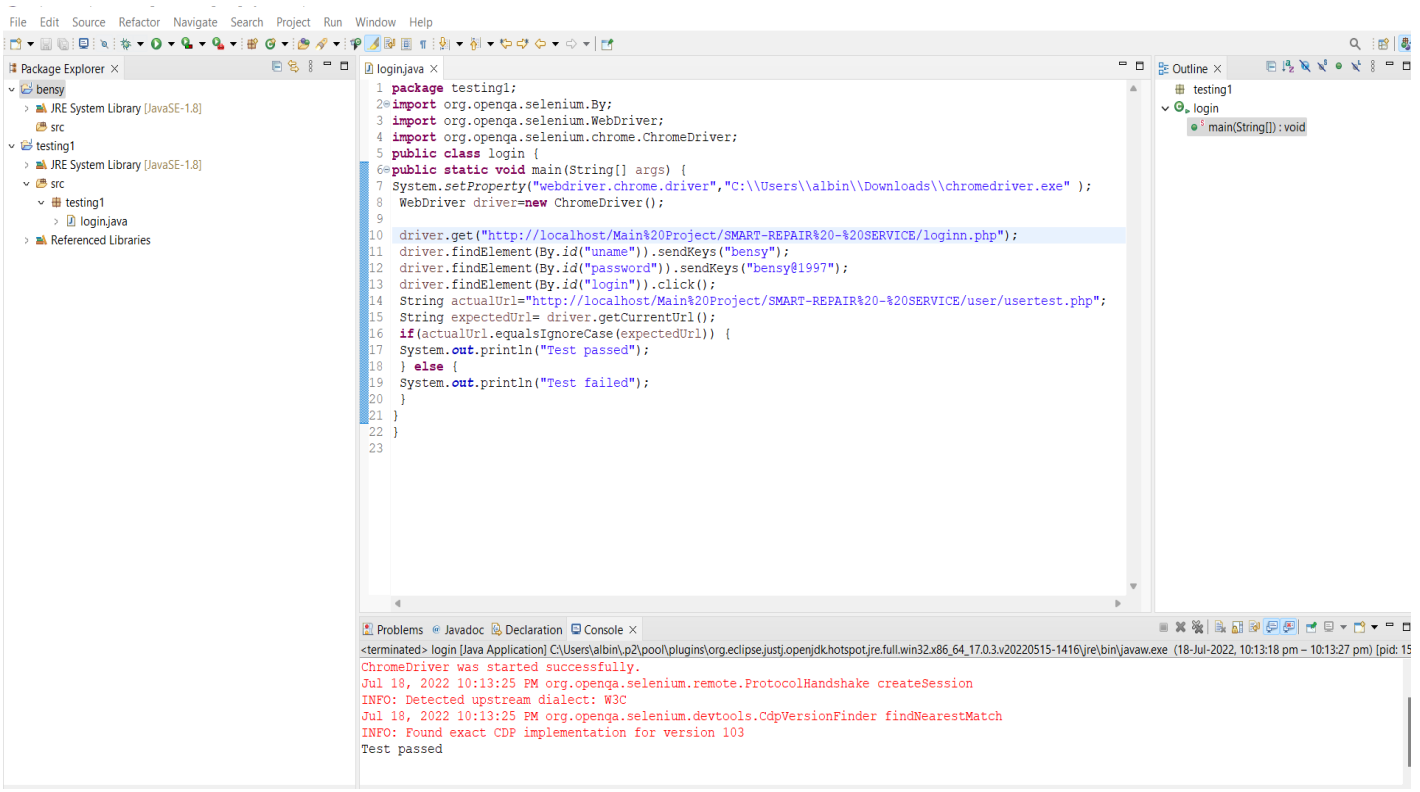
Test cases for a Login Page

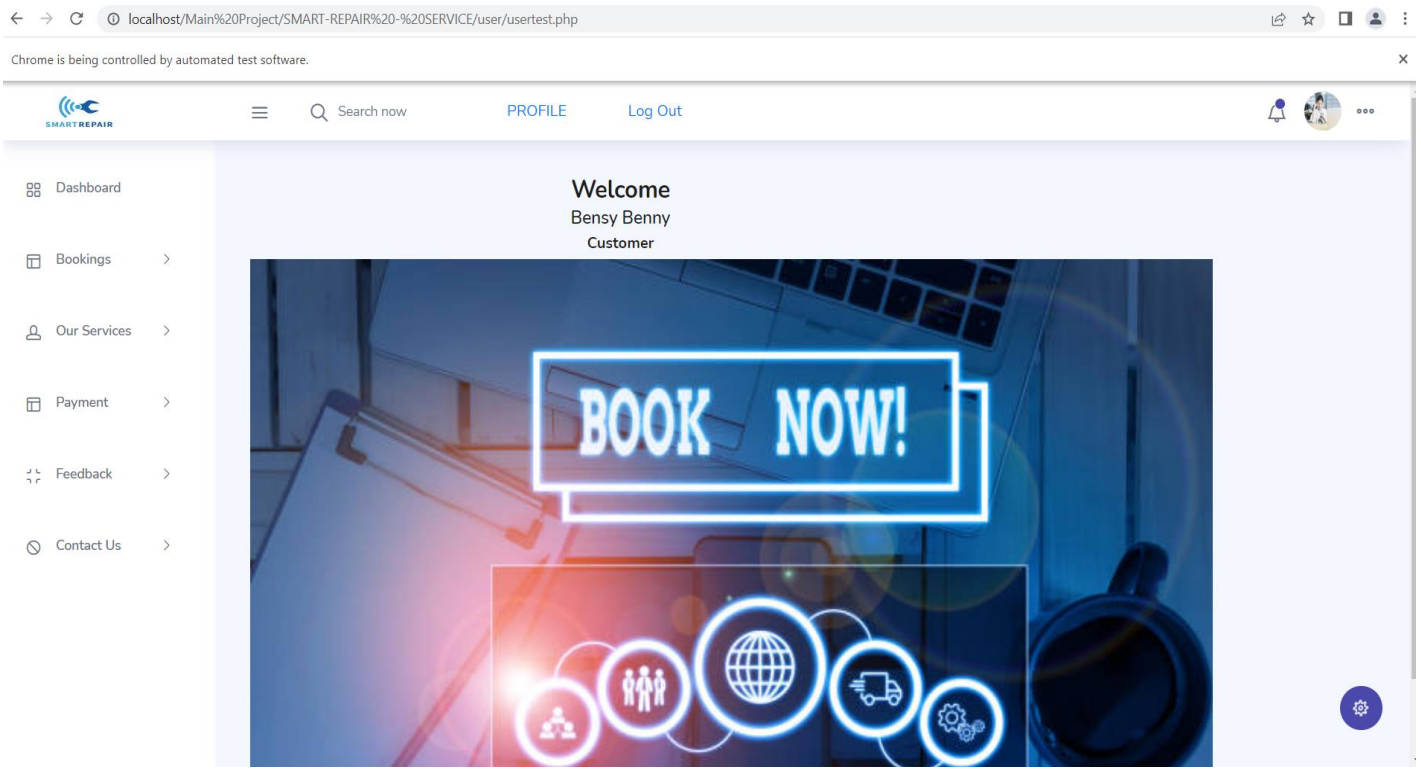
<table>
<tr><td colspan="6">Project Name: SMART REPAIR</td></tr>
<tr><td colspan="6">Login Test Case</td></tr>
<tr><td colspan="3">Test Case ID: repair_1</td><td colspan="3">Test Designed By: Bensy Benny</td></tr>
<tr><td colspan="3">Test Priority (Low/Medium/High): High</td><td colspan="3">Test Designed Date:19-07-2022</td></tr>
<tr><td colspan="3">Module Name: Login Screen</td><td colspan="3">Test Executed By: Ms. Sona Maria Sebastian</td></tr>
<tr><td colspan="3">Test Title: Verify login with valid username and password</td><td colspan="3">Test Execution Date: 18-07-2022</td></tr>
<tr><td colspan="3">Description: Test the Login Page</td><td colspan="3"></td></tr>
<tr><td colspan="6">Pre-Condition: User has valid username and password</td></tr>
<tr><td>Step</td><td>Test Step</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1</td><td>Navigation to Login Page</td><td></td><td>Login Page should be display ed</td><td>Login page displayed</td><td>Pass</td></tr>
<tr><td>2</td><td>Provide Valid username</td><td>Username : bensy</td><td rowspan="3">User should d be able to Login</td><td rowspan="3">User Logged in and navigated to User Dashboard</td><td rowspan="3">Pass</td></tr>
<tr><td>3</td><td>Provide Valid Password</td><td>Password:bensy@1997</td></tr>
<tr><td>4</td><td>Click on Sign In button</td><td></td></tr>
<tr><td>5</td><td>Provide Invalid username orpassword</td><td>Username:rose
Password: rose@2004</td><td rowspan="3">User should not be able to Login</td><td rowspan="3">Message for enter valid email id or password displayed</td><td rowspan="3">Pass</td></tr>
<tr><td>6</td><td>Provide Null username or Password</td><td>Username : null
Password: null</td></tr>
<tr><td>7</td><td>Click on Sign In button</td><td></td></tr>
</table>

Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database.

Code package

```
package testing1;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login {
public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\albin\\Downloads\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();

 driver.get("http://localhost/Main%20Project/SMART-REPAIR%20-%20SERVICE/loginn.php");
 driver.findElement(By.id("uname")).sendKeys("bensy");
 driver.findElement(By.id("password")).sendKeys("bensy@1997");
 driver.findElement(By.id("login")).click();
 String actualUrl="http://localhost/Main%20Project/SMART-REPAIR%20-
%20SERVICE/user/usertest.php";
 String expectedUrl= driver.getCurrentUrl();
 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
 System.out.println("Test passed");
 } else {
 System.out.println("Test failed");
 }
}
}

}
```


File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
bensy
JRE System Library [JavaSE-1.8]
src
testing1
JRE System Library [JavaSE-1.8]
src
testing1
login.java
Referenced Libraries
login.java
1 package testing1;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 public class login {
6 public static void main(String[] args) {
7 System.setProperty("webdriver.chrome.driver","C:\\Users\\albin\\Downloads\\chromedriver.exe");
8 WebDriver driver=new ChromeDriver();
9
10 driver.get("http://localhost/Main%20Project/SMART-REPAIR%20-%20SERVICE/loginn.php");
11 driver.findElement(By.id("uname")).sendKeys("bensy");
12 driver.findElement(By.id("password")).sendKeys("bensy@1997");
13 driver.findElement(By.id("login")).click();
14 String actualUrl="http://localhost/Main%20Project/SMART-REPAIR%20-%20SERVICE/user/usertest.php";
15 String expectedUrl= driver.getCurrentUrl();
16 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
17 System.out.println("Test passed");
18 } else {
19 System.out.println("Test failed");
20 }
21 }
22 }
23
Outline
testing1
login
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> login [Java Application] C:\Users\albin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (18-Jul-2022, 10:13:18 pm – 10:13:27 pm) [pid: 15
ChromeDriver was started successfully.
Jul 18, 2022 10:13:25 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 10:13:25 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed


localhost/Main%20Project/SMART-REPAIR%20-%20SERVICE/user/usertest.php
Chrome is being controlled by automated test software.
SMARTREPAIR
Search now
PROFILE
Log Out
Dashboard
Bookings
Our Services
Payment
Feedback
Contact Us
Welcome
Bensy Benny
Customer
BOOK NOW!

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 meansconvening 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

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. 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 examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. It is crucial for the participants to have faith in their roles in the new system in order to achieve the goal and benefits anticipated from a computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

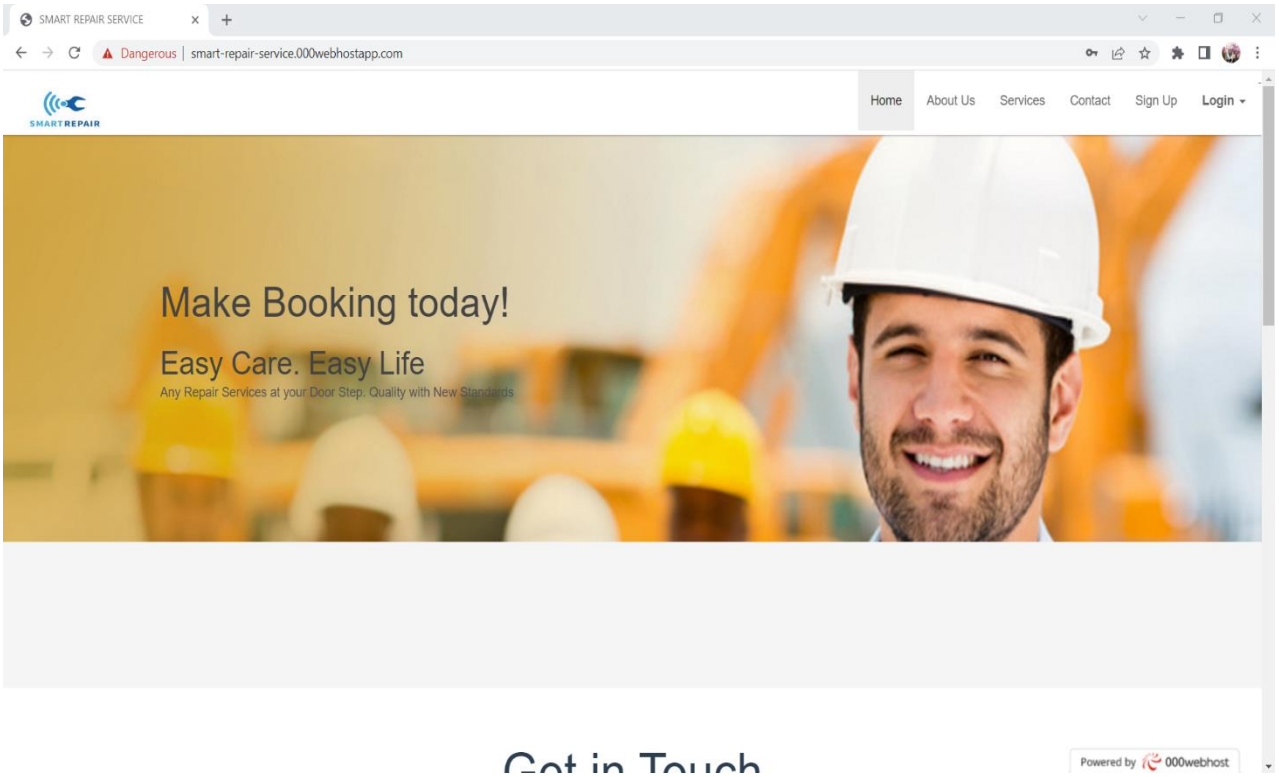
After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the 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".

6.2.4 Hosting

000webhost is a free website hosting solution that provides an array of valuable features, including a website builder, WordPress support, and no ads.



6.2.5 Hosting Method Steps

Go to the 000webhost homepage

• Sign up to make an account.

• Make sure you verify your email.

• Click + button to create a new website.

• Enter your details and click create.

• Click Manage Website to start building website.

• Choose a platform for your website.

• In website Builder, Drag and Drop the project files.

• Insert database in Database Management.

• Get the custom Domain name.

• Publish the website.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In comparison to previous experiences, No modern technologies exist in such situation rather giving proper Repair service system. And when every activity related to the repair service was very time taken to provide services to the users. To overcome the above problem, this system helps to reduce burden in finding solutions for the users required services, the proposed system provides several services by providing employers at your doorstep in one click.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the admin add necessary services and posts informational to the users.
- Users can able to give ratings to employers based on their services.
- Users can able to request for more services.
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- Pankaj Jalote, "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
- http://ieeeprojectcentre.in/blog/online-electrical-complaint-management-system.php
- https://ukdiss.com/examples/android-app-for-household-services.php
- https://app.diagrams.net
- http://stackoverflow.com
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample

Index Page

```
<?php
include_once 'assets/conn/dbconnect.php';session_start();

if (isset($_POST['login']))
{
$username = mysqli_real_escape_string($con,$_POST['username']);
$password  = mysqli_real_escape_string($con,$_POST['password']);

$res = mysqli_query($con,"SELECT * FROM tbl_login WHERE Username =
    '$username' andPassword='$password'");
$row=mysqli_fetch_array($res,MYSQLI_ASSOC);
$cc=mysqli_num_rows($res);if ($cc==1)
{
if($row['UserType_id']==1)
{
$_SESSION['customerSession'] = $row['Login_id'];
?>
<script type="text/javascript">alert('Login Success');
</script>
<?php
header("Location: user/user.php");
}
else if($row['UserType_id']==2)
{
$_SESSION['customerSession'] = $row['Login_id'];
?>
<script type="text/javascript">alert('Login Success');
</script>
<?php
header("Location: user/user.php");
}
else if($row['UserType_id']==3)
{
$_SESSION['customerSession'] = $row['Login_id'];
?>
<script type="text/javascript">alert('Login Success');
</script>
<?php
header("Location: user/user.php");
}
else if($row['UserType_id']==0)
{
$_SESSION['customerSession'] = $row['Login_id'];header("Location:
    template/admindashboard.php");
}
}
```

```
else
{
?>
<script> alert('wrong input ');
</script>
<?php
}
}

?>
<?php
if (isset($_POST['signup'])) {
$customerFirstName =
    mysqli_real_escape_string($con,$_POST['customerFirstName']);
$customerLastName  =
    mysqli_real_escape_string($con,$_POST['customerLastName']);
$customerAddress        =
    mysqli_real_escape_string($con,$_POST['customerAddress']);
$customerAadhar         =
    mysqli_real_escape_string($con,$_POST['customerAadhar']);
$customerMobile         =
    mysqli_real_escape_string($con,$_POST['customerMobile']);
$idcustomer      = mysqli_real_escape_string($con,$_POST['idcustomer']);
$password        = mysqli_real_escape_string($con,$_POST['password']);
$userType        = mysqli_real_escape_string($con,$_POST['userType']);
$Username        = mysqli_real_escape_string($con,$_POST['Username']);

$query = " INSERT INTO tbl_registration( Login_id, Fname, Lname, Address,
    Mobile_no,Aadhar_no, Status)
VALUES ( '$idcustomer','$customerFirstName',
    '$customerLastName','$customerAddress','$customerMobile','$customerAadhar','A
    ctive' ) ";
$result = mysqli_query($con, $query);if( $result )
{
$query2="INSERT INTO tbl_login(Login_id, UserType_id, Username, Password)
    VALUES ('$idcustomer','$userType','$Username','$password')";
$result2 = mysqli_query($con, $query2);
?>
<script type="text/javascript">
alert('Register success. Please Login to make a Booking.');
</script>
<?php
}
else
{
?>
<script type="text/javascript">
alert('User already registered. Please try again');
</script>
<?php
}
```

```
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>SMART REPAIR SERVICE</title>
<link href="assets/css/style.css" rel="stylesheet">
<link href="assets/css/style1.css" rel="stylesheet">
<link href="assets/css/blocks.css" rel="stylesheet">
<link href="assets/css/date/bootstrap-datepicker.css" rel="stylesheet">
<link href="assets/css/date/bootstrap-datepicker3.css" rel="stylesheet">
<link rel="stylesheet" href="https://formden.com/static/cdn/font-
    awesome/4.4.0/css/font-awesome.min.css" />
<link href="assets/css/material.css" rel="stylesheet">
</head>
<body>

<nav class="navbar navbar-default navbar-fixed-top" role="navigation">
<div class="container-fluid">

<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
    target="#bs-example-navbar-collapse-1">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="index.php"><img alt="Brand"
    src="assets/img/logo.jpg" height="40px"></a>

</div>
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">


<ul class="nav navbar-nav navbar-right">
<li class="active"><a href="index.php">Home</a></li>
<li class="dropdown">

<a href="#" data-toggle="dropdown" class="dropdown-toggle">About Us <b
    class="caret"></b></a>
<ul class="dropdown-menu">
<li><a href="about.html">Company</a></li>
<li><a href="#">Our Team</a></li>
<li><a href="#">News</a></li>
<li><a href="#">Investors</a></li>
```

```
</ul>
</li>
<li><a href="services.html">Services</a></li>
<li><a href="contact.html">Contact</a></li>
<li><a href="#" data-toggle="modal" data-target="#myModal">Sign Up</a></li>


<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown"><b>Login</b>
<span class="caret"></span></a>
<ul id="login-dp" class="dropdown-menu">
<li>
<div class="row">
<div class="col-md-12">
<form class="form" role="form" method="POST" accept- charset="UTF-8" >
<div class="form-group">
<label class="sr-only" for="idcustomer">Username</label>
<input type="text" class="form-control" name="username"

placeholder="Username" required>
</div>
<div class="form-group">
<label class="sr-only" for="password">Password</label>
<input type="password" class="form-control" name="password"
    placeholder="Password" required>
</div>
<div class="form-group">
<button type="submit" name="login" id="login" class="btn btn-primary btn-
    block">Sign in</button>
</div>
</form>
</div>
</div>
</li>
</ul>
</li>
</ul>
</div>
</div>
</nav>
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-
    labelledby="myModalLabel">
<div class="modal-dialog" role="document">
<div class="modal-content">

<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
    aria-hidden="true">&times;</span></button>
<h3 class="modal-title">Sign Up</h3>
</div>
```

```
<div class="modal-body">


<div class="container" id="wrap">
<div class="row">
<div class="col-md-6">

<form action="<?php $_PHP_SELF ?>" method="POST" accept-charset="utf-8"
    class="form" role="form">
<h4>Smart Repair at your Door Step.</h4>
<div class="row">
<div class="col-xs-6 col-md-6">
<input type="text" name="customerFirstName" value=""class="form-control input-lg"
    placeholder="First Name" required />
</div>
<div class="col-xs-6 col-md-6">
<input type="text" name="customerLastName" value=""class="form-control input-lg"
    placeholder="Last Name" required />
</div>
</div>
<input type="text" name="customerAadhar" value="" class="form-controlinput-lg"
    placeholder="Your Aadhar No"  required/>
<input type="text" name="customerMobile" value="" class="form-controlinput-lg"
    placeholder="Your Mob Number"  required/>

<input type="number" name="idcustomer" value="" class="form-controlinput-lg"
    placeholder="Login ID "  required/>
<input type="text" name="Username" value="" class="form-control input-lg"
    placeholder="Username "  required/>
<input type="password" name="password" value="" class="form-controlinput-lg"
    placeholder="Password"  required/>



<span class="help-block">By clicking Create my account, you agree toour Terms and
    that you have read our Data Use Policy, including our Cookie Use.</span>

<button class="btn btn-lg btn-primary btn-block signup-btn"type="submit"
    name="signup" id="signup">Create my account</button>
</form>

</div>
</div>
</div>
</div>
</div>
</div>
</div>

<section id="promo-1" class="content-block promo-1 min-height-600px bg-offwhite">
<div class="container">
```

```
<div class="row">
<div class="col-md-5">
<h2>Make Booking today!</h2>
<h3>Easy Care. Easy Life
<p>Any Repair Services at your Door Step. Quality with New Standards  </p>



<div id="txtHint"><b> </b></div>


</div>

</div>

</div>
</section>

<section id="content-1-9" class="content-1-9 content-block">
<div class="container">
<div class="underlined-title">
<h1>Get in Touch</h1>
<hr>
<h2>We Discover, You Explore</h2>
<h3>Just a Click for Smart Service</h3>
</div>
<section class="jumbobox">
<div class="container">
<div class="row">
<div class="col-lg-12">
<div><center><h1>Services</center></h1>One Call can Solve allyour Problems, The
    Quality you Expect, The Service You Deserve!<br> We Bring New Life to Broken
    Things</div>

                    </div>
                <
<                   /
    /               d
    d               i
    i               v
    v               >
    >
```

```
</section>


<section id="content">


<div class="container">
<div class="row">
<div class="skill-home"> <div class="skill-home-solid clearfix">
<div class="col-md-3 text-center">
<div class="box">
<span class="icons c1"><i class="icon-settings icons"></i></span> <divclass="box-
    area">
<h3>Plumber</h3> <p>Done Right the First Time, Acting is not an important Job in
    the Scheme of Things.Plumbing is!</p> <p><a href="#">Learn
    More</a></p></div>
</div></div>
<div class="col-md-3 text-center">
<div class="box">
<span class="icons c2"><i class="icon-login icons"></i></span> <div class="box-
area">
<h3>Electrician</h3> <p>An Electrical Shock can tell you the importance of an
Electrician in the electrical Line!</p> <p><a href="#">Learn More</a></p></div>
</div></div>
<div class="col-md-3 text-center">
<div class="box">
<span class="icons c3"><i class="icon-user icons"></i></span> <div class="box-
area">
<h3>Home Cleaning</h3> <p>Directly Employed, Fully insured and Vetted,
Flexible Cleaning Plans, Call Today</p> <p><a href="#">Learn More</a></p></div>
</div></div>
<div class="col-md-3 text-center">
<div class="box">
<span class="icons c4"><i class="icon-home icons"></i></span> <div class="box-
area">
<h3>Vehicle Repair</h3> <p>Genuine Collision Services, Get back to your Road,
Great Communication and Time Sensitive</p> <p><a href="#">Learn
    More</a></p></div>
</div></div>
</div></div></div>
</div>


</div>
</section>


<div class="copyright-bar bg-black">
<div class="container">

<p class="pull-right small"><a href="adminlogin.php">admin</a></p>
```

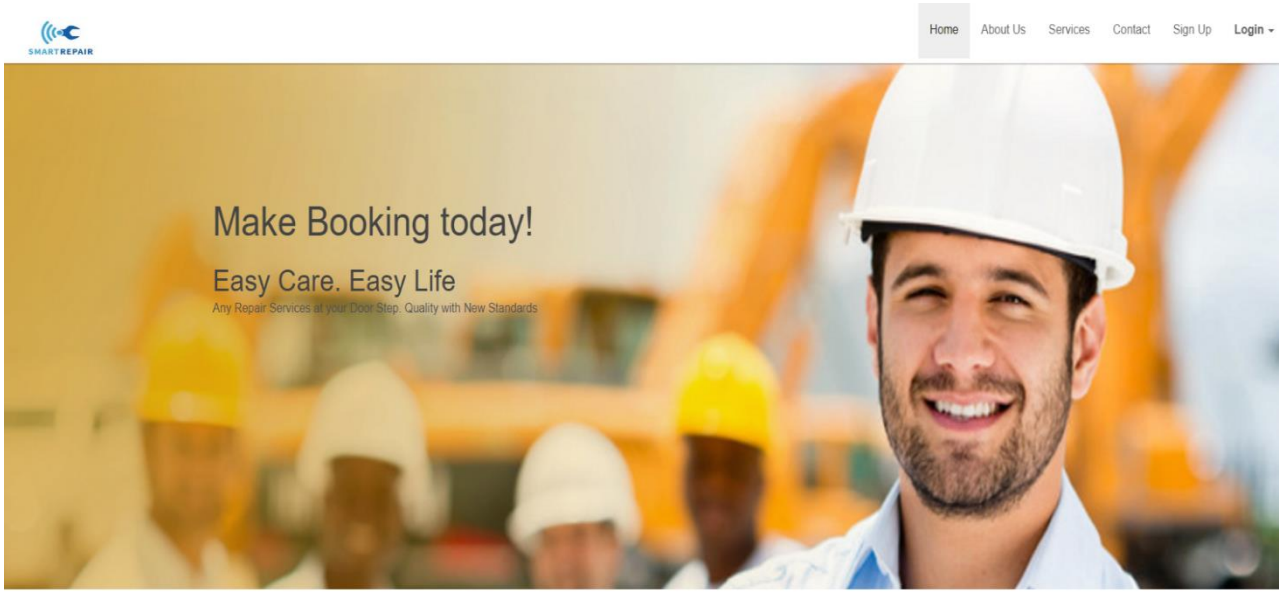
```
<li><a href="template/admindashboard.php">DEMO</a></li>
</div>
</div>

</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/date/bootstrap-datepicker.js"></script>
<script src="assets/js/moment.js"></script>
<script src="assets/js/transition.js"></script>
<script src="assets/js/collapse.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script type="text/javascript">
$('#myModal').on('shown.bs.modal', function () {
$('#myInput').focus()
})
</script>


<script>
```

```
$(document).ready(function(){
var date_input=$('input[name="date"]');
var container=$('.bootstrap-iso form').length>0 ? $('.bootstrap-iso form').parent() : "body";
    date_input.datepicker({
format: 'yyyy-mm-dd',container: container, todayHighlight: true, autoclose: true,
})

})

</script>



</body>
</html>
```

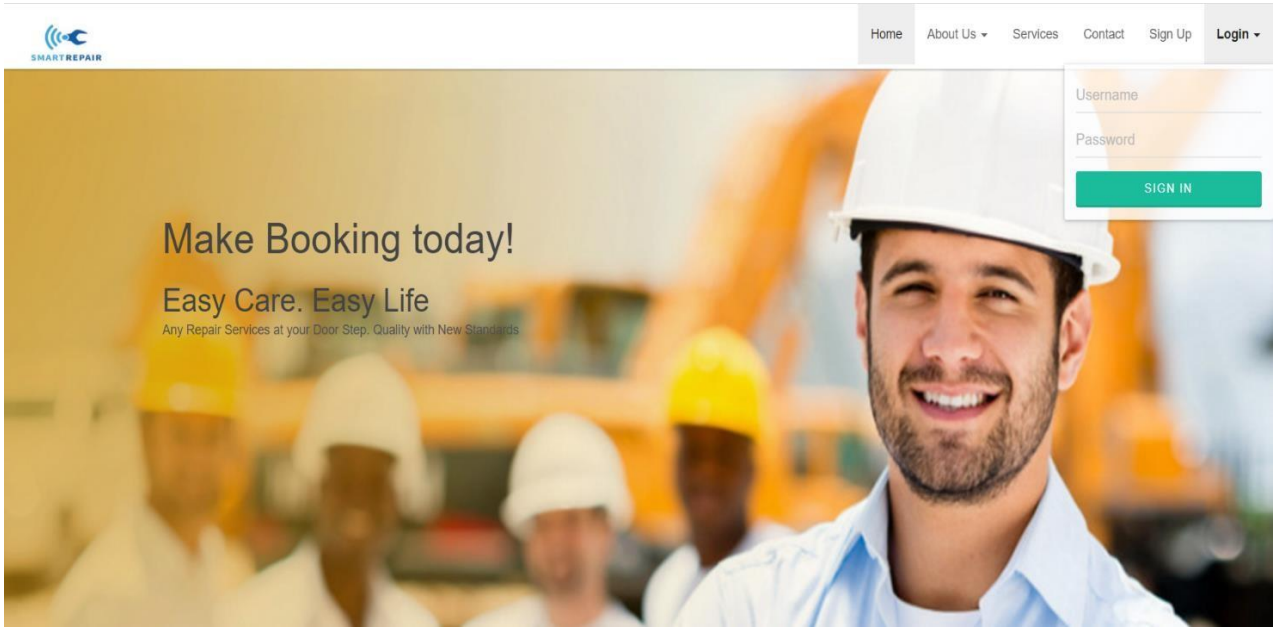
9.2 Screen Shots

USER PAGES User

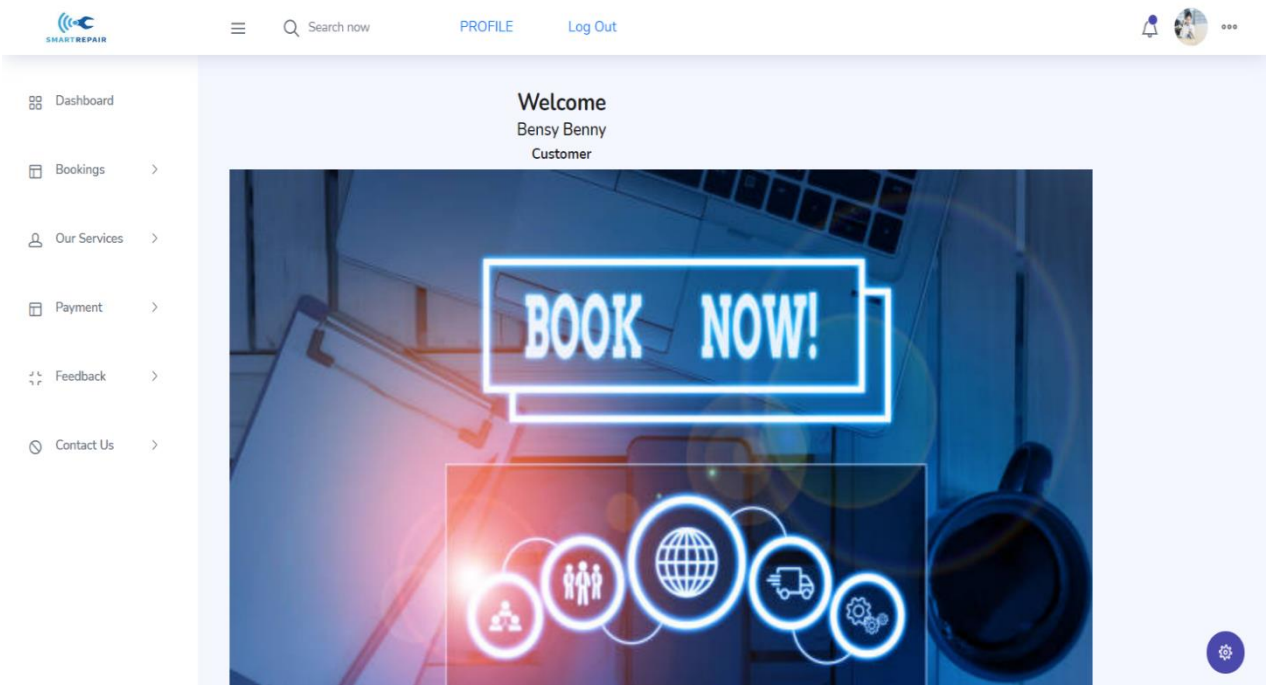
Home page



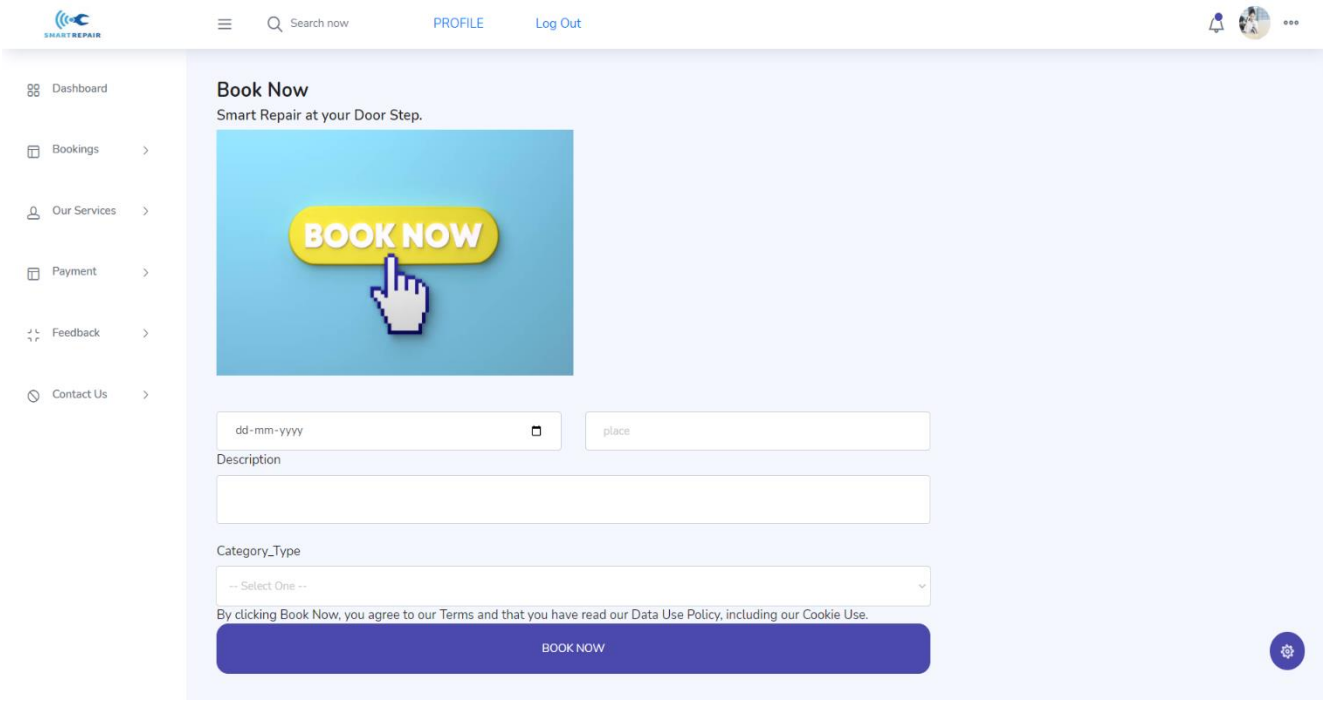
Login page



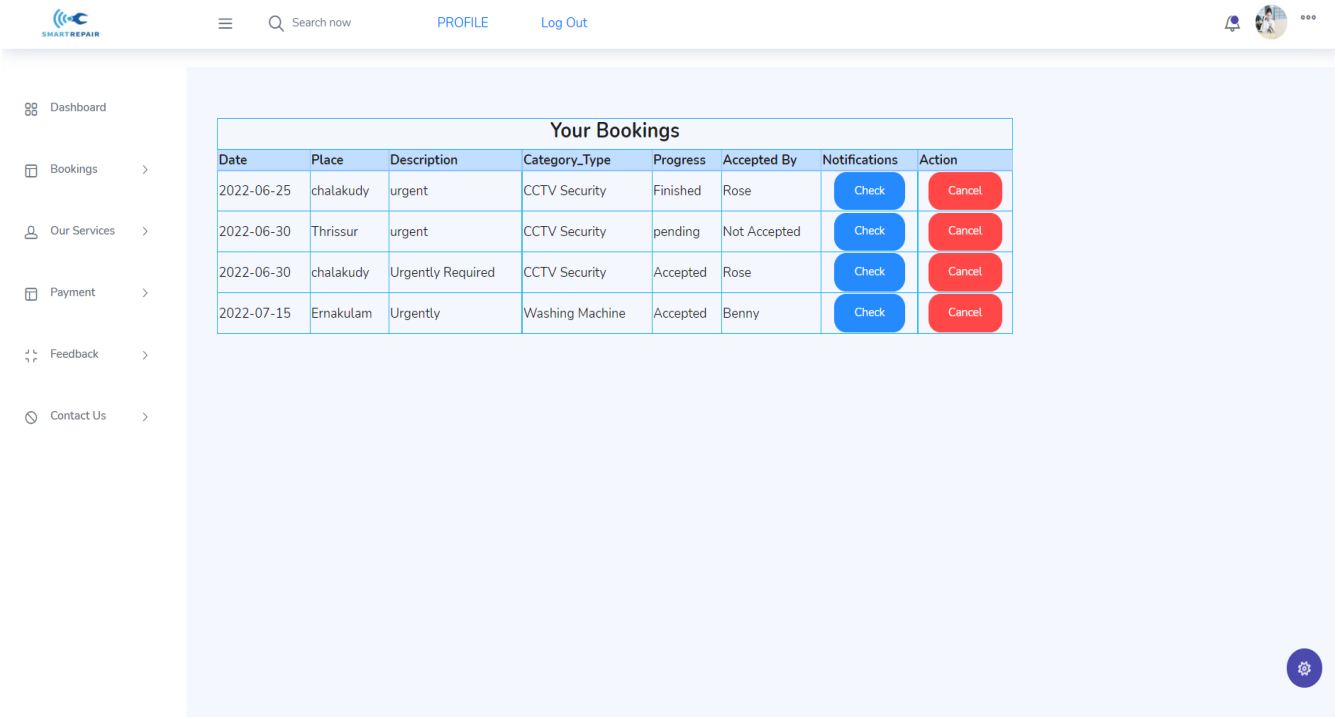
Customer Dashboard page



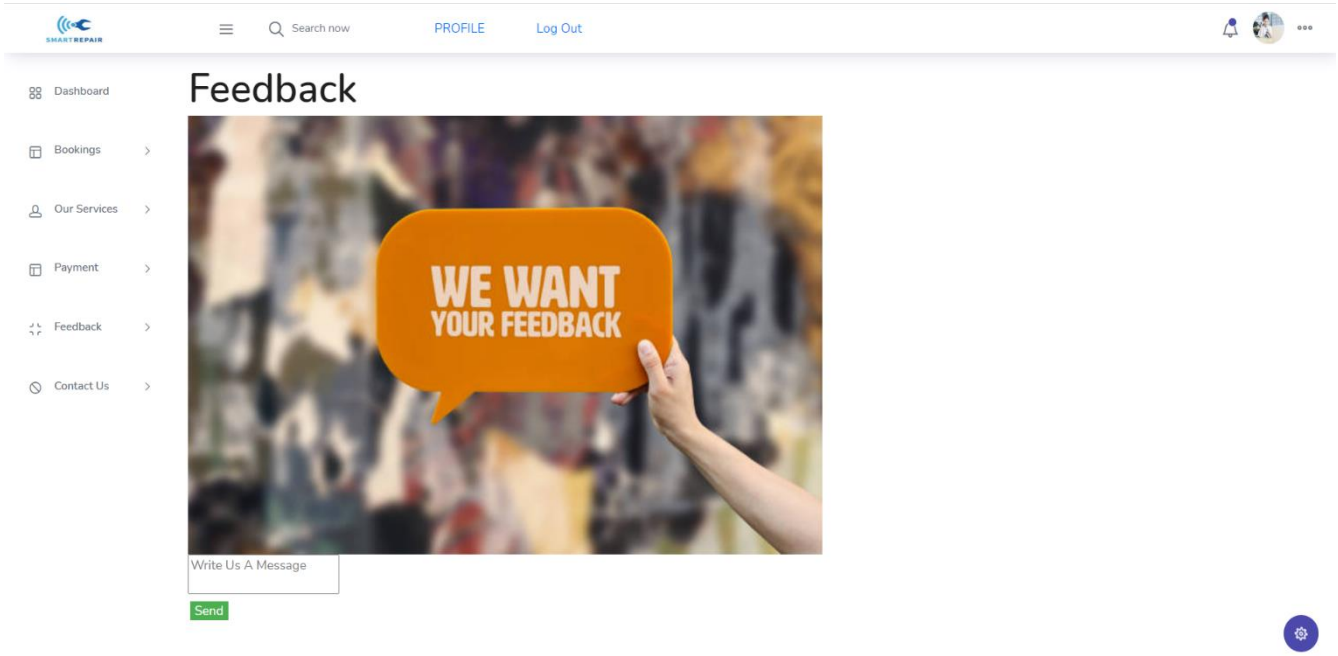
Customer Booking Page



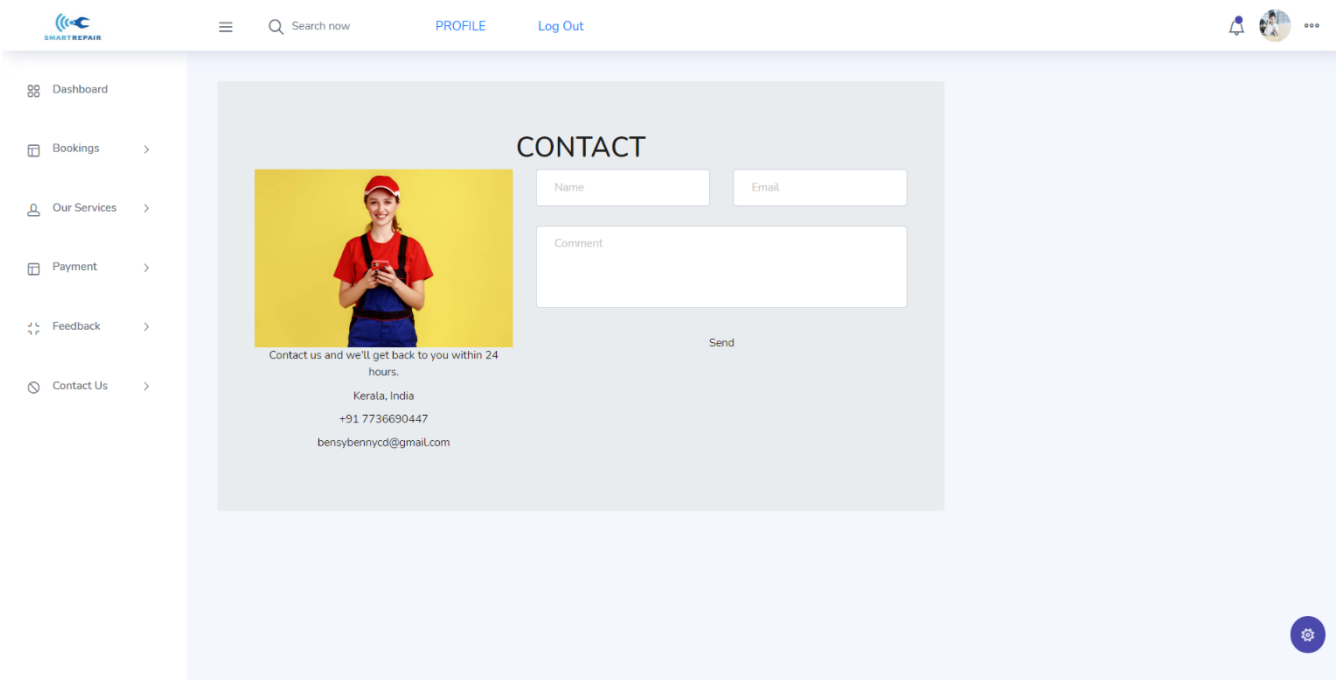
Customer Bookings



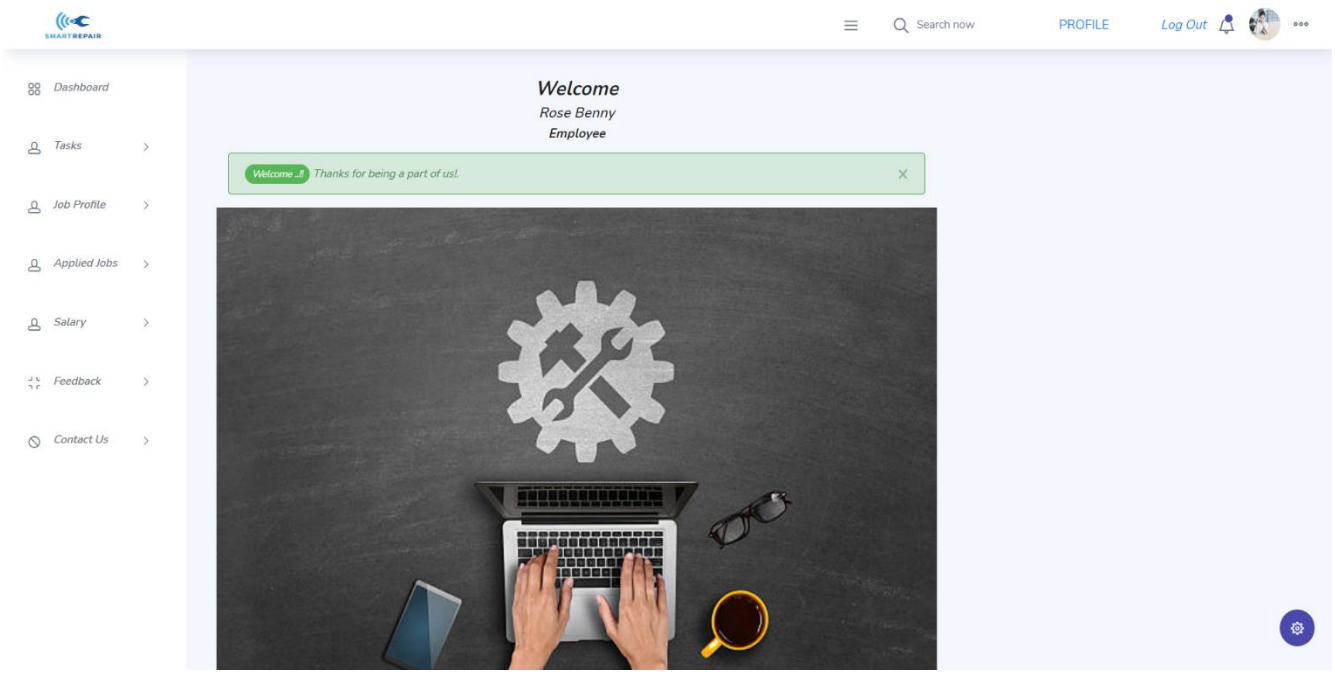
Customer Feedback



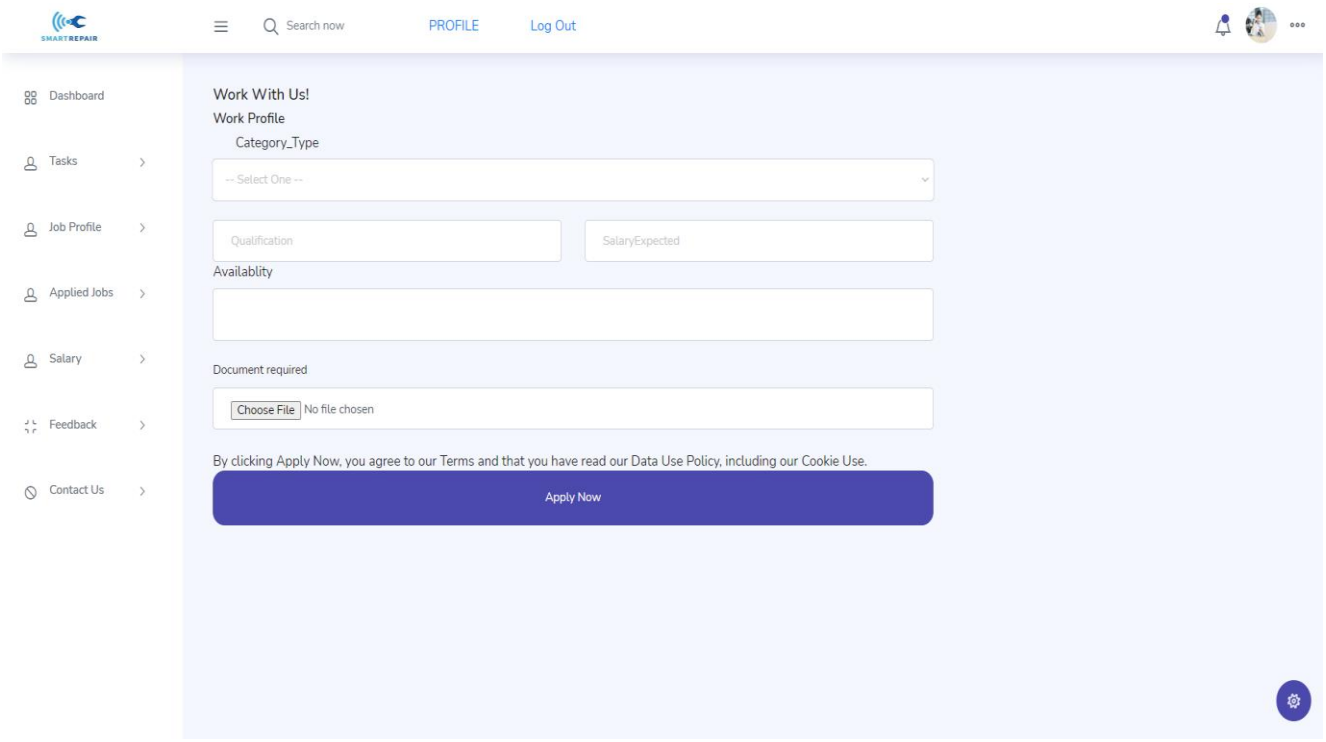
Customer Contact Us Page



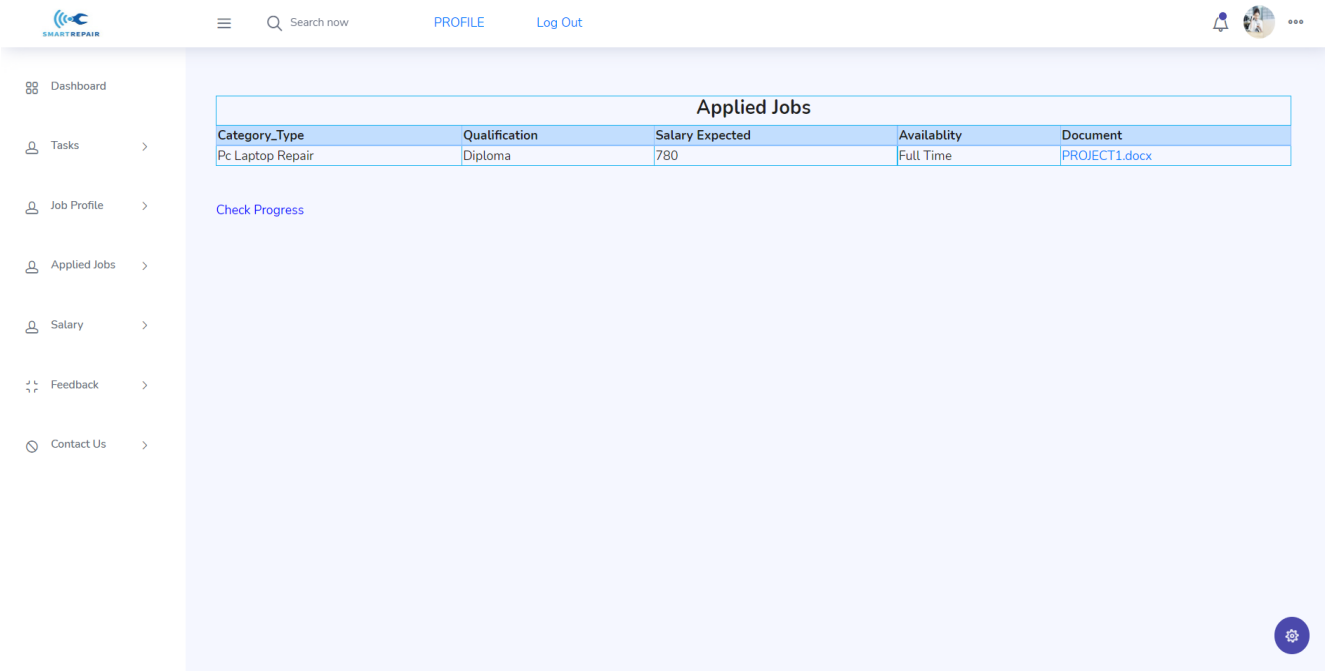
Employee Dashboard Page



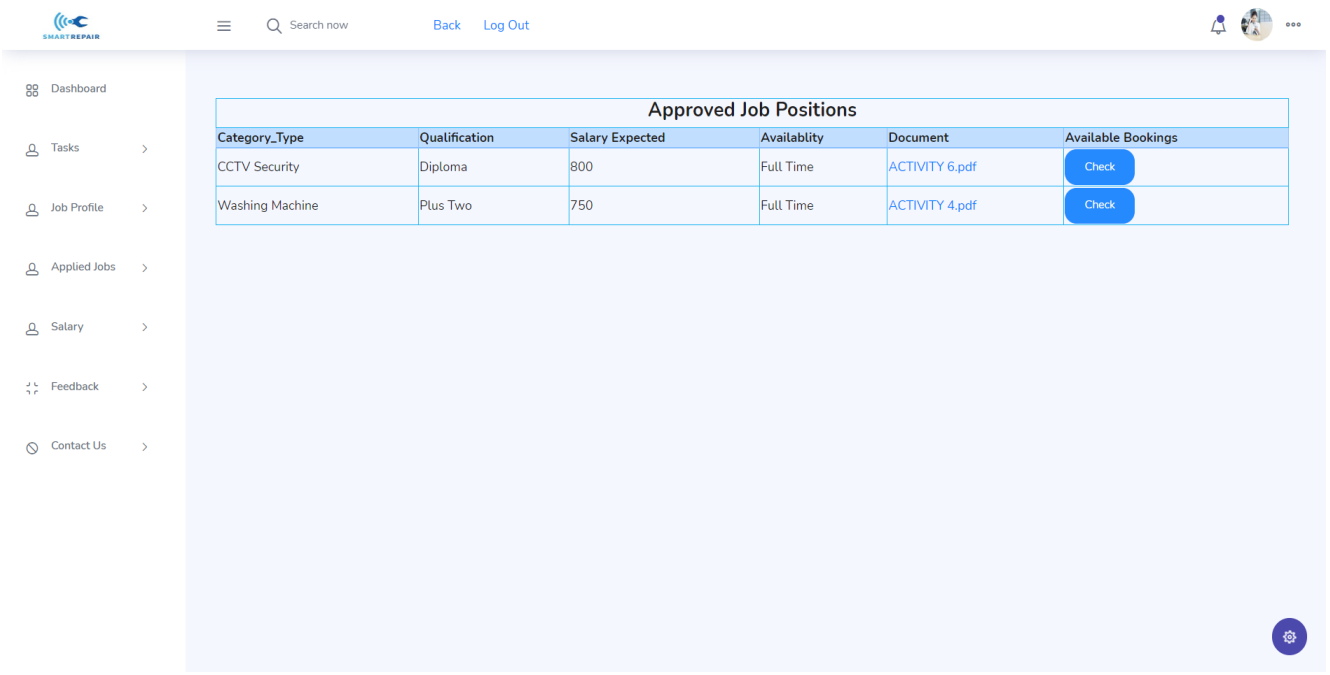
Employee Job Profile Page



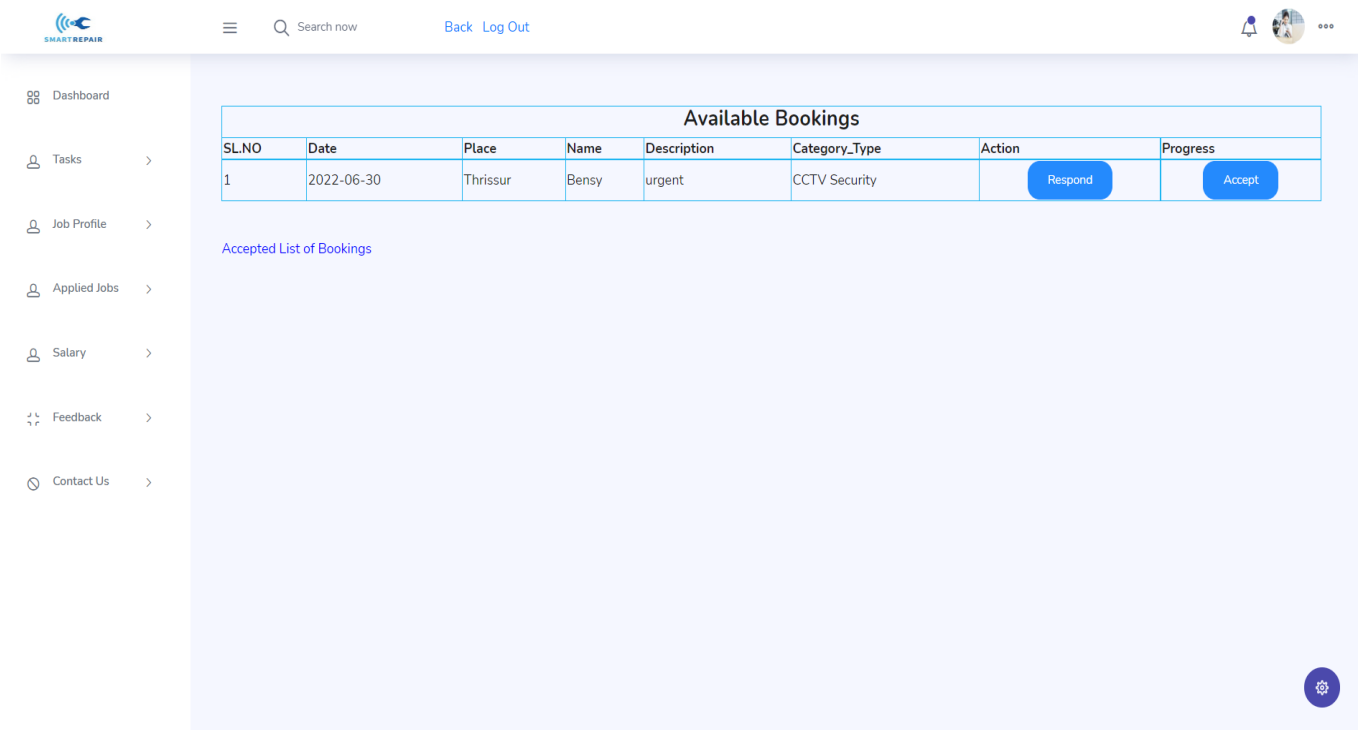
Employee Applied Jobs Page



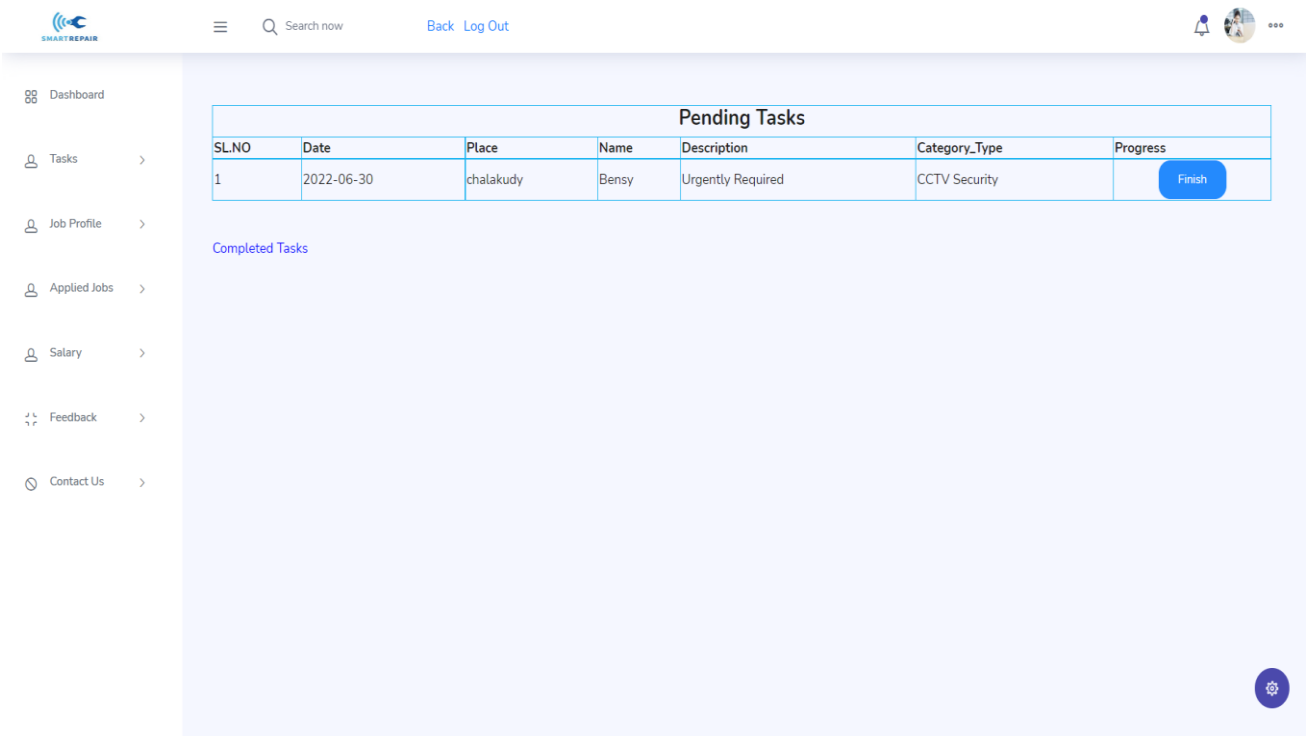
Employee Approved Jobs Progress Page



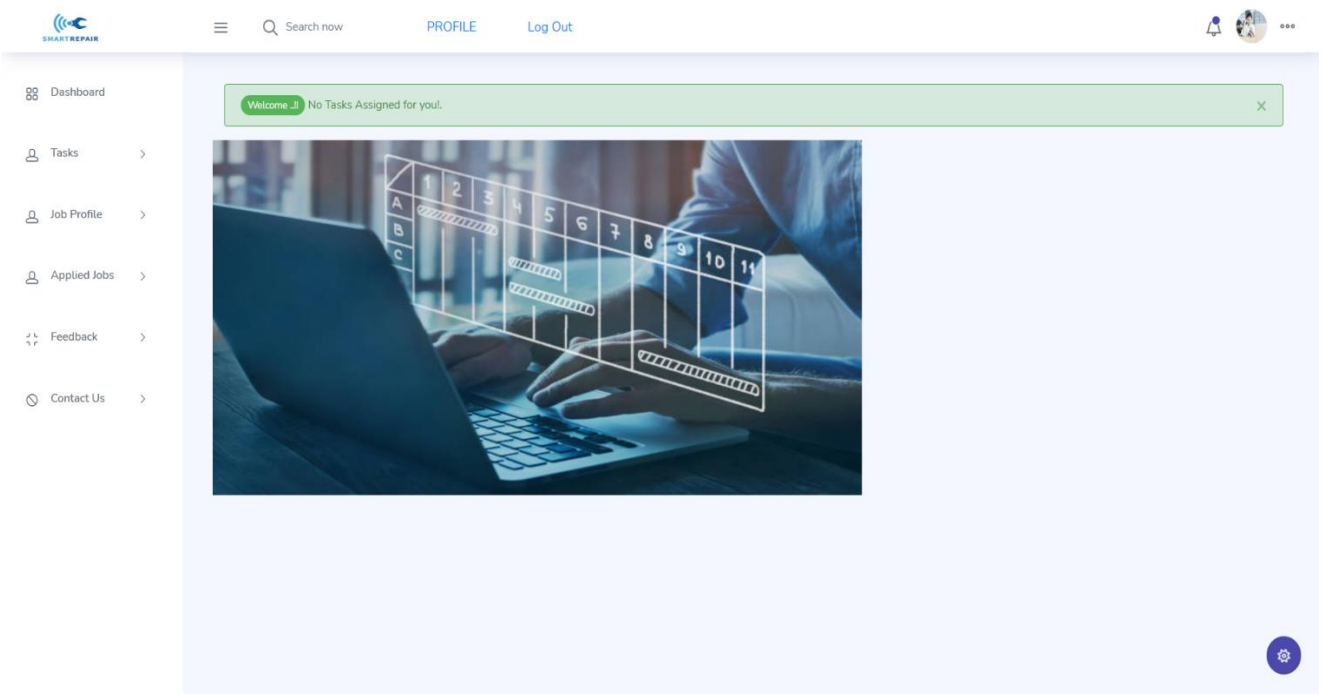
Employee Available Bookings Page



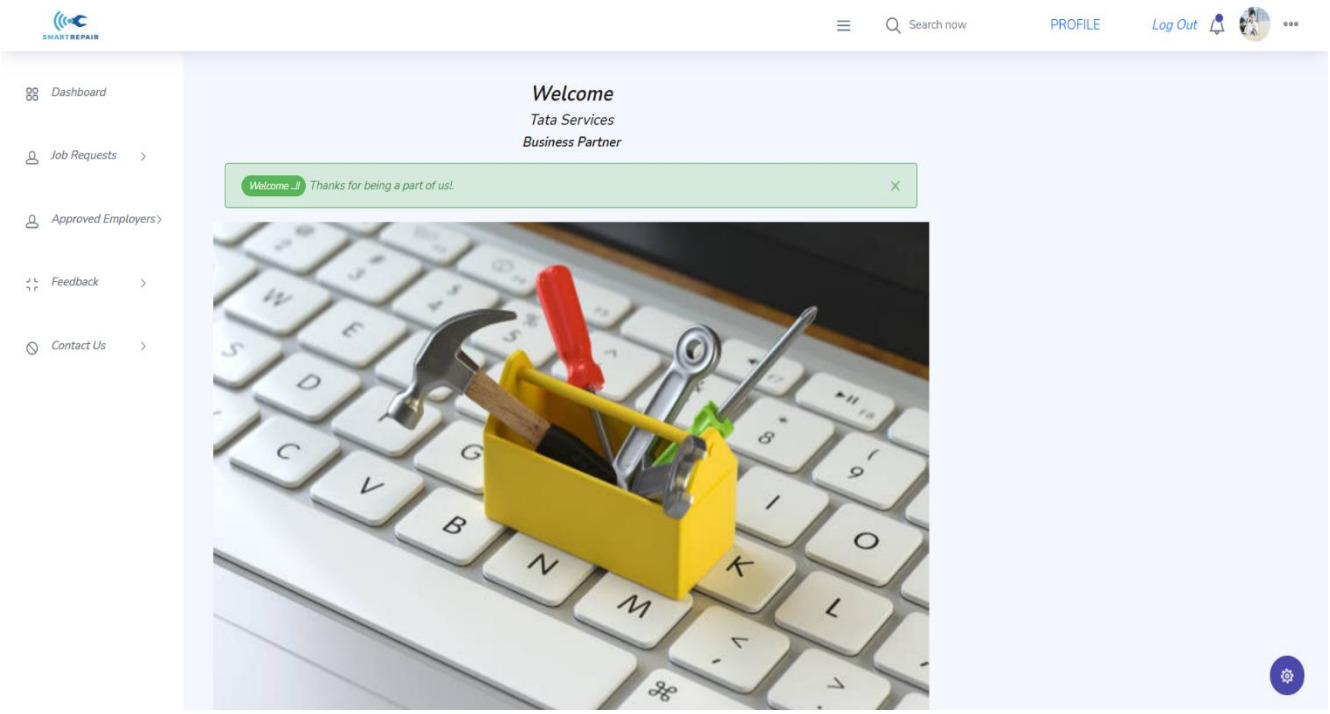
Employee Accepted List of Bookings



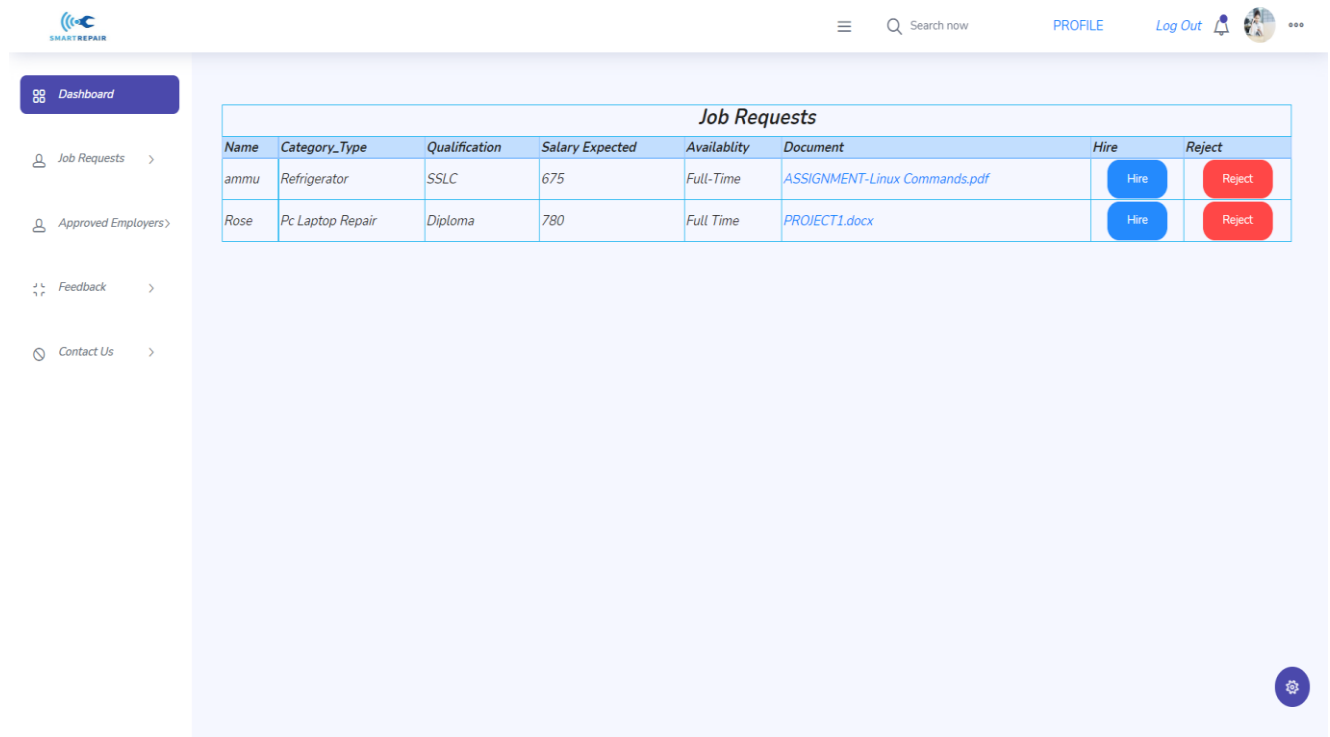
Employee Tasks Page



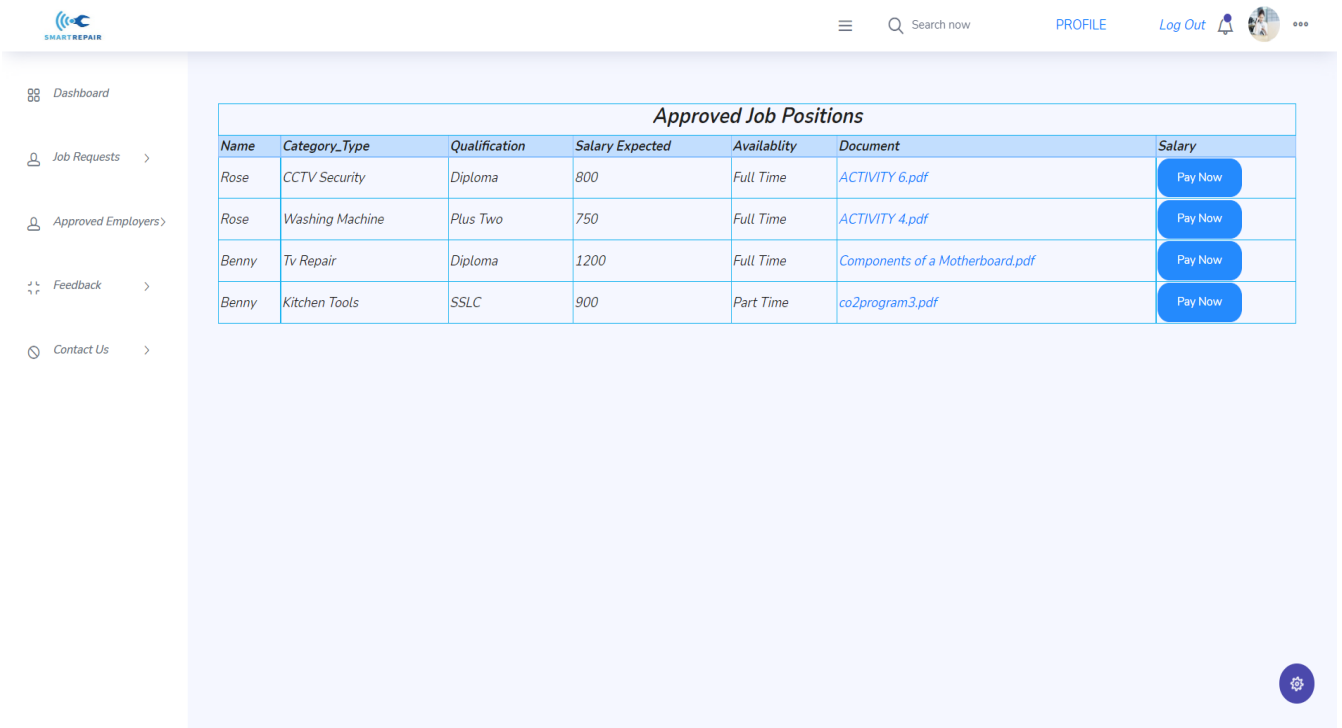
Buisness Partner Dashboard



View Job Requests

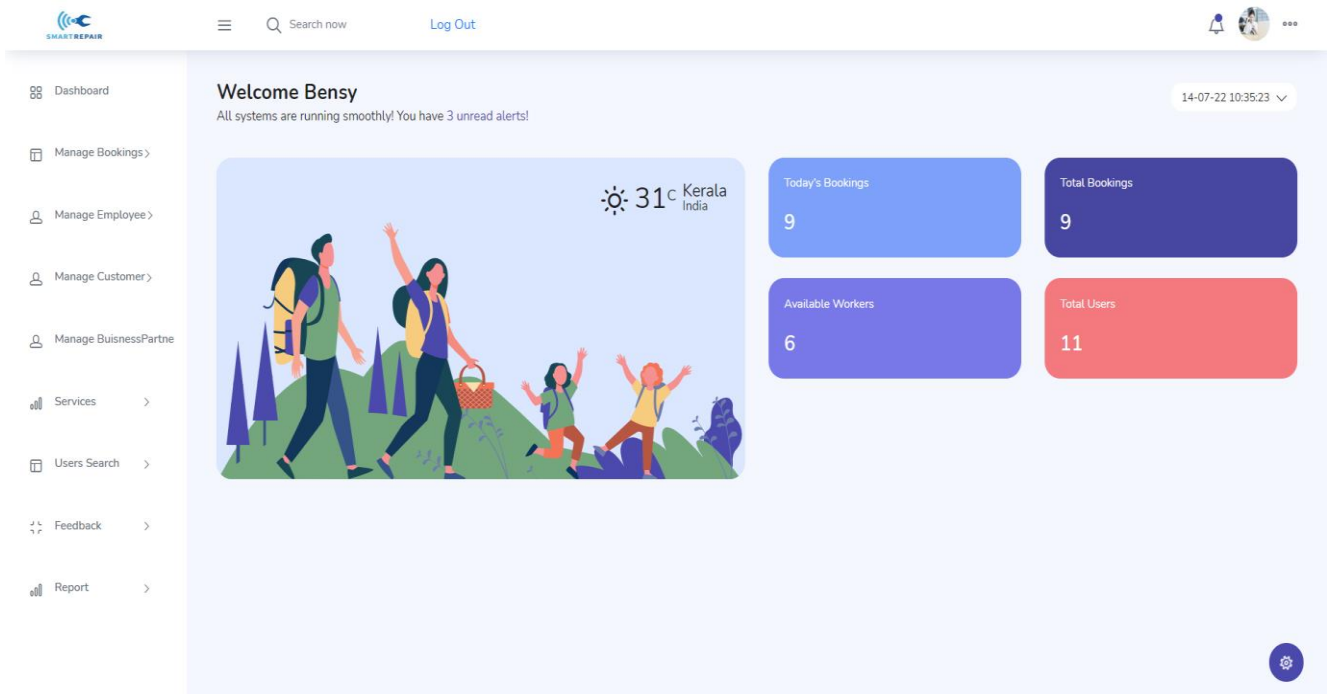


View Approved Employers

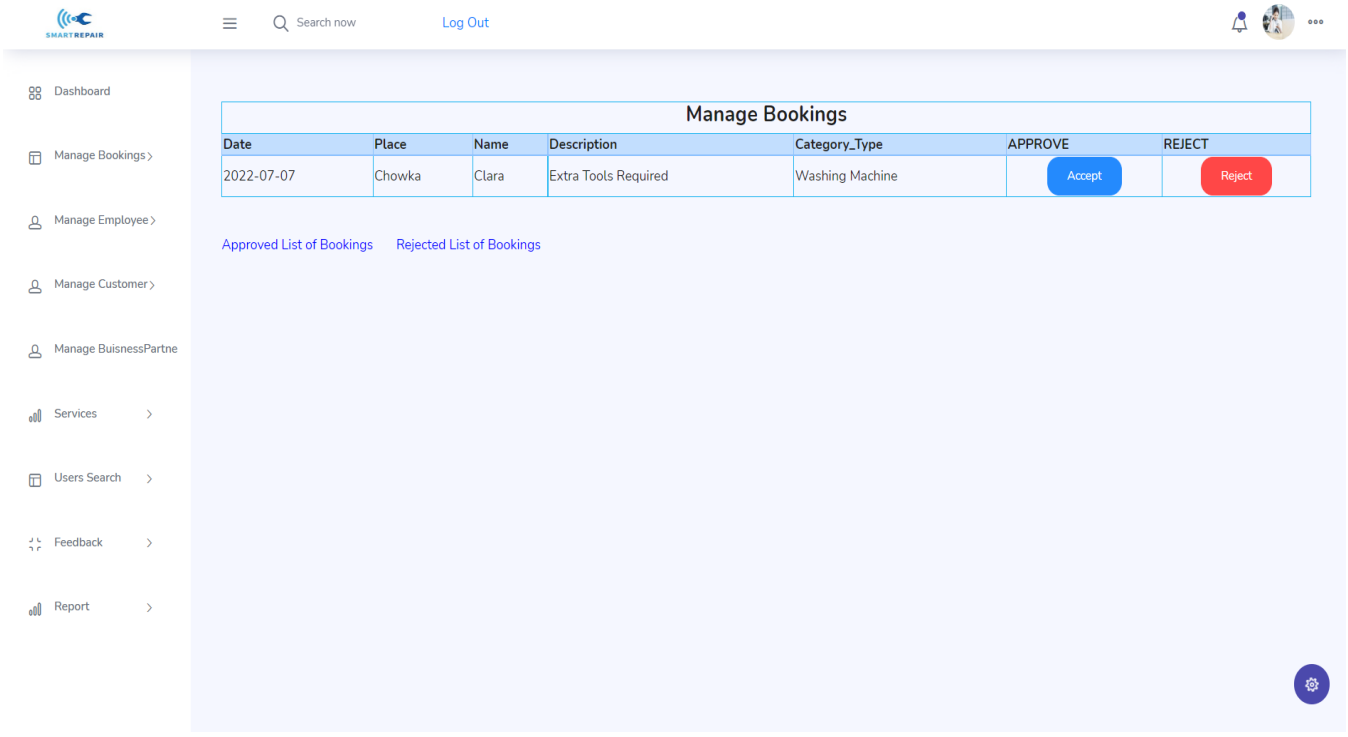


ADMIN PAGES

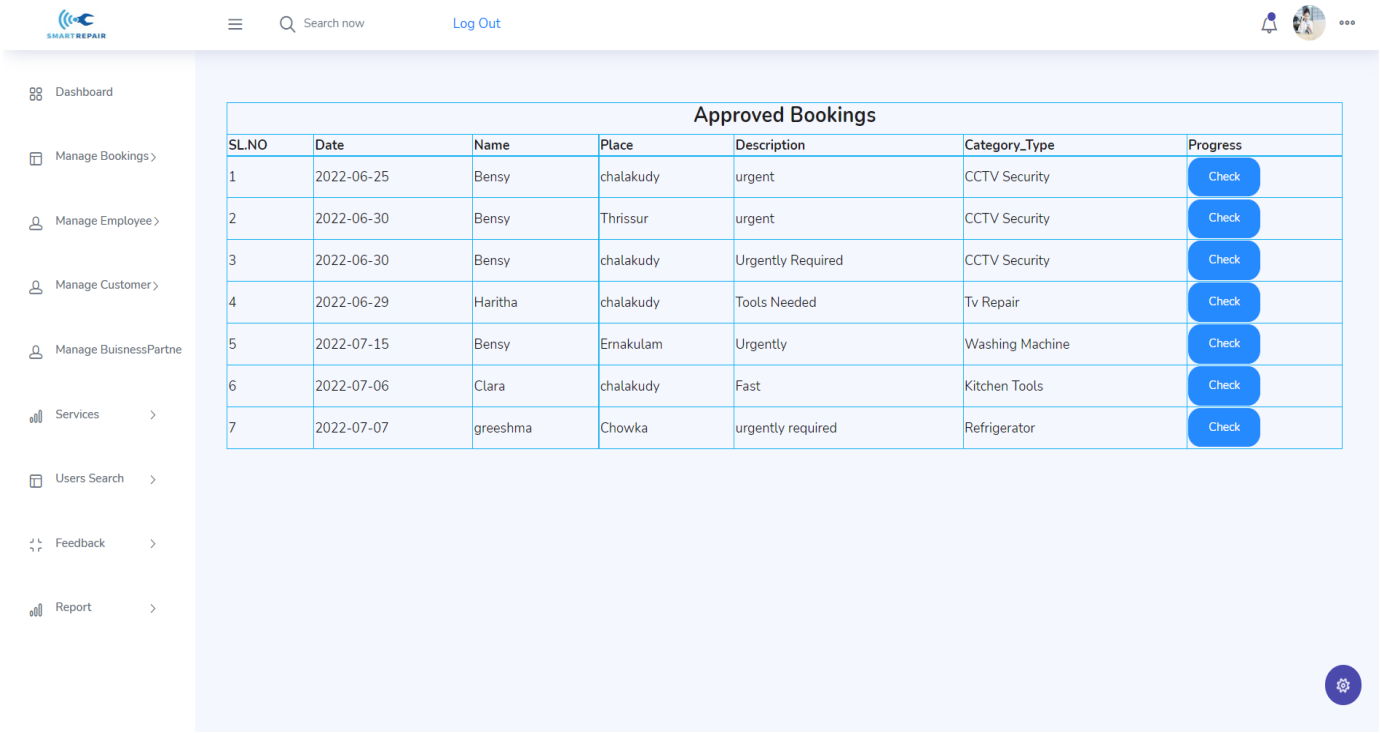
Admin Dashboard Page



View Manage Bookings



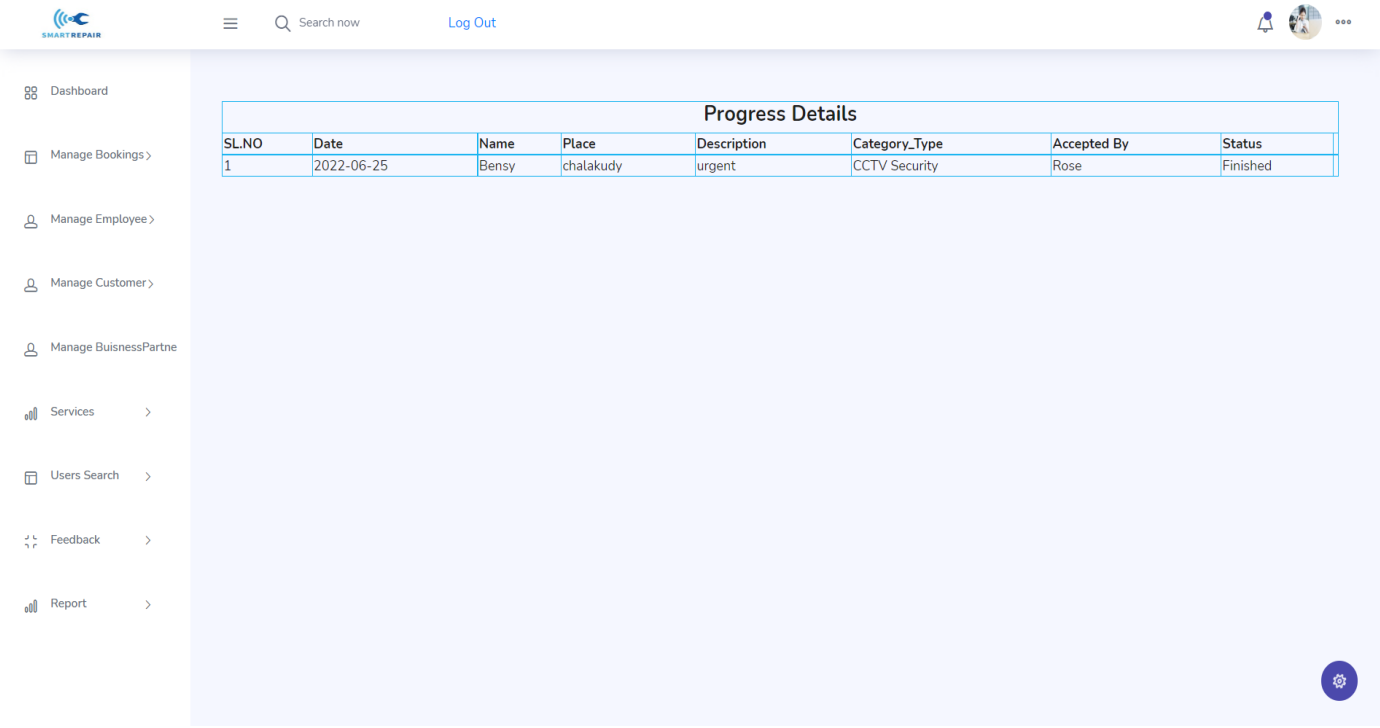
View Approved Bookings



Approved Bookings

SL.NO	Date	Name	Place	Description	Category_Type	Progress
1	2022-06-25	Bensy	chalakudy	urgent	CCTV Security	Check
2	2022-06-30	Bensy	Thrissur	urgent	CCTV Security	Check
3	2022-06-30	Bensy	chalakudy	Urgently Required	CCTV Security	Check
4	2022-06-29	Haritha	chalakudy	Tools Needed	Tv Repair	Check
5	2022-07-15	Bensy	Ernakulam	Urgently	Washing Machine	Check
6	2022-07-06	Clara	chalakudy	Fast	Kitchen Tools	Check
7	2022-07-07	greeshma	Chowka	urgently required	Refrigerator	Check

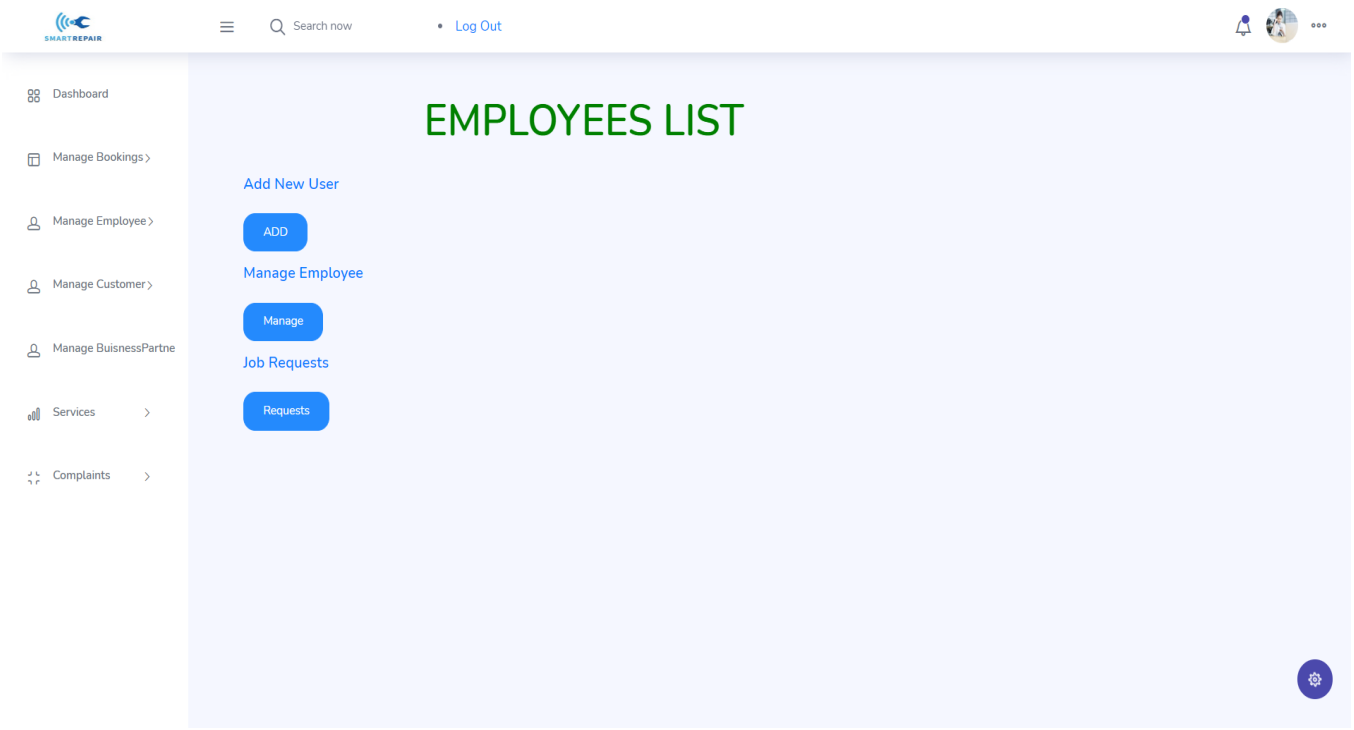
View Progress



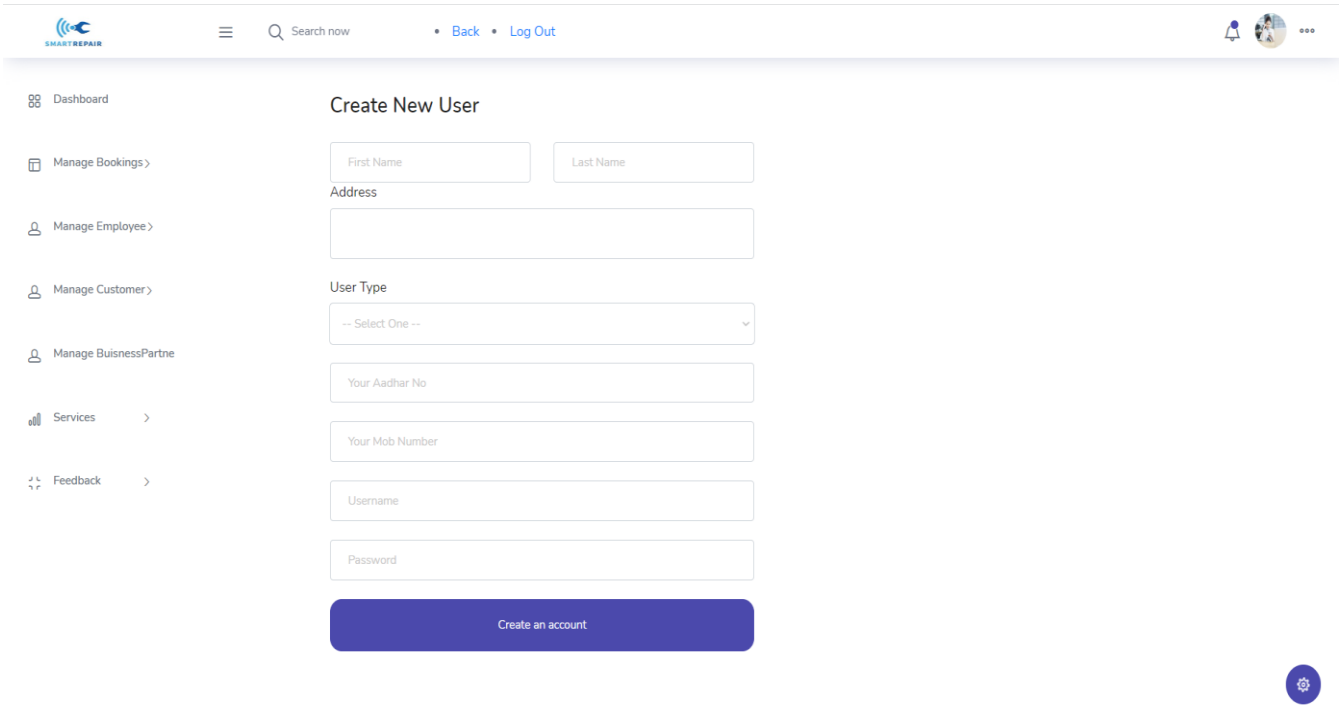
Progress Details

SL.NO	Date	Name	Place	Description	Category_Type	Accepted By	Status
1	2022-06-25	Bensy	chalakudy	urgent	CCTV Security	Rose	Finished

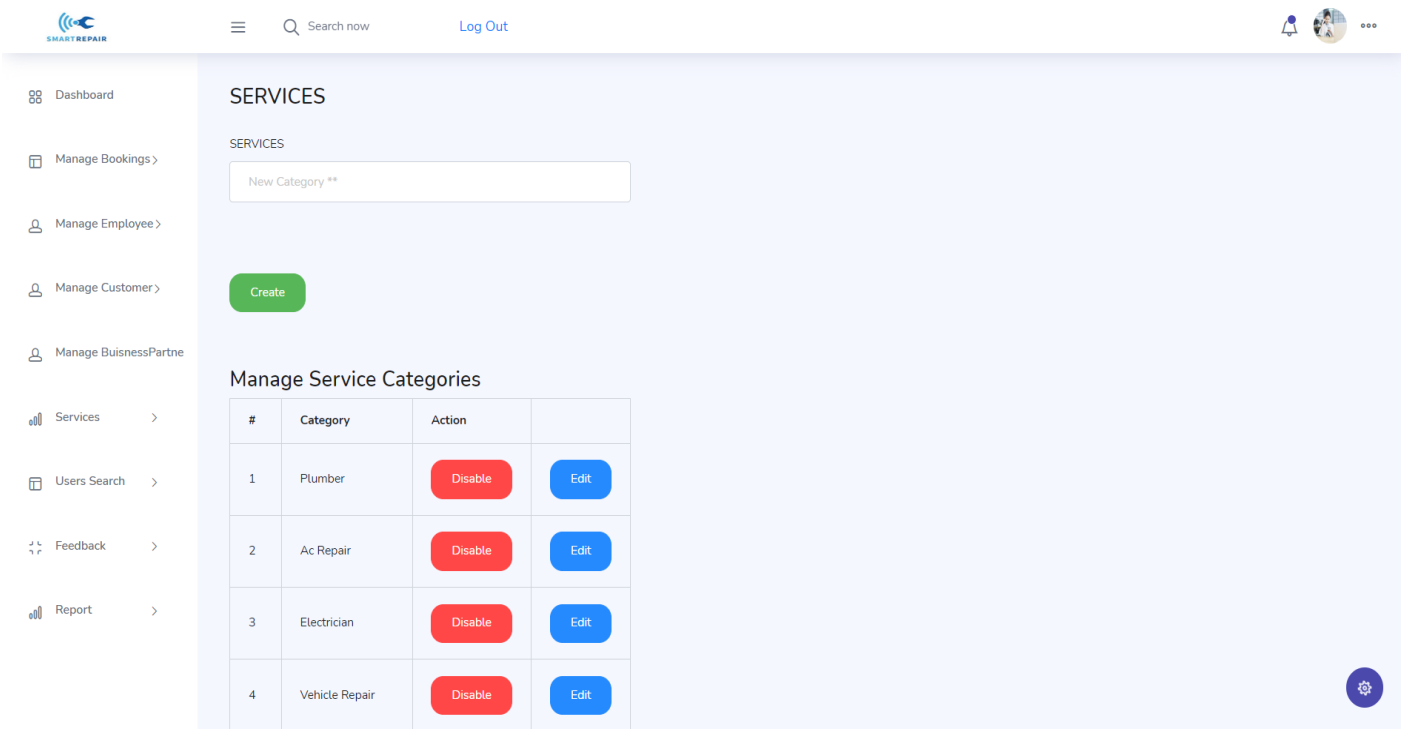
View Manage Employee



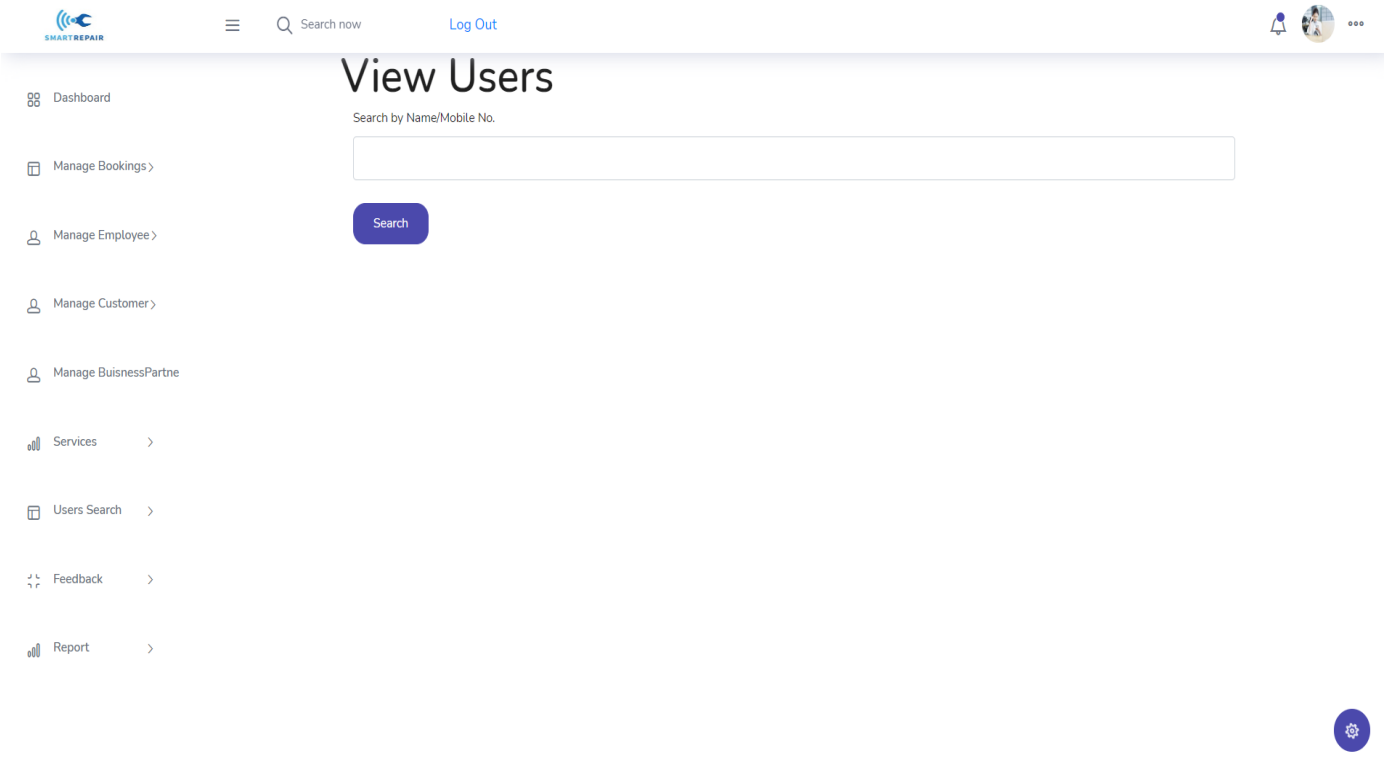
View Add New User



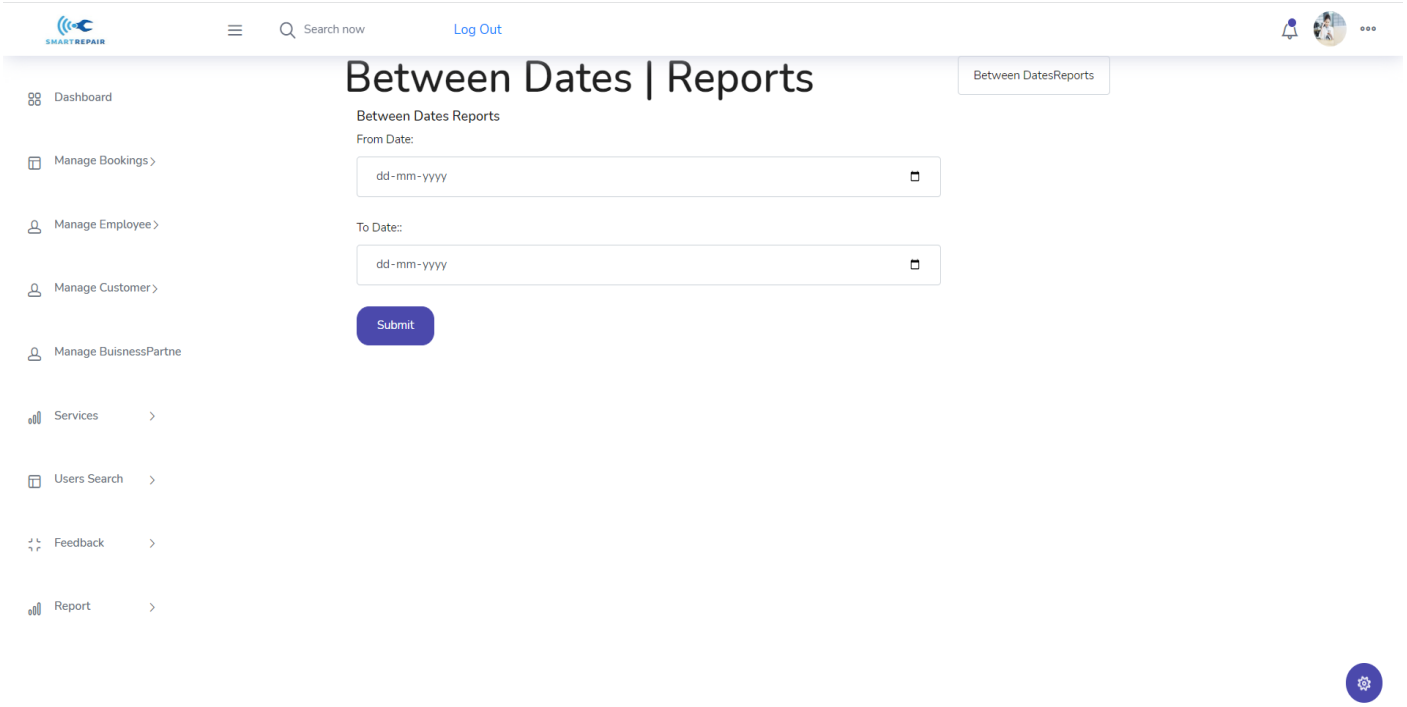
View Manage Services



View User Search



View Reports



9.3 Plagiarism Report

turnitin Similarity Report ID oid:10159:20020580

PAPER NAME	AUTHOR
smartreport (1).pdf	Bensy Benny

WORD COUNT 9023	CHARACTER COUNT 48164
Words	Characters
PAGE COUNT 53	FILE SIZE
Pages	1.2MB
SUBMISSION DATE	REPORT DATE
Jul 22, 2022 11:20 AM GMT+5:30	Jul 22, 2022 11:21 AM GMT+5:30

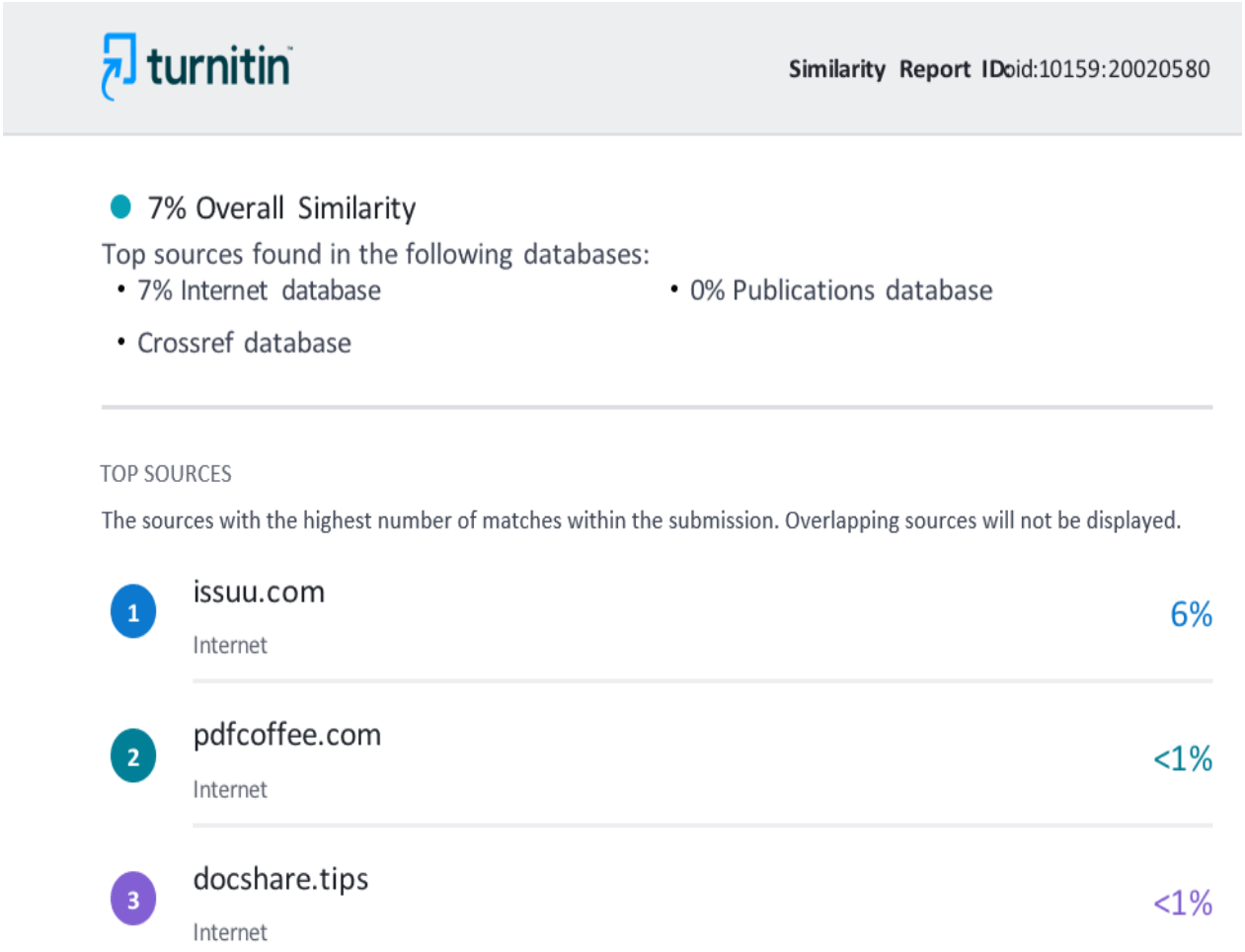
7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 0% Publications database
- Crossref database

Excluded from Similarity Report

- Small Matches (Less then 50 words)



turnitin

Similarity Report ID oid:10159:20020580

7% Overall Similarity

Top sources found in the following databases:

- 7% Internet database
- 0% Publications database
- Crossref database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	issuu.com Internet	6%
2	pdfcoffee.com Internet	<1%
3	docshare.tips Internet	<1%