**04. System Design**

**4.1 System Architecture**

**4.1.1 Class Diagram**

**4.2 Logical Design**

**4.2.1 ER Diagram**

**4.2.2 Data Flow Diagram**

**4.3 Physical Design**

**4.4 Interfaces/GUI**

**5.** [**System Implementation**](#_Toc63077680)

[5.1 Project Architecture](#_Toc63077681)

[5.1.1 Cake Design Customization](#_Toc63077682)

[5.1.2 Customer Registration Page](#_Toc63077683)

[5.1.3 Customer Login Page](#_Toc63077684)

[5.1.4 User Interface (GUI)](#_Toc63077685)

[5.1.5 Src Image Section](#_Toc63077686)

[5.1.6 Order Section](#_Toc63077687)

5.1.7 Logo

[5.1.8 Admin Panel](#_Toc63077689)

[5.2 Coding Section](#_Toc63077690)

[5.2.1 User Panel](#_Toc63077691)

[5.2.2 Admin Panel](#_Toc63077692)

**6.** [**System Testing**](#_Toc63077694)

[6.1 Test Plan](#_Toc63077695)

[6.2 Resources](#_Toc63077696)

[6.2.1 Human Resources](#_Toc63077697)

[6.2.2 Testing Environment](#_Toc63077698)

[6.2.3 Testing Tools](#_Toc63077699)

[6.3 Test Schedule](#_Toc63077700)

[6.4 Testing Method](#_Toc63077701)

[6.4.1 Black Box Testing](#_Toc63077702)

[6.5 Level Of Testing](#_Toc63077703)

[6.5.1 Unit Testing](#_Toc63077704)

[6.5.2 Integration Testing](#_Toc63077705)

[6.5.3 System Testing](#_Toc63077706)

[6.5.4 Gantt Table](#_Toc63077707)

[6.6 Test Cases](#_Toc63077708)

**7.** [**Conclusion**](#_Toc63077710) **and future work**

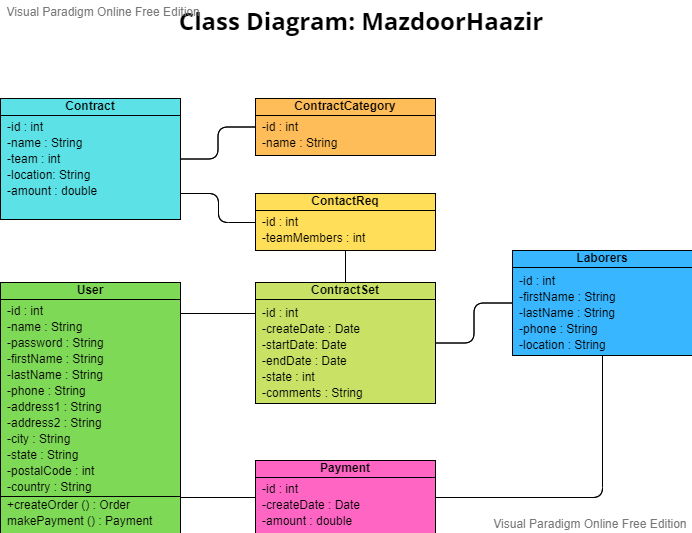
[7.1 Conclusion](#_Toc63077711)

[7.2 Future Work](#_Toc63077712)

[References](#_Toc63077713)

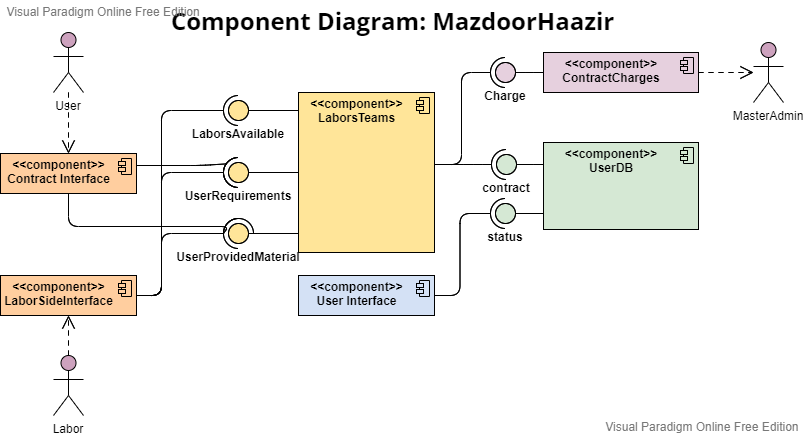
**Class Diagram**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.



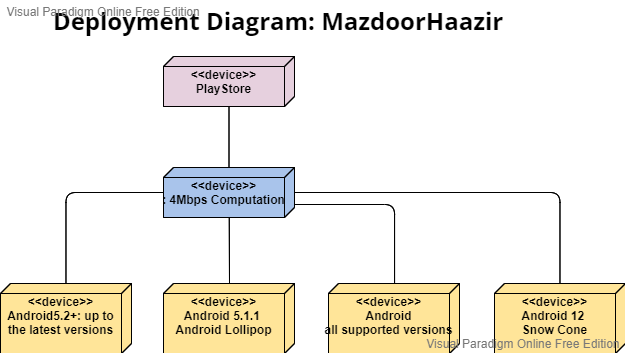
**Component Diagram**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.



**Deployment Diagram**

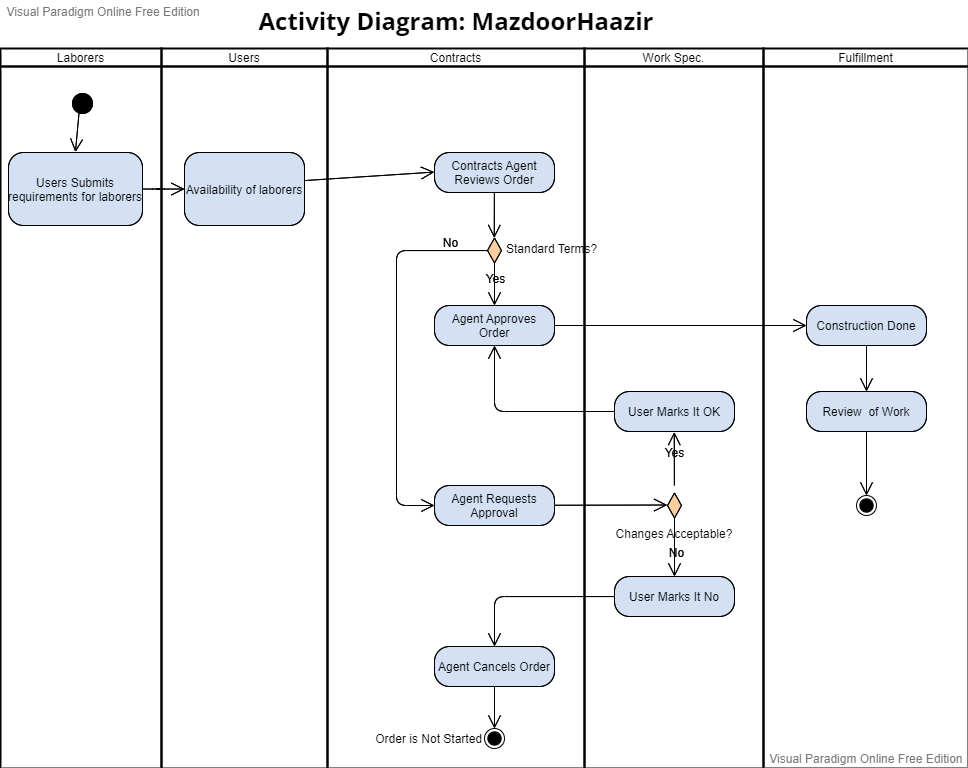
This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.



**Use Case Diagram**

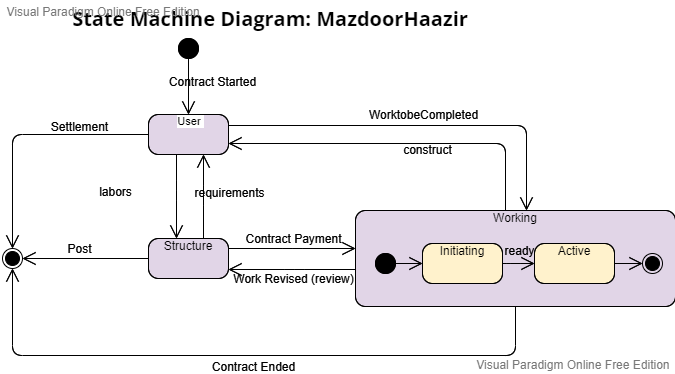
This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.

**Activity Diagram**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.

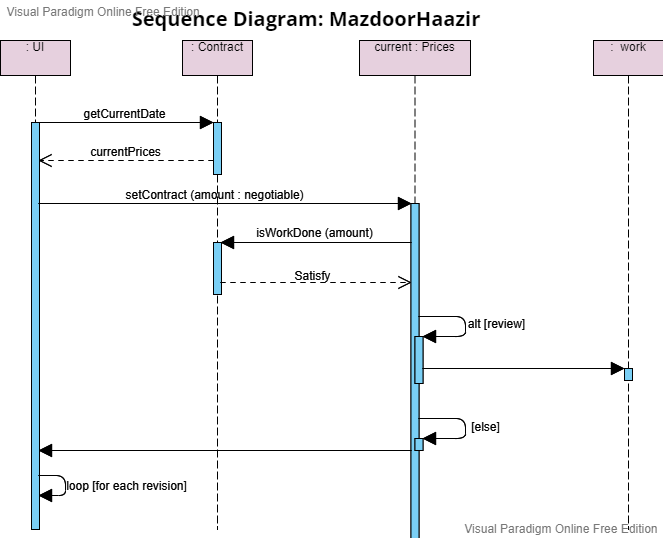
**State Machine Diagram**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.



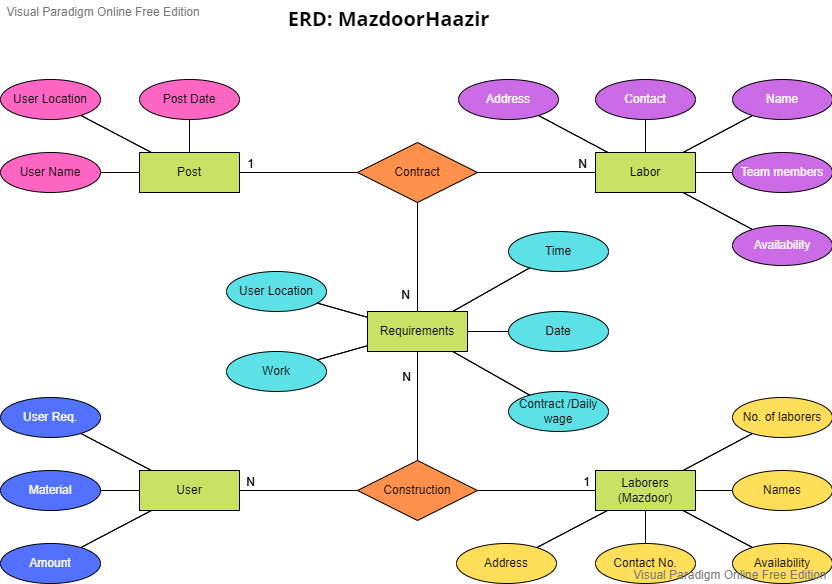
**Sequence Diagrams**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.



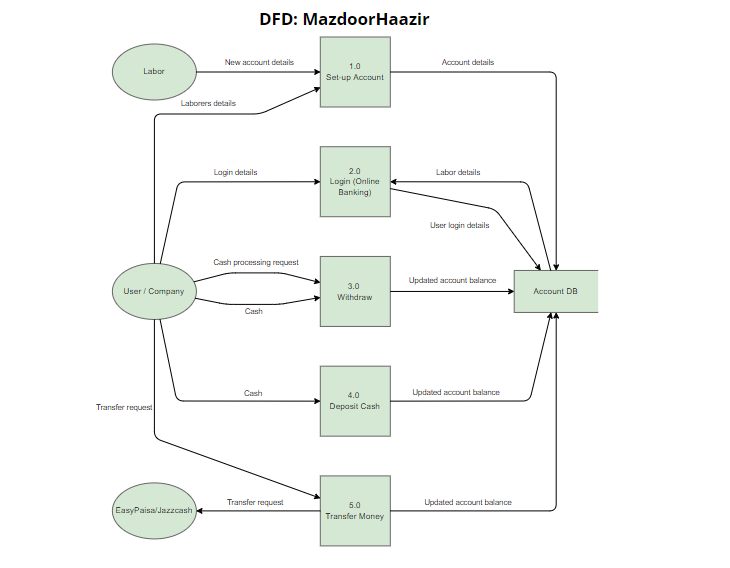
**Entity Relationship Diagram**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.



**Data Flow Diagram**

This Entity Relationship (ER) Diagram shows the model. The entity relationship diagram represents all the optical important of database tables as well as the relations between the id, Name, Order Type, Flavour, Phone Number, Email and Address etc. It uses the structure data for defining the relationship between the structured data groups of functionalities.

****

#### **4.11.1 EXTERNAL ENTITY**

External entities are on the outside of system or process which gets the data from the structured system. They also send the data to the structured system. They are also called the sources, terminators, actors as well as sinks. An external entity, they are also the sources of information as well as destination of information. They are sited on the edges of the diagram.

#### **4.11.2 PROCESS**

It is a procedure or plan which operate or handle the data as well as its flow through taking, changing as well as constructing the next data. The process will do this by using logic as well as performing the computation to classify the data as well as change its flow of the direction. Processes will normally begin from the top left of the dataflow diagram finish on the bottom right of the dataflow diagram.

#### **4.11.3 DATA STORE**

Data store is used for holding the details to use. For example, the file is staying to be prepared. Data input flows by the process and then the data stores. Data output flows out from the data store and then by the process.

#### **4.11.4 DATA FLOW**

Data flow is avenue the information of system which is taken from the external entities by the processes and then the data stores. By using the labels as well as arrows, the dataflow diagram can display us the indication of the dataflow. We must follow the 4 thumb rules to make a valid dataflow diagram before we start the mapping dataflow diagrams.

# **CHAPTER 5****: System Implementation**

## **5.1 Project Architecture**

The architecture for this project consists of:

#### **5.1.1 Labor Profile**

* **Input:** Labor Name, Labor Contact, Team members, Location, Availability
* **Read** user input for the labor profile specification
* **Validate** user inputs; if valid, proceed
* **Establish** connection to database
* **Store** the user or laborers profile data in database

This page considers various types of user input, including text, numbers, and image files. Users are able to create their profile and manage their profile and users have a functionality to add the post to the platform and labors will see them on their application interfaces.

Graphical user interface, text, application

Description automatically generated

#### 

#### **5.1.2 Customer Registration Page**

This page has the name, email, profile picture as well as password. These all attributes are design for the creating for profile but importantly user/labor have to separately entered the details to register their profile for the view on the application. This information will help the user to hire/connect with the labors through their profile data.

Graphical user interface

Description automatically generated

#### **5.1.3 User Login Page**

This page has the mail as well as password fields. User will write the mail and password which they have given in the time of registration. The email as well as password which he/she gave matched with the stored data in the database so they can access with their account.

A picture containing text, indoor, food, dessert

Description automatically generated

#### **5.1.4 User Interface (GUI)**

This is the Graphical user interface of the profile creation which contains the user profile picture and information of labor which has its name, location, contact number and number of team members.

Graphical user interface, application

Description automatically generated

A screenshot of a video game

Description automatically generated

#### **5.1.5 Laborers Profile Section**

This section will contain the laborers profile that is selected by the user this pic will show in this section so that the system can view the profile and contact them.

#### **5.1.6 Job Post Section**

This section will allow the users to post their job requirements and this job will be appeared to the laborers. Labors will be able to apply on these job posts according to the criteria set by the user.

Graphical user interface, website

Description automatically generated

#### **5.1.7 Logo**

The logo of our project is a character which holds the tools which help him in construction work and also our project title is also their “MazdoorHaazir”. It is designed in the photoshop.

Logo

Description automatically generated

#### **5.1.8 Admin Panel**

This section will be managed by the admins of the “MazdoorHaazir”. By the admin, the profiles, jobs or other information can be updated. All the application/website data will be handled by the admin. This section will allow these operations.

Graphical user interface, application

Description automatically generated

## **5.2 Coding Section**

This is the coding section which has the user panel as well as labor view. There are 2 different interfaces of the application. We have added dependencies in pubspec.yml file because this is the file which is our application totally depends on.

#### **5.2.1 User Panel**

Text

Description automatically generated

Text

Description automatically generated

#### **5.2.2 Admin Panel**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

# **Chapter 6** **- System Testing**

In this chapter, we will substantiate and validate our application system. In order to do this, we will use many different techniques of testing. Oversetting, we will check either our system is working well or not? We will set some testing criteria to assess the system. We will give some inputs and then check their productivities. This chapter will be consisted of test plan, unit testing, system testing, integration testing as well as test cases, etc.

## **6.1 Test Plan**

Testing plan has the great importance and it is one of the modules of the design analysis as well as module of the system. Testing is an evaluative phase. The system will go for the testing if the system is ready. The developer of the system needs to undertake that which thing will not be avoided before the final implementation of the system.

If the testing will be completed successfully so it means that the system has no error. Furthermore, the testing also displays that the functions of the software represent to be the working according to the requirements of the user as well as specifications. Secondly, the information or detail which is gathered as testing, gives the acceptable demonstration of reliability of the software and some demonstration of quality of the software.

Testing is able to show us the errors which will be in the software. It cannot show us the absence of faults. Before beginning the testing, the testing should be designed. After completing the coding phase, testing phase will start.

During the testing of the software, the software will be used provisionally so that we can assure that the software is not fail, connections are fine, the result is inspected as well as the test information are input.

## **6.2 Resources**

We discussed various resources which are used in testing phase like what human resources are needed? What will be testing environment? Tools and other resources.

#### **6.2.1 Human Resources**

In this part, we will discuss about their sources about testing. Unit testing, integration testing will be covered by my group member (Danish Ul Hassan), I am responsible to prepare, design, manage, execute, resolving the related issues as well as test activities.

#### **6.2.2 Testing Environment**

All stuff related to testing environment (such as developers, testers, operations staff, testing services, etc.) will be done according to the flutter official documentation, which mainly includes the Flutter widget test which is separately done by flutter for each widget.

#### **6.2.3 Testing Tools**

Flutter allows us to write unit tests as well as integration test, but the best part is that it has the functionality to check each widget using widget test which verifies the working of each widget. All testing methods or levels will be covered as under (manually).

## **6.3 Test Schedule**

A test schedule species the tasks and related dates and modules.

|  |  |  |
| --- | --- | --- |
| **Tester Name** | **Module** | **Schedule** |
| **Danish Ul Hassan** | Main forms | January 2022 |
| **Danish Ul Hassan** | Collecting data | January 2022 |
| **Danish Ul Hassan** | Code writing (UI) | February 2022 to March 2022 |
| **Danish Ul Hassan** | Code writing (Backend) | March 2022 to April 2022 |
| **Danish Ul Hassan** | Finding bugs | April 2022 |
| **Danish Ul Hassan** | Finding errors | May 2022 |
| **Danish Ul Hassan** | Running proper code | May 2022 |
| **Danish Ul Hassan** | Finalizing the code | June 2022 |

## **6.4 Testing Method**

We conducted Black-Box testing throughout testing phase.

#### **6.4.1 Black Box Testing**

Black box testing is the technique in which the internal design or structure of the product, which is be tested is not familiar with the tester. these tests will be functional or nonfunctional. Black box testing is also called the Behavioral testing.

## **6.5 Level of Testing**

There are couple of testing levels which are mentioned below

* System testing
* Integration testing
* Unit testing
* Widget testing (by Flutter)

#### **6.5.1 Unit Testing**

When smallest unit of code of our project is developed, then we can test this smallest unit of testing. This testing will be called as unit testing.

**EXAMPLE**: first we will apply unit testing on sign-in module whether it is behaving according to requirements or not?

Then we will consider sign-up module for unit testing, and so other core modules will be gone through unit testing.

#### **6.5.2 Integration Testing**

If developer’s team is developed some unit of codes, then first we will test separately each unit of code and then we will integrate these unit of codes and perform integration testing.

In unit testing we were testing each module separately, now in integration-testing, we will combine modules (like sign-in & sign-up module) and then apply integration testing, and in this manner, we will go through all other core modules.

#### **6.5.3 System Testing**

When your complete project is completed in terms of code, now here is a time to test whole system, called as system testing.

Now we will combine all modules (such assign-in, sign-up modular module, user module & admin module and other modules) as whole system and will apply system testing.

#### **6.5.4 Gantt Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Feb** | **Mar** | **April** | **May** | **June** | **July** | **Aug** | **Sep** | **Oct** | **Nov** | **Dec** | **Jan** |
| **Do the Literature Survey** | **✓** | **✓** | **✓** | **✓** |  |  |  |  |  |  |  |  |
| **Analyzing Classified website** | **✓** | **✓** | **✓** |  |  |  |  |  |  |  |  |  |
| **Planning Language to be used** |  | **✓** | **✓** |  |  |  |  |  |  |  |  |  |
| **Planning Platform used to Create website** | **✓** | **✓** | **✓** |  |  |  |  |  |  |  |  |  |
| **Planning Data Mining Algorithms** |  |  | **✓** | **✓** | **✓** |  |  |  |  |  |  |  |
| **Planning User Interface** |  | **✓** | **✓** | **✓** |  |  |  |  |  |  |  |  |
| **Analyzing Users of Our website** |  |  |  |  | **✓** | **✓** | **✓** | **✓** | **✓** |  |  |  |
| **Analyzing Server & Hosting** |  |  |  |  |  |  | **✓** | **✓** | **✓** | **✓** |  |  |
| **Developing Graphical Interface(UI/UX)** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** |  |  |
| **Integrating Server to GUI** |  |  |  |  |  |  |  | **✓** | **✓** | **✓** | **✓** |  |
| **Building Database** |  |  |  |  |  |  |  | **✓** | **✓** |  |  |  |
| **Building Data Mining Algorithms** |  |  |  |  |  |  |  | **✓** | **✓** |  |  |  |
| **Black-Box Testing** |  |  |  |  |  |  |  |  | **✓** | **✓** | **✓** |  |
| **Integration** |  |  |  |  |  |  |  |  |  | **✓** | **✓** |  |
| **Validation** |  |  |  |  |  |  |  |  |  |  | **✓** | **✓** |
| **Deployment** |  |  |  |  |  |  |  |  |  |  |  | **✓** |
| **Write the Final Report** |  | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** |
| **Prepare for the Presentation** |  |  |  |  |  |  |  |  |  |  | **✓** | **✓** |
| **Design the Project Poster** |  |  |  |  |  |  |  |  |  |  |  | **✓** |

## **6.6 Test Cases**

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. We have written some test cases too through which we have tested our application as per the test steps.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC ID** | **DESCRIPTION** | **TEST-STEPS** | **EXPECTED RESULT** | **PASS/FAIL** |
| **TC\_001** | Slider of product images is working fine or not? Can it be hovered and clicked | Home page | Each image is slided line by line and it is hovered as well as clicked form going on the order page | Passed. |
| **TC\_002** | Images of the profile, detail of laborers, price and location information are related to the goal. | Labor Profile detail page | All information is related to the laborers profile and GUI is best based on the application successfully | Passed |
| **TC\_003** | Adding profile to the platform, with name, contact no, location, profile picture functionality is working fine? | Profile Data | Platform is able to add in the profile and name, location, contact no, profile picture functionality will work according to the data. | Passed |
| **TC\_004** | Post the jobs, and labors can view the job on their interfaces | After Job Posted | The job posted by the user will enable the laborers to see the job and then they can apply on them based on the criteria set by the user. | Passed |
| **TC\_005** | Login, FAQS, contact-us page and customer services section will tested | Other tests | All pages and sections are related to the website as well as product which is selected by us. | Passed |
| **TC\_006** | Membership or login page testing that the user who is logging, is a part of membership. | Membership or login | Verified that the user is member of website or not by their username and password | Passed |

# **Chapter 7** **- Conclusion And Future Work**

## **7.1 Conclusion**

The concept of “**MazdoorHaazir**” is to create work opportunities for the illiterate community, The MazdoorHaazir platform is for the people who are struggling to find good Mistry’s and Labors to build their house. The aim is to set up a platform where people can connect with the labors and contractors and hire them according to the need. In “**MazdoorHaazir**” platform (application), in which they (mistris and labors) can register their self along with their other team members (labors and workers) and get the jobs (work contract) through the application just creating and managing their profile. They can share their contact details, area preference of work (availability where they are looking for work).“**MazdoorHaazir**” is a platform that connects customers to the skilled, experienced and reliable service professionals in their own locality. “Mazdoor Haazir” is the one-stop destination for all areas needs which includes cemetery, concrete laborers at your doorstep.

## **7.2 Future Work**

The future work which we must do in our website is a mentioned below:

* We have a plan to implement make a database in which system will show the different categories of laborers which includes painters, plumbers, electricians, motor or bike mechanics etc.
* We will work on interface and will make it more elegant and user friendly.
* We will also add trending features on the system.
* We will reduce the manual labour or human efforts.
* Major operations which are done manually, we will do within a matter of second.
* On plus the potential org/companies can display their ads on the platform who are looking for laborers (Ad-based)
* Other types of laborers profile section