

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Online cement bricks is a web application which is meant to help the customers to buy cement based bricks and other building materials. The most basic building material for construction of houses is the conventional brick. The rapid growth in today's construction industry has obliged the civil engineers in searching for more efficient and durable alternatives far beyond the limitations of the conventional brick production. The proposed system includes two users they are administrator, user. Registered customers can login to the site and can search the building construction items. The proposed system enables the user to buy bricks and other building materials. There are different type of bricks are available on the basis sizes. Not only bricks other building materials such as interlocks, cements, concrete windows, M sand, will delivered to the users. This system enables admin to monitor the user and add products. The customer can also see the status of the ads posted. Customers can also chat with the agents if they are looking to buy a property. This system also enables the Employee who is working in this company to login and register. They can enter there details to verify them. And also they can apply leave if they have any medical urgancies. All employee can know there salary details on every end of the month.

1.2 PROJECT SPECIFICATION

The proposed system is made to help the customers to buy the basic construction materials. The speaciality of this project is agent and admin is same role. This system will also act as a bridge between the customer and agent. We will also provide facility for users to post their review about the product. They can also view details of products and quality of the products. The system includes modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add product, view the users, update can see the selling. Admin can also view and manage the user details, manage all the system work.

2. Customer Module

Customer can register and login to the system. Search the product needed. They can also see ratings and review of all the products. Also they can chat with the agents if necessary.

3. Employee Module

Employee can register and login to the system. They can enter there details to verify them. And also they can apply leave if they have any medical urgancies. All employee can know there salary details on every end of the month.

CHAPTER 2

SYSTEM STUDY

3.1 INTRODUCTION

Online cement bricks is a web application which is meant to help the customers to buy cement based bricks and other building materials. The most basic building material for construction of houses is the conventional brick. The rapid growth in today's construction industry has obliged the civil engineers in searching for more efficient and durable alternatives far beyond the limitations of the conventional brick production. The proposed system includes three users they are administrator, user. Registered customers can login to the site and can search the building construction items. The proposed system enables the user to buy bricks and other building materials. There are different type of bricks are available on the basis sizes. Not only bricks other building materials such as interlocks, cements, concrete windows, M sand, will delivered to the users. This system enables admin to monitor the user and add products.

To managing a productive and cost efficient site efficient material management is very essential. Research has shown that construction materials and equipment may constitute more than 70% of the total cost for a typical construction project. Therefore the proper management of this single largest component can improve the productivity and cost efficiency of a project and help ensure its timely completion. One of the major problems in delaying construction projects is poor materials and equipment management.

The cement bricks store attempts to ensure that the right quality and quantity of materials are appropriately selected, purchased, delivered and handled on site in a timely manner and at a reasonable cost. The cement bricks store is the system for planning and controlling all of the efforts necessary to ensure that the correct quality and quantity of materials are properly specified in a timely manner, are obtained at a reasonable cost and most importantly are available at the point of use when required.

3.2 EXISTING SYSTEM OF PROJECT MANAGEMENT SYSTEM

Existing system is not a fully automated system. Customer can register and they can buy their products. Each customer can create their own profile. The proposed system rectifies 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 customers can post their review or they can approach admin for selling their products faster. If any complaint about the product they can register complaint in complaint

register option. They can also search for a particular product the customer wish to buy. Customers can also chat with the admin if they are looking to buy a product.

3.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient in posting the ad details including finding an agent nearest to their location.
- Human effort is needed.
- More manual hours need to generate required reports

3.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. The goal of this system is to ensure that construction materials are available at their point of use when needed. The online cement brick store system attempts to insure that the right quality and quantity of materials are appropriately selected, purchased, delivered and handled on site in a timely manner and at a reasonable cost. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the users. It allows customers to buy the products and do their transactions by using online payment method. Users of this proposed system are admin, customer. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work.

3.5 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:

- **Better security:** -

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of 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. It will also provide data security as we are using the secured databases for maintaining the documents.

- **Ensure data accuracy: -**

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

- **Better service: -**

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

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

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

3.1.1 Economical Feasibility

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

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

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

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

The cost of project, Online cement bricks system was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

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

Technical issues raised during the investigation are:

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

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

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

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

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

Online cement bricks system, GUI is simple so that users can easily use it. Online cement bricks system 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, SCSS

Backend - PHP

Client on PC - Windows 10 and above.

Technologies used - JS, HTML5, AJAX, jQuery, PHP, CSS, SCSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

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

3.3.2 MySQL

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

•MySQL databases are relational.

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

“SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

•MySQL software is Open Source.

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

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

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

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

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

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

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

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML stands for Unified Modelling Language. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modelling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design.

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML, a standard notation forth modelling of real- world objects and systems.

A use case diagram contains four components.

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

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

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

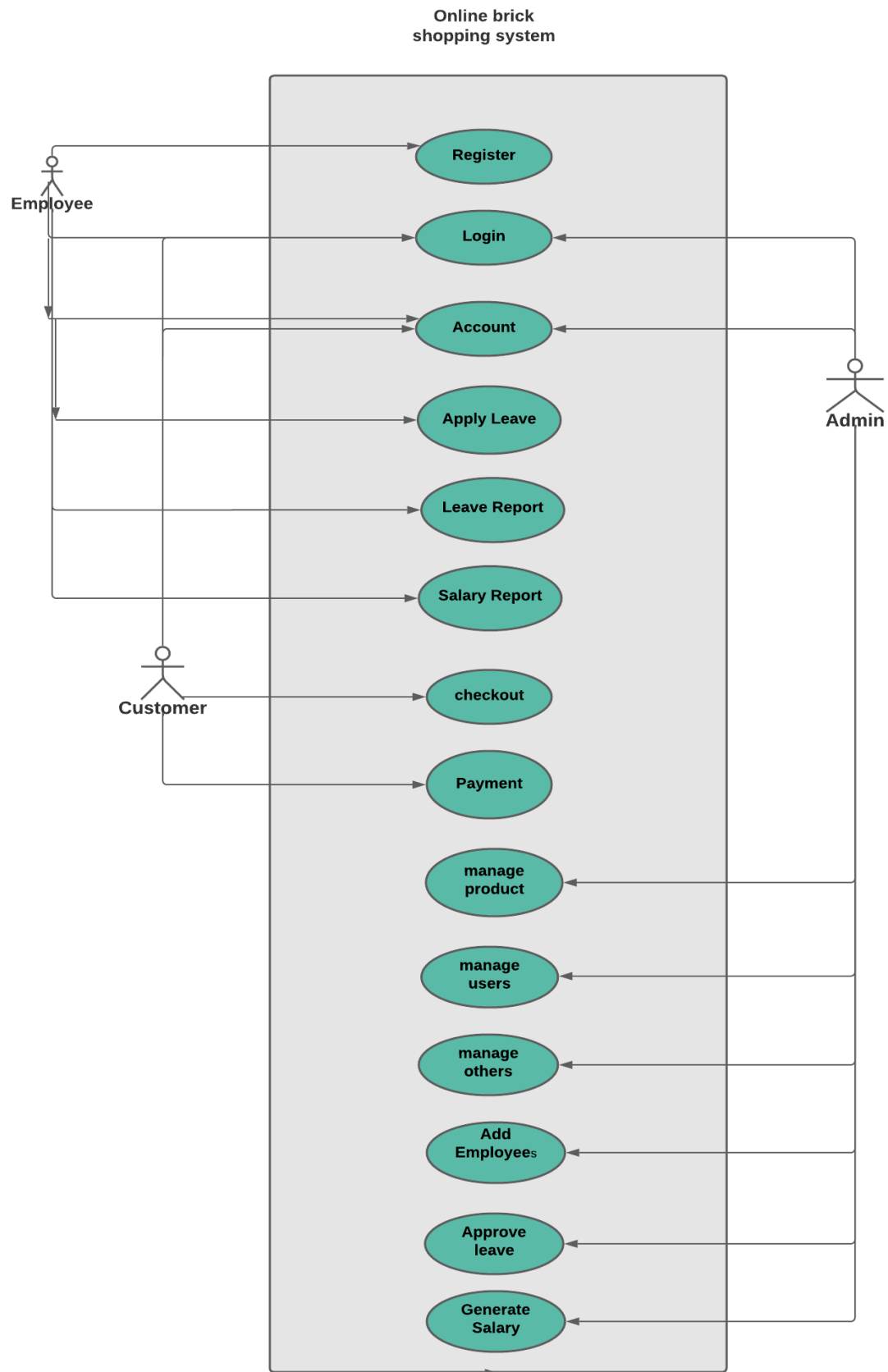


Fig 1: Use case diagram for Online cement brick store

4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
- iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

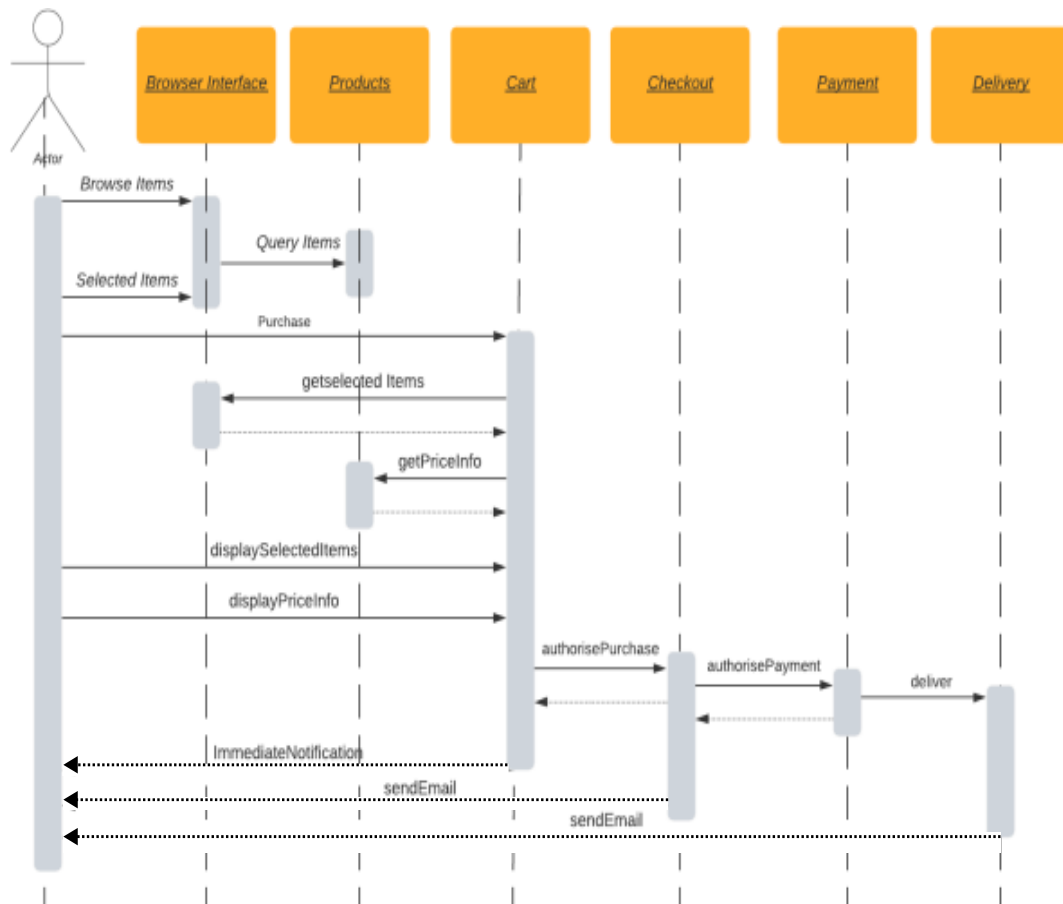


Fig 2: Sequence diagram for Online cement bricks store

4.2.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The following points should be remembered while drawing a class diagram –

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing, it should be understandable to the developer/coder.

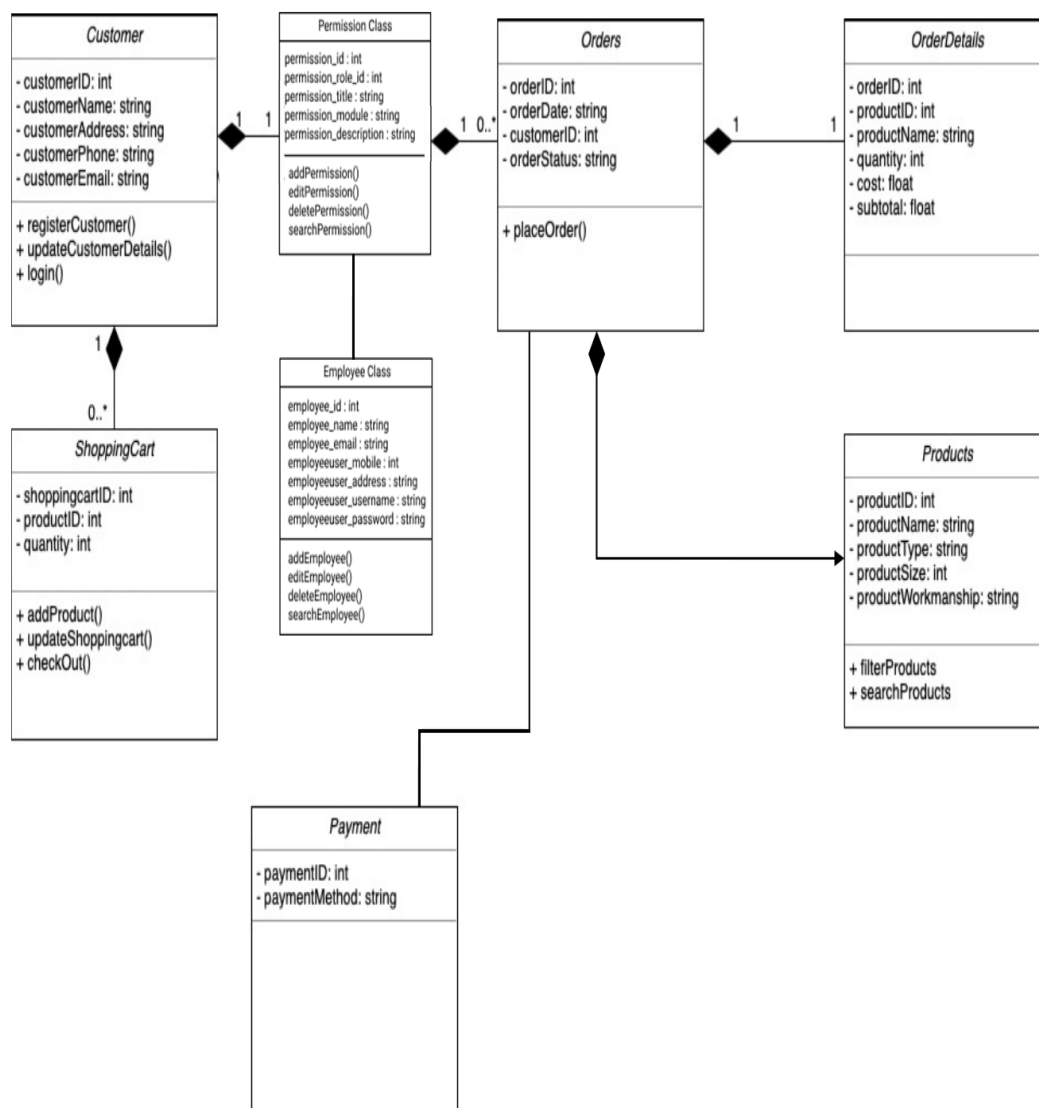


Fig 3: Class diagram for Online cement brick system

4.2.4 ACTIVITY DIAGRAM

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

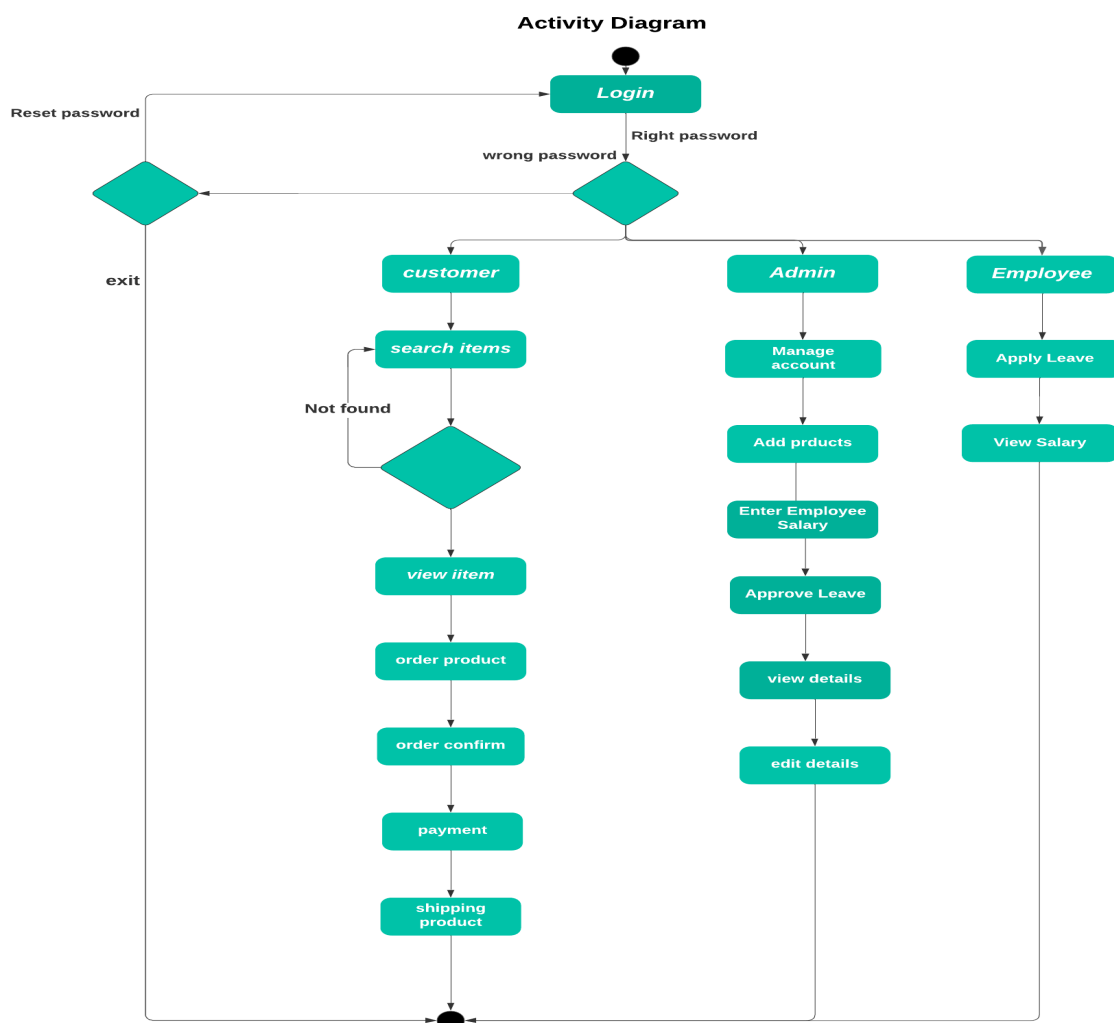


Fig 4: Activity diagram for online cement bricks store

4.2.5 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.

The purpose of deployment diagrams can be described as –

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

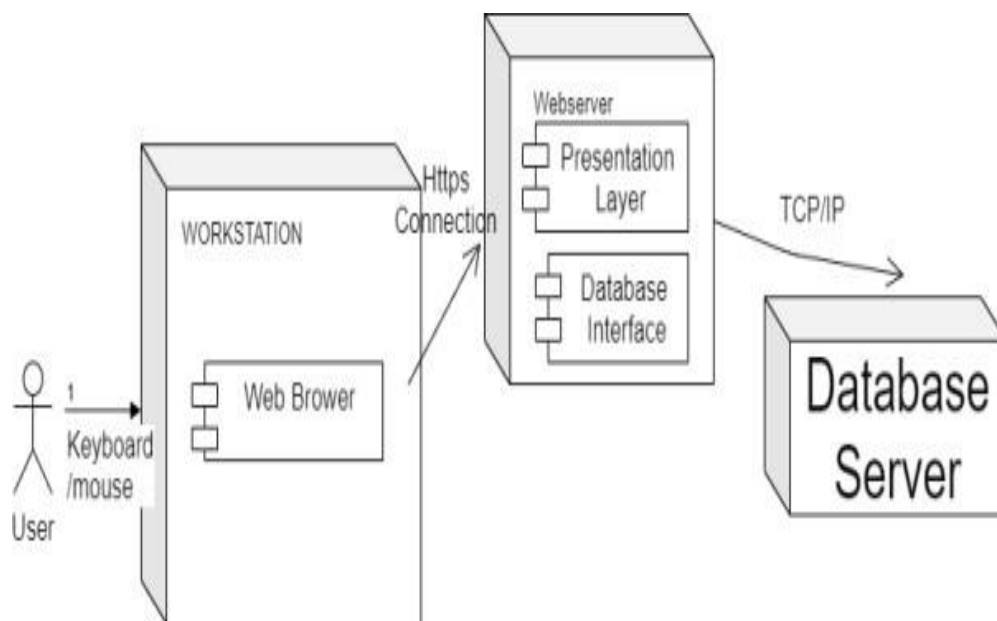


Fig 5: Deployment diagram for online cement bricks store.

4.2.6 STATE DIAGRAM

A state diagram is used to represent the condition of the system or part of the system at finite behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we every class using State diagrams. We prefer to model the states with three or more states.

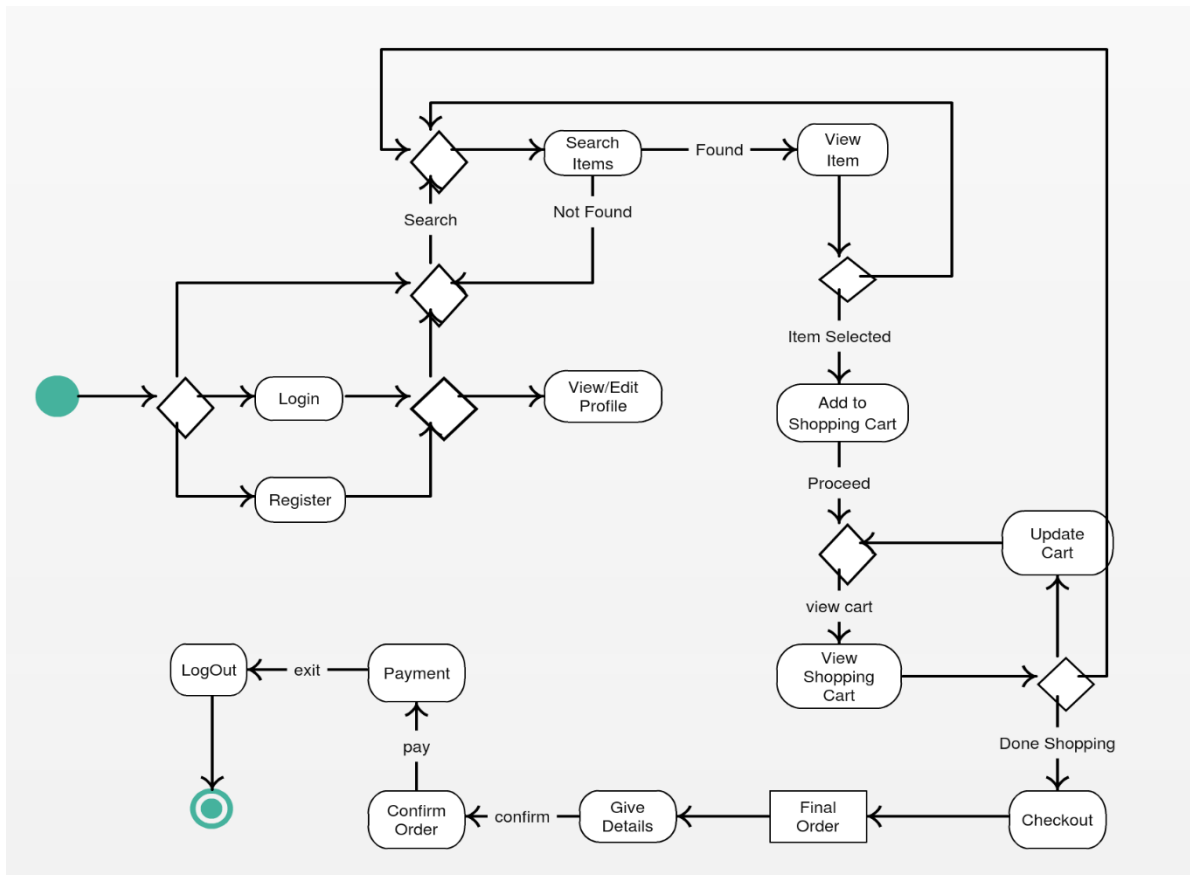


Fig 6: Deployment diagram for online cement bricks store.

4.2.7 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

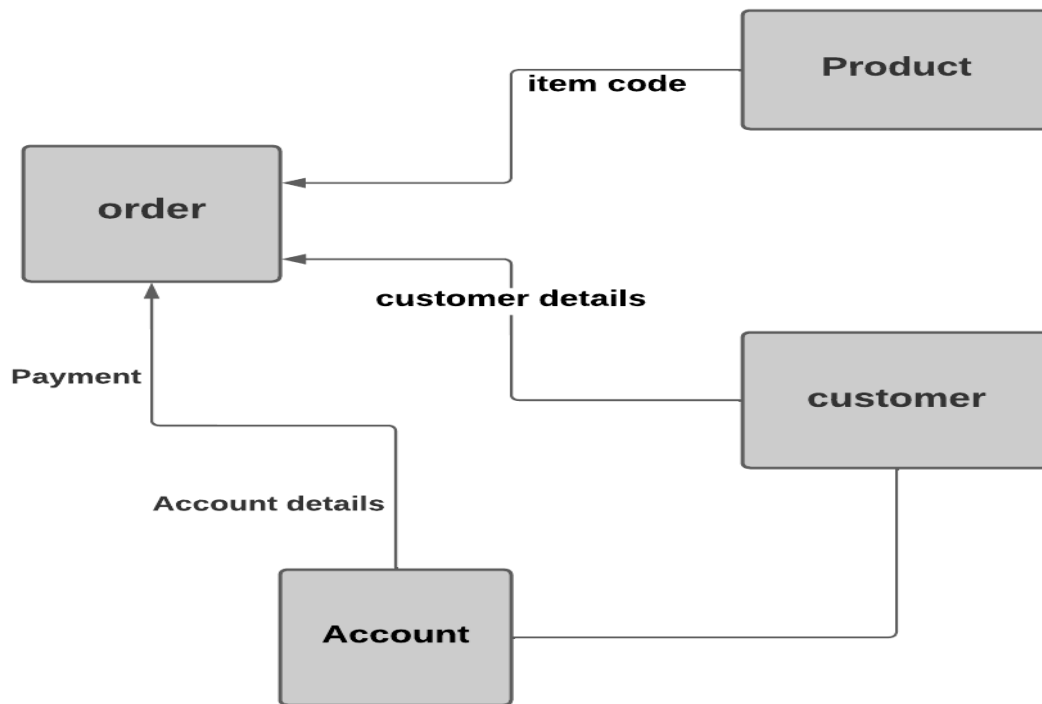


Fig 7: component diagram for online cement bricks store.

4.2.8 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

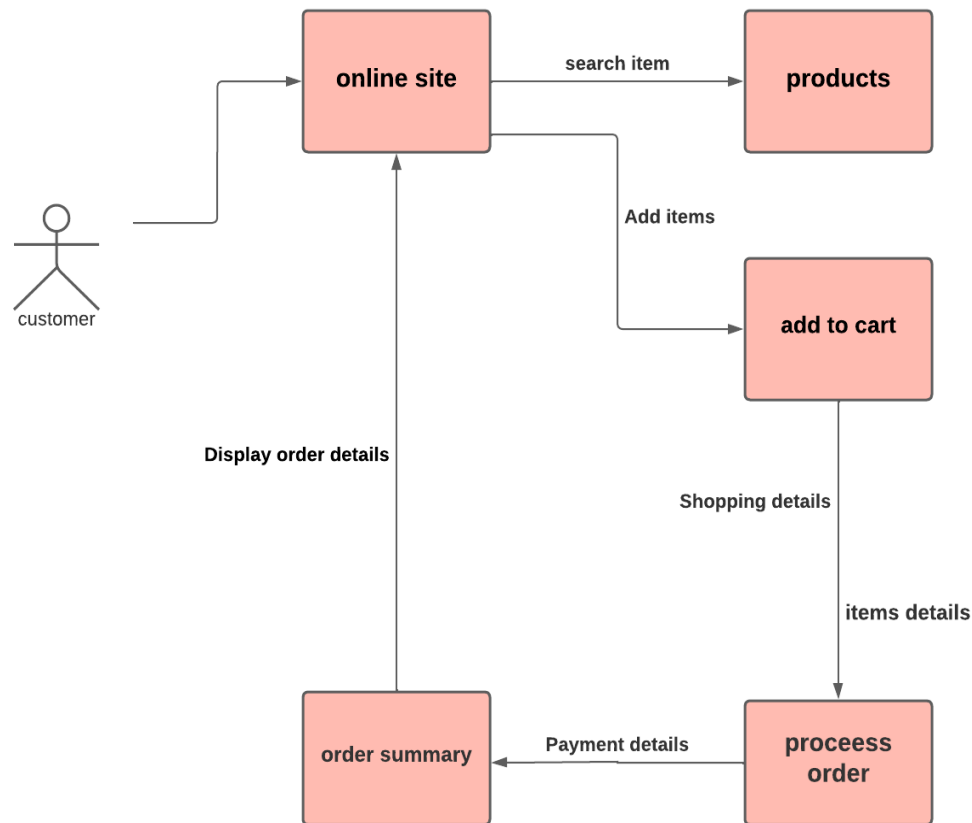


Fig 8: Collaboration diagram for online cement bricks store.

4.2.9 OBJECT DIAGRAM

Object Diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance. The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams. The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature. It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment.

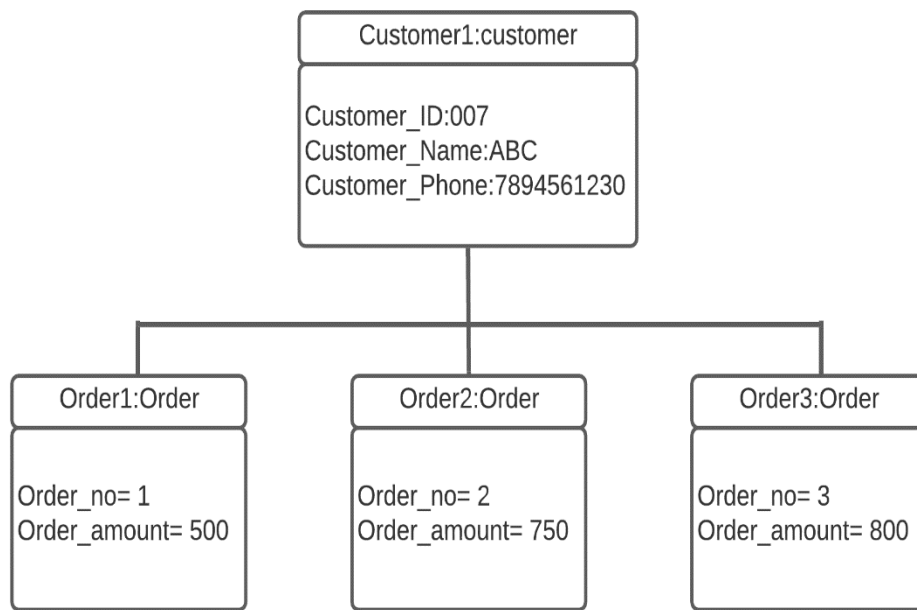
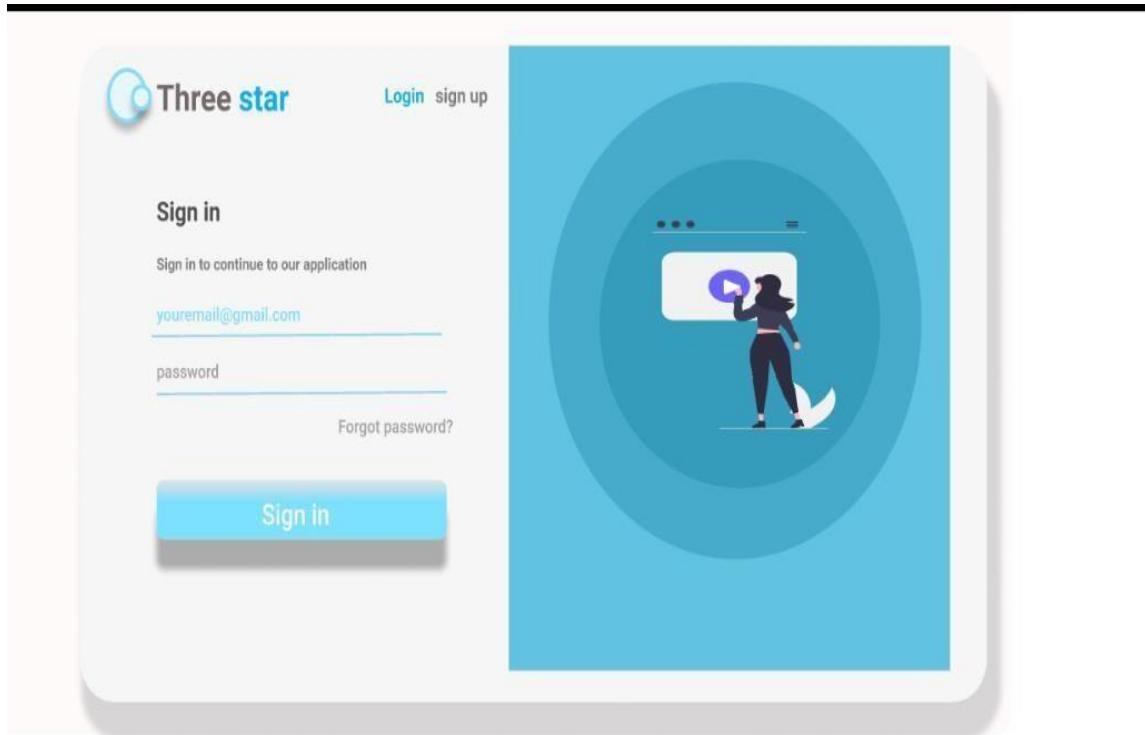


Fig 9: Object diagram for online cement bricks store

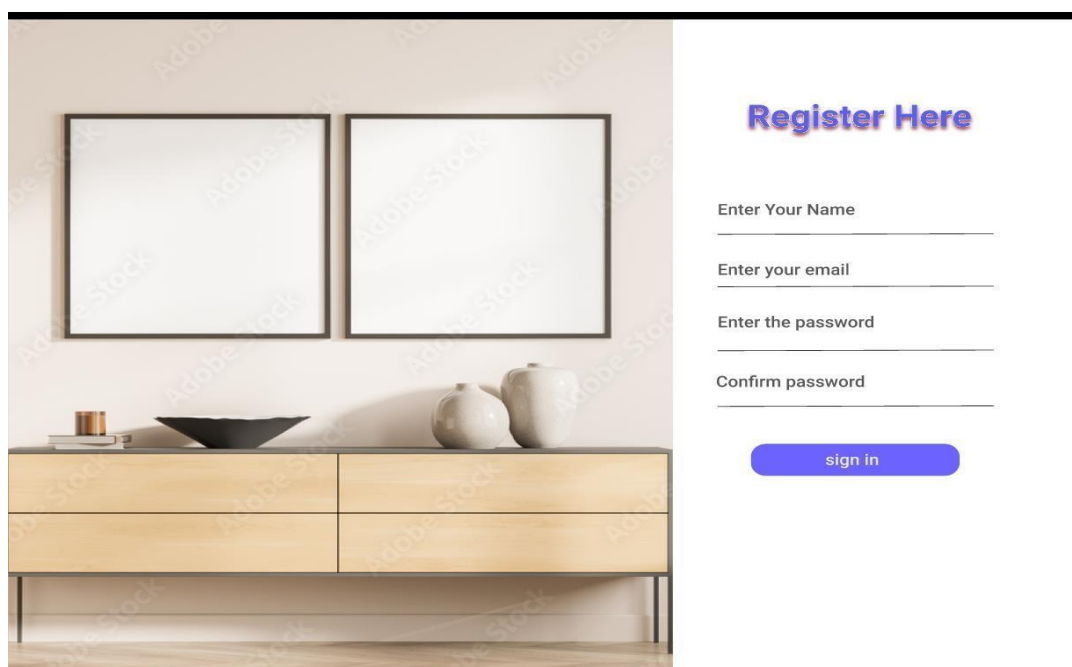
4.3 USER INTERFACE DESIGN

4.3.1 DESIGN PROTOTYPE

Login Page




Registration Pages



Book product

Three star Hollow and solid Bricks	Home	CementBased Bricks	Other building materials	Logout
------------------------------------	------	--------------------	--------------------------	--------



6x solid brick

Enter your name:

Select the date for delivery:

Contact number:

Enter no. of bricks:

Adress of delivery place:

Product detail

Three star Hollow and solid Bricks	Home	CementBased Bricks	Other building materials	Login
------------------------------------	------	--------------------	--------------------------	-------



Featured Products	Bricks	Cements	concrete windows	Door frames	Inter locks
-------------------	--------	---------	------------------	-------------	-------------



Hollow Bricks



Solid Bricks



Concrete windows



cements



Door Frames



Interlocks




M sand

Book product

Three star Hollow and solid Bricks

HomeCementBased BricksOther building materialsLogout



6x solid brick

Enter your name:

Select the date for delivery:

Contact number:

Enter no. of bricks:


Adress of delivery place:

Place order

Product details


Three star Hollow and solid Bricks


HomeCementBased BricksOther building materialsLogin





Featured Products


BricksCementsconcrete windowsDoor framesInter locks



Hollow Bricks



Solid Bricks


Concrete windows


cements


Door Frames


Interlocks


M sand

4.3.2 FORM DESIGNS

Form Name: Add product

Add products

Product Name	<input type="text"/>
Size	<input type="text"/>
Stock	<input type="text"/>
Price	<input type="text"/>
Image	<input type="button" value="Choose Files"/> No file chosen
Category	<input type="text" value="Cement based bricks"/> ▼
<input type="button" value="upload"/>	
<input type="button" value="Back"/>	

Form Name: Register

Create Account


or use your email for registration

Sign Up


Hello, Friend!

Enter your personal details and start journey with us

Form Name: Payment**Credit Card Checkout**

Name on the card	Amount of bricks Ordered
<input type="text"/>	98
Card Number	Payment amount
<input type="text"/>	₹2548
	<input type="button" value="PAY"/>
Expiry date	
<input type="text"/>	
CVV/CVC	
<input type="text"/>	

Form Name: Booking Page



6 Inch solid block
Available Stock:700

Enter your name	<input type="text"/>
Enter your delivery address	<input type="text"/>
Select your date for delivery	<input type="text" value="dd-mm-yyyy"/>
Contact number	<input type="text"/>
Enter the quantity of bricks	<input type="text"/>

Proceed to Pay

4.4 DATABASE DESIGN

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

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

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

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

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

Relations, Domains & Attributes

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

Relationships

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

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

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

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or

indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

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

Third Normal Form

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

TABLE DESIGN**Table No. 01****Table Name : tbl_userdetails****Primary Key : user_id****Table Description : To store Users details**

Field Name	Data Type	Size	Description
user_id	Int	50	Primary key
name	Varchar	50	User's Name
user_email	Varchar	200	Email of user
user_address	varchar	200	Address of user
user_Mobile	Varchar	10	User Phone number
status	Int	5	Status of the User

Table No. 02**Table Name : tbl_login details****Primary Key : login_id****Foreign Key : user_id****Table Description : To store Login details**

Field Name	Data Type	Size	Description
login_id	int	50	primary key
user_id	int	50	Foreign key
user_name	varchar	50	users name
password	varchar	50	user password

Table No 03

Table Name : tbl_booking

Primary Key : booking_id

Foreign Key :user_id

Table Description: To store booking details

Fields	Type	Constraints
user_id	int	Foreign key
booking_id	int	Primary key auto increment
user_name	Varchar	Users name
user_mobile	varchar	User mobile number
user_email	varchar	User email
deliver_date	Date/time	Date of delivery product
no_of_quantity	int	Quantity of product
user_address	varchar	Address of user
user_status	int	As default1

Table No 04

Table Name : tbl_payment

Primary Key : payment_id

Foreign Key :user_id

Table Description : To store payment details

Fields	Type	Constraints
pyment_id	int	Primary key
user_id	int	Foreign key
card_type	Varchar	User card type
card_number	int	User card number
card_holdername	varchar	Card holder name
card_expiry_date	Date/time	Expiry date of card

card_cvv	int	Card cvv
----------	-----	----------

Table No 05

Table Name : tbl_product

Primary Key : product_id

Foreign Key :user_id

Table Description: To store product details

Fieldname	Data Type	Size	Description
product_id	Int	10	primary key
user_id	Int	20	Foreign key
product_name	Varchar	50	Name of the product
product_price	Int	10	Price of the product
product_category	Varchar	50	Category of product
product_size	Date/time	30	Size of the product

Table No 06

Table Name : tbl_empdet

Primary Key : emp_id

Table Description : To store Employee details

Fieldname	Data Type	Size	Description
Employee_id	Int	10	primary key
Employee_fname	Varchar	20	Name of employee
Employee_lname	Varchar	50	Last name of employee
Employee_country	Varchar	10	Employee country
Employee_state	Varchar	50	Employee state
Employee_district	Varchar	30	Employee District
Employee_pincode	Int	50	Employee Pincode
Employee_address	Varchar	200	Employee address
Employee_phone number	Int	100	Employee phone number
Employee_dob	Date/Time	200	Employee dob

Table No 07

Table Name : tbl_empsalary

Primary Key : salary_id

Foreign Key :Employee_id

Table Description: To store Salary details

Fieldname	Data Type	Size	Description
Salary_id	Int	10	primary key
Employee_id	Int	20	Foreign Key
Salary_date	Varchar	50	Salary entry date
Basic Salary	Int	100	Entering basic salary
Bonus	Varchar	50	Entering bonus
Special Allowance	Varchar	30	Entering special allowance
Rent allowance	Int	50	Entering rent allowace
Total_Salary	Varchar	200	Total Salary

Table No 08

Table Name : tbl_leave

Primary Key : leave_id

Foreign Key :Employee_id

Table Description: To store Employee Leave details

Fieldname	Data Type	Size	Description
Leave_id	Int	10	primary key
Employee_id	Int	20	Foreign key
Employee_name	Varchar	100	First name of employee
Employee_number	Int	100	Employee number
Leave_type	Varchar	100	Type of leave
Leave_Sdate	Varchar	30	Leave start date
Leave_edate	Varchar	50	Leave end date

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

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

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

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

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

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

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

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers are always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent

test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

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

5.2.1 Unit Testing

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

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

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

5.2.1.2 Test Case

Project Name: Online cement bricks system					
Login Test Case					
Test Case ID: Login			Test Designed By: Lijin santhosh		
Test Priority (Low/Medium/High): High			Test Designed Date: 17-05-2022		
Module Name: Login Screen			Test Executed By: Ms. Akitha Philip		
Test Title: Verify login with username and password			Test Execution Date: 18-05-2022		
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Login Page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid email	Email: lijinsanthosh@mca.ajce.in	User should be able to login	User logged in and navigated to respective home page	Pass
3	Provide valid password	Password: lijin@123			
4	Click on Login button				
5	Provide invalid email or password	Username: @lijingmail Password: lijin@1234	User should not be able to login	Message about Invalid Credentials is displayed	Pass
6	Provide NULL email or password	Username: (NULL) Password: (NULL)			
7	Click on Login button				
Post-Condition: User is validated with database and successfully logged in to their account. The account session details are logged in database.					

Code

```
package testcases;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
```

```
import chromedriver.DriverSetup;
```

```
public class Login {
    public static WebDriver driver;
```

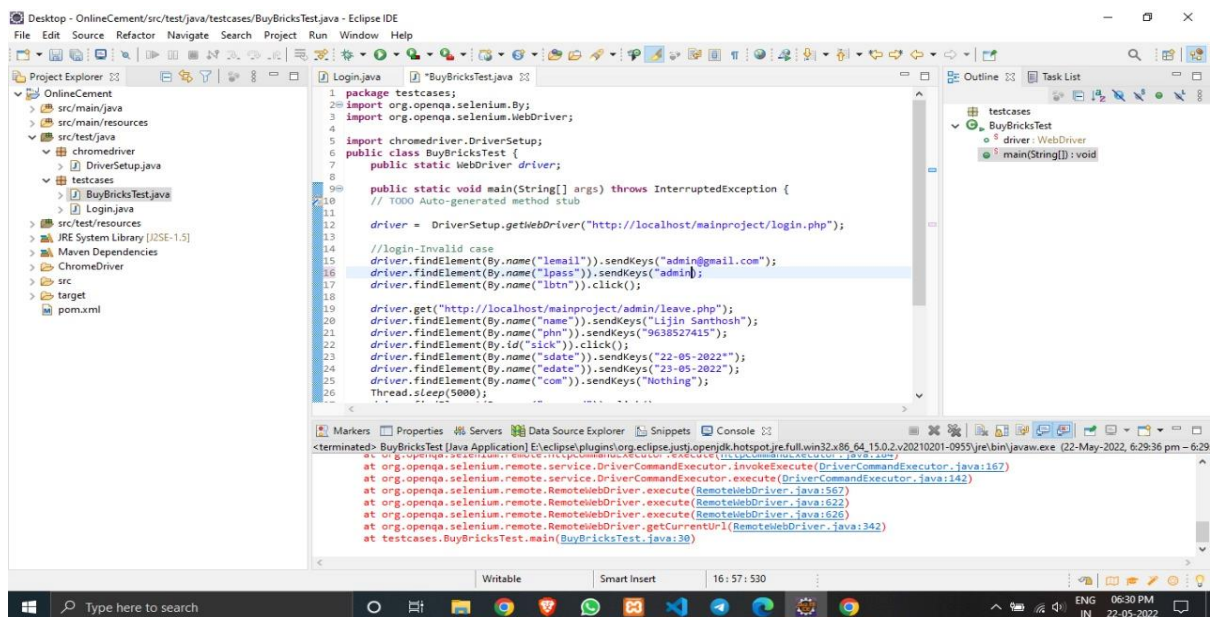
```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```
        driver = DriverSetup.getWebDriver("http://localhost/mainproject/login.php");
```

```
        //login-Invalid case
```

```
        driver.findElement(By.name("lemail")).sendKeys("lijinanthosh@mca.ajce.in");
        driver.findElement(By.name("lpass")).sendKeys("lijin123*");
        driver.findElement(By.name("lbtn")).click();
```

```
        String actualUrl="http://localhost/mainproject/index.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed"); } else { System.out.println("Test failed");
        }
        driver.quit();
    }
}
```



5.2.1.3 Test Case

Project Name: Online cement bricks system					
Employee Leave Apply Test Case					
Test Case ID: Apply Leave			Test Designed By: Lijin santhosh		
Test Priority (Low/Medium/High): High			Test Designed Date: 17-05-2022		
Module Name: Leave Screen			Test Executed By: Ms. Akitha Philip		
Test Title: Leave apply page			Test Execution Date: 18-05-2022		
Description: Test the Leave apply Page					
Pre-Condition: Employee should register					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Employee Page		Employee page displayed	Employee page displayed	Pass
2	Entering leave details	Employee name	Employee should able to apply	Employee applied	Pass
3	Entering Employee name	Bhushan			
4	Click on Apply button				
5	Type of Employee	Entering type of Employee	Employee should not be able to register	Message about Invalid Credentials is displayed	Pass
6	Provide NULL type	Providing null type			
7	Click on Submit button				
Post-Condition: Employee should have registered account					

Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;

public class LeaveApplyTest {
    public static WebDriver driver;

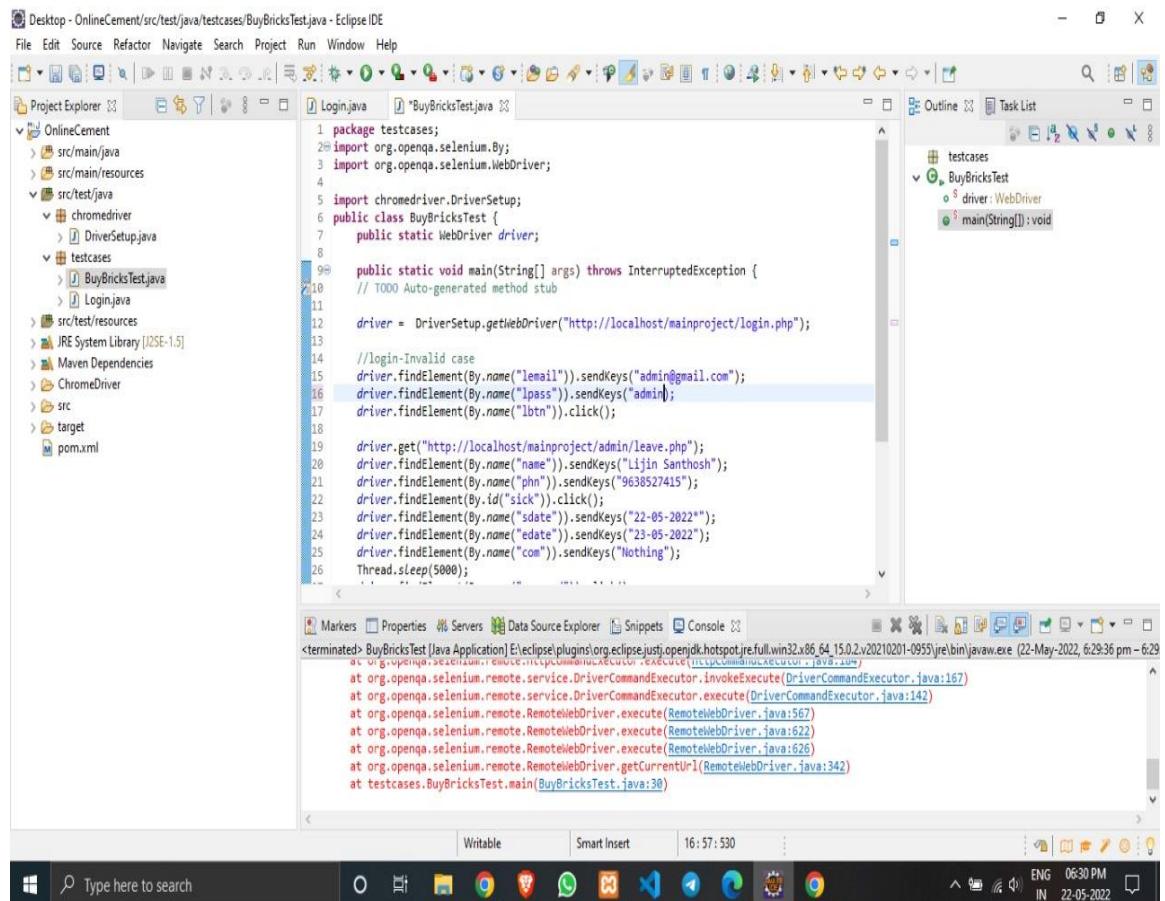
    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub

        driver = DriverSetup.getWebDriver("http://localhost/mainproject/login.php");

        //login-Invalid case
        driver.findElement(By.name("lemail")).sendKeys("admin@gmail.com");
        driver.findElement(By.name("lpass")).sendKeys("admin");
        driver.findElement(By.name("lbtn")).click();

        driver.get("http://localhost/mainproject/admin/leave.php");
        driver.findElement(By.name("name")).sendKeys("Lijin Santhosh");
        driver.findElement(By.name("phn")).sendKeys("9638527415");
        driver.findElement(By.id("sick")).click();
        driver.findElement(By.name("sdate")).sendKeys("22-05-2022*");
        driver.findElement(By.name("edate")).sendKeys("23-05-2022");
        driver.findElement(By.name("com")).sendKeys("Nothing");
        Thread.sleep(5000);

        driver.findElement(By.name("proceed")).click();
        String actualUrl="http://localhost/mainproject/admin/leave.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed"); } else { System.out.println("Test failed");
        }
        driver.quit();
    }
}
```



5.2.2 Integration Testing

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

5.2.3 Validation Testing or System Testing

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

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

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

5.2.4 Output Testing or User Acceptance Testing

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

- Input Screen Designs
- Output Screen Designs

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

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

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

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

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

The implementation state involves the following tasks:

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

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development

project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

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

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

6.2.1 User Training

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

6.2.2 Training on the Application Software

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

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance

is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to view and buy the products.

It has 24-hour service so people who work nights that aren't able to go purchase then they are able to do purchase the brick product at any time of the day or night. Before online shopping there was only shopping in store but as the world has got more technology, shopping online has become more popular.

They could improve online shopping by making the delivery time a lot quicker or allowing people to make a delivery time exactly when they want it. They could also improve online shopping's security levels with this they could gain a lot more customers. So less able people who aren't able to get out very much, it gives them some sort of freedom because they can do anything online without help. In store shops have very limited amount of stock they can hold, but online has an infinity amount of stock.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode.
- Customers can able to do advanced search options
- Customers can able to add complaints and feedbacks etc.
- 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, “Software engineering: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES

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

CHAPTER 9

APPENDIX

9.1 SAMPLE CODE

customer

Home.php

```
<?php

include 'connect.php';

if (isset($_SESSION['sessionId'])) {

?>

<!doctype html>

<html>

<title></title>

<head>

<meta charset="utf-8">

<style>

.flex-container {

display: flex; background-color: #fff; align-items: stretch;

}

.flex-container>div { background-color: #f1f1f1; margin: 20px;

padding: 30px; font-size: 14px; text-align: center; line-height: 75px;

}

<link rel="stylesheet" type="text/css" href="style.css">

<body>

<div class="topnav">

<ul>

<li><a class="active" href="#">Three Star Hollow and Solid Bricks</a></li>
```

```
<li><a href="">Home</a></li>

<li><a href="cmt.php">Cement Based Bricks</a></li>

<li><a href="">Other Building Materials</a></li>

</style>

        <li><a class="log" href="logout.php">Logout</a></li>

<?php
if (isset($_SESSION['login_user']))
?>

<li><a href="profile.php"> <?php echo $_SESSION['userName']; ?> </a></li>

</ul>

</div>

<div class="slider1">

</div>

<div class="topnav1">

<ul>

<li><a class="active" href="">Featured Products</a></li>

<li><a href="">Bricks</a></li>

<li><a href="">Cements</a></li>

<li><a href="#about">Concrete Windows</a></li>

<li><a href="#about">Door Frames</a></li>

<li><a href="#about">Well Bricks</a></li>

<li><a href="#about">InterLocks</a></li>

</ul>
```

```
</div><br><br>

<div class="flex-container">

<div>

<P>

<h2>16" x 8" x 4" Hollow Bricks</h2>

</P>

</div>

<div>

<P>

<h2>16" x 8" x 6" solid Bricks</h2>

</P>

</div>

<div>

<P>

<h2>16" x 8" x 8" solid Bricks</h2>

</P>

</div>

<div>

<P>

<h2>Cements</h2>

</P>

</div>

<div>

<P>
```

```
<h2>concrete windows</h2>

</P>

</div>

</div>

<div class="flex-container">

<div>

<P>

<h2>M sand</h2>

</P>

</div>

</div>

<div>

<P>

<h2>InterLocks</h2>

</P>

</div>

</div>

<style> ul {

list-style-type: none; margin: 0;

padding: 0; overflow: hidden;

background-color: #fff; text-color: black;

}

.topnav1 input[type=text]

{ float: right;
```

```
padding: 6px;

border: none; margin-top: 8px; margin-right: 16px; font-size: 17px;

}

li {

float: left;

. li a {

display: block; color: #000000;

text-align: center; padding: 14px 16px; text-decoration: none;

}

li a:hover { color: #fff;

}

active {

background-color: #4CAF50;

}

</style>

<script>

var slideIndex = 1; showSlides(slideIndex);

function plusSlides(n) { showSlides(slideIndex += n);

}

function currentSlide(n) { showSlides(slideIndex = n);

}

function showSlides(n) {

var i;

var slides = document.getElementsByClassName("mySlides");
```

```
var dots = document.getElementsByClassName("dot");

        }

slides[slideIndex - 1].style.display = "block";

dots[slideIndex - 1].className += " active";

}

</script>

</body>

</html>
```

Bookproduct.php

```
-<?php

include 'connect.php';

?>

<!doctype html>

<html>

<title></title>

<head>

<meta charset="utf-8">

<style>

div.polaroid { width: 50%;

background-color: white;

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

margin-bottom: 30px;

}

div.container {
```

```
text-align: center;

div.container {

text-align: center; padding: 10px 20px;

}

</style>

<link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

<script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

<script type="text/javascript" src="./registration.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

</head>

<link rel="stylesheet" type="text/css" href="style.css">

<script>

function redirect() { window.location = "login.php";

}

function isalph(evt) {

var charCode = (evt.which) ? evt.which : event.keyCode

if ((charCode == 32) || (charCode > 64 && charCode < 91) || (charCode > 96 && charCode
< 123))

return true;

return false;
```

```
fQybjgWLrvvRgtW6bFIB7jaZrFsaBXjsOMm/tB9LTS58ONXgqbR9W8oWht/amnpF"
crossorigin="anonymous">

}

function isnum(evt) {

var charCode = (evt.which) ? evt.which : event.keyCode if (charCode > 31 && (charCode <
48 || charCode > 57))

return false;

return true;

}

</script>

<body style="background-color: #fefbd8;">

<div class="topnav">

<ul>

<li><a class="active" href="#">Three Star Hollow and Solid Bricks</a></li>

<li><a href="index.PHP">Home</a></li>

<div class="dropdown">

<button class="dropbtn">Cement based bricks

<i class="fa fa-caret-down"></i>

</button>

<div class="dropdown-content">

</div>

</div>

<li><a href="">Other Building Materials</a></li>

<li><a class="log" href="login.php">Logout</a></li>

<?php
```

```
if (isset($_SESSION['login_user']))

?>

<li><a href="</ul>

</div>

<?php

$Id = $_GET['i'];

$sel = "select * from tbl_product where id='$Id'";

$res = mysqli_query($con, $sel);

$itemCount = 0;

while ($ar = mysqli_fetch_array($res)) {

$itemCount = $ar["stock"];

?>

<center>

?>">

<div class="polaroid"><?php echo $ar["name"]; ?></h4>

<h4>Available Stock:

<?php echo

</center><br>

</div>

</div>

<?php

}

?>
```

```
<form method="post" action="#">

<input type="hidden" name="itemCount" value="<?php echo $itemCount; ?>">

<input type="hidden" name="itemId" value="<?php echo $id; ?>">

<center>

<table border="2" cellspacing=10px cellpadding=15px>

<tr>

<td><b>Enter your name</b></td>

<td>

<input type="text" id="name" name="name" onkeypress="return isalph(event)"

    required="">

</td>

</tr>

<tr>

<td><b>Enter your delivery address</b></td>

<td>

<input type="text" id="addres" name="addres" onkeypress="return isalph(event)"

    required="">

</td>

</tr>

<td><b>Select your date for delivery</b></td>

<td>

<input type="date" name="date" min="<?php echo date('Y-m-d'); ?>

" id="dat" required="">

</td>
```

```
</tr>

<tr>

<td><b>Contact number</b></td>

<td>

<input type="text" name="contactnumber" id="num" maxlength="10" onkeypress="return
isnum(event)" required="">

</td>

</tr>

<tr>

<td><b>Enter the quantity of bricks</b></td>

<td>

<input type="number" name="amount" id="num" required="">

</td>

</tr>

</table><br>

</div>

<button class="submit" type="submit" name="proceed" value="<?php echo $id; ?>"
id="proceed" style="background-color:#fefbd8; height:50px; width:200px; border-
radius:10px; ">Proceed to Pay</button>

</form><br><br>

</center>

<br><br>

</center>

<div class="fixed-footer">

<div align="center" class="container">
```

```
<center>

copyright &copy; owned by lijn@2021.all rights reserved

</center>

</div>

</div>

</body>

</html>

<?php

if (isset($_POST['proceed'])) {

$itemCount = $_POST['amount'];

$itemId = $_POST['itemId'];

$_SESSION['kid'] = $_POST['proceed'];

$bname = $_POST['name'];

$baddress = $_POST['address'];

$bdate = $_POST['date'];

$bcontact = $_POST['contactnumber'];

$bamt = $_POST['amount'];

$sql = "INSERT INTO `tbl_book` ( `Name`, `address`, `date`, `contact`, `amt_bricks` )

VALUES('$bname','$baddress','$bdate','$bcontact','$bamt')";

if (mysqli_query($con, $sql)) {

echo $updateQuantity = "UPDATE `tbl_product` SET `stock`=stock-'$itemCount' WHERE

id='$itemId'";

if (mysqli_query($con, $updateQuantity)) { echo ("<script LANGUAGE='JavaScript'>

window.location.href='payment.php';

</script>");
```

```
}  
  
}  
  
}  
  
?>
```

Payment.php

```
<?php  
  
include 'connect.php';  
  
$s = $_SESSION['userName']  
  
?>  
  
<?php  
  
$a = "SELECT * FROM `tbl_book` WHERE id=(select max(id) from tbl_book)";  
  
$res = mysqli_query($con, $a);  
  
$ar = mysqli_fetch_array($res);  
  
$prodnum = $ar['amt_bricks'];  
  
$kid = $_SESSION['kid'];  
  
$final_rate = $prodnum * $kid;  
  
?>  
  
<!doctype html>  
  
<html>  
  
<title></title>  
  
<head>  
  
<linkhref="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"  
rel="stylesheet"integrity="sha384-  
1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"  
crossorigin="anonymous">
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

<link rel="stylesheet" type="text/css" href="style8.css">

<script type="text/javascript" src="./payment.js"></script>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" />

<script <head>

  <meta charset="utf-8" />

  <meta http-equiv="X-UA-Compatible" content="IE=edge" />

  <title></title>

  <meta name="description" content="" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link rel="stylesheet" href="./stylesheets/css/style.css" />

type="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js">

</script>

<linkhref="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css" />

<script type="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

<body>

<div class="container-fluid px-1 px-md-2 px-lg-4 py-5 mx-auto">

<div class="row d-flex justify-content-center">

<div class="col-xl-7 col-lg-8 col-md-9 col-sm-11">

<div class="card border-0">
```

```
<div class="row justify-content-center">

<h3 class="mb-4">Credit Card Checkout</h3>

</div>

<div class="row">

<div class="col-sm-7 border-line pb-3">

<div class="form-group">

<p class="text-muted text-sm mb-2">Name on the card</p> <input

type="text" name="name" placeholder="Name" size="15">

</div>

<div class="form-group">

<p class="text-muted text-sm mb-2">Card Number</p>

<div class="row px-3">

<input type="text" name="card-num" placeholder="0000 0000 0000 0000" size="18"

id="cr_no" minlength="19" maxlength="19">

<pclass="mb-0ml-3">

/</p><imgclass="image ml-1" src="https://i.imgur.com/WIAP9Ku.jpg">

</div>

</div>

<div class="form-group">

<p class="text-muted text-sm mb-3">Expiry date</p> <input type="text" name="exp"

placeholder="MM/YY" size="6" id="exp" minlength="5" maxlength="5">

</div>

<div class="form-group">

<p class="text-muted text-sm mb-3">CVV/CVC</p> <input type="password" name="cvv"

placeholder="000" size="1" minlength="3" maxlength="3">
```

```
</div>

<div class="form-group mb-2">

<div class="custom-control custom-checkbox custom-control-
inline"></div>

</div>

</div>

<div class="col-sm-5 text-sm-center justify-content-center pt-4 pb-4">

<small class="text-sm text-muted">Amount of bricks Ordered</small>

<h5 class="mb-3">

<h4><?php echo $ar["amt_bricks"]; ?></h4>

</h5> <small class="text-sm text-muted">Payment amount</small>

<div class="row px-3 justify-content-sm-center">

<h2 class=""><span class="text-md font-weight-bold mr-2">₹</span><span class="text-
danger"><?php echo "$final_rate" ?></span></h2>

</div> <a href="succes.php"><button type="submit" class="btn btn-red text- center mt-
4">PAY</button></a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

    </div>

</div>
```

BY Admin**Add Product.php**

```
<?php

include 'connect.php';

?>

<html>

<head>

<title>Login</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

<script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

<style> h1 {

padding-bottom: 10px;

}

table, th,

td {

border: 1px solid black;

}

</style>

</head>
```

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
<body>
```

```
<div class="topnav">
```

```
font-weight: bold;font-family: Impact, Haettenschweiler, 'Franklin Gothic Bold', 'Arial Black',  
'sans-serif';color:#00FF24;">Add products</p>
```

```
<div class="">
```

```
<form name="form1" action="#" method="post" enctype="multipart/form-data">
```

```
<table class="pdetails" border="2px">
```

```
<div>
```

```
<tr>
```

```
<td>Product Name</td>
```

```
<td><input type="text" name="name"></td>
```

```
</tr>
```

```
</div>
```

```
<div>
```

```
<td>Size</td>
```

```
<td><input type="text" name="size"></td>
```

```
</tr>
```

```
</div>
```

```
<div>
```

```
<tr>
```

```
<td>Stock</td>
```

```
<td><input type="text" name="stock"></td>
```

```
</div>
```

```
<div>
```

```
<tr>

<td>Price</td>

<td><input type="text" name="price"></td>

</tr>

</div>

<div>

<tr>

<td>Image</td>

<td><input type="file" name="imageurl" multiple style="color:red;"></td>

</tr>

</div>

<div>

<tr>

<td>Category</td>

<td><select name="category" value="category">

<option value="Cement_based_bricks">Cement based bricks</option>

<option value="other_building_materials">other building materials</option>

</select>

</td>

</tr>

</div>

<div>

<tr>

<td colspan="2" align="center"> <input type="submit" name="addproductSubmit"
id="buttton" value="upload"></td>
```

```
</tr>

</div>

<div>

<tr>

<td colspan="2" align="center"> <a href="admin.php"><input type="button" name="back"
id="buttton" value="Back"></a></td>


</tr>

</div>

</table>

</form>

</div>

</center>

</div>

<div class="fixed-footer">

<div align="center" class="container">

<center>

copyright &copy; owned by lijn@2021.all rights reserved

</center>

</div><br><br>

</div>

</body>

</html>

<?php

include 'connect.php';
```

```
if (isset($_POST['addproductSubmit'])) {

    $pname = $_POST['name'];

    $psize = $_POST['size'];

    $pstock = $_POST['stock'];

    $pprice = $_POST['price'];

    $img = $_FILES['imageurl']['name'];

    $pcat = $_SESSION['category'] = $_POST['category'];

    $fnm = $_FILES['imageurl']['tmp_name'];.php

<?php

    $dst = 'product_image/' . $_FILES['imageurl']['name']; move_uploaded_file($fnm, $dst);

    $sql = "INSERT INTO `tbl_product`(`name`, `size`, `stock`, `price`, `imageurl`,`category`)
        VALUES ('$pname','$psize','$pstock','$pprice','$dst','$pcat')";

    $res = mysqli_query($con, $sql);

    if ($pcat == 'Cement_based_bricks') { echo ("

<script>

    alert('Product Added'); window.location.href='cmt.php';

</script>");

    }

}

?>
```

User

Profile.php

```
<?php

include 'connect.php';
```

```
?>

<!DOCTYPE html>

<html>

<head>

<title>profile</title>

<link rel="stylesheet" type="text/css" href="style.css">

<link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

<scriptsrc="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous">

</script>

<style>

.wrapper { width: 400px;

margin: 0 auto;

background-color: #b2ccd3;

}

</style>

</head>

<body style="background-color:#b2ccd3;">

<div class="topnav">

<ul>

<li>
```

```
<a class="active" href="#">Three Star Hollow and Solid Bricks</a></li>
```

```
<li><a href="index.PHP">Home</a></li>
```

```
<li><a href="">Cement Based Bricks</a></li>
```

```
<li><a href="">Other Building Materials</a></li>
```

```
<li><a class="log" href="login.php">Logout</a></li>
```

```
</ul>
```

```
</div><br>
```

```
<div class="container">
```

```
<form action="" method="post">
```

```
</form>
```

```
<div class="wrapper">
```

```
<h2 style="text-align:center;">My Profile</h2>
```

```
<center>
```

```

```

```
<div>Welcome <?php echo $_SESSION['userName']; ?></div>
```

```
<br><br>
```

```
<table border='3'>
```

```
<?php
```

```
$sql = 'SELECT * FROM registration WHERE `id` = ' . $_SESSION['userId']
```

```
$result = mysqli_query($con, $sql); if (mysqli_num_rows($result) > 0) {
```

```
    $row = mysqli_fetch_array($result);
```

```
    echo "
```

```
<tr>
```

```
<td>Name</td>
```

```
<td>" . $row['name'] . "</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Email</td>
```

```
<td>" . $row['email'] . "</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Phone</td>
```

```
<td>" . $row['phno'] . "</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Password</td>
```

```
<td>" . $row['password'] . "</td>
```

```
</tr> ";
```

```
}
```

```
?>
```

```
</table><br>
```

```
<button class="btn-2" type="button" data-bs-toggle="modal" data-bs-  
target="#staticBackdrop">Edit</button>
```

```
<div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-  
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">
```

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

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

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

```
<h5 class="modal-title" id="staticBackdropLabel">Update your
details</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>

</div>

<form class="modal-body" action="#" method="post">

<table border='2'>

<?php
$sql = 'SELECT * FROM registration WHERE `id` = ' .
$_SESSION['userId'] . ' ';

$result = mysqli_query($con, $sql); if (mysqli_num_rows($result) > 0) {
$row = mysqli_fetch_array($result); echo "

<tr>

<td>Name</td>

<td><input type='text' value=" . $row['name'] . " name='name' id='name'></td>

</tr>

<tr>

<td>Phone</td>

<td><input type='text' value=" . $row['phno'] . " name='phone' id='phone'></td>

</tr>

<tr>

<td>Password</td>

<td><input type='text' value=" . $row['password'] . " name='cpass' id='pass'></td>

</tr>
```

```
<td>Enter new Password</td>

<td><input type='text' name='pass'></td>

</tr>

}

?>

</table>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>

<input type="submit" name="updateprofile" id="updateprofile" class="btn btn-
primary"></button>

</div>

</div>

</div>

</div>

</div>

</center>

</div>

</div>

</form>

</body><?php

if (isset($_POST['updateprofile'])) {

$name = $_POST['name'];

$phone = $_POST['phone'];

$password = $_POST['pass'];

$uid = $_SESSION['userId'];

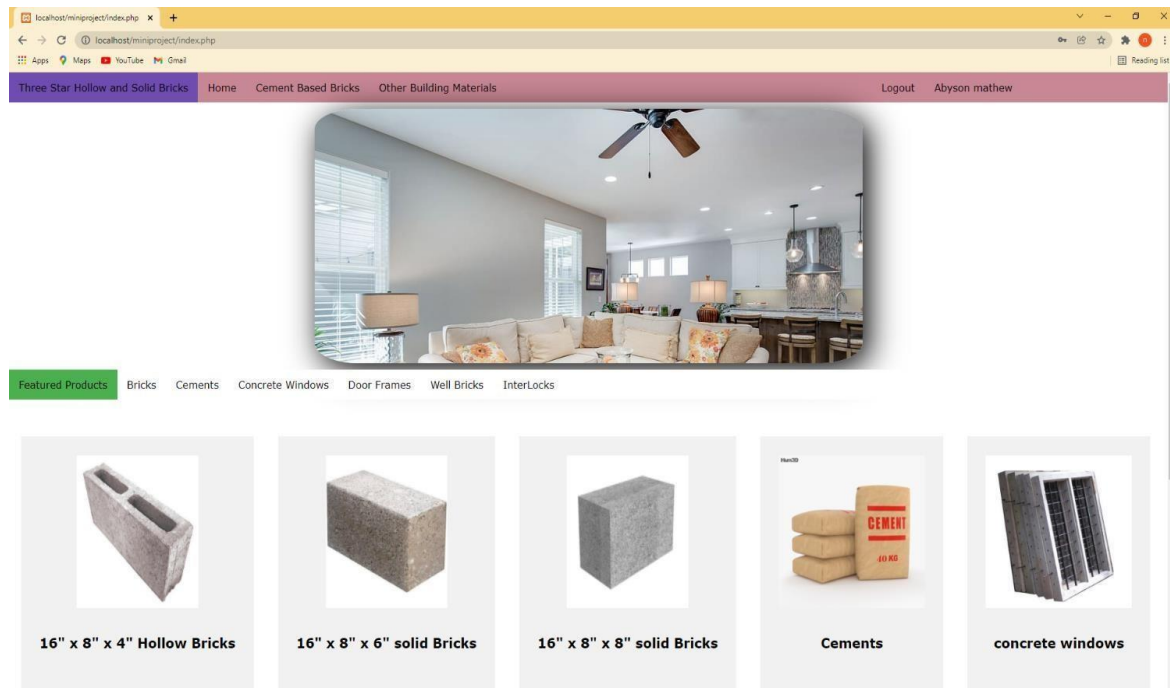
$sql = "UPDATE `registration` SET
```

```
`name`='$name',`phno`='$phone',`password`='$password' WHERE `id`='$uid';  
  
$res = mysqli_query($con, $sql); if ($res != 0) {  
  
header('location:profile.php');  
  
} else {  
  
echo "not inserted";  
  
}  
  
}  
  
?>
```

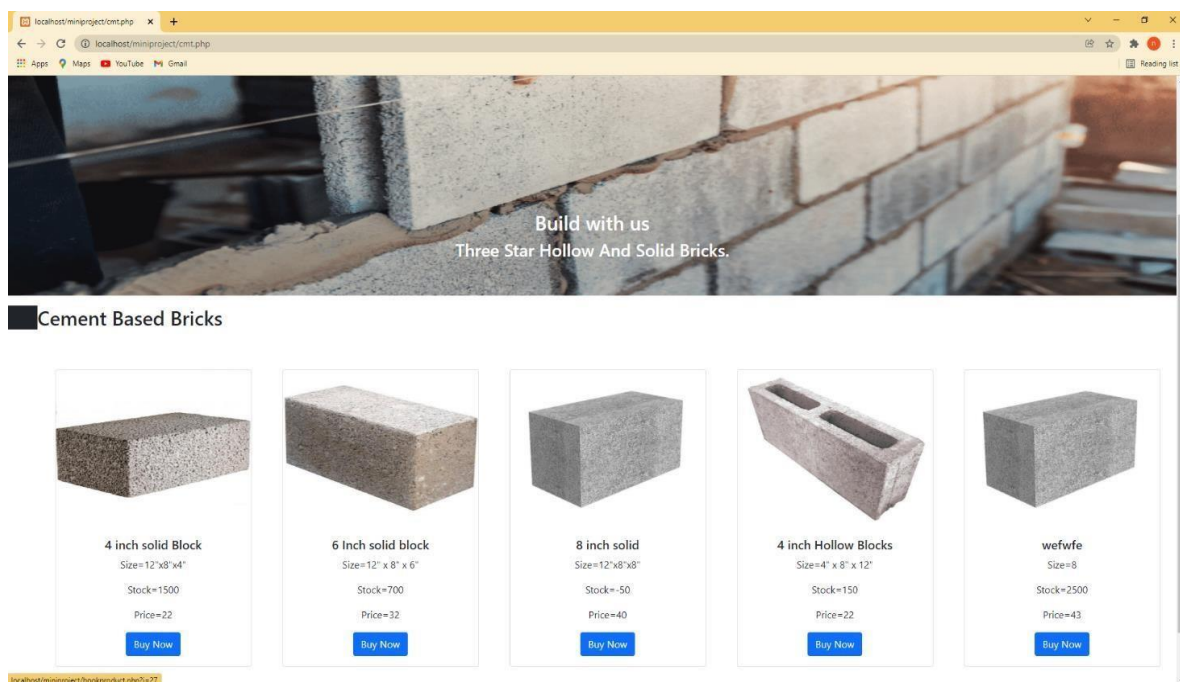
9.2 SCREENSHOTS

CUSTOMER PAGES

Home page



Product page



Booking page

The screenshot shows a web browser window with the URL `localhost/miniproject/bookproduct.php?i=25`. The page has a navigation bar with links: "Three Star Hollow and Solid Bricks", "Home", "Cement based bricks", "Other Building Materials", "Logout", and "Abyson mathew". The main content area features a 3D image of a grey brick. Below the image, the text reads "6 Inch solid block" and "Available Stock:700". A form for booking is located below the stock information, containing the following fields:

- Enter your name:
- Enter your delivery address:
- Select your date for delivery:
- Contact number:
- Enter the quantity of bricks:

Below the form is a button labeled "Proceed to Pay".

Payment Page

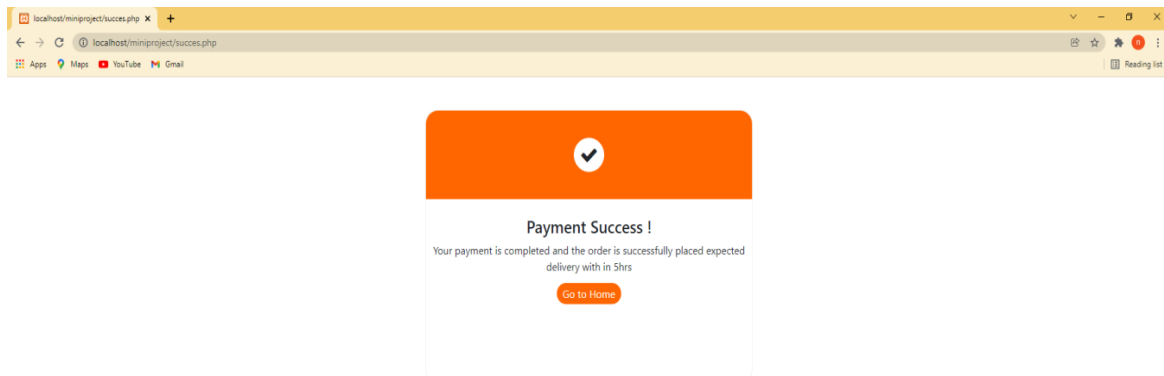
The screenshot shows a web browser window with the URL `localhost/miniproject/payment.php`. The page has a red background. A white box in the center contains the "Credit Card Checkout" form. The form is divided into two columns. The left column contains the following fields:

- Name on the card:
- Card Number:
- Expiry date:

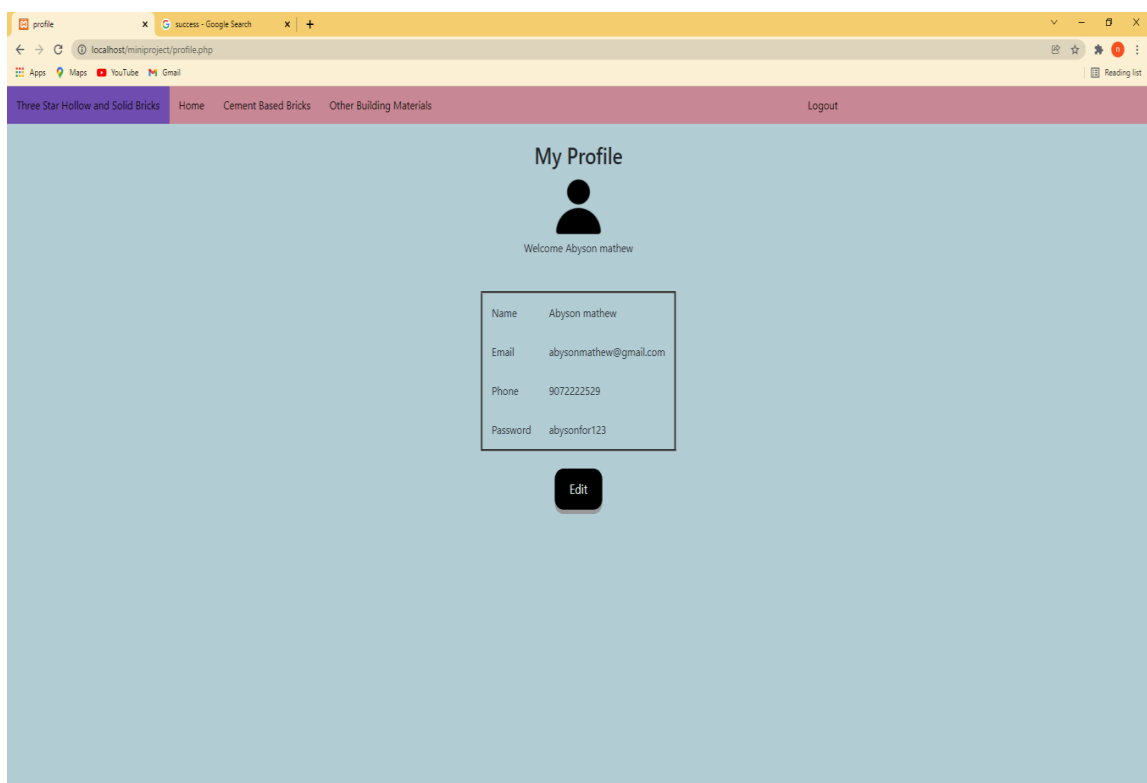
The right column contains the following information:

- Amount of bricks Ordered: 98
- Payment amount: ₹2548
- A red button labeled "PAY"

Success Page



Profile page



Add product page

Add products

Product Name	<input type="text"/>
Size	<input type="text"/>
Stock	<input type="text"/>
Price	<input type="text"/>
Image	<input type="button" value="Choose Files"/> No file chosen
Category	Cement based bricks ▾
<input type="button" value="upload"/>	
<input type="button" value="Back"/>	

copyright © owned by lijn@2021.all rights reserved

User details page

User_Details

NAME	EMAIL	PASSWORD	STATUS	Delete
frank	frank@gmail.com	frank123*	1	<input type="button" value="Delete"/>
akhil	akhil@gmail.com	akhil645433	1	<input type="button" value="Delete"/>
Ujini Santhosh	ljinsanthosh@gmail.com	ljjinfor123	1	<input type="button" value="Delete"/>
ebin johnson	ebin@gmail.com	ebinfor123	1	<input type="button" value="Delete"/>
Abyson mathew	abysonmathew@gmail.com	abysonfor123	1	<input type="button" value="Delete"/>
Amal joy	amaljoy@gmail.com	amalfor123	1	<input type="button" value="Delete"/>
Manas	manas@gmail.com	Qwerty@123	1	<input type="button" value="Delete"/>

Sign in page

The screenshot shows a web browser window with the URL `localhost/miniproject/login.php`. The page title is "Three Star Hollow and Solid Bricks". The main content area is split into two vertical panels. The left panel is white and contains the "Sign in" heading, a subtext "or use your account", and two input fields for "Email" and "Password". Below these fields is a link "or sign up" and a red "SUBMIT" button. The right panel is pink and contains the "Welcome Back!" heading, a subtext "Enter your Credentials", and a white "SIGN UP" button.

Sign up page

The screenshot shows a web browser window with the URL `localhost/miniproject/register.php`. The page title is "Three Star Hollow and Solid Bricks". The main content area is split into two vertical panels. The left panel is white and contains the "Create Account" heading, a subtext "or use your email for registration", and five input fields for "Name", "Email", "Phone", "Password", and "Confirm Password". Below these fields is a red "SUBMIT" button and a link "Sign Up". The right panel is pink and contains the "Hello, Friend!" heading, a subtext "Enter your personal details and start journey with us", and a white "SIGN IN" button.