**MINUTE MAINTENANCE**

**A PROJECT REPORT**

***Submitted by***

**PATEL DEV N.(194520307030)**

**PATEL NISARG R.(194520307031)**

**MEET .B. RATHOD(194520307035)**

**SHRIJI .M. NAKRANI(194520307039)**

***In partial fulfillment for the award of the degree***

***of***

**DIPLOMA ENGINEERING**

**in**

**COMPUTER ENGINEERING**

***Guided by***

**Prof. Nayanaba Gohil**

****

**Gyanmanjari Diploma Engineering College, Bhavnagar**

**Gujarat Technological University, Ahmedabad**

**September, 2021**

**GUJARAT TECHNOLOGICAL UNIVERSITY AHMEDABAD**

**GYANMANJARI DIPLOMA ENGINEERING COLLEGE**

**BHAVNAGAR**

****

Certificate

This is to certify that Mr./Ms. Patel Dev N. from **Gyanmanjari Diploma Engineering College, Bhavnagar** having Enrollment No: 194520307030 has completed **Project Report** having title MINUTE MAINTENANCE, in a group consisting of 4 persons under the guidance of the faculty guide Prof. Nayanaba Gohil .

Institute Guide-UDP Head of Department

**GUJARAT TECHNOLOGICAL UNIVERSITY AHMEDABAD**

**GYANMANJARI DIPLOMA ENGINEERING COLLEGE**

**BHAVNAGAR**

****

Certificate

This is to certify that Mr./Ms. Meet B Rathod from **Gyanmanjari Diploma Engineering College, Bhavnagar** having Enrollment No:194520307035 has completed **Project Report** having title MINUTE MAINTENANCE, individually/ in a group consisting of 4 persons under the guidance of the faculty guide Prof. Nayanaba Gohil .

Institute Guide-UDP Head of Department

**GUJARAT TECHNOLOGICAL UNIVERSITY AHMEDABAD**

**GYANMANJARI DIPLOMA ENGINEERING COLLEGE**

**BHAVNAGAR**

****

Certificate

This is to certify that Mr./Ms. Patel Nisarg R. from **Gyanmanjari Diploma Engineering College, Bhavnagar** having Enrollment No: 194520307031 has completed **Project Report** having title MINUTE MAINTENANCE, individually/ in a group consisting of 4 persons under the guidance of the faculty guide Prof. Nayanaba Gohil .

Institute Guide-UDP Head of Department

**GUJARAT TECHNOLOGICAL UNIVERSITY AHMEDABAD**

**GYANMANJARI DIPLOMA ENGINEERING COLLEGE**

**BHAVNAGAR**

****

Certificate

This is to certify that Mr./Ms. Shriji M Nakrani from **Gyanmanjari Diploma Engineering College, Bhavnagar** having Enrollment No:194520307039 has completed **Project Report** having title MINUTE MAINTENANCE, individually/ in a group consisting of 4 persons under the guidance of the faculty guide Prof. Nayanaba Gohil .

Institute Guide-UDP Head of Department

**STUDENT INFORMATION SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Patel Dev Nipulbhai | | |
| **Surname Name Father’s Name** | | |
| **Enrollment Number** | 194520307030 | | |
| **Contact Numbers** | **Mob:8200467191 / 9601541759** | **Landline:NA** | |
| **Email ID** | [devpatel4k2@gmail.com](mailto:devpatel4k2@gmail.com) / meetrathod178@gmail.com | | |
| **College Name** | Gyanmanjari Diploma Engineering College, Bhavnagar | | **College Code:** 452 |
| **Branch** | Computer Engineering | | **Semester :** V |
| **Student Team** | **Name** | | **Enrollment Number** |
| Patel Nisarg Riteshbhai | | 194520307031 |
| Rathod Meet Bharatbhai | | 194520307035 |
|  | Nakrani Shriji Manishbhai | | 194520307039 |
| **Student Signature** |  | |  |

**ACKNOWLEDGEMENT**

We, the group of Minute Maintenance project, would like to thank almighty God for giving us the strength and strong willpower to complete this project.

We would also like to thank our project guide, **Prof. Nayanaba Gohil** for giving us proper guidance and helping us to create a successful project. Last but not least we would like to thank our H.O.D **Prof. Amit Maru** and other faculty who supported us and gave us ideas for this project. We would also like to thank our mentor who helped us in every way he can in this project and gave his precious time to us.

Finally, We would like to thank our college, **GYANMANJARI DIPLOMA ENGINEERING COLLEGE** for providing us with the proper knowledge and skill to create this project.

**ABSTRACT**

Minute Maintenance is a web application designed primarily for house-based services. This system will allow workers and users a seamless experience and will provide the scope of business for the laborers and technicians.

In this system, a user can find a worker within a couple of minutes and he/she can have the work done within a couple of hours. With this system, our web application can also save both users and technicians time.

Users can also communicate with technicians with our chat box feature. As a result, their contact details and confidential information is safe and secure.

# **INDEX**

Index Pg No

1. Introduction……………………………………………………… ………………..…2

1.1 Project Introduction……………………………………………………..…...3

1.2 Purpose…………………………………………………………………..…..4

1.3 Scope…………………………………………………………………..…….5

2. System Requirement Analysis…………………………………… …………..……...6

2.1 Current System Study……………………………………………….………7

2.2 Weakness of Current System……………………………………………......7

2.3 Problem Identification / Definition………………………………….............8

2.4 Requirement of New System…………………………………………….......8

2.5 Feasibility Study………………………………………………………..........9

2.5.1 Technical……………………………………………………….......9

2.5.2 Economical……………………………………………….…..…...10

2.5.3 Operational………………………………………………...……...10

2.5.4 Schedule (Timeline Chart)……………………………..…..……...11

2.6 Development model (Software Process Model).............................................12

2.7 Requirement Validation……………… …………………………….............13

2.8 Tools and Technology / Minimum Hardware and Software

Requirements………………………………….……………………............13

2.9. System Architecture…………………………………………………...........14

2.10 Data Flow Diagram…………………………………………………….......15

2.11 Use –case Diagram……………………………………………………..…..18

2.12 Activity Diagram…………………………………………………......….…19

3. System Design……………………………………………………………………..….20

3.1 Database Design………………………………………………………..…....21

3.1.1 Data Dictionary………………………………………………...….22

3.1.2 Entity-Relationship Diagram……………………………….…...…23

3.2 GUI Design (Self-Created GUI Screen shot)..................................................24

4. Testing………………………………………………………………...……….…...…26

4.1 Testing Plan………………………………………………………….............27

4.2 Testing Strategy...............................................................................................28

4.3 Testing Methods…………………………………………………….……….29

4.4 Testing Cases……………………………………………………....………...30

5. Future Enhancement, Limitation and Conclusion………………………….…….…...39

5.1 Limitations……………………………………………………...…….……...40

5.2 Future Enhancement……………………………………………...…….…....40

6. Conclusion…………………………………………………………………….…...….41

6.1 Conclusion……………………………………………………………......….42

Appendix…………………………………………………………....................................43

Reference…………………………………………………………...................................44

**1**

**Chapter # 1: Introduction**

* 1. **Project introduction**
  2. **Purpose**
  3. **Scope**

## 

## **1.1 Project introduction:**

## Minute Maintenance is a website that will help users to provide laborer’s and technicians. This web application has such features which make it distinguished from other

service provider sites meanwhile some of those are :

* Workers get near you.
* Minimal user interface.
* Direct communication with workers with no third-party involvement.
* Variety of all home appliance services.
* Rating of workers so you get the best person who will solve your problem.

So by using this web application all the above features and lots more are at your fingertips. This website is not just a web application but it helps people. However, it is difficult for people to find a job, so it is a non-profitable web application so all workers, as well as technical, get an opportunity to work on this website.

Customer and workers will have a direct connection with each other and no third party involvement. They can also pay worker with UPI or cash. Many a times a worker cannot reach to the location at required time. By this we gave the worker to see the address of the customer and then they can accept or reject the job.

## 

## **1.2 Purpose:**

This project is associated with the current system where everyone can access web applications that can make a person’s problems solved easily. This project is associated with the goal of the solution to a particular problem.

* The customer can find their problem like fixing the AC or Refrigerator and will be able to find a suitable person for the job.
* The customer will get a nearby feature from which he/she will be able to find the mechanic or service provider.
* There will be an online mode of payment from which a customer can give the money from any UPI id or any other payment method.
* Customers can find their problem’s solution easily and can give a job to particular specialized technician and can have a job done.
* We saw technicians such as electricians, plumber etc. weren’t getting jobs and so we developed this website for their ease also.
* We have made a dedicated website for workers so that they can see the job easily and can accept and reject the job.

**1.3 Scope:**

In this project, a user will have to search for a particular service. If a user will use our web application then he/she will get a particular service with an experienced technician in a couple of minutes and work done in a couple of hours.

The workers will get a specific benefit while using our web application. They can get a job much faster. They can also fix a time of work and they will also get daily expenses which will be more than they usually earn. They can manage their jobs and can prevent the collision of two jobs. As a result, they can get more customers than they usually get.

We have also added some charts for the users to show their statistical analysis of their work which is divided in days, weeks and months. Each and every day a worker gets a job and if he accepts this job he will get a notification of the chart and the daily goal which he has set at the starting of the day. By this chart a worker can get the idea of how much work he has done throughout the day. We have also given a chart which is based on the salary of the worker. The more he does the job, the more money he will get.

**2**

**Chapter # 2: System Requirement Analysis**

**2.1 Current System Study**

**2.2 Weakness Of Current System**

**2.3 Problem Identification / Definition**

**2.4 Requirement of the new system**

**2.5 feasibility Study**

**2.5.1 Technical**

**2.5.2 Economical**

**2.5.3 Operational**

**2.5.4 Schedule(Timeline Chart)**

**2.6 Development Model(Software Process Model)**

**2.7 Requirement Validation**

**2.8 Tools And Technology/ Minimum Hardware and Software Requirement**

**2.9 System Architecture**

**2.10 Data Flow Diagram**

**2.11 Use Case Diagram**

**2.12 Activity diagram**

**2.1 Current System Study:**

Throughout the system analysis, an in-depth study of end-user information is conducted, for producing functional requirements of the proposed system.

Data about the existing services required are collected in a database through several fact-finding techniques such as website visits and document reviews, at the beginning of the stage. The data collected facilities information required during detailed analysis.

A study on the current system is performed based on the collected data. As a result, the user requirements of the proposed systems are determined. At the end of this stage, requirement specification is produced as deliverable. Many customer have a issue with time management and as a result a worker have to wait at the doorstep of the customer. Therefore, we have solved their problem by giving the phone no of the worker. It will only be displayed when a customer confirms the order.

**2.2 Weakness of Current System:**

The existing system happens to be a non-computerized operating system where all operations are done manually by the user carrying a phone and making a phone call to the technician. This system is a weakness from our side because while a user makes a phone call he/she doesn’t know the specific time of when the technician will come.

This leads to many kinds of mistakes such as timing management of technicians is not maintained. The users have to wait and have no guarantee of whether the technician will come or not. This could be embarrassing because the user could not take this lightly with the technician which may raise some misunderstandings.

**2.3 Problem Identification / Definition:**

When we have issues related to home appliances we generally find a technician’s number and sometimes it's difficult to get a proper technician in which we can trust. As a result we have to be dependent or rely on only one or two options we had interacted with in the past.

This leads to many kinds of mistakes such as conflict of timing, payment, accountability. As a result sometimes a customer is not satisfied with the work of a technician and can raise some problems and vice versa. This system is non technical and has no data saved in any records. Therefore, it was mandatory for us to create a system that can save records and can give our user a proper solution to his/her problem.

**2.4 Requirement of New System:**

* The new system is required to have a longer life span and have a secure environment for our users.
* If there is not any online web portal available like this then we face many problems like we cannot search in our nearby places, negotiate/call with them without going anywhere and sitting at home.
* We have to rely on somebody’s advice for selecting a random service/repairer without comparing/knowing what other customers are reviewing about them & any past experience.
* Our System is secure. We believe that a secure system makes a user believe that his data is secure in our servers.
* Respecting our users' suggestions and needs we have built a system that has a longer life span and can be used by our customers from anywhere around the globe.
* Customers' transactions are secure when he/she is doing any kind of transaction such as net banking, Net Banking, UPI ID payment and wallets such as Paytm wallet etc. He/She can also pay the amount by Cash with our Cash On Delivery system.

**2.5 Feasibility Study:**

It is the technique of confirmation of whether the errand justifies doing or not. Feasibility considerations are endeavored inside tight time restrictions and routinely completed the cycle in a formed and oral reasonableness report. It will help in making decisions, for instance, which programming to use, hardware blends etc.

**2.5.1 Technical Feasibility:**

* Technical feasibility is focused on the present technical resource available in the organization.
* It studies if the technical resources including the technical term are capable of converting the idea into a working system.
* It also evaluates the hardware and the software requirement of the proposed system.
* Inputs are carried out by the person who is in the project and implementation are then carried out.
* It also represents the technical aspects of the project and how a project is made.
* We can say that it is a analysis to study the project basically from technician’s angle. The main aspect to be considered under this study are technology of the project:

1. Size of project.
2. Strength of project

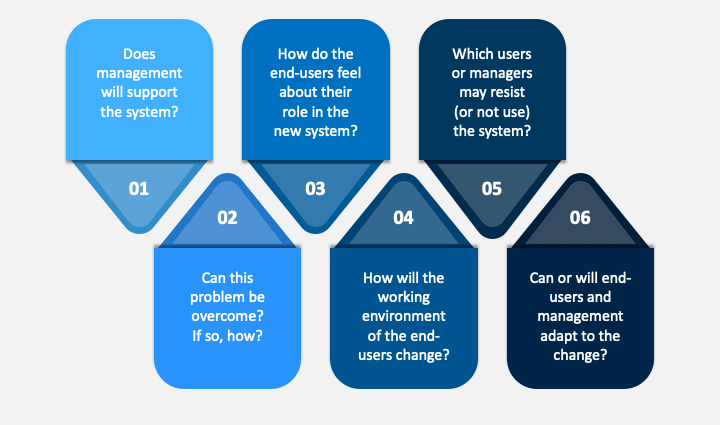
* A working system is created in a project by a person or a developer of that project or a team that is involved in a project by fist analyzing the Technical feasibility of the idea. In our system we have included this term because we have analyzed all the terms of the project.

**2.5.2 Economical Feasibility:**

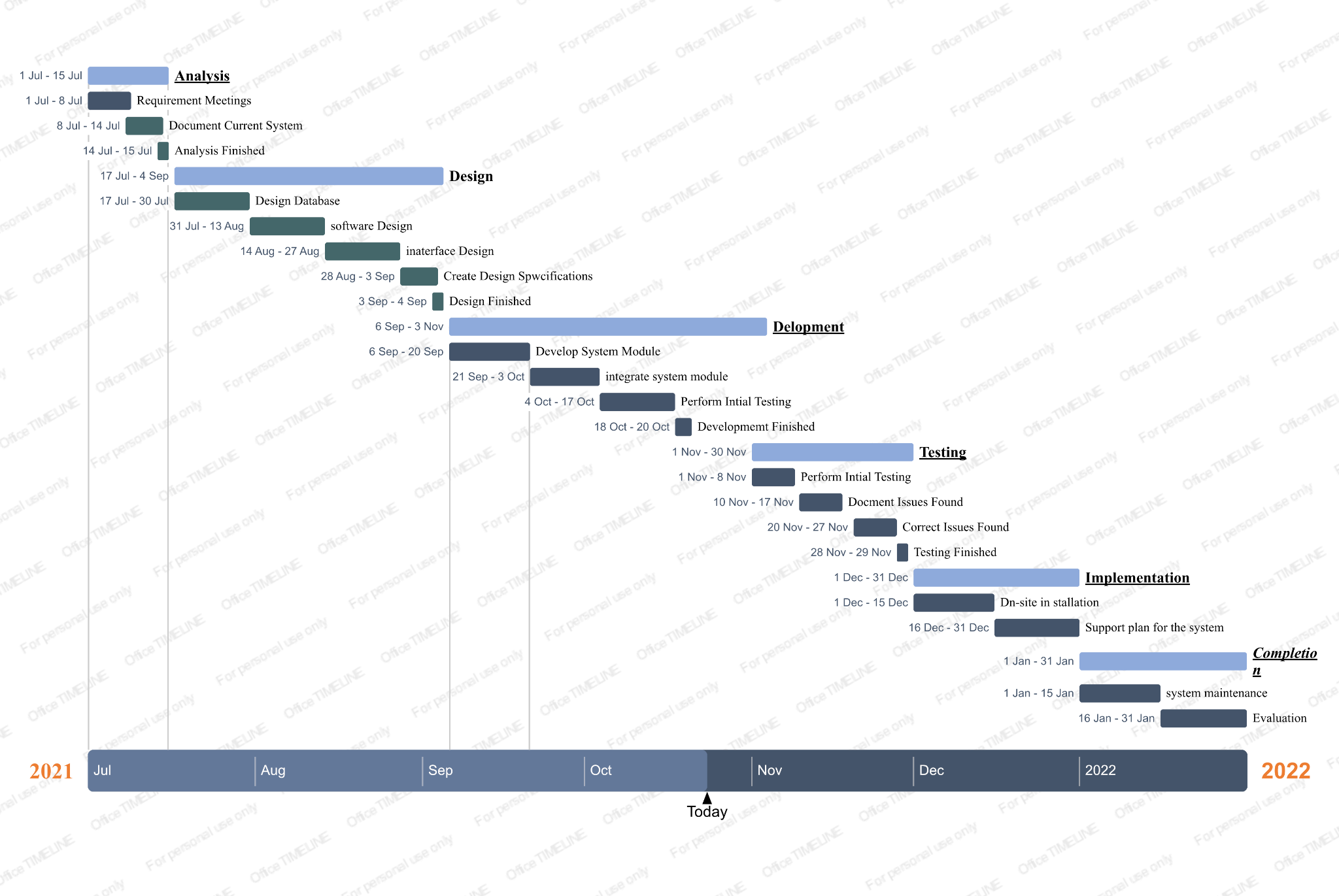
* Economical feasibility has a great importance in the making of applications.
* Here a developer has to keep a fixed budget in mind and hase to work accordingly.
* The developer has to keep one thing in mind that his application or software is cost-efficient and will be futuristic.
* This program has no update needed on the external part and is easy to set up .

**2.5.3 Operational Feasibility:**

* Operational feasibility is the system which will provide all the necessary information to the user.
* The cost of such a system will be low and have access to up-to-date data.
* Correct installation or better running of programs will give users a valid result.

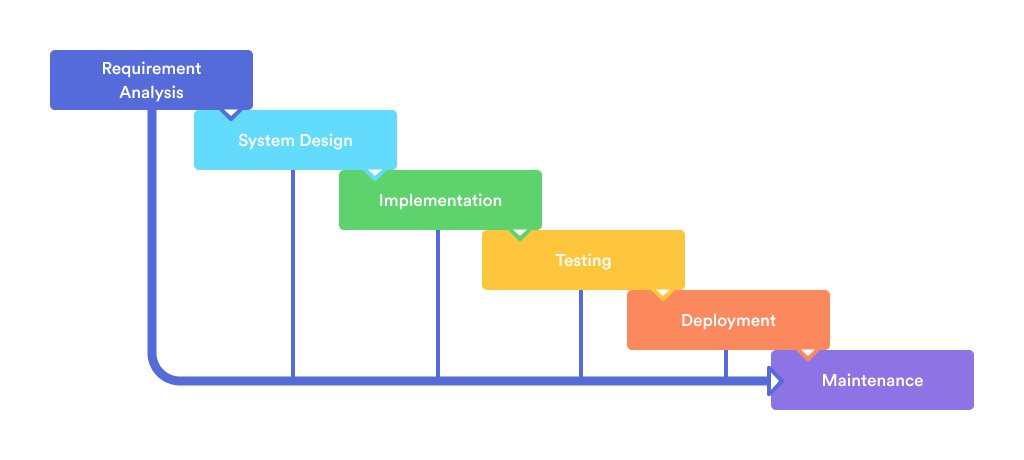


**2.5.4 Schedule (Time Chart):**



**Fig 2.1 (Time Chart)**

**2.6 Development Model(Software Process Model):**

****

**Fig 2.2 (Software Process Model)**

**Requirement Analysis:** It is based on what are the requirements to create a new system which has a longer life span and is secure.

**System Design:** After That comes the system design. Design is a core component of the system that makes a user use the system seamlessly.

**Implementation:** After a design is ready the implementation process starts and the implementation work is done by seeing the necessary system requirements.

**Testing:** Testing of the system is done by the company’s software engineer to see the faults and the bugs in the software.

**Deployment:** After the software is stable and has no bugs the software is deployed by the company on Platforms like Google Play Store, App Store etc.

**Maintenance:** After a stable release is deployed major software updates and security updates are given to the users.

**2.7 Requirement Validation:**

* Requirements validation is the process of checking that requirements define the system that the customer really wants.
* It overlaps with elicitation and analysis, as it is concerned with finding problems with the requirements.
* Requirements validation is critically important because errors in a requirements document can lead to extensive rework costs when these problems are discovered during development or after the system is in service.
* The cost of fixing a requirements problem by making a system change is usually much greater than repairing design or coding errors.
* A change to the requirements usually means that the system design and implementation must also be changed.

**2.8 Tools and Technology /**

**Minimum Hardware and Software Requirements:**

These are the minimum and maximum system requirements for our web application to run. The given below are recommended for a great user experience.

* **Supported Operating Systems:**Windows,Linux,Mac,Android.,IOS
* **Supported Web browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.
* **Version:** 32 bit As well as 64 bit.
* **Processor:** 1.5 GHz core 2 Duo/AMD Processor or more (Recommended).
* **RAM:** Minimum 1 GB (Recommended - 4 GB).
* **Hard Disk:** upto 1 GB available space.
* **Display:** 1366 x 768 (Recommended).

**Tools and Technology Used**

These are the major technologies used in the development of the web application.

**UI design:** HTML, CSS, Bootstrap

**Front-end:** JavaScript

**Back-end:** PHP

**Database:** MySQL

**Server:** Apache

**2.9 System Architecture:**

* The architecture is a member of the team that is responsible for designing and building a system
* The architecture contribution comes in the very early stages of the systems process
  + When the operational concept is defined
  + The basic structure of the system is conceptualized
* A system architecture, not only knows about the individual components, but also understands the interrelationships among the components
* Process of creating complex, reliable systems
* Building systems in today’s world is weak.
* Requirements of the marketplace are ill-defined
* Rapidly evolving technology provides new services at a global level instantly
* Uncertainty is increasing about the way the system will be used, the components that will be incorporated and the interconnections that will be made

**Components:** A basic approach to architecture is to separate work into components. These may be designed to be reusable. Components also serve to reduce extremely complex problems into small manageable problems. The difference between a costly, unstable, low performance system and a fast, cheap and reliable system often comes down to how well it has been architecture into components.

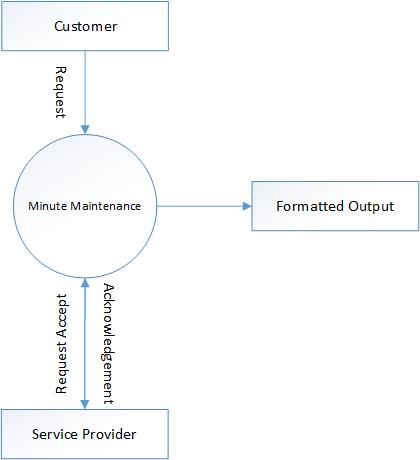
**Layers:** It is common to separate components into layers. Components in different layers are loosely coupled such that they hide their implementation behind an interface. This allows for complexity reduction and can reduce the cost of the future changes.

**Services:** A service is a piece of functionality that can be separately deployed and managed. Services are loosely coupled such that you can rework a service without impacting the rest of your architecture.

**Deployment:** As services are separately deployed, they allow for extreme scalability and reliability. Services can also cut your computing cost as they allow large systems to be deployed to many instances of inexpensive hardware.

**2.10 Data Flow Diagram:**

**LEVEL 0**

****

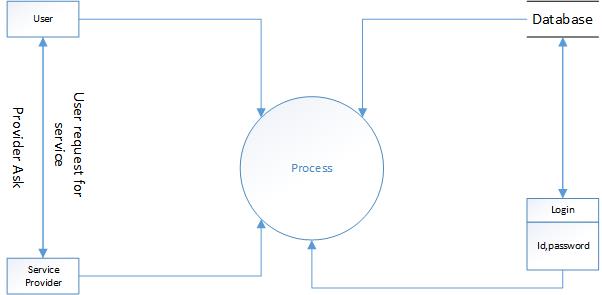
**Fig 2.3 (DFD LEVEL 0)**

* 0 level DFD is also called fundamental system model or a context model

.

* It represents the entire software element as a single process with input or output data.
* In this diagram we used each and every component of the current required system and made a easy to understand system.

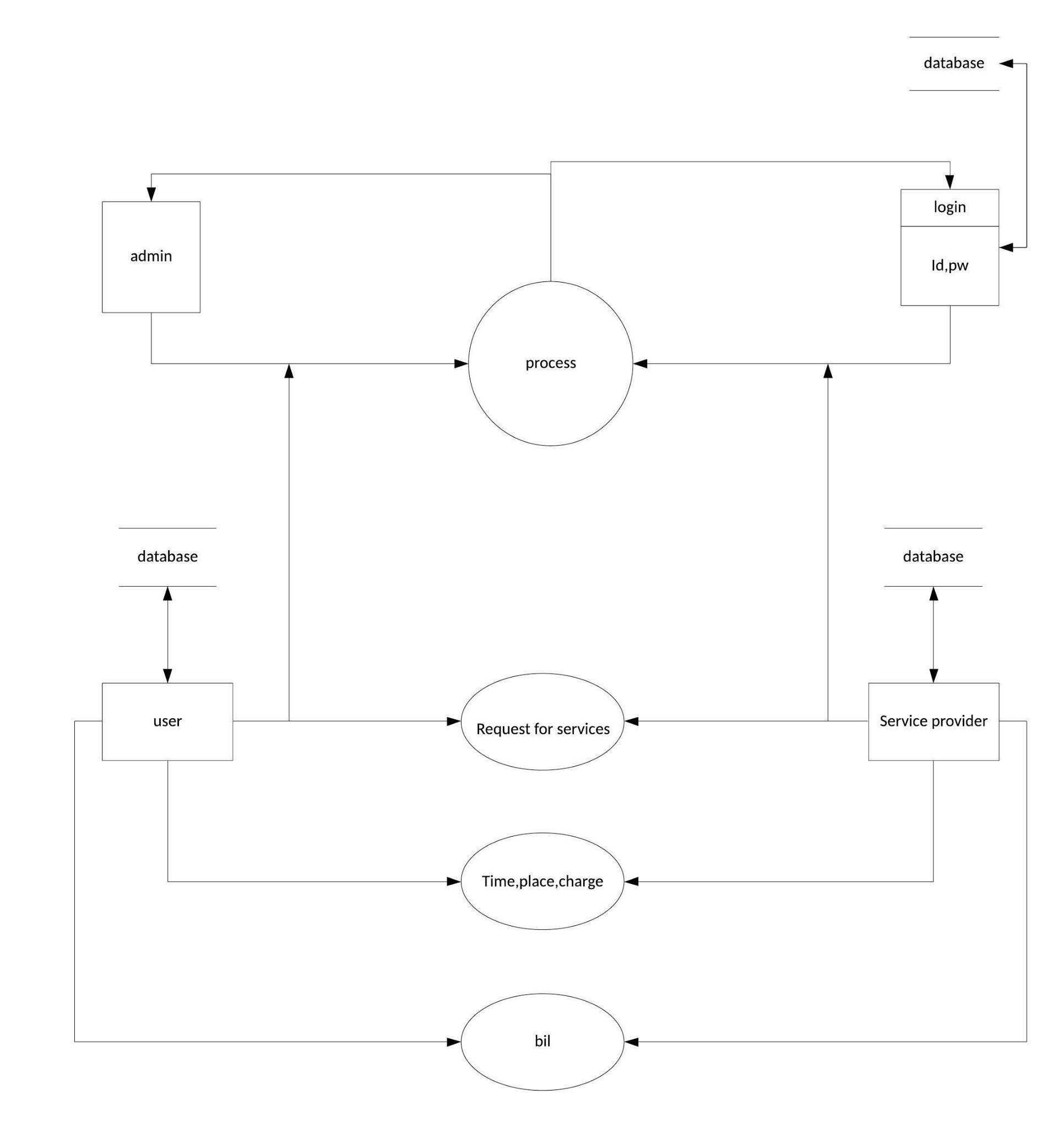
**LEVEL 1**

****

**Fig 2.4(DFD LEVEL 1)**

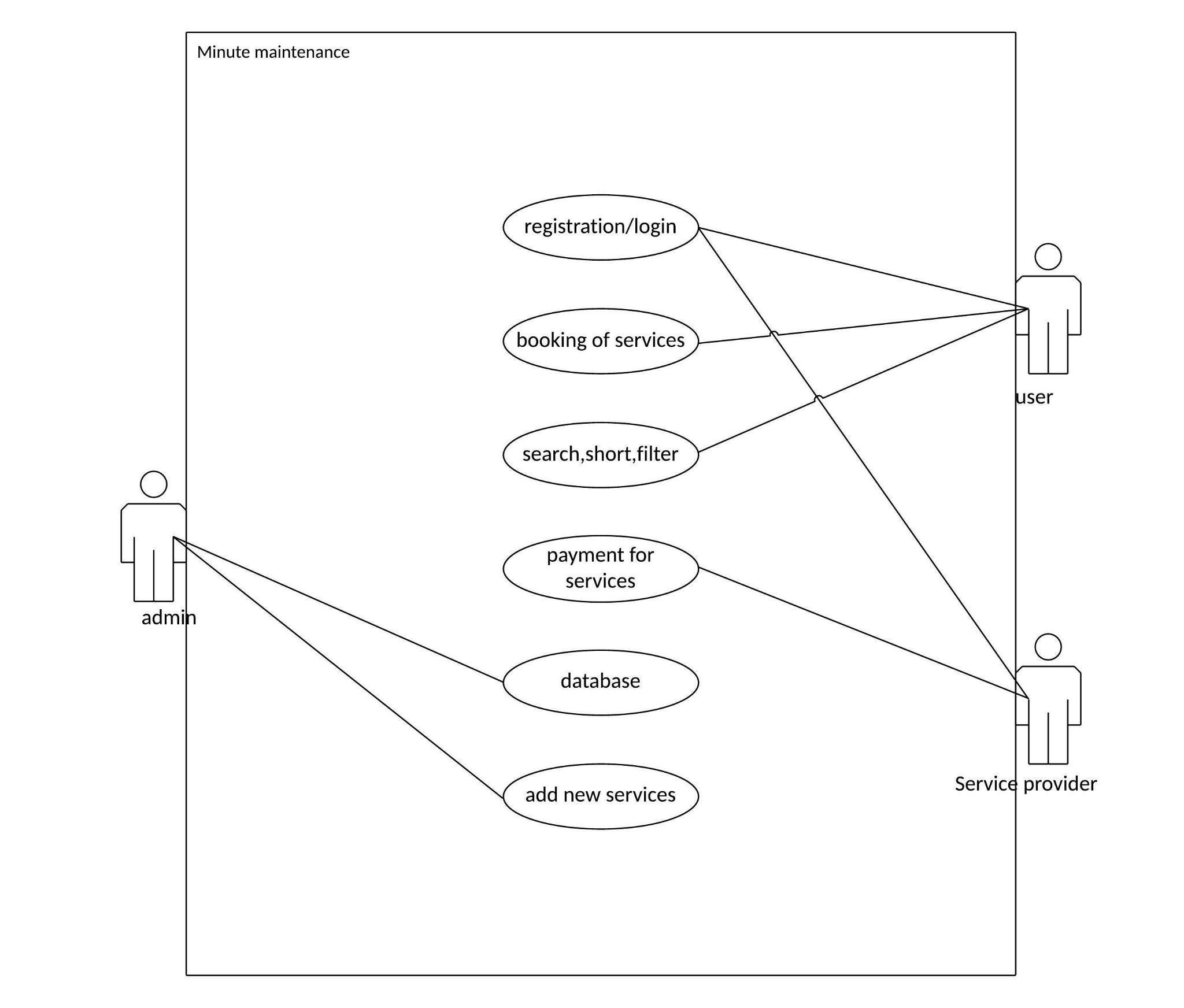
* The level 1 DFD we construct is a more refined version of the context diagram.
* It covers the entire system, all the main processes are shown in the above figure.
* While creating this diagram we were aware of balancing each and every component which is used by us in this current system.

**LEVEL 2**

****

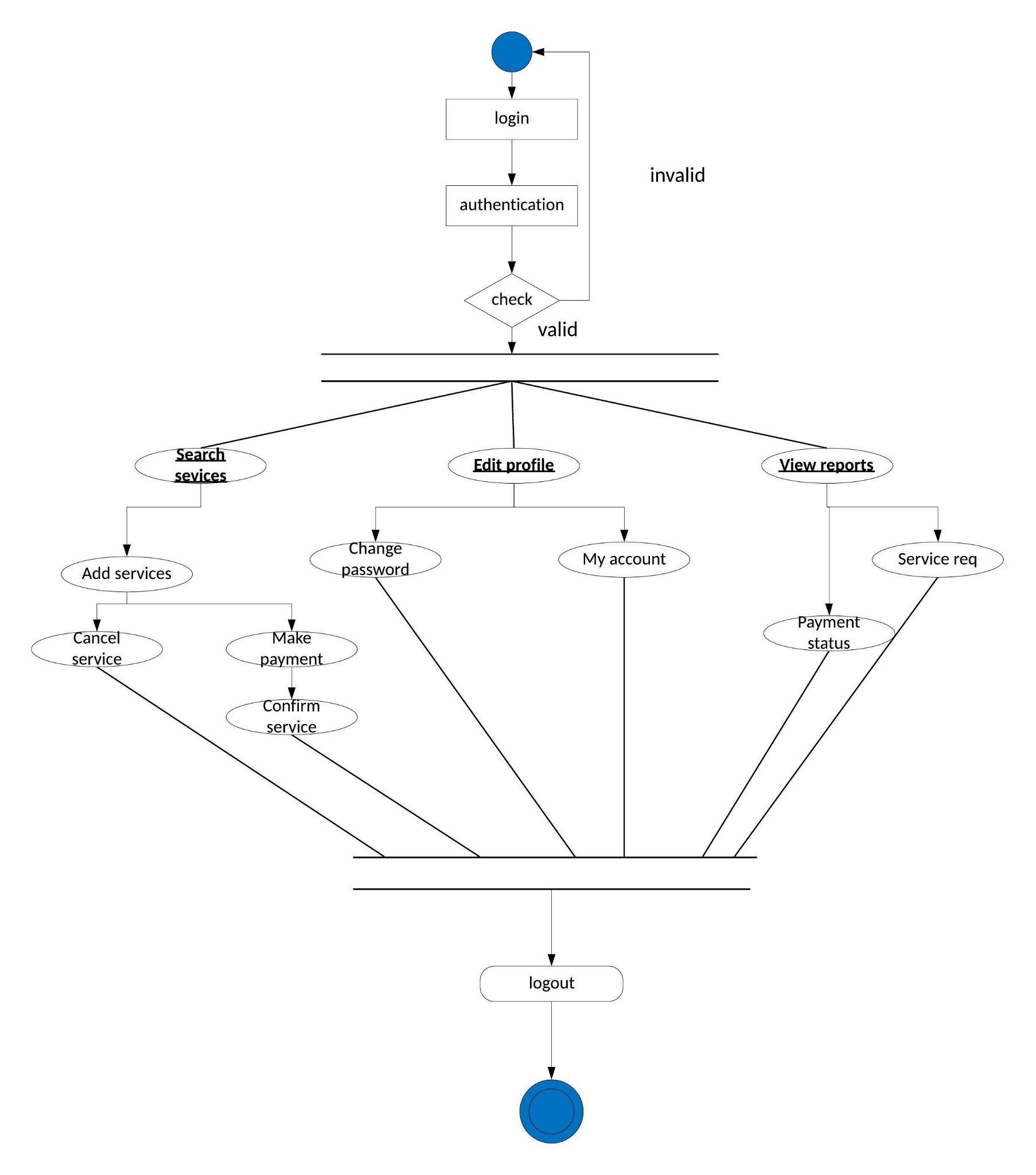
**Fig 2.5 (DFD LEVEL 2)**

* In this, the process which are needed to be expanded are shown.
* We have described all the process of DFD Level 1 in this Level 2 diagram.

**2.11 Use-case Diagram**

**Fig 2.6 ( Use-case Diagram)**

**2.12 Activity Diagram**



**Fig 2.7 ( Activity Diagram)**

**3**

**Chapter # 3: System Design**

**3.1 Database Design**

**3.1.1 Data Dictionary**

**3.1.2 Entity-Relationship Diagram**

**3.2 GUI Design (Self-Created GUI Screen shot)**

**3.1 Database Design:**

**Database**: The name of our database is **Login**. All the related data of our users are stored in that particular database. Any information such as email, id etc is stored in that database and a user does not need to make a new account as long as his/her database is stored in our servers.

**Tables**: We have named our table as **Users** to make sure that our backend developers do not need to find the database. Here all the login information of the user is stored and is encrypted by our backend developers.

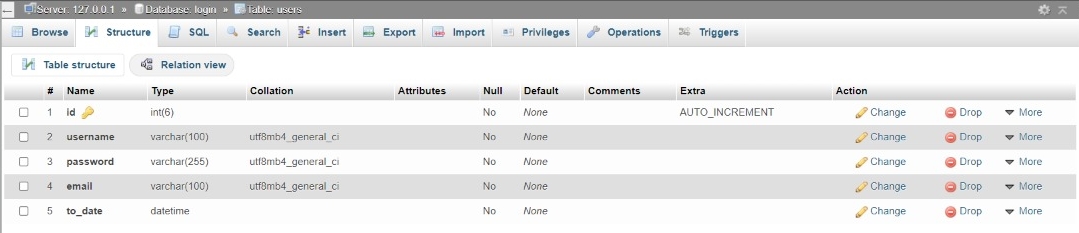
**Database** **Fields**: There are mainly 4 important fields in our table.

1. **Email:** email of the user which he/she gives when he/she makes an account.
2. **Id:** A unique id of the user to identify him/her from other users differently.
3. **Username:** A username to identify a specific user from millions of users. Username is generally given to a user to prevent a clash of the same name in the web application.
4. **Password:** Password is stored in our database and is encrypted by us to respect our user’s privacy.

**3.1.1 Data dictionary**

* A data dictionary is a data structure that stores metadata, i.e., (structured) data about data. The software package for a stand-alone data dictionary or data repository may interact with the software modules of the DBMS, but it is mainly used by the designers, users and the definitions of all schema objects in the database
* How much space has been allocated for, and is currently used by, the schema objects
* Default values for column
* Integrity constraint information
* The names of Oracle users Privileges and roles each user has been granted.

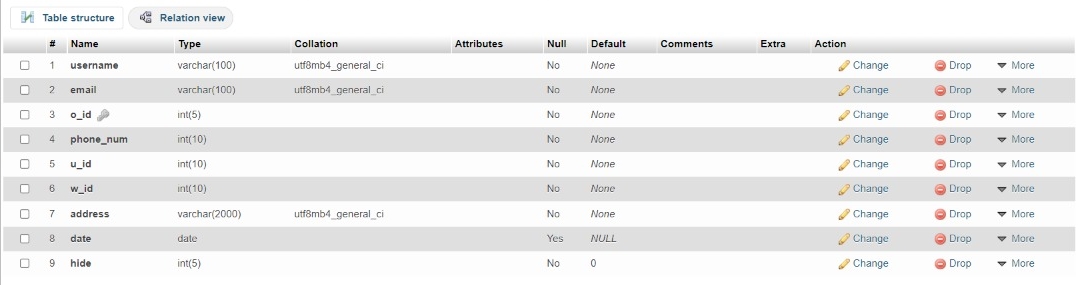
* Auditing information, such as who has accessed or updated various schema objects
* Other general database information Administrators of a computer system for information resource management.



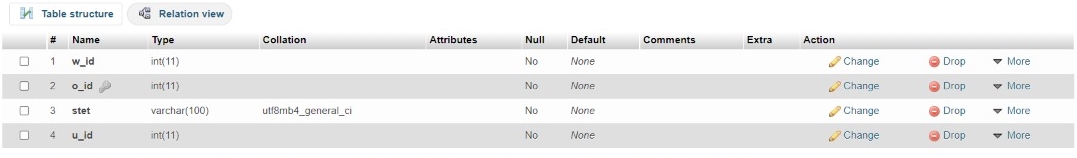
**Fig 3.1 (Name of the users stored in a separate Database with data types)**



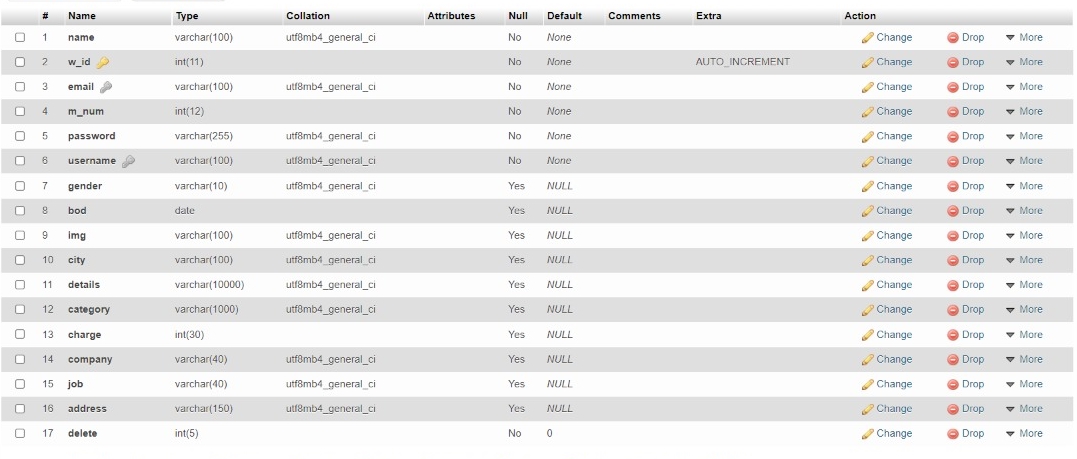
**Fig 3.2 (Name of the users stored in a separate Database)**

****

**Fig 3.3 (Name of the users stored in a the order table)**

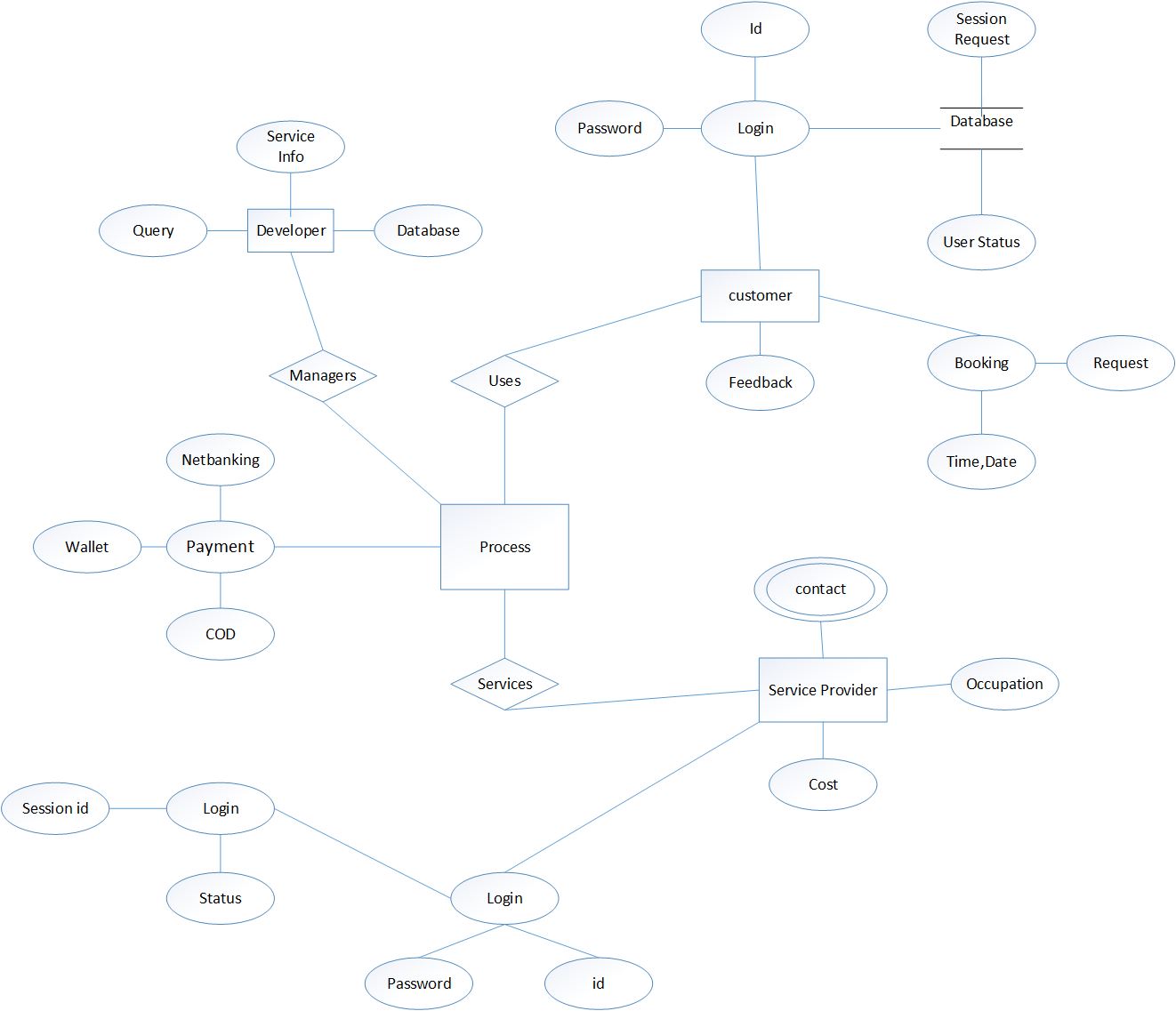
****

**Fig 3.4 (Name of the accepted order in a the accept table)**

****

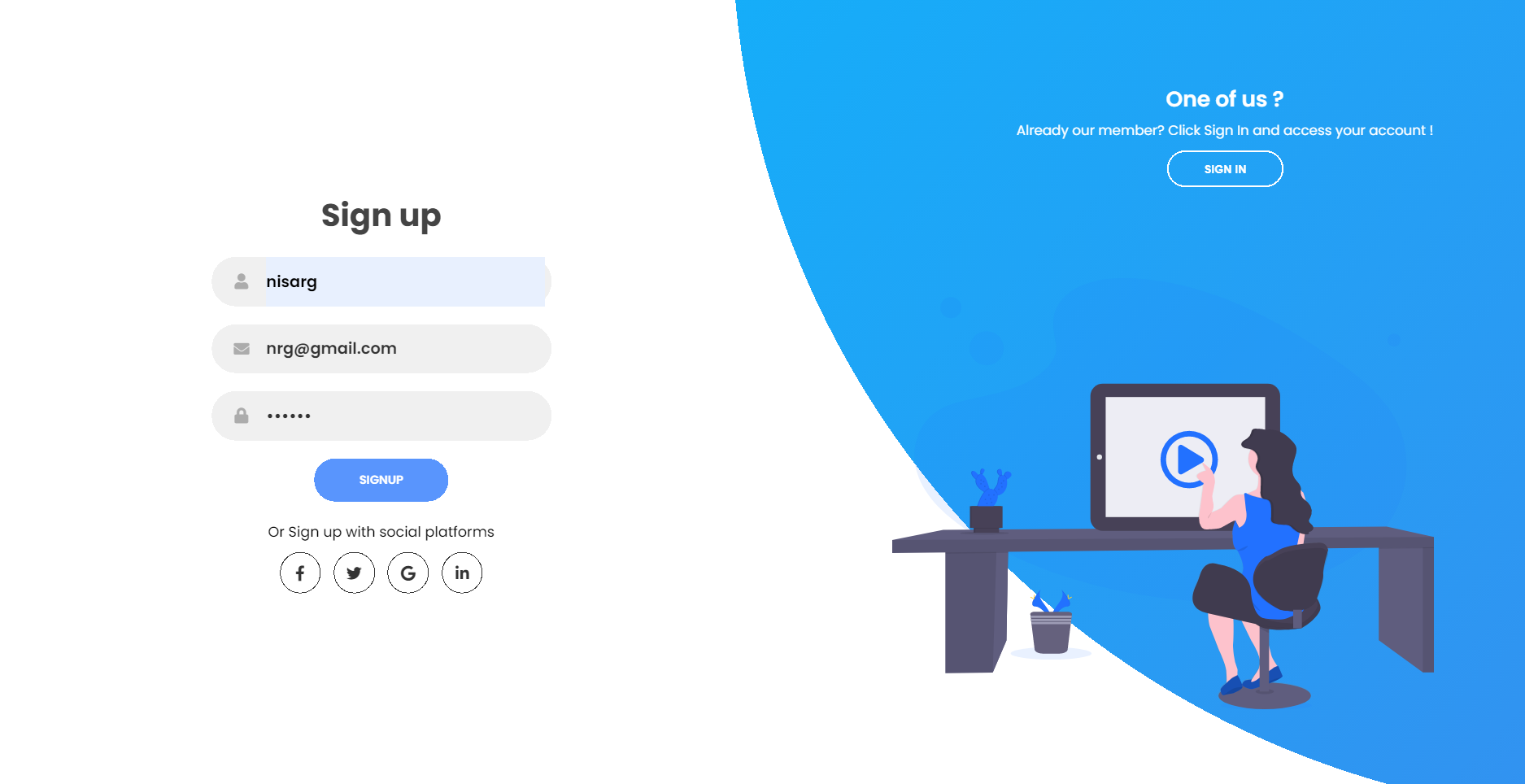
**Fig 3.5 (details of the worker when he creates the account)**

**3.1.2 Entity-Relationship Diagram**

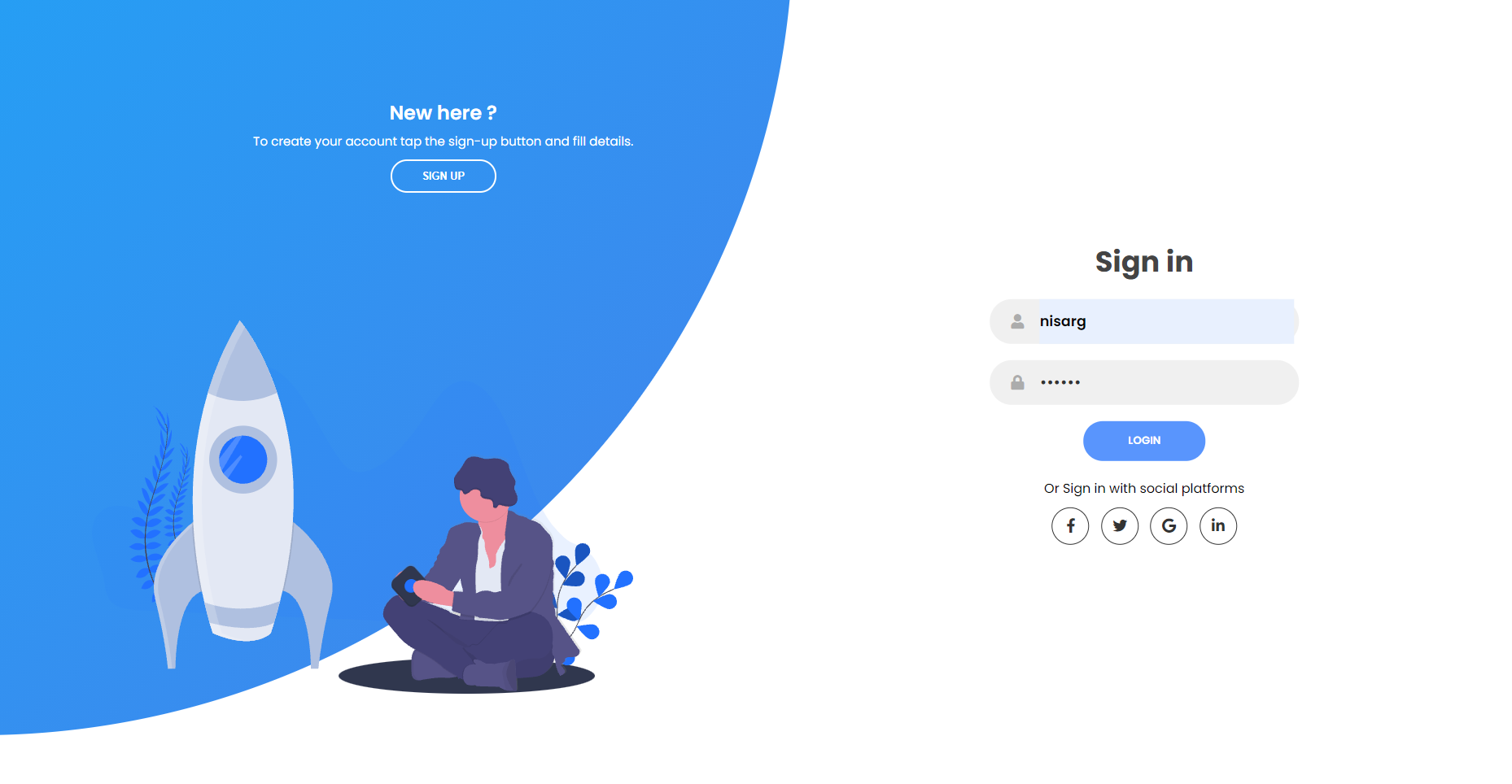
****

**Fig 3.6(Entity-Relationship Diagram)**

**3.2 GUI Design (Self Created GUI Screen shot)**

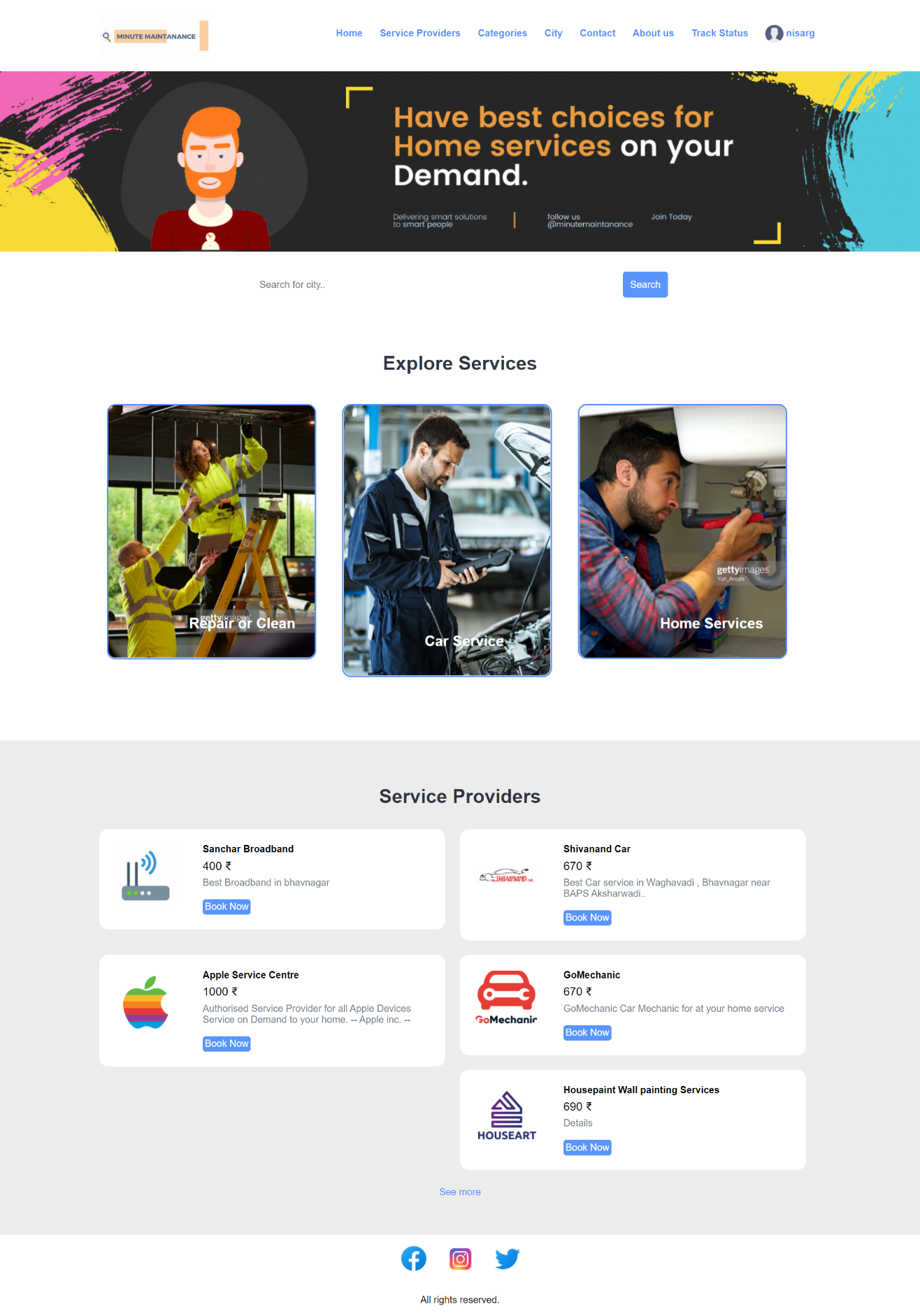
* **Sign up**

**Fig 3.7(Sign up)**

* **Log in**

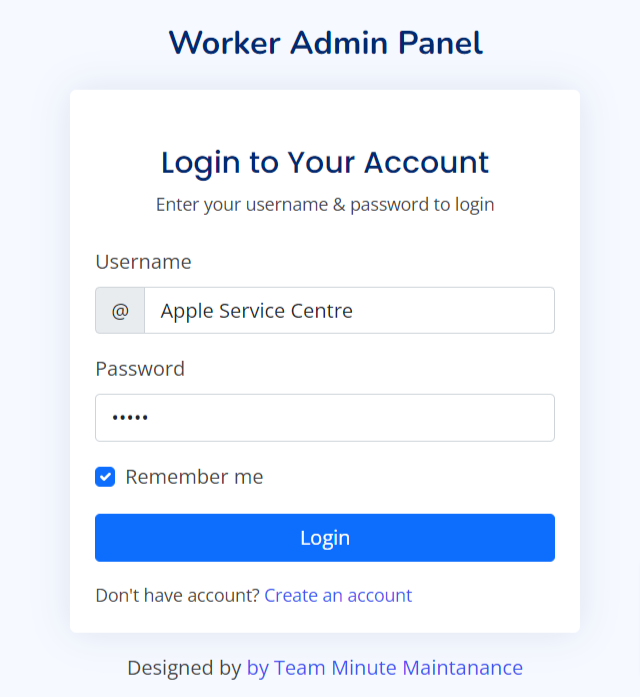
**Fig 3.8(Log in)**

* **Home Page**

****

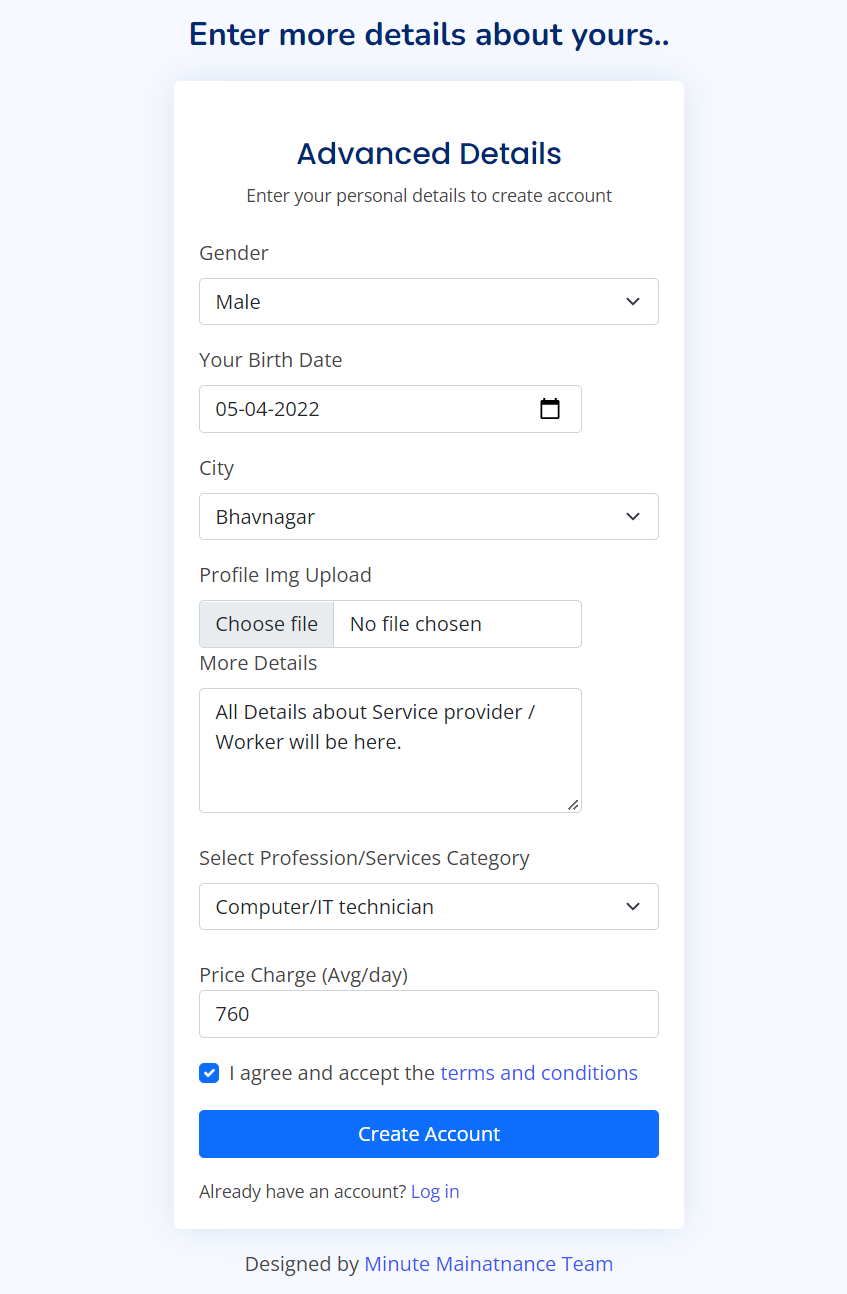
**Fig 3.9( Home Page)**

* **Log in Page For Workers:**

****

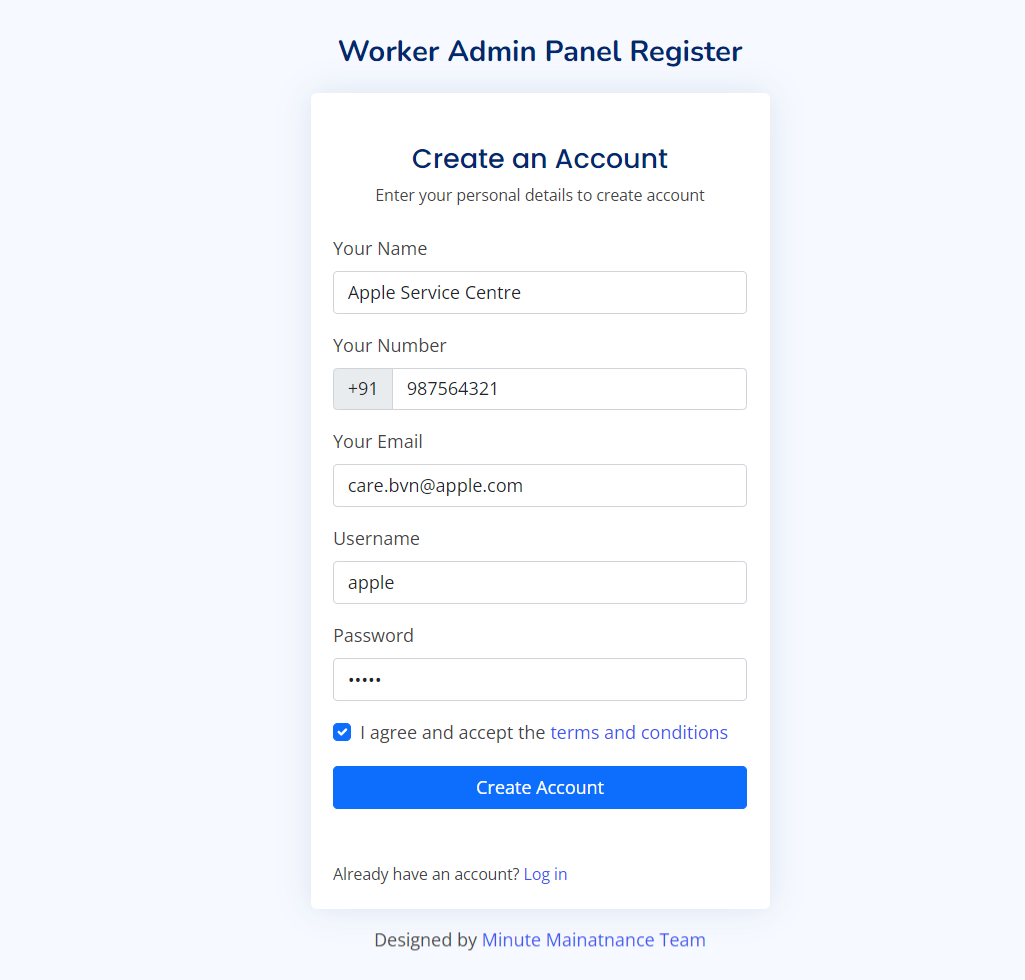
**Fig 3.10(Log in Page For Workers)**

* **Additional Details Page for Workers**

****

**Fig 3.11(Additional Details Page for Workers)**

* **Worker Side Signup page:**

****

**Fig 3.12(Sign Up Page for Worker side)**

**Home Page For Workers:**

**Fig 3.13(Home Page For Workers)**

**4**

**Chapter # 4: TESTING**

**4.1 Testing Plan**

**4.2 Testing Strategy**

**4.3 Testing Methods**

**Testing**

* Testing presents an interesting anomaly for the software engineer. During Earlier software engineering the engineer attempted to build software from an abstract concept to a tangible product.
* Now comes testing. The engineer creates a series of test cases that are intended to “demolish” the software that has been built.
* In fact, testing is the one step in the software process that could be viewed as destructive rather than constructive.

**4.1 Testing Plan**

Testing Plan Iterative performance Testing Activities:

1. Understand Project Vision and Context.

2. Identify Reasons for Testing Performance.

3. Identify Value of Testing performance.

4. Configure Test Environment.

5. Identify and coordinate tasks.

6. Execute Tasks.

7. Analyze Result and Report.

8. Reprioritize tasks.

**4.2 Testing Strategy**

* System testing is a critical process that takes as much as 50% of the system development time. The common view of testing held by users is that it is performed to prove that there are no errors in a program.
* Testing is too important for the success of the system. If all parts of the system are correct, the goal will be achieved successfully.
* System testing is also called the “put it all together” phase where the elements of the system are put tighter to examine its validity and reliability. Hardware, Software, Manpower and live data are combined in an effort to produce the necessary result. The variety of tests are conducted as,

o Output testing.

o Hardware and Software testing.

o Recovery and Security testing.

**The test strategy will include five different types of testing as described below:**

**Logical Testing**

* This is used to test every aspect of each form and query as soon as it is implemented. In this test results are compared with expected results.

**Functional Testing**

* Each menu item should be tested in the form to ensure that no functionalities have been missed out.

**System Testing**

* When the system is complete, the whole range of tests should be carried out again to ensure that no errors have been introduced.

**Recovering Testing**

* The computer will be re-booted while the database is open to ensure that data is not lost or corrupted in the event of power failure.

**4.3 Testing Methods**

1. White Box Testing.

2. Black Box Testing.

**White Box Testing**

* White box testing, sometimes called glass box testing, is a test case design method.
* That uses the control structure of the procedural design to derive to test cases, Using.
* White box testing methods, the software engineer can derive test cases that

1. Guarantee that all independent paths within a module have been exercised.

2. Exercise all logical decisions on their true and false sides.

3. Execute all loops at their boundaries and within their operational bounds, and.

4. Exercise internal data structures to ensure their validity.

**Black Box Testing**

* Black box testing, also called behavior testing, focuses on the functional requirement of the software.
* That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.
* Black box testing is not an alternative to white box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white box methods.
* Black box testing attempts to find errors in the following categories:

1. Incorrect or missing function,

2. Interface errors,

3. Errors in data structures or external database access

4. Behavior or performance errors, and

5. Initialization and termination errors.

## **Test Cases:**

#### **Test Case 1: Login**

Actor: User

Precondition: Username & password is available in the database.

Main Scenario:

1. User starts the application.
2. The splash screen is displayed on the monitor.
3. After that the login screen appears asking for Username & Password.
4. User logs in to the system the next page is available for further processes.

Alternate Scenario:

1. Either username or password is invalid or empty.
2. User may not enter into the system.

#### **Test Case 1.1: Login (Workers)**

Actor: Workers

Precondition: Username & password is available in the database.

Main Scenario:

1. Workers starts the web page.
2. The splash screen is displayed on the monitor.
3. After that the login screen appears asking for Username & Password.
4. User logs in to the system the next page is available for further processes.

Alternate Scenario:

1. Either username or password is invalid or empty.
2. User may not enter into the system.

#### **Test Case 2: Creating a New User Account.**

Actor: User Site.

Precondition: Users have to make an account on the basics of this the login in scenario will be available.

Main Scenario:

1. User clicks on the link showing “Sign Up”.
2. User enters the new Username & Password.
3. The new user is created. The user can now use the system or website.

#### **Test Case 2.1: Creating a New Worker Account.**

Actor: Worker Site.

Precondition: Worker have to make an account on the basics of this the login in scenario will be available.

Main Scenario:

1. User clicks on the link showing “Sign Up”.
2. User enters the new Username & Password.
3. The new user is created. The user can now use the system or website.

#### **Test Case 3: Delete a user**

Actor: User and Worker both.

Precondition: User and worker data has to be present in the database.

Main Scenario:

1. Admin can view the users of the system.
2. Admin can delete a user by clicking the Delete button.
3. That user cannot access the system now.

#### **Test Case 4: Booking for Services**

Actor: User only

Precondition: User data has to be present in the database.

Main Scenario:

1. User can book the required services.
2. Confirm of booking is the next step done by the user.
3. Order id is given to the user by us.

#### **Test Case 3: Accepting services by the Worker.**

Actor: Worker only.

Precondition: worker data has to be present in the database.

Main Scenario:

1. Worker can accept the users service request from his own database.
2. Worker can confirm the location and time of the user.

**5**

**Chapter # 5: Future Enhancement, Limitation and Conclusion**

**5.1 Limitation**

**5.2 Future Enhancement**

**5.1 Limitations:**

* It is easily scalable at National level but it is hard at International Level.
* If the selected service is not present or the worker has not selected the service then the user will not be able to see that particular service at that point of time.

**5.2 Future Enhancement:**

The project has a very vast scope in future. The project can be implemented on internet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

**6**

**Chapter # 6: Conclusion**

**6.1 Conclusion**

**6.1 Conclusion:**

Our website Minute Maintenance give a user an option to fix his/her problem from our site and as a result he/she can book a service from a single web site. We have also created a dedicated Workers website from which they can get in touch with the customers and get their job done in hours.

**Appendix**

**List Of Figures:** **Page.NO:**

**Fig 2.1 (Time Chart)**  l1

**Fig 2.2 (Software Process Model)**  12

**Fig 2.3 (DFD LEVEL 0)**  16

**Fig 2.4(DFD LEVEL 1)**  17

**Fig 2.5(DFD LEVEL 2)**  18

**Fig 2.6 ( Use-case Diagram)** 19

**Fig 2.7 ( Activity Diagram)** 20

**Fig 3.1 (Name of the users stored in a separate Database as Data types)** 23

**Fig 3.2 (Name of the users stored in a separate Database)** 24

**Fig 3.3 (Name of the users stored in a the order table)** 24

**Fig 3.4 (Name of the accepted order in a the accept table)** 24

**Fig 3.5 (details of the worker when he creates the account)** 24

**Fig 3.6 (Entity-Relationship Diagram)**  25

**Fig 3.7(Sign UP)** 26

**Fig 3.8(Sign IN)** 26

**Fig 3.9( Home Page)** 27

**Fig 3.10(Sign Up Page for Worker side)** 29

**Fig 3.11(Log in Page For Workers)** 28

**Fig 3.12(Sign Up Page for Worker side)** 30 **3.13(Home Page For Workers)** 31

**Reference**