UNIVERSITY OF MUMBAI

A Project Report on

ONLINE INTERIORS SHOPPING SYSTEM

SUBMITTED BY

MOHIT DEVESH VERNEKAR

SATISHKUMAR ARUNKUMAR VISHWAKARMA

ACADEMIC YEAR 2017-2018

Under the guidance of

Mrs. INDRANI SEN

Submitted in partial fulfilment of the requirements for qualifying

B.Sc.IT Semester-VI Examination April 2018

NIRMALA MEMORIAL FOUNDATION COLLEGE OF SCIENCE AND COMMERCE

90 Feet Road, Asha Nagar, Thakur Complex, Kandivali (E), Mumbai-400101

INDEX

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude to management of N.M.F College for giving me this opportunity to accomplish this project work.

I am very much thankful to Dr. T.P Madhu Nair, Principal of N.M.F College for his kind co-operation during the completion of my project.

A special vote of thanks to our HOD, Prof. S.B. Patil and our coordinator Prof. Vaishali Mishra for their cooperation and motivation and Prof. Indrani Sen, who is our professor & also our project guide for her most sincere, useful and encouraging contribution throughout the project span.

Finally, I would like to thank all my friends & entire information technology department who directly or indirectly helped me in completion of this project & to my family without whose support, motivation & encouragement this would not have been possible.

INTRODUCTION

An online interiors shop that allows users to check for various interior decors available at the online store and purchase online. The project consists of a list of interior decor products displayed in various models and designs. The users may browse through these products as per categories or search for products by applying various filters. If the users like a product, they may add it to their shopping cart. Once, the users wish to check out, they must register on the website first or they can also proceed as a guest. The users can then login using the same email id and password next time. In case, the users forget their login credentials, they can easily reset their password Users will be able to register themselves with their basic personals like name, address, email id, etc. After this, an OTP will be sent to the user's specified email id, which the users need to check and enter the OTP on the website for completing the registration process. Now, the users may pay through a credit card/debit card/e-wallet or by cash on delivery. Credit card/debit card details will be checked if they are valid or not. From e-wallet, the users can add money and pay money and its history will be viewed to the users. Once, the users make a successful transaction, they get a copy of the shopping receipt in their email id. Transaction history like the orders which they have placed previously, payment history (credit card/debit card details) will be shown to the users. The users can submit their feedback/suggestions about the overall experience with the website.

Here, we use HTML, JSP, Servlets and Bootstrap framework to make the frontend and SQL serves as a backend to store furniture lists, customer details and inventory data. Thus, the online interiors shopping project brings an entire interior designing shop online and makes it easy for both buyer and seller to make deals on interior decor.

SYNOPSIS

Topic: ONLINE INTERIORS SHOPPING SYSTEM

Front End: HTML, JSP, Servlets, Bootstrap

Back End: MYSQL

The website allows the user to view different templates and design for different rooms, select one of those designs and order for the same. User also has an option to enhance or customize the design by selecting products like curtain, furniture etc. User at last can make payment via different payment modes like COD, CC/DC.

Description of Modules

1. **Login Module**: User will be able to login using user id and password after they are finished with registration process.

2. **User Registration:**
   - Users will be able register themselves with their basic personal like name, address, email id, etc.
   - After this an OTP will be sent to the user's specified email id which user needs to check and enter the OTP on the website for completing registering process.

3. **Guest User Shopping:**
   - Guest User can view different products on the website but cannot place order for any products.

4. **Registered User Shopping:**
   - User can view different products on the website and place order for some products.
   - As the user is already registered in our system, he need not enter those details again.
   - They can view the history and transaction of previous orders and payment as well.
   - After the successful completion of the process, a confirmation email will be sent to the user's registered email id.

6. **Payment Facilities:**
   - Payment details like credit card /debit card/e-wallet details will be taken from the user.
   - Credit card/debit card details will be checked if they are valid or not.
   - From e-wallet user can add money and pay money and its history will be viewed to the user.

5. **Transaction History:**

- Transaction history like the orders which it has placed previously, payment history (credit card/debit card details) will be shown to the user.

6. Service Module:

- This module allows the user to place a service request for any complaints regarding computer or any peripherals.
- The admin will receive the request placed by the user and take necessary action based on the complaint of the user and notify the same to the user via email.

7. Admin:

- Admin can add products/designs, remove products/designs or alter products details.
- Admin can check the order details like how many orders are placed for which dates and placed by which users.
- Admin can also view the feedback and suggestions.
- Admin handles all the service request of different users.

8. Feedback/Suggestion:

- Feedback and suggestions will be taken from the users and submitted to admin.

9. Forgot Password module:

- This module is used by the admin or employee to reset the password.
- This is done by sending an OTP to the user's email id which needs to be entered in the application to reset the password.
- Maximum 3 attempts will be given to the user to enter the OTP.

10. Search/Filter module:

- This module can be used by admin or user to search for various products by applying various filters or orders and many some more

HARDWARE & SOFTWARE REQUIREMENT

Hardware Requirements:

- ✓ **Processor type:**
- Minimum: Intel Pentium3/ Pentium4
- Recommended: Core 2 Duo or higher.
- Monitor: VGA Monitor
- ✓ **RAM:**
- Minimum: 512 MB
- Recommended: 1 GB or more
- ✓ **Processor speed:** 1 GHz or higher
- ✓ **Hard disk*:*** 40 GB or more

Software Requirements:

- ✓ **Operating system:** Microsoft Windows 2000/XP/Vista/7/8/10
- ✓ **Front End Tool:** Java Technology with Servlet
- ✓ **Back End Tool:** MY SQL Server 2008
- ✓ **Web Browser:** Google Chrome, Mozilla Firefox, etc.

SCOPE

- Helps in reducing the file work.
- A lot of time will be saved.
- Database is maintained so properly so that managing and keeping records is very easy.
- System is fully integrated.
- Quicker access of information to the administrator.
- All types of information are available at anywhere and at any time as the system is now online.
- As it is now on the internet, it will surely increase the growth of the business and profits will automatically increase.
- Ultimately, customers are the center points in this system.
- Booking system is very much effective and maintains only one data so that there is no duplication of data.

OBJECTIVES

- The objective of this project is to make a website to allow the customers to purchase items from an existing shop.
- To build such a website, complete web support need to be provided.
- A complete and efficient website which can provide the online shopping experience is the basic objective of this project.
- The goal of this project is to reduce the work load with the increased efficiency and to speed up the activities of interiors decor retail shop by bringing it online.
- The project is developed to cope up with the current issues and problems of online shopping by making improvements in control and performance.
- It is aimed to save cost and time by computerizing the system and reducing the overall human force required.

TECHNOLOGIES

Software Development Environment	**CLR (Java Runtime Environment)**
Presentation Layer	**HTML, CSS, Bootstrap**
Logic Layer	**JSP, Servlet**
Back-End	**MY SQL 2008**
Server	**Glassfish Server**

FRONT END: -

JSP

Java Server Page (JSP) is a technology for controlling the content or appearance of web pages using servlets, small programs that are specified in the web page and run on the web server to modify the web page before it is sent to the user who requested it. Sun Microsystems, the developer of Java, also refers to the JSP technology as the servlet application program interface (API). JSP is comparable to Microsoft's Active Server Page (ASP) technology. Whereas a Java Server Page calls a Java program that is executed by the web server, an Active Server Page contains a script that is interpreted by a script interpreter (such as a VBScript or JScript) before the page is sent to the user.

Servlets

A Java servlet processes or stores a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol. Thus, "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL rewriting.

To deploy and run a servlet, a web container must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container

is responsible for managing the lifecycle of servlets, mapping a URL to a servlet and ensuring that the URL requester has the correct access rights.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of the web container and a servlet.

JAVA

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plugin for applets).

The latest version is Java 10, released on March 20, 2018, which follows Java 9 after only six months in line with the new release schedule. Java 8 is still supported but there will be no more security updates for Java 9. Versions earlier than Java 8 are supported by companies on a commercial basis.

BACK END: -

MySQL

- MySQL is an open-source relational database management system(RDBMS).
- Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.
- The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.
- MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.
- For proprietary use, several paid editions are available, and offer additional functionality.
- MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks).
- LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python".

- Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines
  Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, largescale websites, including Google, Facebook, Twitter, Flickr, and YouTube.

PROBLEM DEFINITION

DRAWBACKS OF PRESENT EXISTING SYSTEM

1. Managing and keeping records of all Customers, Wish-list are not included, and they keep some record manually.
2. Business is not on the web which was biggest Disadvantages.
3. Customer can Order the Product by visiting to the office or to the nearest branch which was a very tedious job.
4. Not good communication between the different branches so there is lacking coordination between them.
5. Customer record is maintained properly.

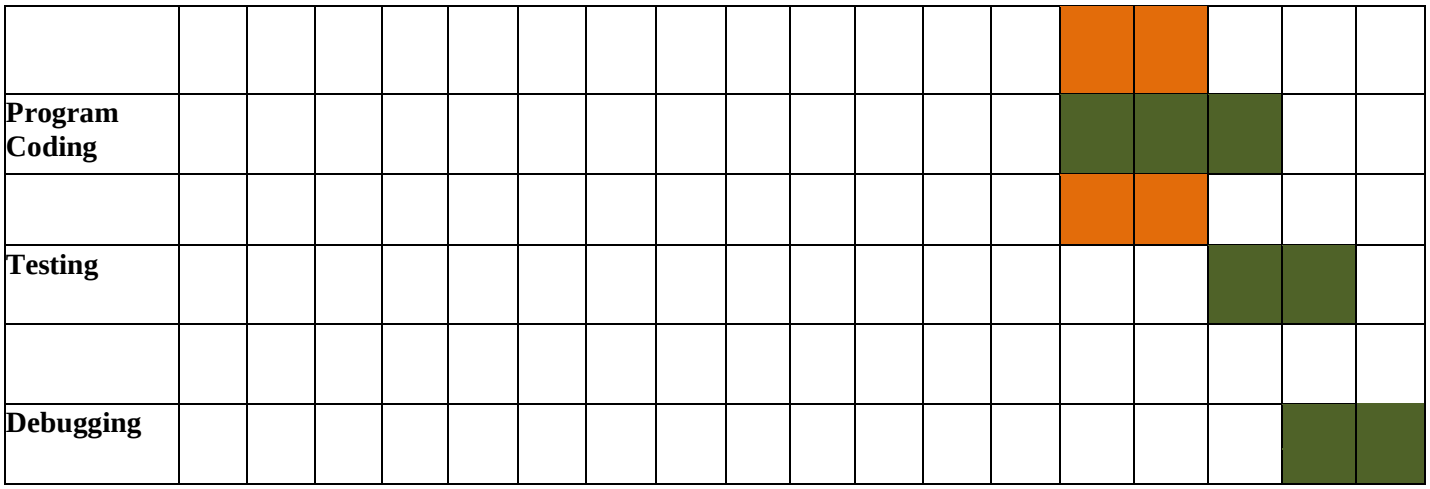
ADVANTAGE OF PROPOSED SYSTEM

1. Time saving.
2. Easy access to information whenever user requires it.
3. Less paper works.
4. Reduction in maintenance cost.
5. Report generation.
6. Security as per the authentication policies is provided.
7. The interface is user friendly.

PLANNING

GANTT CHART

WEEKS TASKS	01-DEC	4-DEC	12-DEC	18-DEC	28-DEC	05-JAN	12-JAN	17-JAN	04-FEB	12-FEB	19-FEB	23-FEB	28-FEB	4-MAR	9-MAR	20-MAR	27-MAR	3-APR
Planning Phase																		
Defining Problem																		
Data Gathering																		
Analysis of data																		
System Design																		
Rework on System Design																		
Database Design																		
Rework on Database design																		

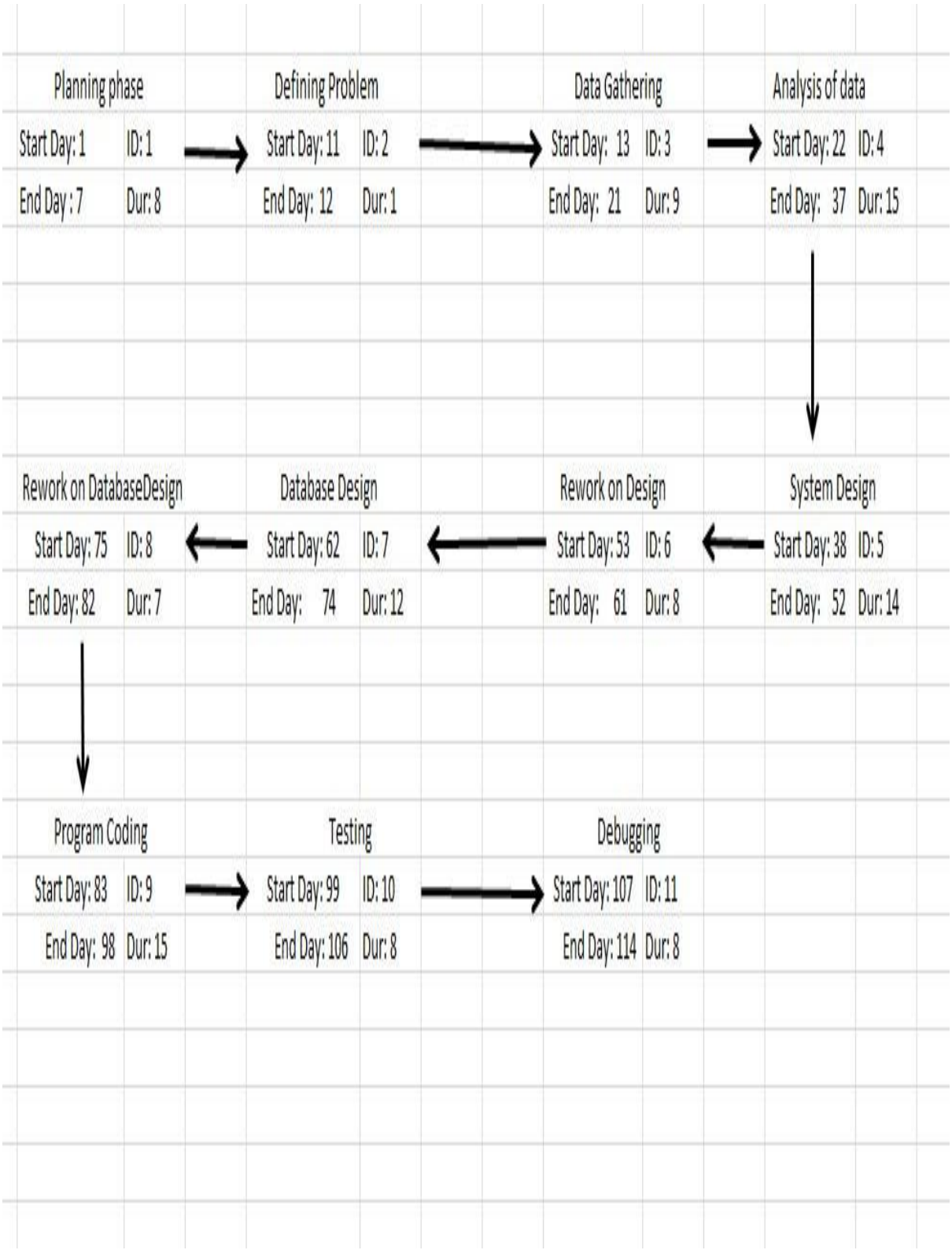


PERT CHART

- The program (or project) evaluation and review technique, commonly abbreviated as PERT, is a model for project management designed to analyze and represent the tasks involved in completing a given project.
- PERT is a method to analyze the involved tasks in completing a given project especially the time needed to complete each task, and identifying the minimum time needed to complete the total project.
- A PERT chart is a project management tool used to schedule organize and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique, is a methodology developed by the US Navy in the 1950s to manage the Polaris submarine missile program. A similar methodology the Critical Path Method (CPM) was developed for project management in the private sector at about the same time.
- A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events or marks in the project. The direction of the arrows on the lines indicates the sequence of tasks**.**

More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system.

Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an outgoing effort that improves in accuracy at each phase of the system life cycle.


Planning phase
Start Day: 1 ID: 1
End Day : 7 Dur: 8
Defining Problem
Start Day: 11 ID: 2
End Day: 12 Dur: 1
Data Gathering
Start Day: 13 ID: 3
End Day: 21 Dur: 9
Analysis of data
Start Day: 22 ID: 4
End Day: 37 Dur: 15
System Design
Start Day: 38 ID: 5
End Day: 52 Dur: 14
Rework on Design
Start Day: 53 ID: 6
End Day: 61 Dur: 8
Database Design
Start Day: 62 ID: 7
End Day: 74 Dur: 12
Rework on DatabaseDesign
Start Day: 75 ID: 8
End Day: 82 Dur: 7
Program Coding
Start Day: 83 ID: 9
End Day: 98 Dur: 15
Testing
Start Day: 99 ID: 10
End Day: 106 Dur: 8
Debugging
Start Day: 107 ID: 11
End Day: 114 Dur: 8

COST & BENEFIT ANALYSIS

Cost Benefit Analysis

Cost-Benefit Analysis (CBA) estimates and totals up the equivalent money value of the benefits and costs to the community of projects to establish whether they are worthwhile. It is important to identify cost and benefit factors. Therefore, a cost-based study was done on Book Hive to identify the factors. Cost and benefits were categorized as follows: -

Development costs, operating costs and cost benefit analysis is done in three steps:

1. The first is to estimate the anticipated development & operational cost. Development costs are those that are incurred during the development of system. Operational costs are those that are anticipated financial benefits.

2. Financial benefits are the expected annual savings or increase in revenue derived from installation of the system.
3. Cost benefits analysis calculated based on the detailed estimates of cost & benefits.

The cost of project management is estimated by using the COCOMO model for estimation which is based on size estimation. The most fundamental calculation in the COCOMO model is the use of the Effort Equation to estimate the number of persons and the months required to develop the project.

COCOMO MODEL 1

COCOMO'81 models depend on the two main equations.

1. Development effort: MM = a * $KDSI^b$ based on MM - man-month / person month / staff-month is one month of effort by one person. In COCOMO'81, there are 152 hours per person month. According to organizations, these values may differ from the standards by 10% to 20%.
2. Effort and development time (TDEV): TDEV = 2.5 * MM^c.

The coefficients a, b and c depend on the mode of the development. There are three modes of development:

Development Mode	Project Characteristics			
	Size	Innovation	Deadline/Constraint	Dev. Environment
Organic	Small	Little	Not light	Stable
Semi-Detached	Medium	Medium	Medium	Medium
Embedded	Large	Greater	Tight	Complex hardware

BASIC COCOMO

The basic COCOMO applies the parameterized equation without much detailed consideration of project characteristics.

Basic COCOMO	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

SM = a * $KDSI^b$.

$$TDEV = c * SM^{d}$$

$$DSI = 4000$$

$$KDSI = DSI/1000$$

$$= 4000/1000$$

$$= 4$$

$$SM = 2.5 * 4^{1.05}.$$

$$= 10.28$$

$$TDEV = 2.5 * 10.28^{0.38}$$

$$= 6.06$$

$$Average\ Staff = SM / TDEV$$

$$= 10.28 / 6.06$$

$$= 1.69$$

$$= 2\ FSP$$

Methodology Adopted (SDLC Model)

Steps for software development: -

Software Development Life Cycle (SDLC) is the overall process of developing information systems through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance. There are many different models:

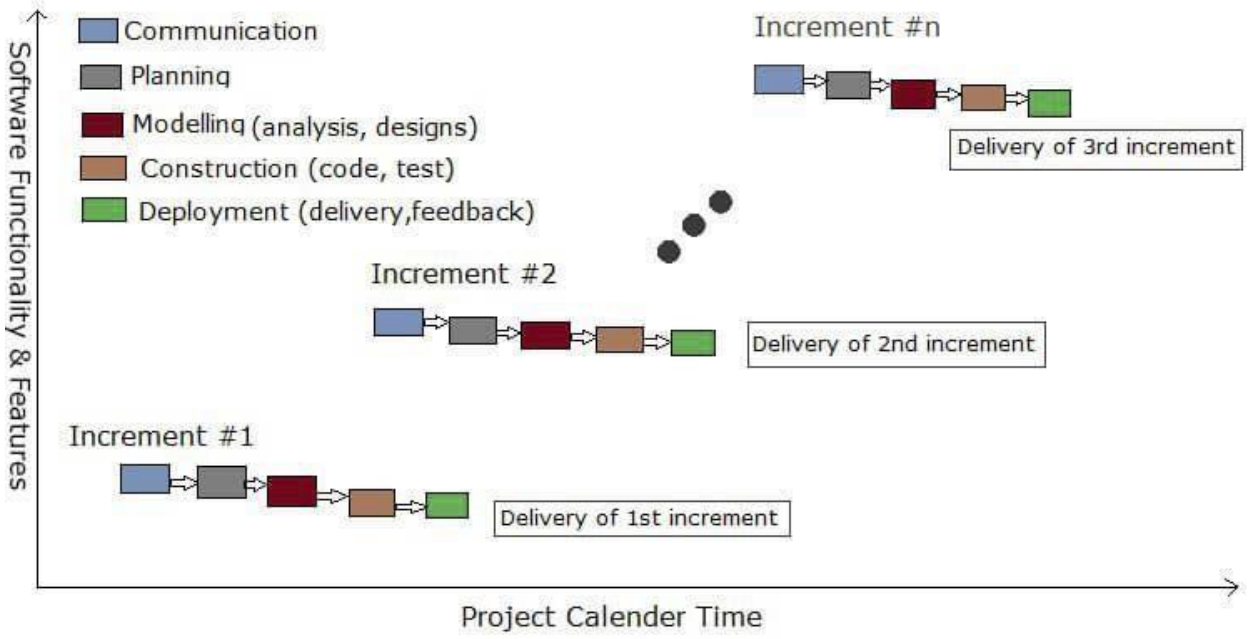
1. Waterfall Model
2. Spiral Model
3. Rapid Prototype Model
4. Incremental Model

In software engineering, the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system.

Waterfall model: Model used for proposed system.

INCREMENTAL MODEL: -

Incremental model in software engineering is a one which combines the elements of waterfall model in an iterative manner. It delivers a series of releases called increments which provide progressively more functionality for the client as each increment is delivered. In the incremental model of software engineering, waterfall model is repeatedly applied in each increment. The incremental model applies linear sequences in a required pattern as calendar time passes. Each linear sequence produces an increment in the work.



As from the diagram, you can see that there are 5 phases (tasks) which are carried out in each increment.
The first increment is often a core product where the necessary requirements are addressed and the extra features are added in the next increments. The core product is used and evaluated by the client. Once, the customer assesses the core product, there is a planned development for the next increment. Thus, in every increment, the needs of the client are kept in mind and more features and functions are added and the core product is updated. This process continues till the complete product is produced.

- Initial product delivery is faster.
- Lower initial delivery cost.

- Core product is developed first i.e. main functionality is added in the first increment.
- After each iteration, regression testing should be conducted. During this testing, faulty elements of the software can be quickly identified because few changes are made within any single iteration.

- It is generally easier to test and debug than other methods of software development because relatively smaller changes are made during each iteration. This allows for more targeted and rigorous testing of each element within the overall product.
- With each release a new feature is added to the product.
- Customer can respond to feature and review the product. Risk of changing requirement is reduced.
- Work load is less.

SYSTEM ANALYSIS & DESIGN

Feasibility Study

Operational Feasibility

- Operational feasibility is to gain an understanding of whether the proposed system will likely solve the business problems or take

advantage of the opportunities or not. It is important to understand how the new system will fit the current day-today operations of the organization.

- We have also conducted an operational feasibility of our system to identify whether our system is able to satisfy the goal of developing the system.
- Here the main goal is not business growth but to make things a little easier for the students.

Technical Feasibility

- Assessing technical feasibility is to evaluate whether the new system will perform adequately and whether an organization has ability to construct a proposed system or not. The technical assessment help answer the questions such as whether the technology needed for the system exists, how difficult it will be to build and whether the firm has enough experience using that technology.
- In case of technical feasibility, we have made a system that match accord to the requirements of the buyers and sellers.
- We have made the project in a language in which we are familiar. The entire project and website is made in Java and MySQL. We have chosen these to as the project is small and, we are familiar with the technologies.

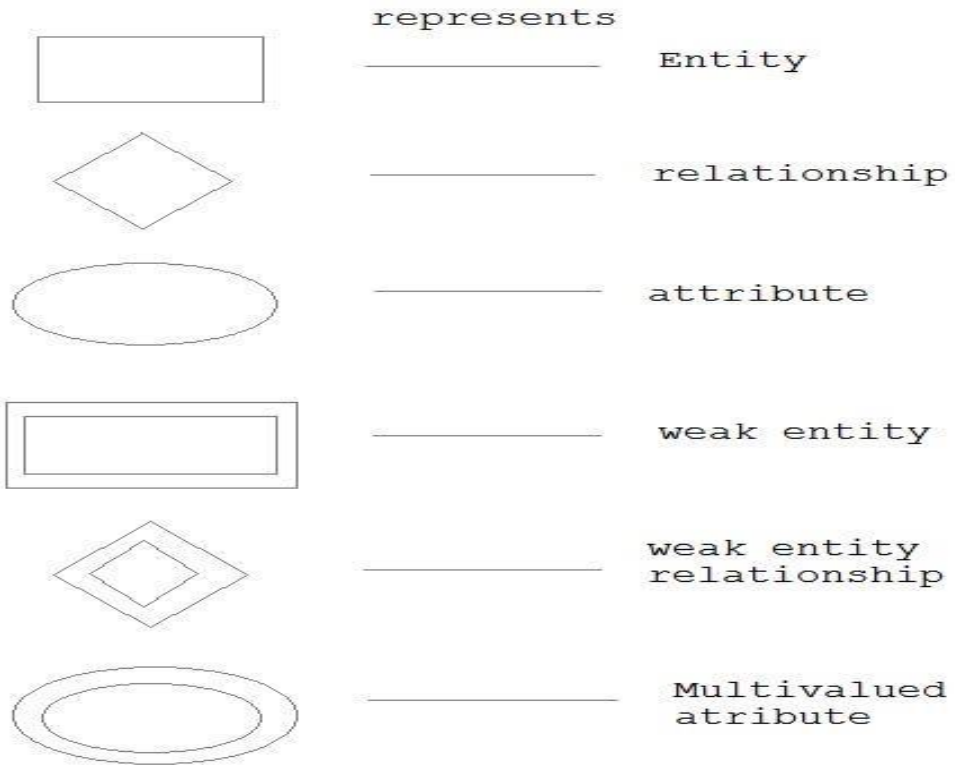
Economic Feasibility

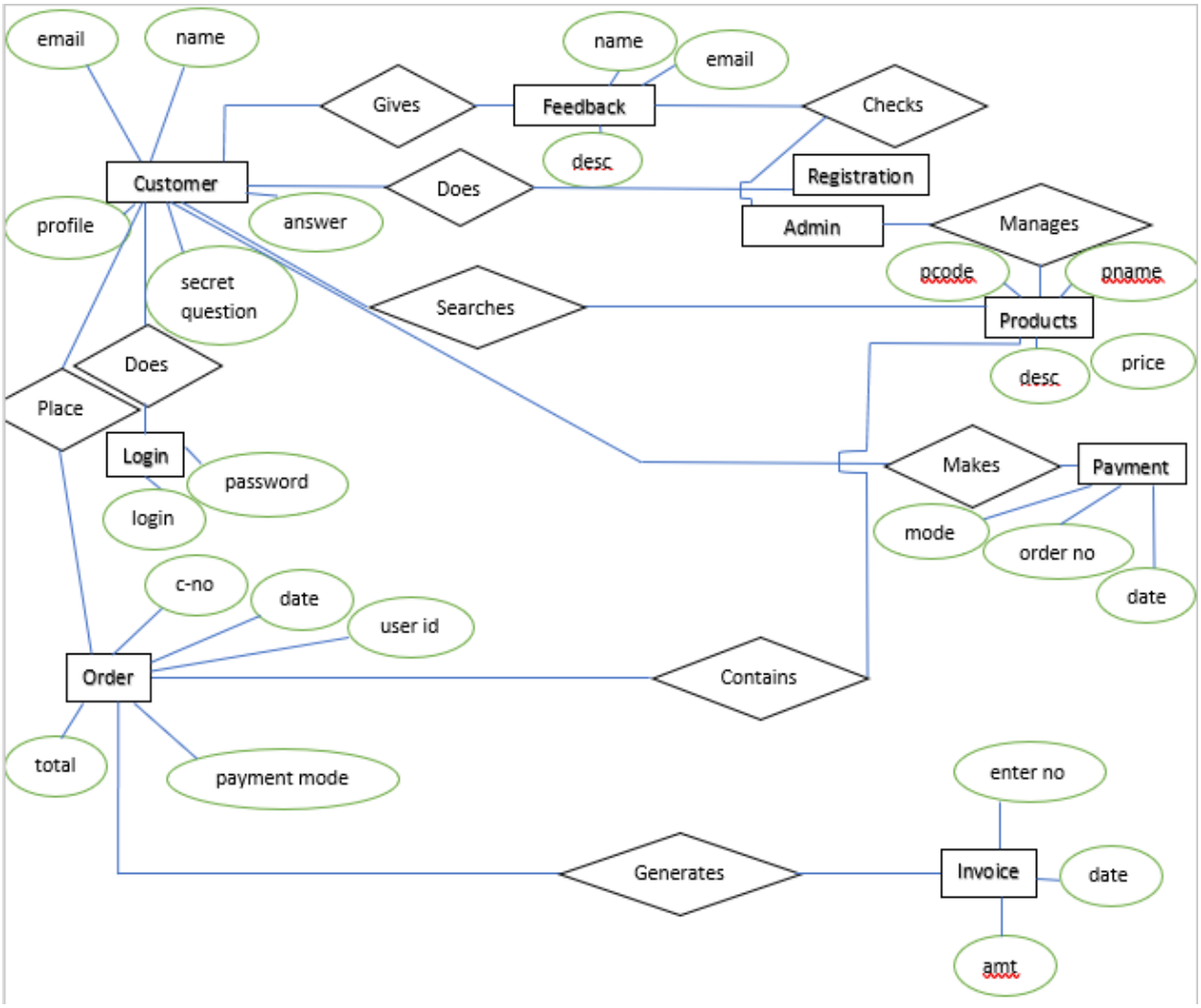
- Economic evaluation is a vital part of investment appraisal, dealing with factors that can be quantified, measured and compared in monetary terms (Chen 1996). The results of an economic evaluation are considered with other aspects to make the project investment decision as the proper investment appraisal helps to ensure that the right project is undertaken in a manner that gives it the best chances of success.
- The system according to us is cost effective as this a small group which is containing little members and only two members are working on this project.

ER Diagram

The Entity Relationship Diagram (Model) is based on perception of a real world that consists of a collection of basic objects called as entities and relationships amongst these objects. Entities in a database are described by a set of attributes.

- A relationship is an association among several entities.
- The set of entities of the same type are called as an entity set.


represents
Entity
relationship
attribute
weak entity
weak entity
relationship
Multivalued
atribute


email
name
Gives
name
email
Feedback
Checks
desc
Customer
Does
Registration
profile
answer
Admin
Manages
pcode
pname
secret question
Searches
Products
Does
price
desc
Place
Login
password
Makes
Payment
login
mode
order no
c-no
date
date
user id
Order
Contains
total
payment mode
enter no
Generates
Invoice
date
amt

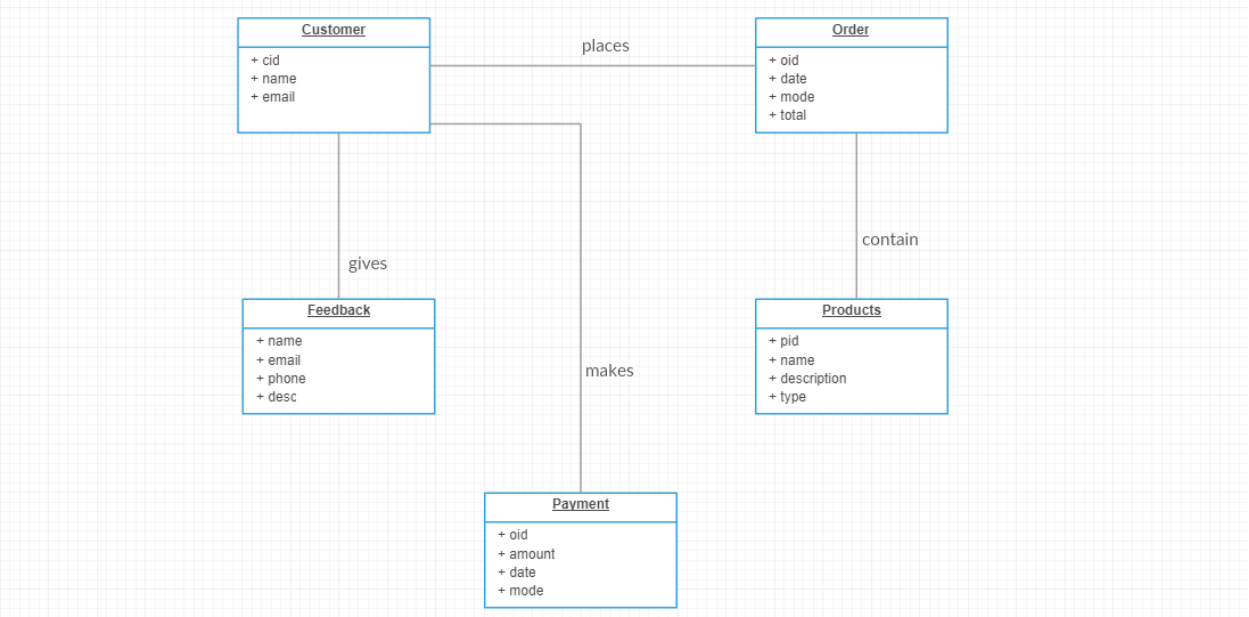
CLASS DIAGRAM

- A class diagram is an illustration that shows the relationship between systems and describes different types of objects.
- The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

- The format of the class is as follows: -

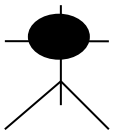
Operation
Name
Attribute

- The first section is name where we define the name of the class. In attribute section we define the attribute with data type. In operation section we define the operations performed by the system.



USE CASE DIAGRAM

- A Use Case represents the sequence of actions that the systems perform to produce something of value to the actor interacting with the system.
- An actor may participate in more than one use case and, conversely, more than one actor may participate in the same use case.
- The Use Case Diagram is a graphical depiction with an accompanying textual description of use cases and the actor that participate in them.
- Use Case Diagrams model the functionality of system by using Actors and Use Cases.
- ACTOR :-

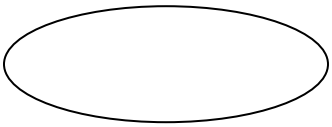
User 

An actor is always outside the automation system boundary of the system. An actor can play several roles.

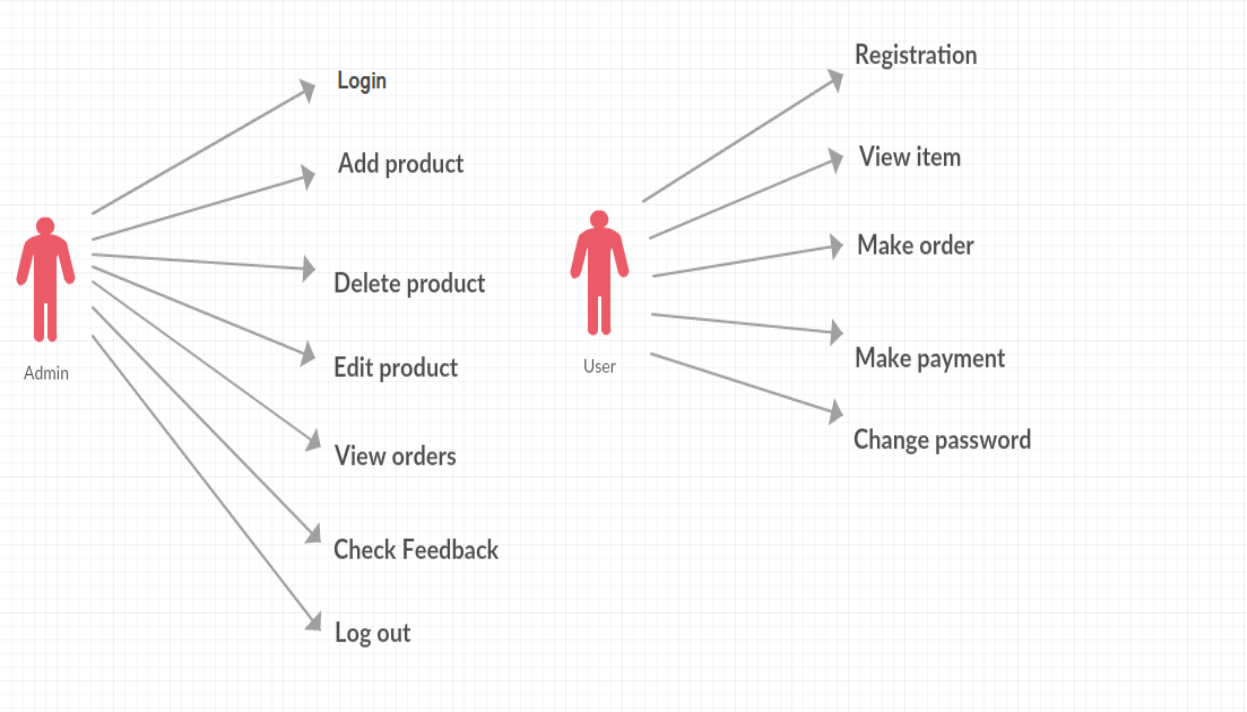
- LINES :-

This indicates that which actor participates with which case.

- USE CASES:



Use Cases are services or functions provided by the system to its users. Each use case describes one logical interaction between the Actor and the system.

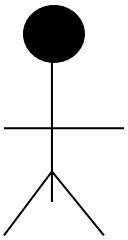

Admin
Login
Add product
Delete product
Edit product
View orders
Check Feedback
Log out
User
Registration
View item
Make order
Make payment
Change password

SEQUENCE DIAGRAM

1. Shows sequence of interactions between objects and flow of events in a single use case.
2. Focuses on message details.
3. Used more frequently in industry.

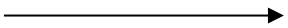
SSD Notation

Actor represented by stick figure person (or role) that "Interacts" with system by entering input data and receiving output data. Object notation is rectangle with name of object underline shows individual object and not class of all similar objects. Lifeline is vertical line under object or actor to show passage of time for object. Messages use arrows to show messages sent or recovered by actor or system.



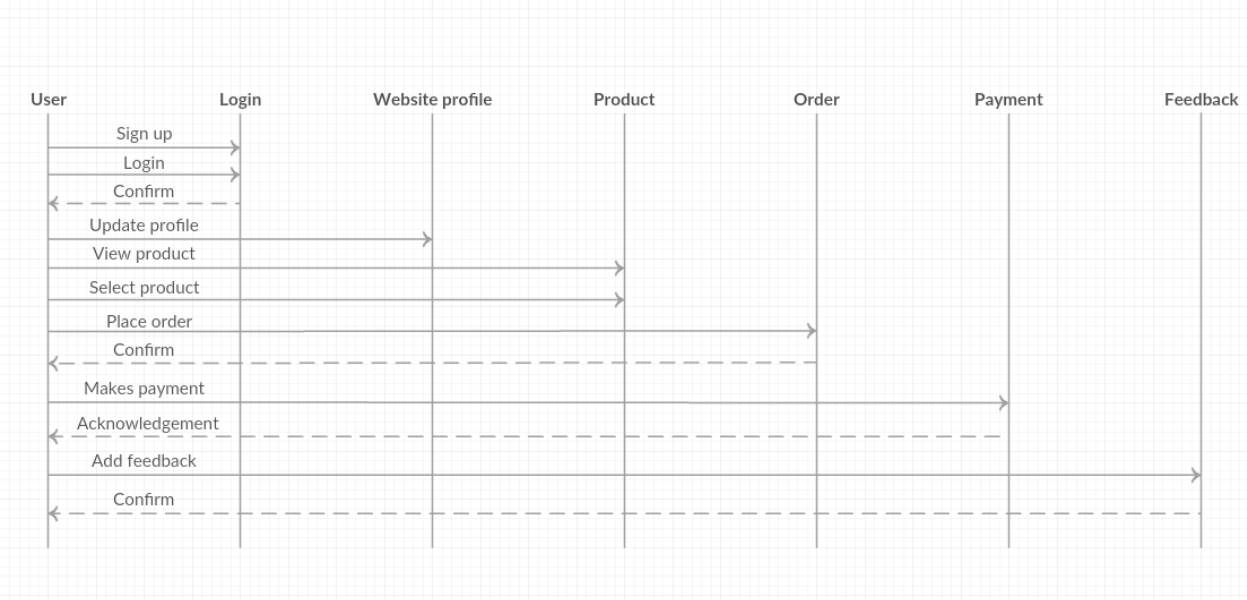
ACTOR SYMBOL

OBJECT SYMBOL

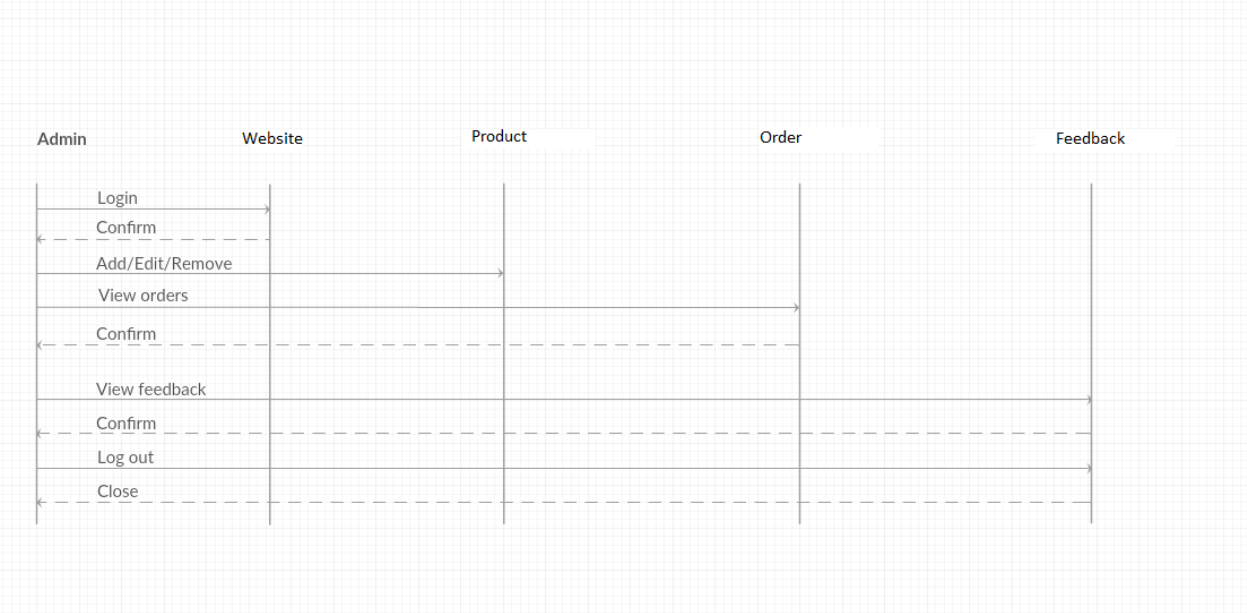


LIFELINE SYMBOL

CUSTOMER



ADMIN

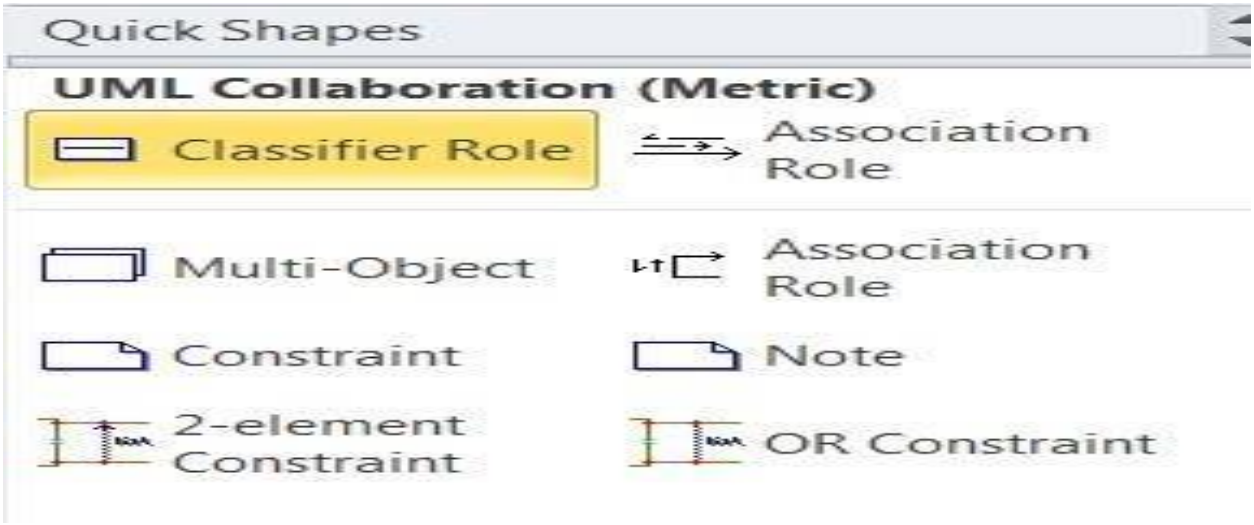


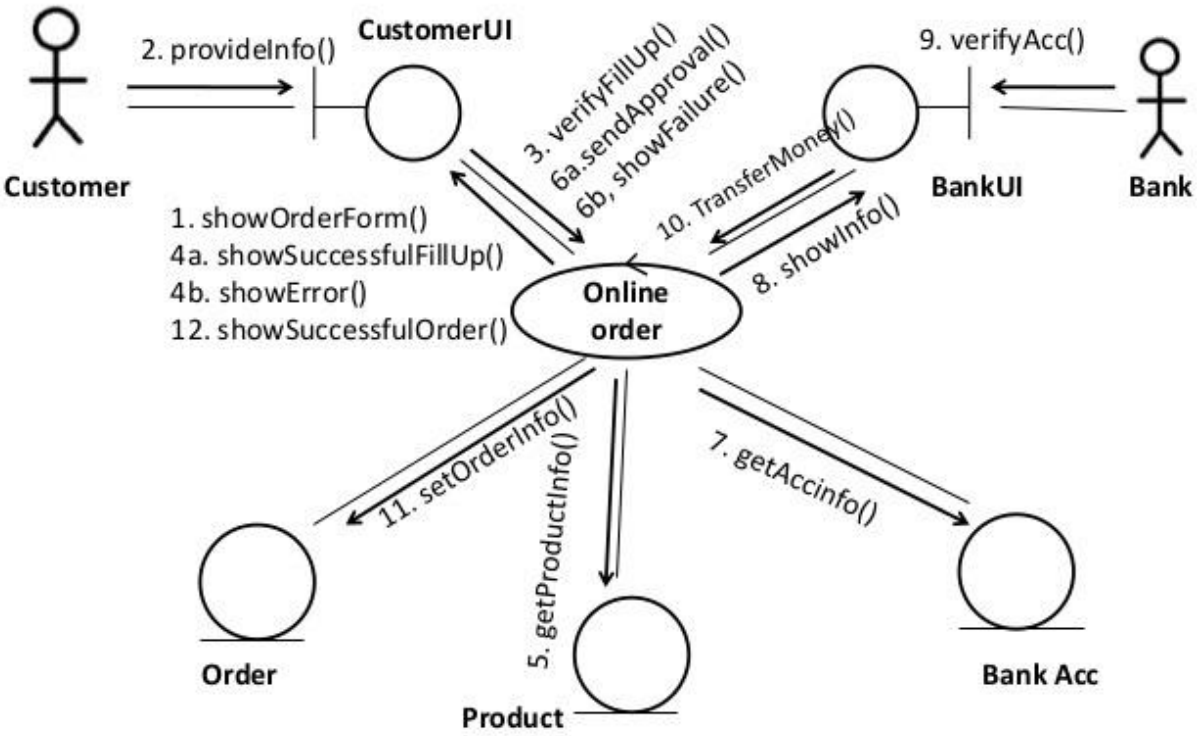
COLLABORATION DIAGRAM

A collaboration diagram is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this collaboration. A collaboration diagram often comes in the form of a visual chart that represents a flow chart.

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling

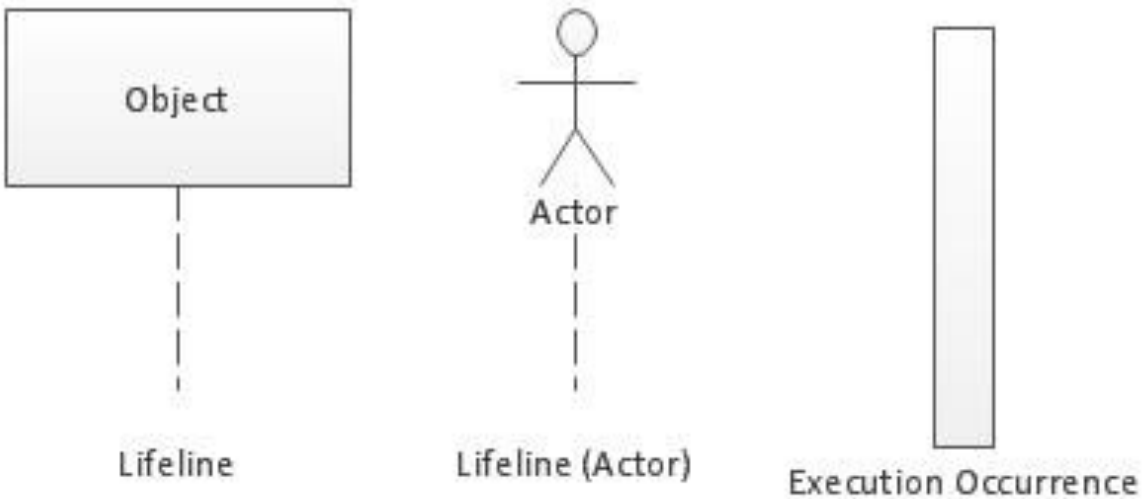
Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

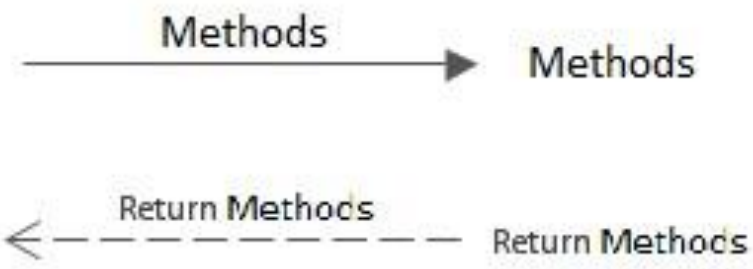


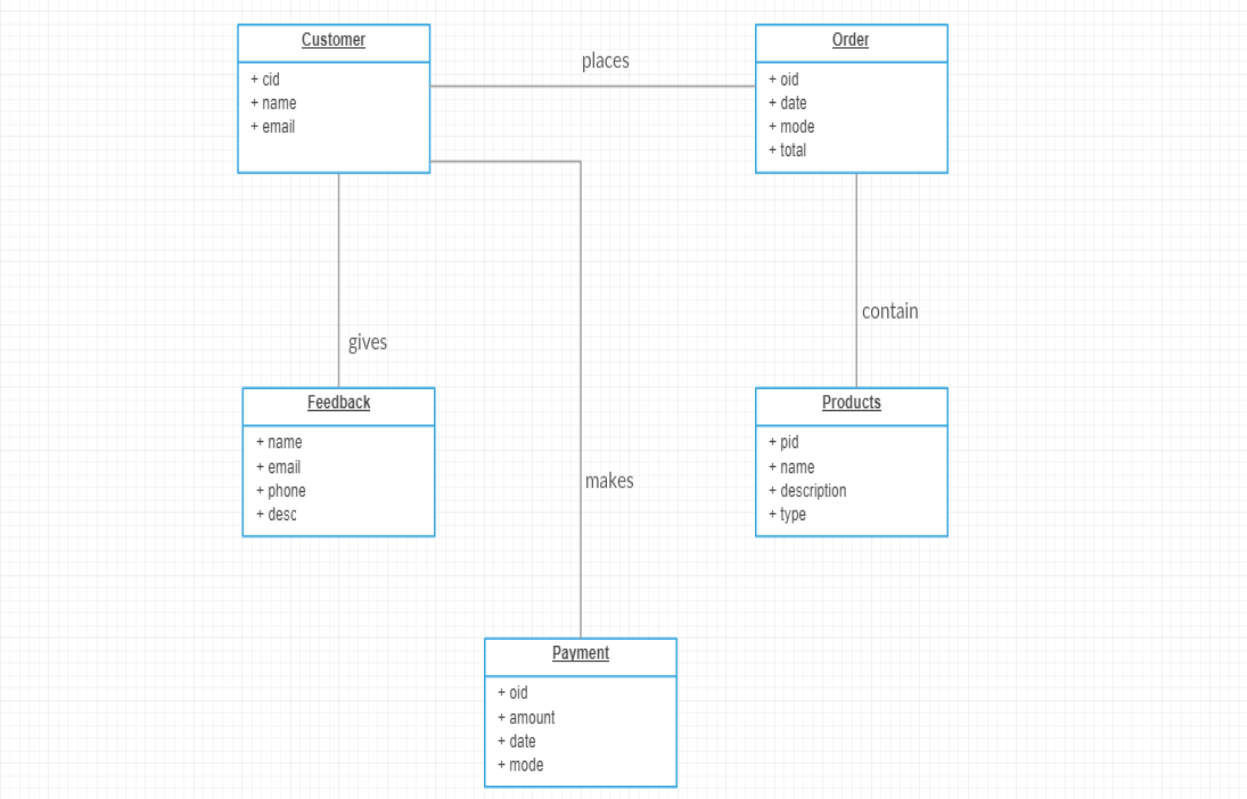

Customer
2. provideInfo()
CustomerUI
3. verifyFillUp()
6a.sendApproval()
6b, showFailure()
9. verifyAcc()
BankUI
Bank
10. TransferMoney()
8. showInfo()
1. showOrderForm()
4a. showSuccessfulFillUp()
4b. showError()
12. showSuccessfulOrder()
Online order
11. setOrderInfo()
5. getProductInfo()
7. getAccinfo()
Order
Product
Bank Acc

OBJECT DIAGRAM

An **object diagram** in the Unified Modelling Language (UML), is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time. In the Unified Modelling Language (UML), an object diagram focuses on some particular set of objects and attributes, and the links between these instances. A correlated set of object diagrams provides insight into how an arbitrary view of a system is expected to evolve over time. Object diagrams and class diagrams are closely related and use almost identical notation.






Customer
+ cid
+ name
+ email
places
Order
+ oid
+ date
+ mode
+ total
gives
contain
Feedback
+ name
+ email
+ phone
+ desc
makes
Products
+ pid
+ name
+ description
+ type
Payment
+ oid
+ amount
+ date
+ mode

.

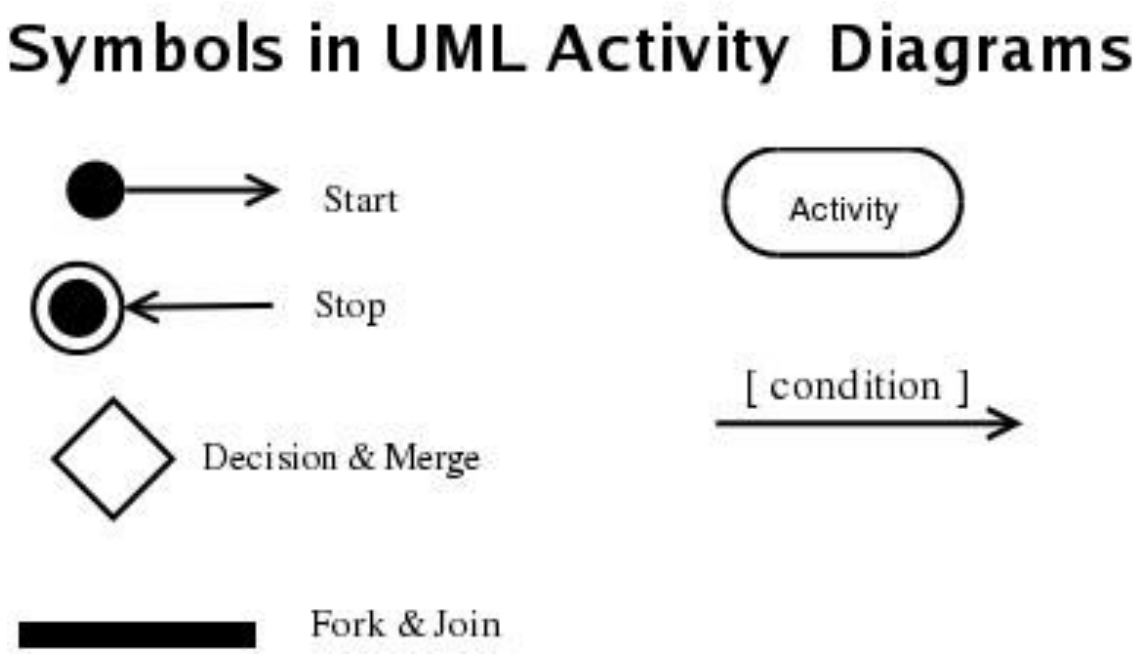
ACTIVITY DIAGRAM

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. Before drawing an activity diagram, we should identify the following elements: Activities

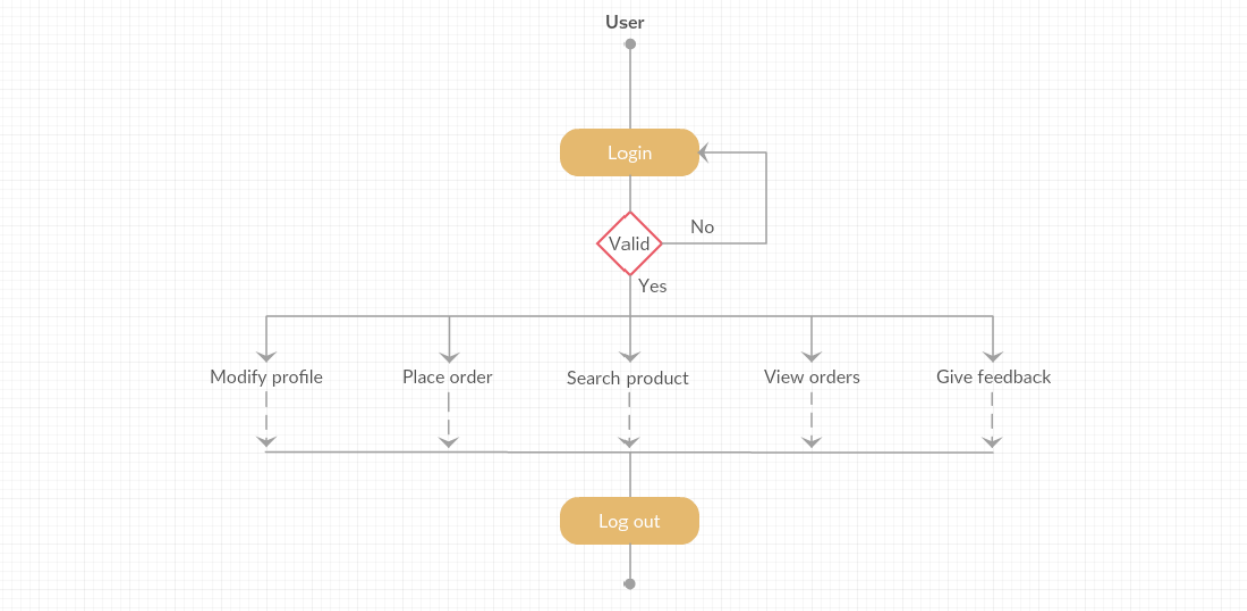
- Association
- Conditions
- Constraints

The most important shape types:

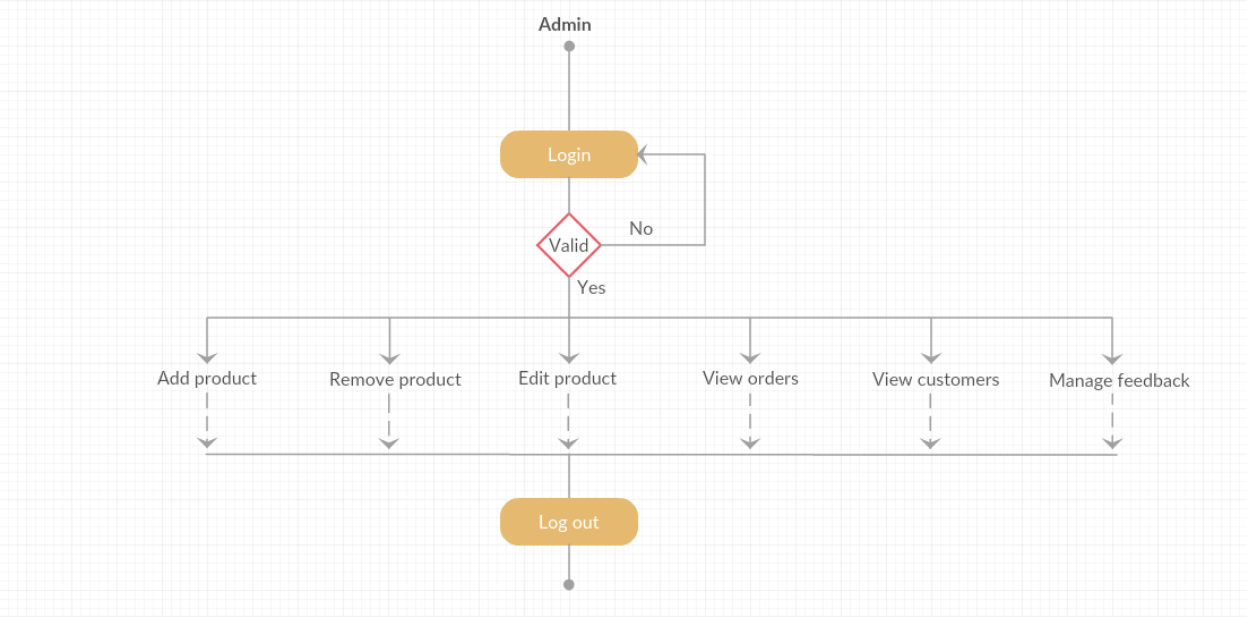
- Rounded rectangles represent activities;
- Diamonds represent decisions;
- Bars represent the start (split) or end (join) of concurrent activities; A black circle represents the start (initial state) of the workflow; An encircled black circle represents the end (final state).



CUSTOMER


User
Login
Valid
No
Yes
Modify profile
Place order
Search product
View orders
Give feedback
Log out

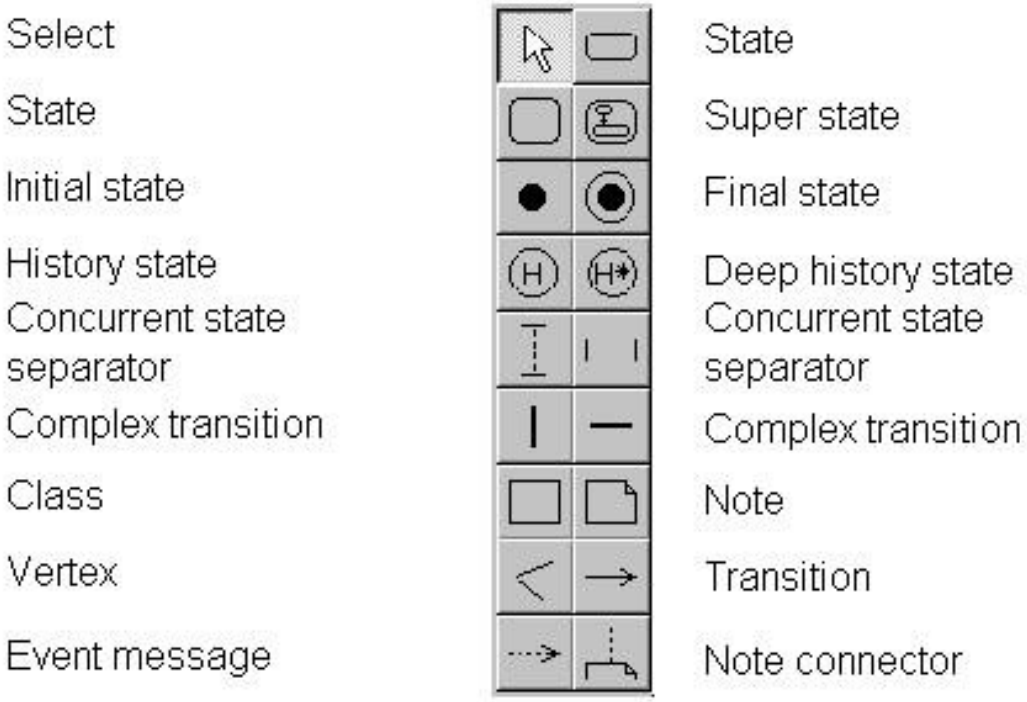
ADMIN

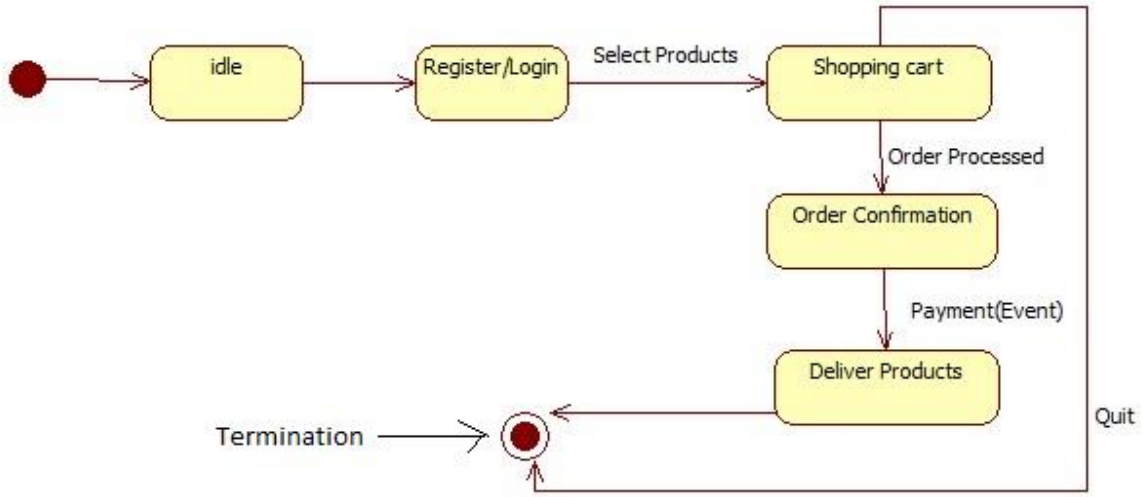

Admin
Login
Valid
No
Yes
Add product
Remove product
Edit product
View orders
View customers
Manage feedback
Log out

STATE CHART

- This diagram shows the life of an object in states and transitions.
- State diagram describes the dynamic behavior of a system in response to external stimuli.
- It helps to model dynamic behavior of objects based on states.
- It also helps to model reactive objects whose states are triggered by specific events.
- It helps to describe passive objects which go through several distinct phases during their lifetime.

Notations:




idle
Register/Login
Select Products
Shopping cart
Order Processed
Order Confirmation
Payment(Event)
Deliver Products
Termination
Quit

COMPONENT DIAGRAM

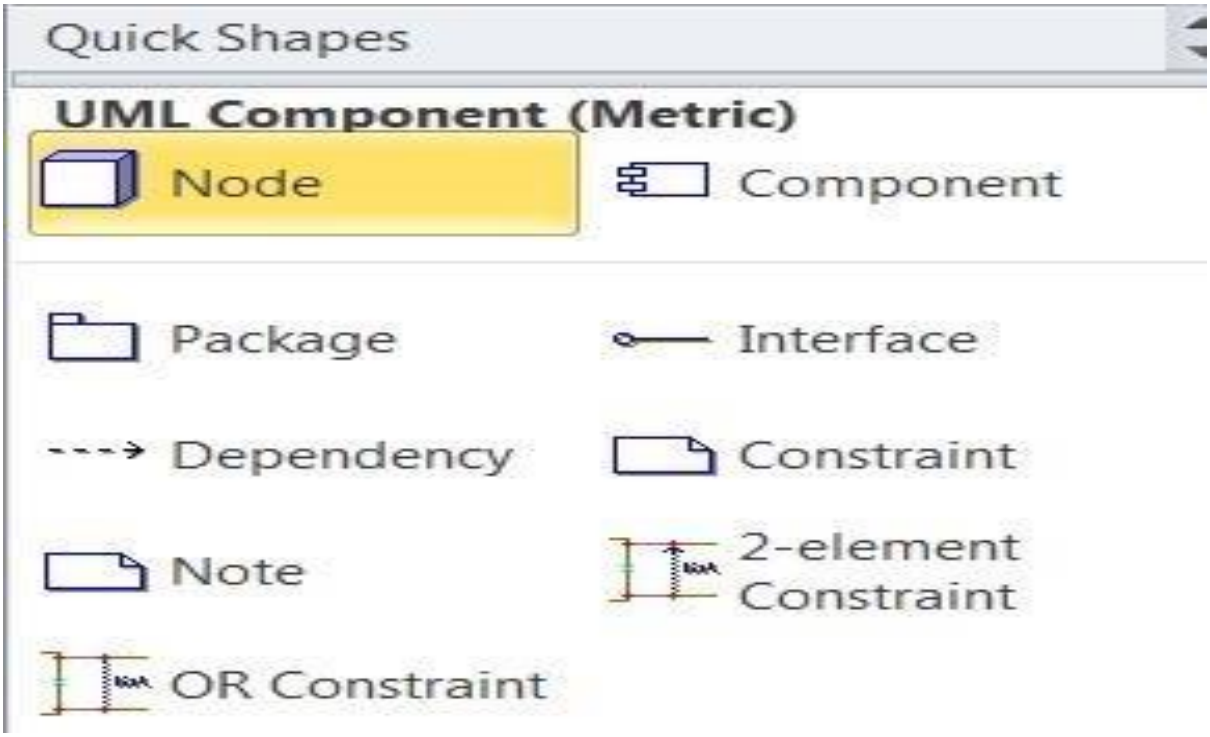
Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

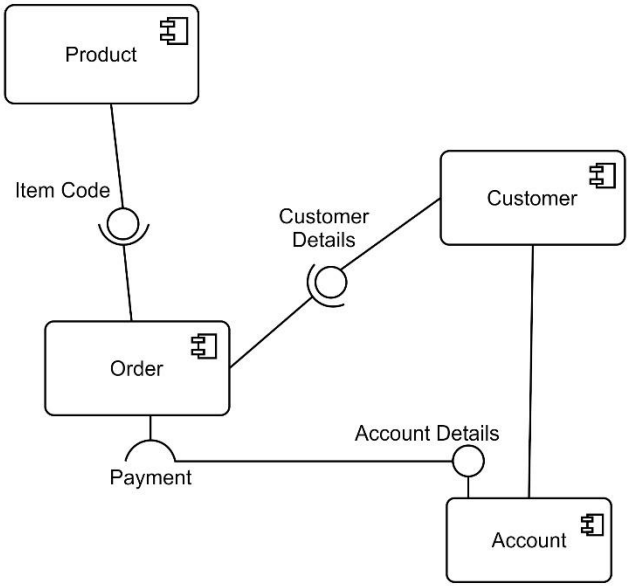
The purpose is different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

The purpose of the component diagram can be summarized as:

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.

Describe the organization and relationships of the components.



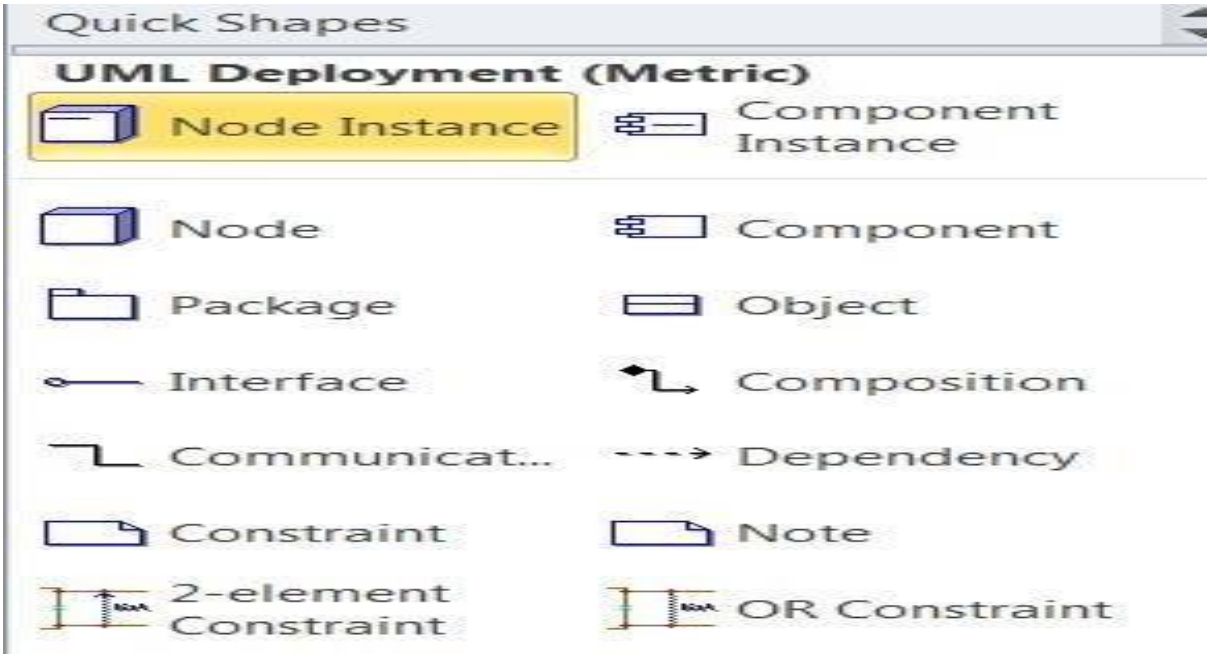

Product
Item Code
Customer
Customer Details
Order
Account Details
Payment
Account

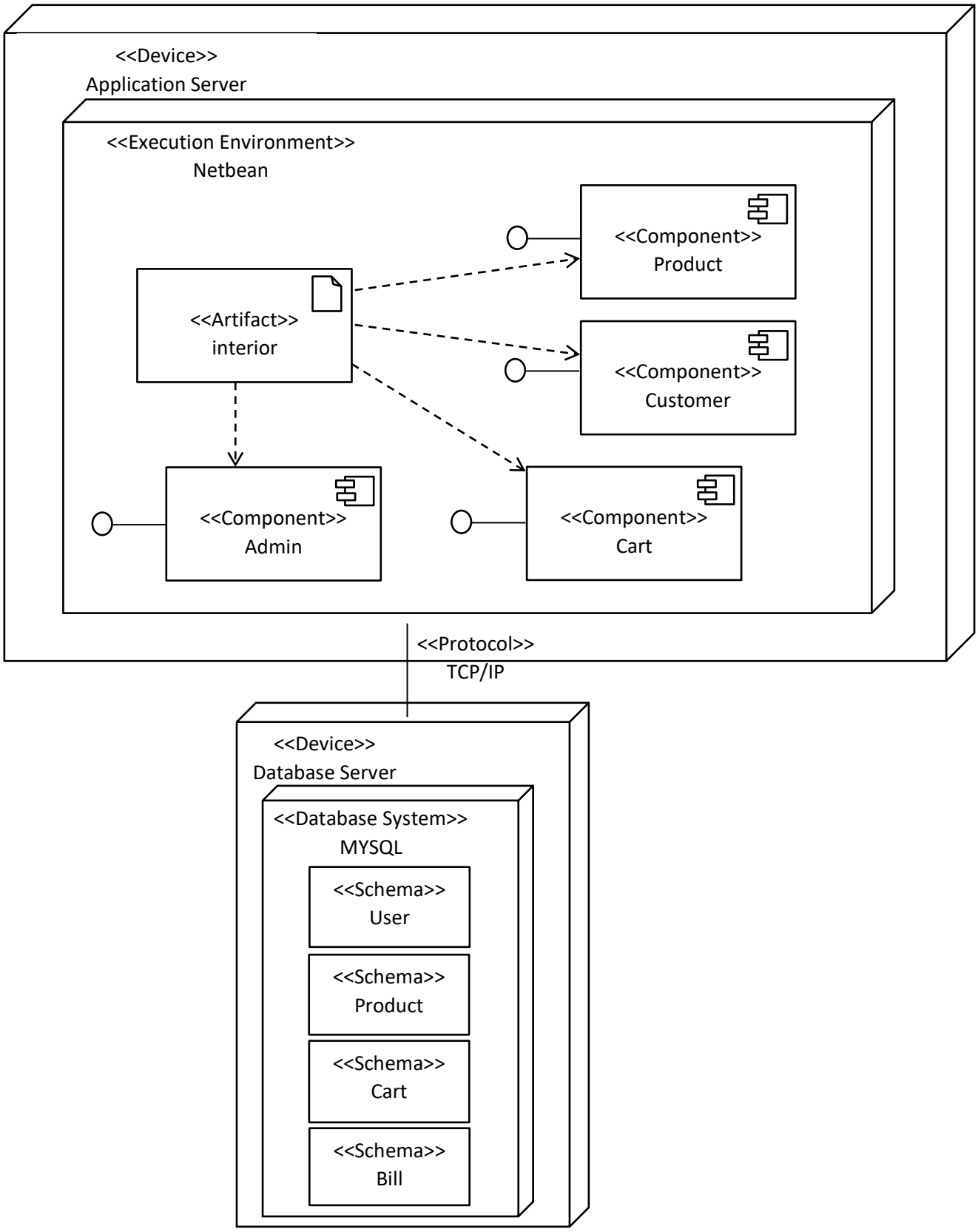
DEPLOYMENT DIAGRAM

A **deployment diagram** the Unified Modelling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected There are two types of Nodes:

- Device Node
- Execution Environment Node

Device nodes are physical computing resources with processing memory and services to execute software, An execution environment node (EEN) is a software computing resource that runs within an outer node.




<<Device>>
Application Server
<<Execution Environment>>
Netbean
<<Component>>
Product
<<Artifact>>
interior
<<Component>>
Customer
<<Component>>
Admin
<<Component>>
Cart
<<Protocol>>
TCP/IP
<<Device>>
Database Server
<<Database System>>
MYSQL
<<Schema>>
User
<<Schema>>
Product
<<Schema>>
Cart
<<Schema>>
Bill

data dictionary:-

metadata repository, as defined in the *IBM Dictionary of Computing*, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format. *Oracle* defines it as a collection of tables with metadata. The term can have one of several closely related meanings pertaining to data bases and database management system (DBMS):

- A document describing a database or collection of databases
- An integral component of a DBMS that is required to determine its structure
- A piece of middleware that extends or supplants the native data dictionary of a DBMS

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

TABLE STRUCTURE

COLUMN NAME	DATATYPE	NULL
USERID	VARCHAR(100)	NOT NULL
PASSWORD	VARCHAR(20)	NOT NULL

CUSTOMER TABLE:

COLUMN NAME	DATATYPE	NULL
FNAME	VARCHAR(50)	NULL
LNAME	VARCHAR(50)	NULL
EMAILID	VARCHAR(100)	NULL
MOBILE	VARCHAR(10)	NULL

PRODUCT_TABLE

COLUMN NAME	DATATYPE	NULL
PCODE	VARCHAR(20)	NOT NULL
PNAME	VARCHAR(50)	NULL
DESCRIPTION	VARCHAR(100)	NULL
TYPE	VARCHAR(100)	NULL
PRICE	VARCHAR(100)	NULL

ORDER DETAILS

COLUMN NAME	DATATYPE	NULL
ORDER_NO	VARCHAR(20)	NOT NULL
ORDER_DATE	VARCHAR(100)	NULL
ADDRESS	VARCHAR(100)	NULL
EMAIL ID	VARCHAR(10)	NULL
GRAND_TOTAL	VARCHAR(100)	NULL
PAYMENT_MODE	VARCHAR(100)	NULL

ORDER_PRODUCT

COLUMN NAME	DATATYPE	NULL
ORDERNO	VARCHAR(20)	NOT NULL

PCODE	VARCHAR(100)	NULL
QUANTITY	VARCHAR(100)	NULL
PRICE	VARCHAR(10)	NULL
GRAND_TOTAL	VARCHAR(100)	NULL

PAYMENT

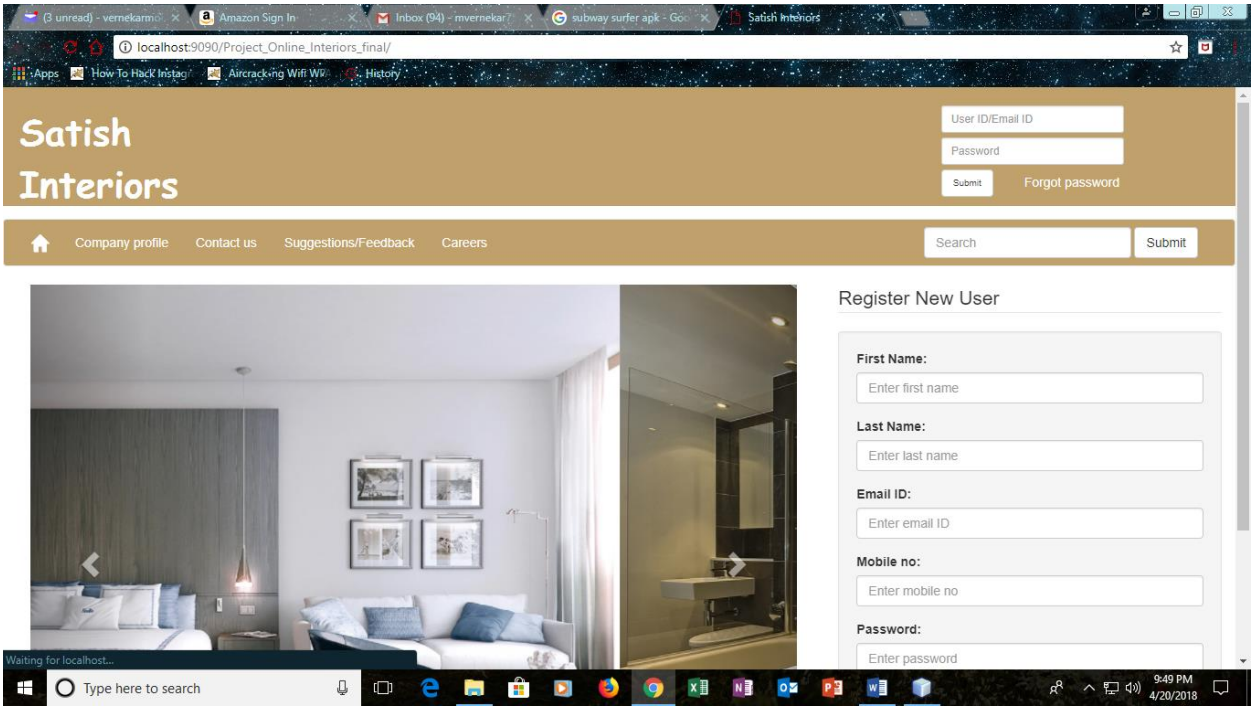
COLUMN NAME	**DATATYPE**	**NULL**
ORDER_NO	VARCHAR(20)	NOT NULL
TOTAL_AMT_PAID	VARCHAR(100)	NULL
PAYMENT_MODE	VARCHAR(100)	NULL

FEEDBACK :

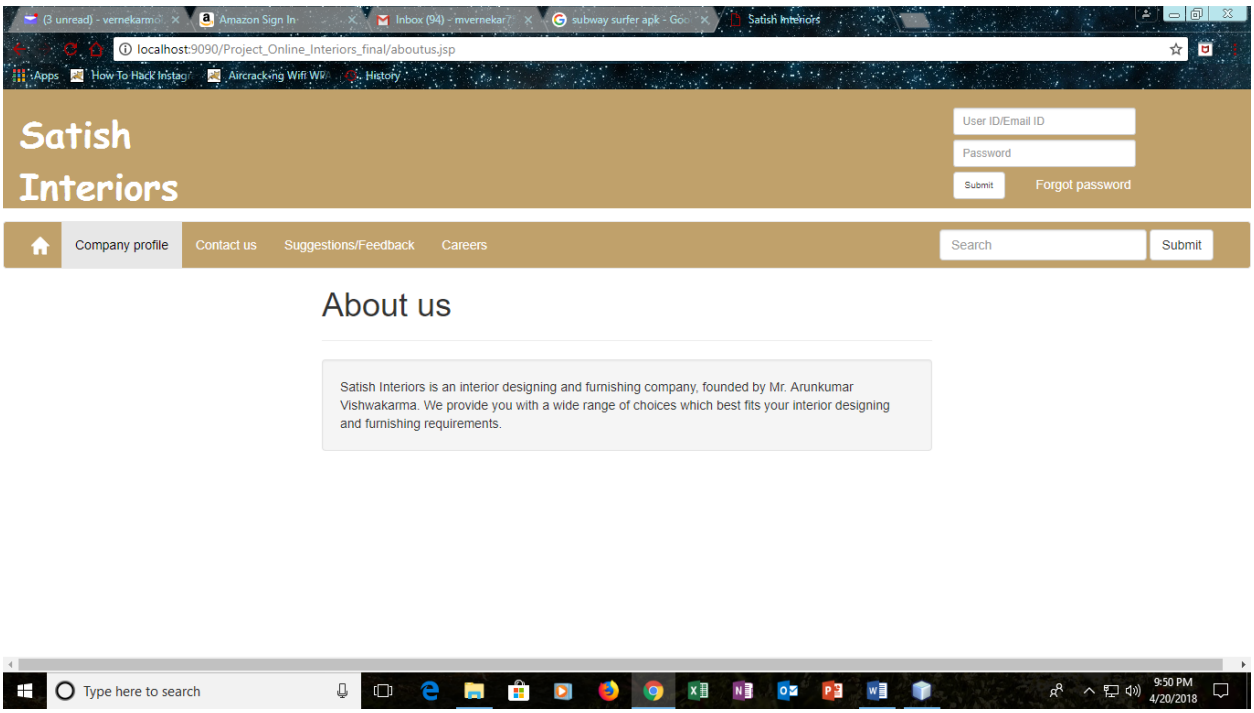
COLUMN NAME	**DATATYPE**	**NULL**
NAME	VARCHAR(100)	NULL
EMAILID	VARCHAR(100)	NULL
MOBILE	VARCHAR(10)	NULL
FEEDBACK	VARCHAR(500)	NULL

SCREENSHOT

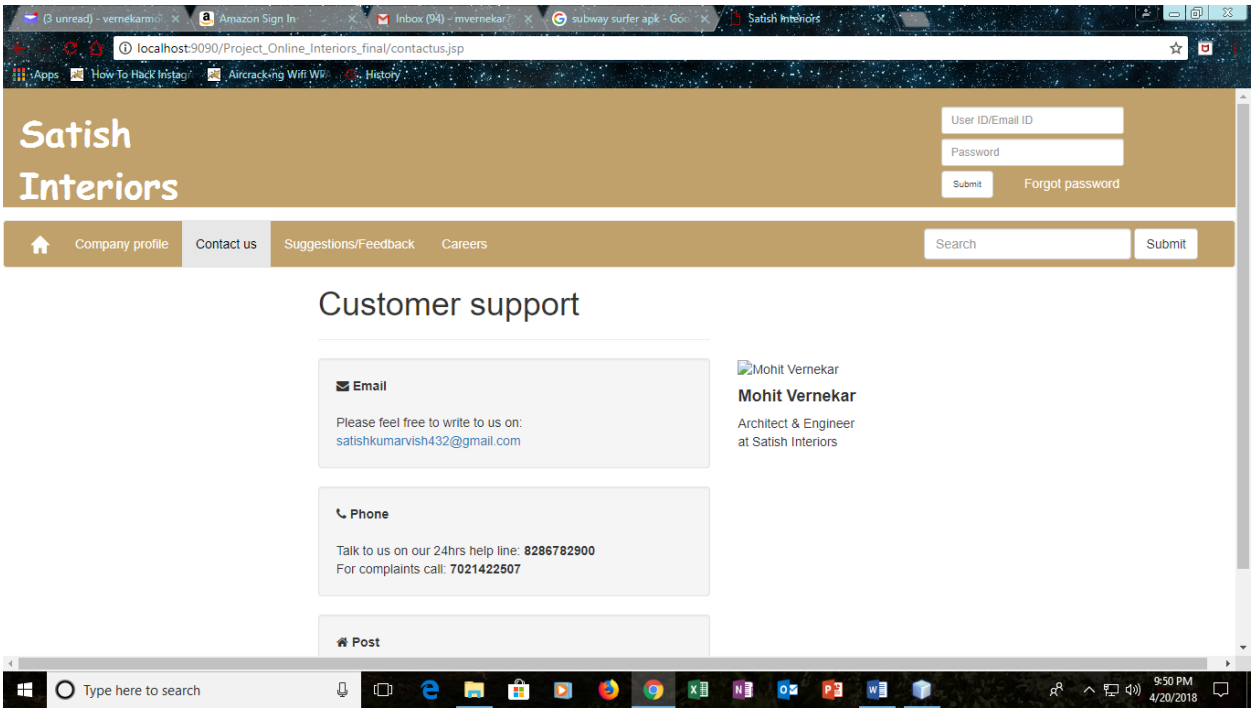
HOMEPAGE



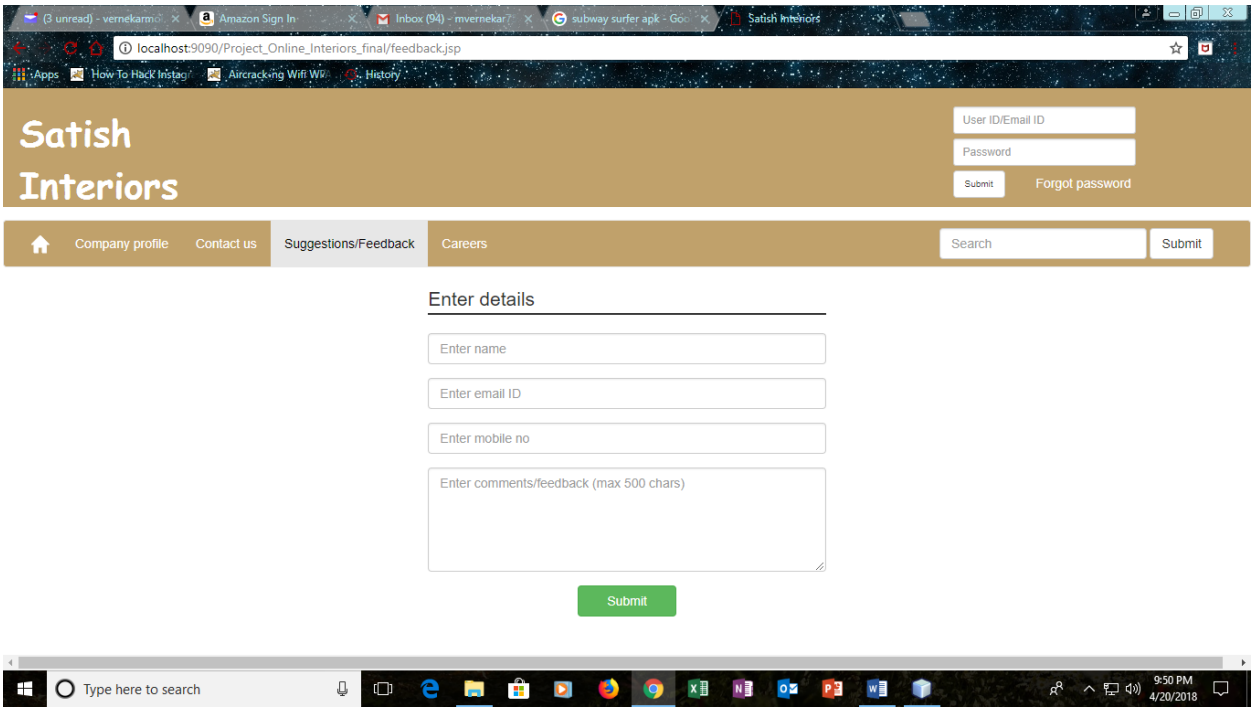
ABOUT US



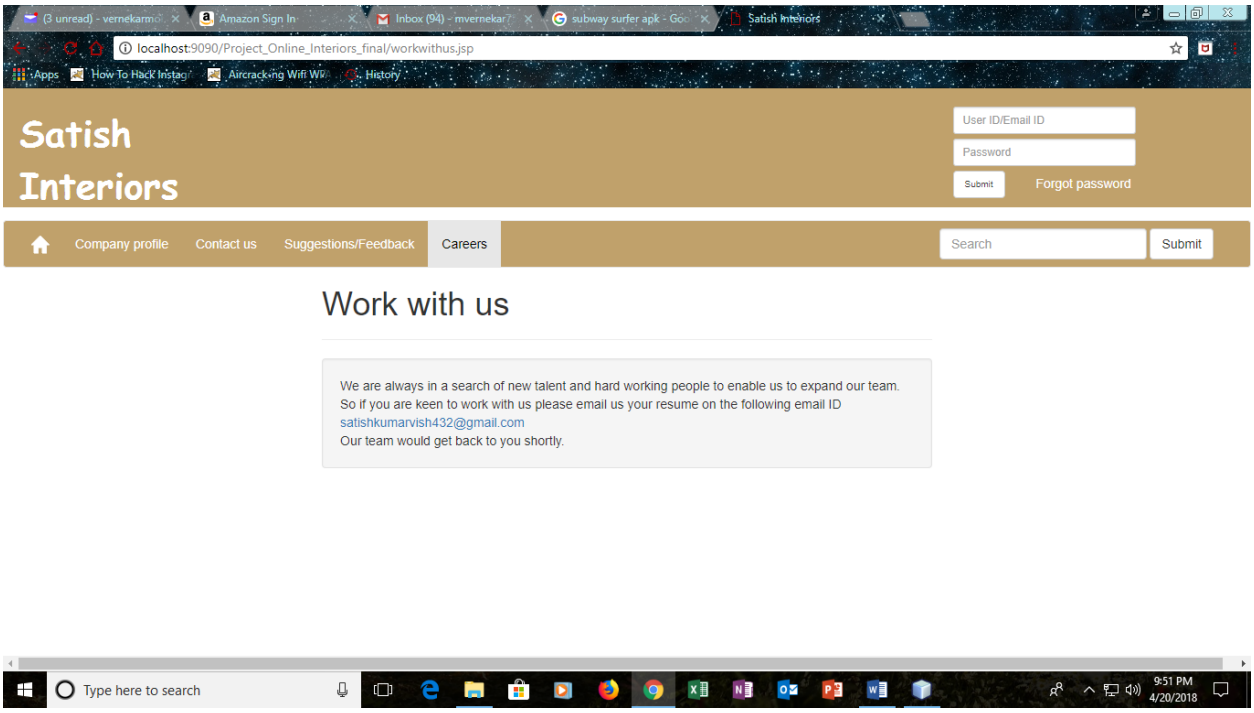
CONTACT US



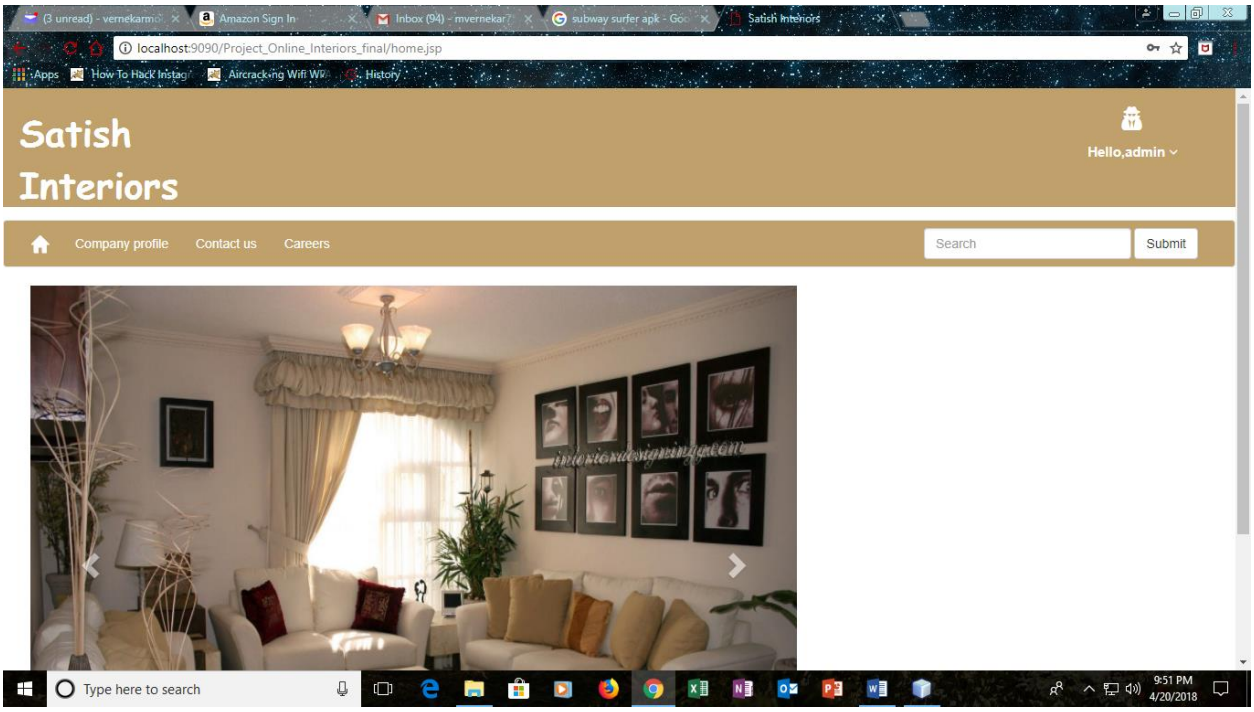
SUGGESTIONS/FEEDBACK



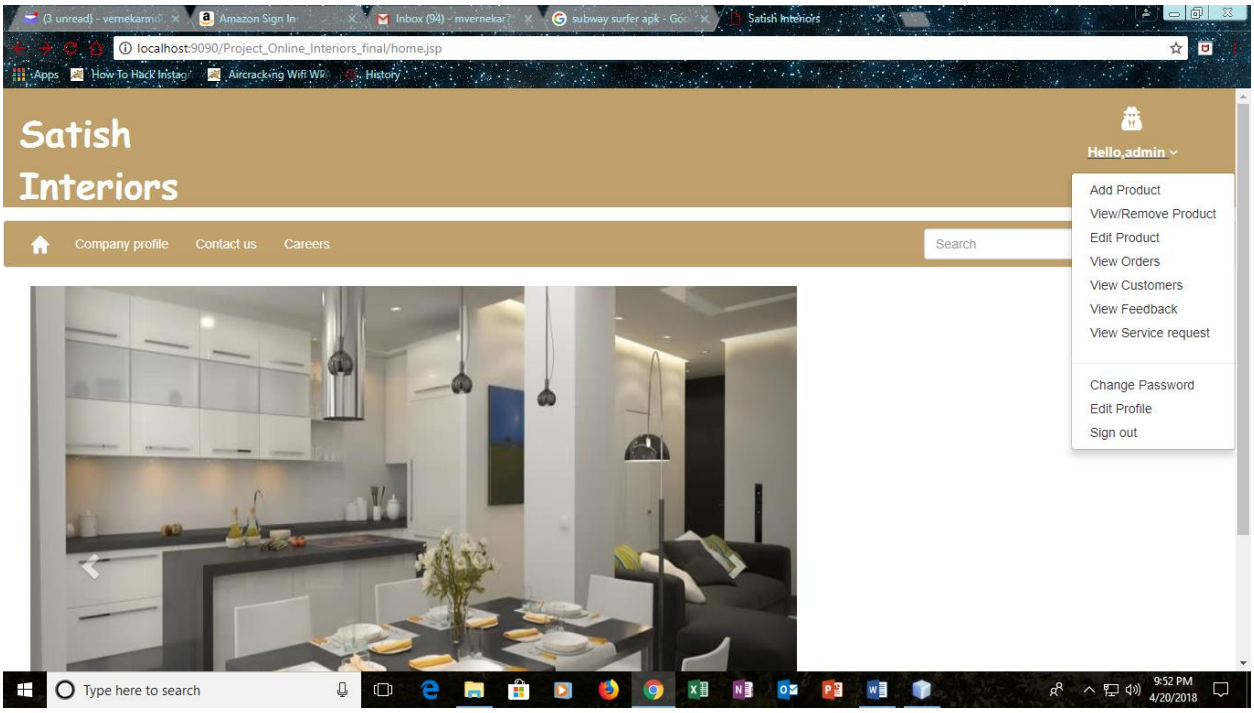
CAREERS



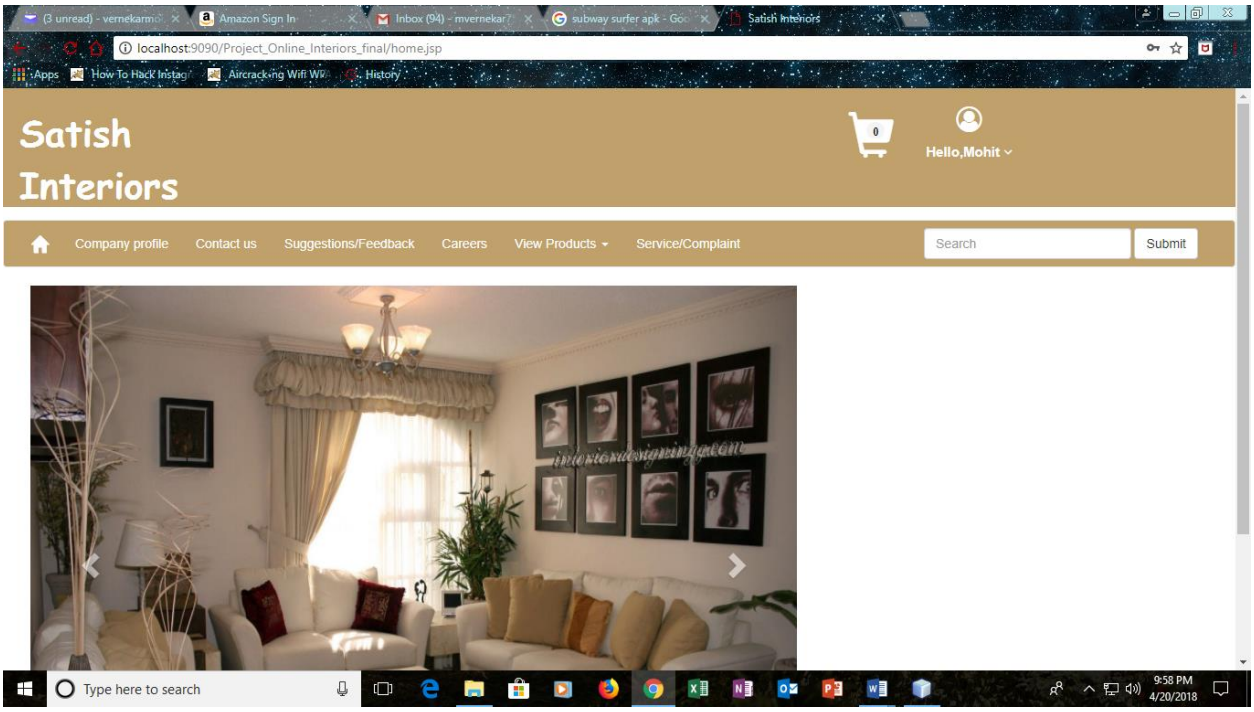
ADMIN HOMEPAGE



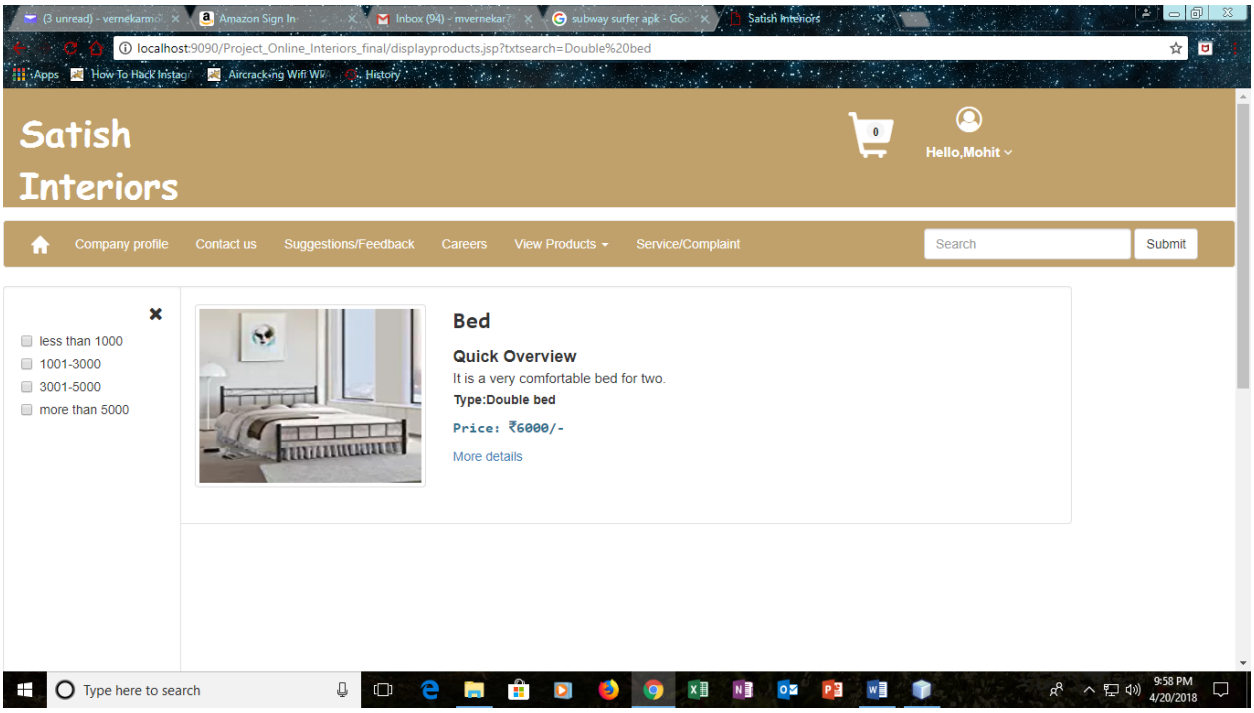
ADMIN FUNCTIONALITIES



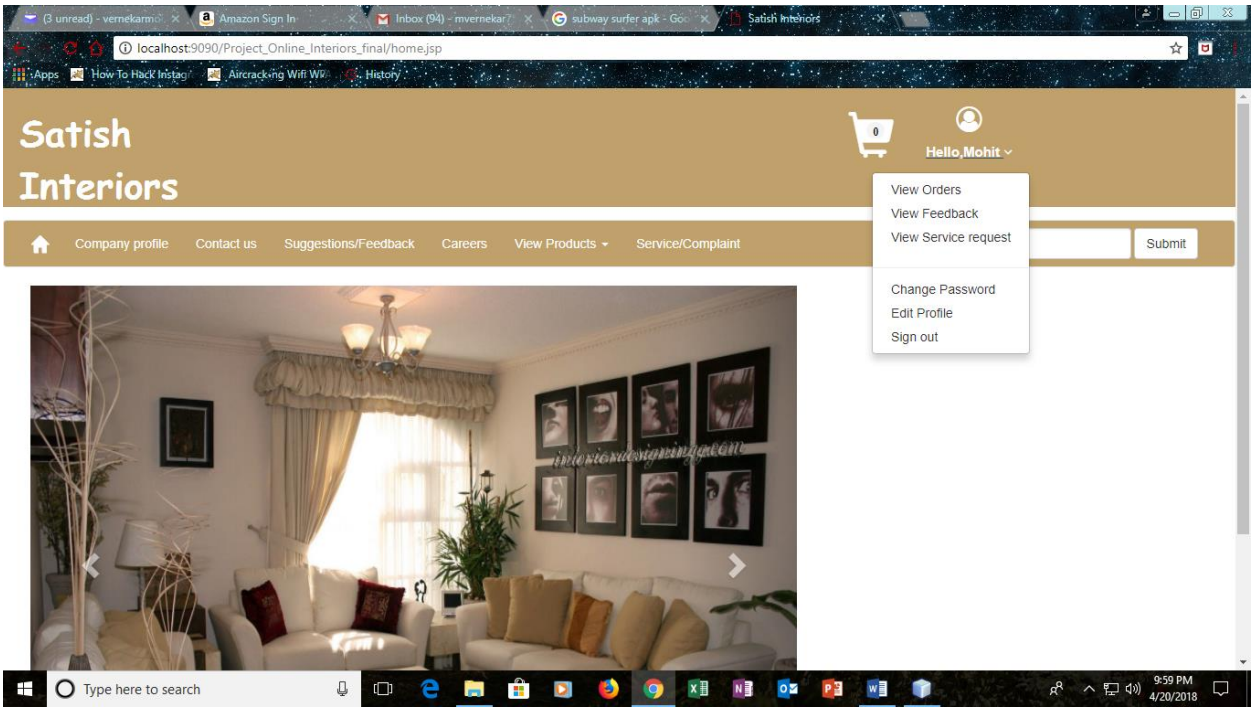
CUSTOMER LOGIN



VIEW PRODUCTS



CUSTOMER FUNCTIONALITIES



PROCESS INVOLVED

Login Module: User will be able to login using user id and password after they are finished with registration process.

User Registration:

1.Users will be able register themselves with their basic personal like name, address, email id, etc.

2.After this an OTP will be sent to the user's specified email id which user needs to check and enter the OTP on the website for completing registering process.

Guest User Shopping:

Guest User can view different products on the website but cannot place order for any products.

Registered User Shopping:

User can view different products on the website and also place order for some products.

As the user is already registered in our system, he need not enter those details again.

They can view the history and transaction of previous orders and payment as well.

After the successful completion of the process, a confirmation email will be sent to the user's registered email id.

Payment Facilities:

Payment details like credit card /debit card/e-wallet details will be taken from the user.

Credit card/debit card details will be checked if they are valid or not.

From e-wallet user can add money and pay money and its history will be viewed to the user

Transaction History:

Transaction history like the orders which it has placed previously, payment history (credit card/debit card details) will be shown to the user.

Service Module:

This module allows the user to place a service request for any complaints regarding computer or any peripherals.

The admin will receive the request placed by the user and take necessary action on the basis of the complaint of the user and notify the same to the user via email.

Admin:

Admin can add products/designs, remove products/designs or alter products details.

Admin can check the order details like how many orders are placed for which dates and placed by which users.

Admin can also view the feedback and suggestions.

Admin handles all the service request of different users.

Feedback/Suggestion:

Feedback and suggestions will be taken from the users and submitted to admin.

Forgot Password module:

This module is used by the admin or employee to reset the password.

This is done by sending an OTP to the user's email id which needs to be entered in the application to reset the password.

Maximum 3 attempts will be given to the user to enter the OTP

TESTING METHODOLOGY

Unit testing:-

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer modules together with associated control data, usage procedures, and operating procedures are tested to determine whether they are fit for use.

Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could unit be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing.

Unit testing is commonly automated, but may still be performed manually. The IEEE does not favor one over the other. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may execute as an integration test case that involves many software components, and thus preclude the achievement of most if not all of the goals for unit testing.

As a consequence, unit testing is traditionally a motivator for programmers to create decoupled and cohesive code bodies. This practice promotes healthy habits in software development. Deign patterns, unit testing, and refactoring often work together so that the best solution may emerge.

Advantages:

1) Find problems early.
2) Facilitates change.
3) Simplifies integration.
4) Documentation. 5) Design.

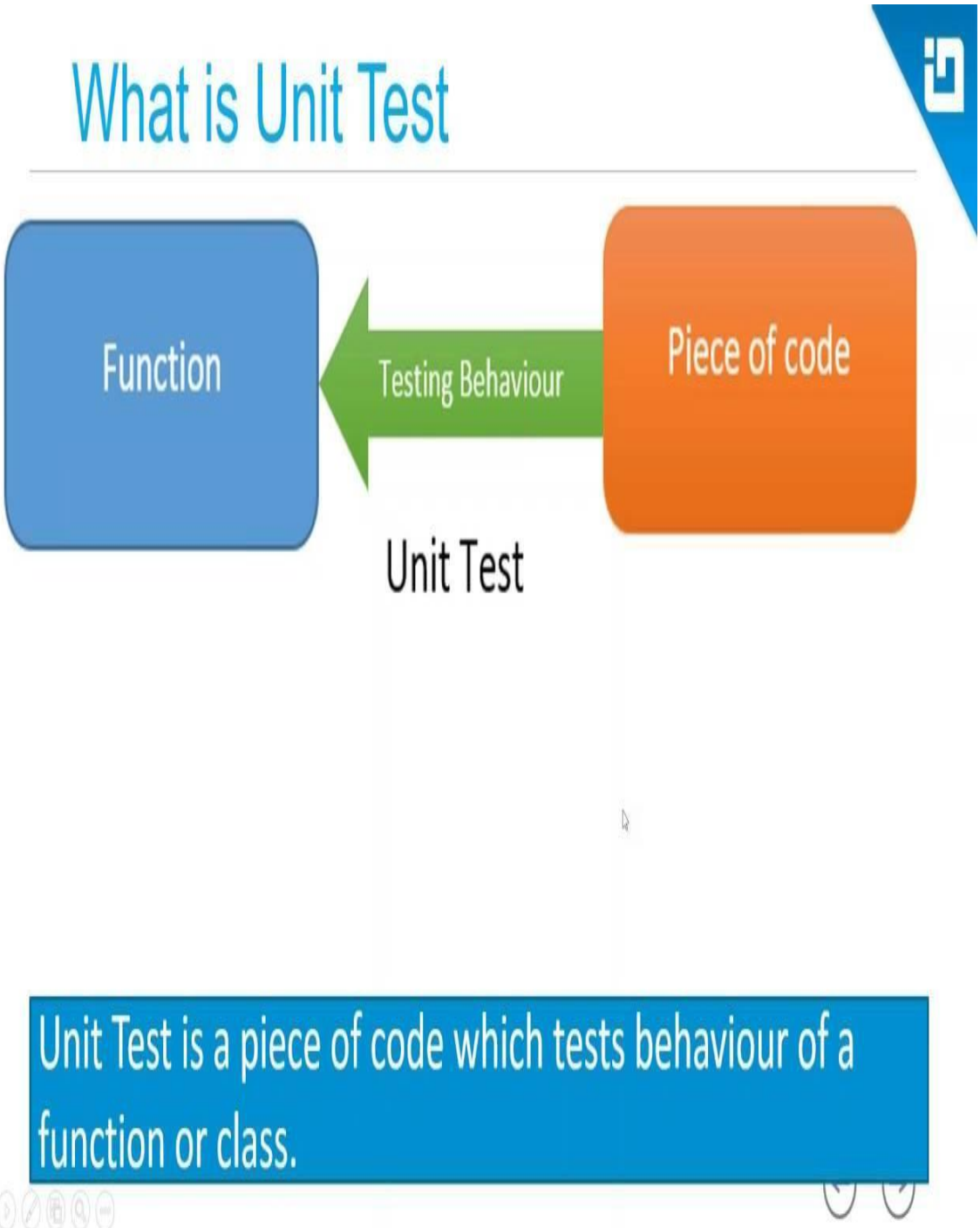
Limitations and Disadvantages:

1) Decision problem.

2) Not integration testing.
3) Combinatorial problem.
4) Realism.
5) Record keeping.
6) Sustainability challenges.
7) Platform differences.
8) External work.

Applications:

1) Extreme programming.
2) Unit test frameworks.
3) Language- level unit testing support.



SYSTEM TESTING:-

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing fails within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

Usually, Black Box Testing method is used. System testing is the third level of software testing performed after Integration Testing and before Acceptance Testing. Normally, independent Testers perform System Testing.

Tasks:

- System Test Plans o Prepare o Review o Rework
    - Baseline

- System Test Cases o Prepare o Review o Rework
    - Baseline

Strategies of testing

A test strategy is an outline that describes the testing approach of the software development cycle. It is created to inform project managers, testers, and developers about some of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment.

Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of testing are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used and occasionally, conceptual design documents may be referred to. Design documents describe the functionality of the software to be enabled in the upcoming release. For every stage of development design, a corresponding design, a corresponding test strategy should be created to test the new feature sets.

- Test Levels :
  The test strategy describes the test levels to be performed.

- Roles and Responsibilities :

  The roles and responsibilities of test leader, individual testers, project manager are to be clearly defined at a project level in this section.

- Environment requirements:

  Environment requirements are an important part of the test strategy. It describes what operating systems are used for testing.

- Testing Tools:

  There are two methods used in executing test cases; manual and automated and the combination of both methods is considered to be the best testing method.

- Risks and mitigation:

  Any risks that will affect the testing process must be listed along with the mitigation. By documenting a risk, its occurrence can be anticipated well ahead of time.

- Test Schedule:

  A test plan should make an estimation of how long it will take to complete the testing phase.

- Regression Test Approach:

  Regression tests will make sure that one fix does not create some other problems in that program or in any other interface.

- Test Groups:

  From the list of requirements, we can identify related areas, whose functionality is similar. These areas are called the test groups.

- Test Priorities:

  Among test cases, we need to establish priorities.

- Test Status Collections and Reporting:

  When test cases are executed, the test leader and the project manager must know, where exactly the project stands in terms of testing activities.

- Test Records and Maintenance:
  When the test cases are executed, we need to keep track of the execution details like when it is executed, who did it, how long it took, what is the result etc.

- Requirements Traceability Matrix :

  In a requirements traceability matrix, the rows will have the requirements. The columns represent each document.

- Test Summary :
- The senior management may like to have a test summary on a weekly or monthly basis. If the project is very critical, they may need it even on daily basis.

Black Box Testing:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

Test design techniques:

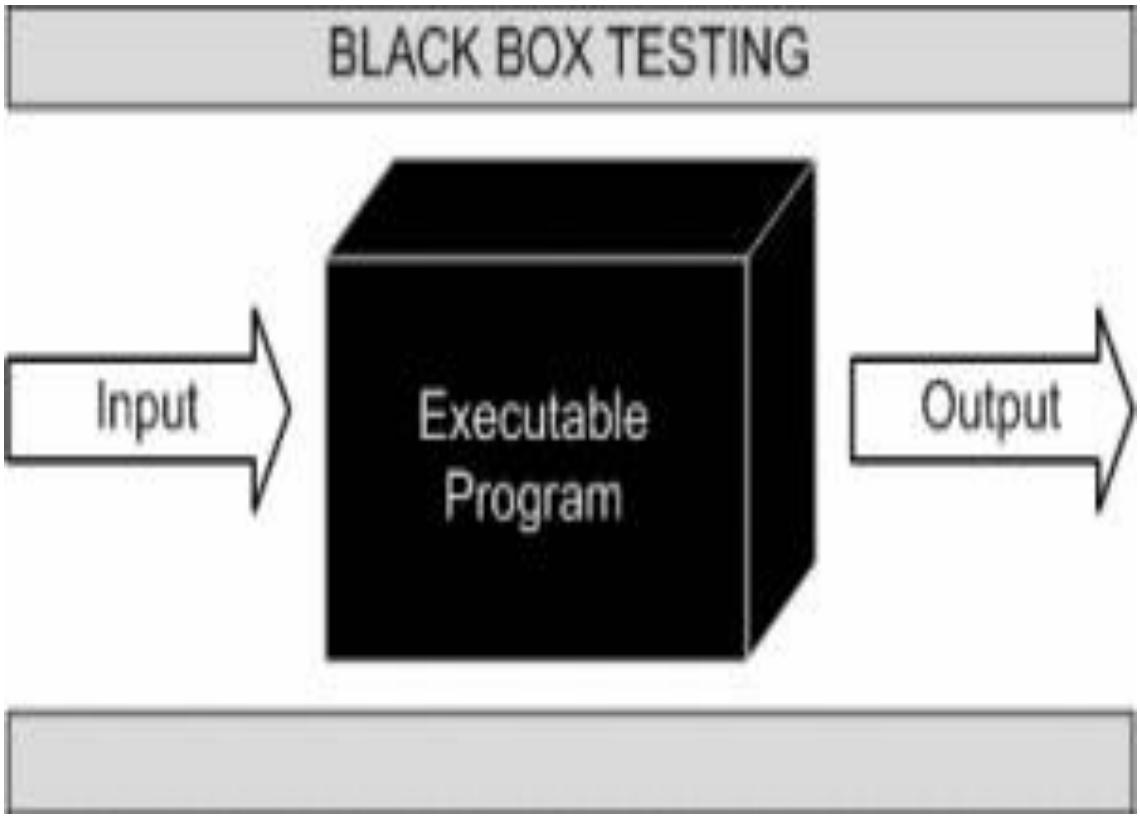
Typical black-box test techniques include:

Decision table testing, Equivalence Testing, Boundary Value Analysis, Cause-effect Graph, Error Guessing, State Transition Testing, Use Case Testing, Domain Analysis, Syntax Testing etc.

Black-box testing also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

- **Advantages**:
    - Tests are done from the user's point of view and will help in exposing discrepancies in the specifications.
    - Tester need not know programming languages or how the software has been implemented. o Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
    - Test cases can be designed as soon as the specifications are complete.

- **Disadvantages:**
    - Only a small number of possible inputs can be tested and many program paths will be left untested. o Without clear specifications, which is the situation in many projects, test cases will be difficult to design.
    - Tests can be redundant if the software designer/ developer has already run a test case.



Equivalence class partitioning:

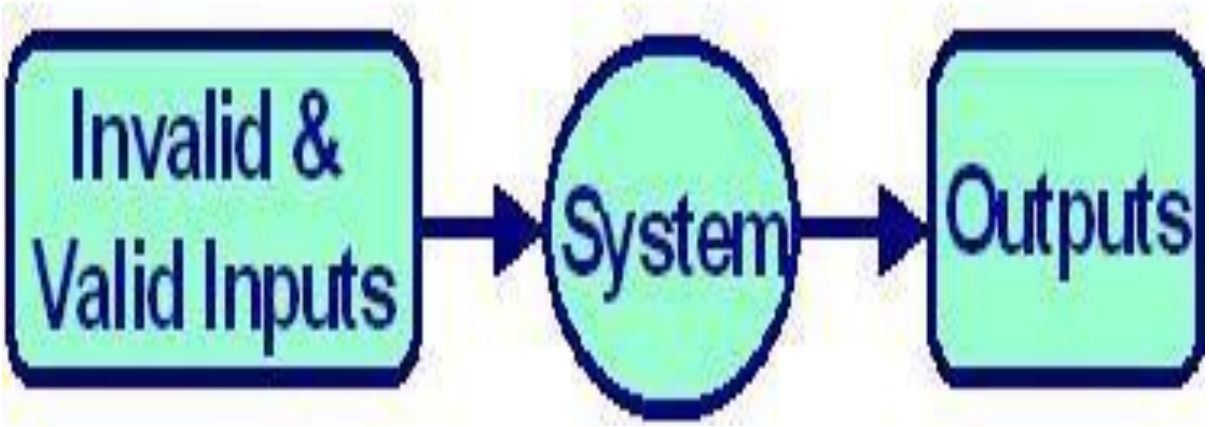
Equivalence partitioning or equivalence class partitioning (ECP) is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once. This technique tries to define test cases that uncover classes of errors, thereby reducing the total number of test cases that must be developed. An advantage of this approach is reduction in the time required for testing a software due to lesser number of test cases.

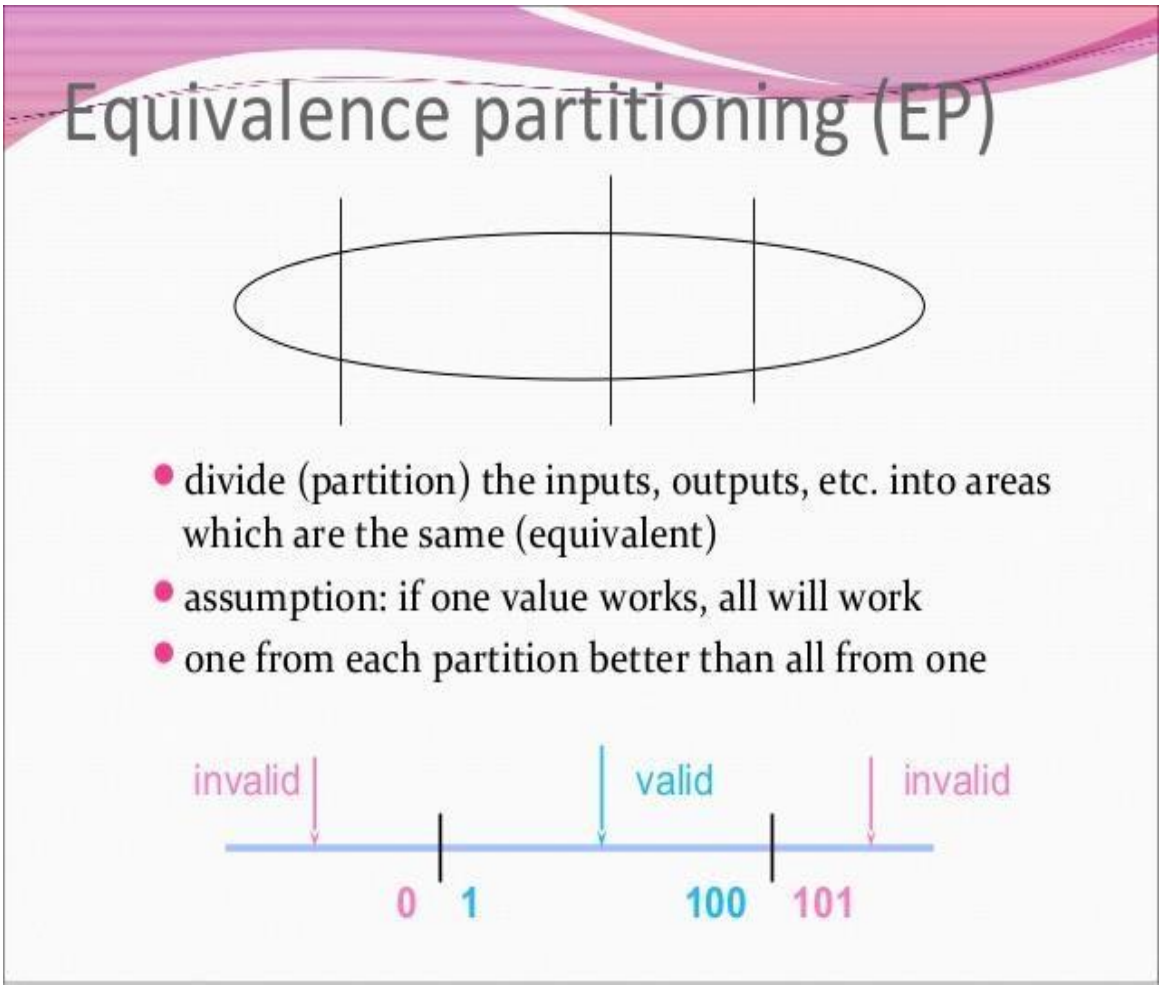
Equivalence partitioning is typically applied to the inputs of a tested component, but may be applied to the outputs in rare cases. The equivalence partitions are usually derived from the requirements specification for input attributes that influence the processing of the test object.

Equivalence Partitioning is a testing technique where input values set into classes for testing.

- Valid Input Class = Keeps all valid inputs.
- Invalid Input Class = Keeps all invalid inputs.
  In equivalence-partitioning technique we need to test only one condition from each partition. This is because we are assuming that all conditions in one partition will be treated in the same way by the software. If one condition in a partition works, we assume all of the conditions in a partition does not work, then we assume that none of the conditions in that partition will work so again there is little point in testing any of these others. Similarly, if one of the conditions in a partition does not work, then we assume that none of the partition will work so again there is little point in more partition.

Equivalence Class Partitioning





White Box Testing.

White-box testing also known as clear box testing or glass box testing or transparent box testing and structural testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs. This is analogous to testing nodes in a circuit. White-box testing can be applied at the unit, integration and system levels of the software testing process. Although Traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a systemlevel test. Though this method of test design can uncover many errors or problems, It has the potential to miss unimplemented parts of the specification or missing requirements.

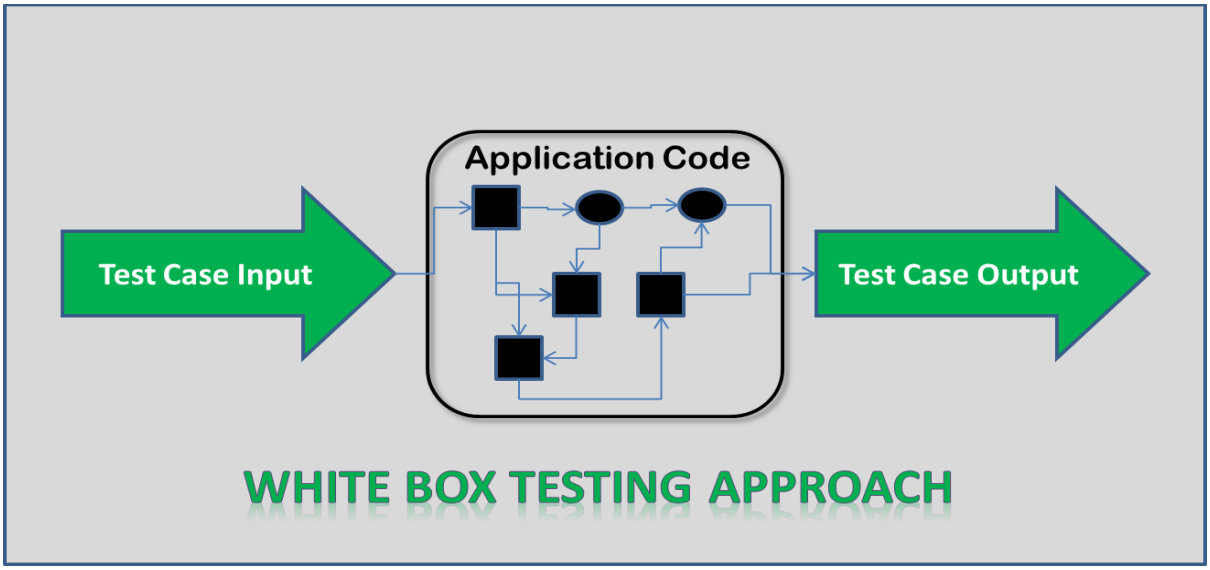
Advantages

1) Side effects of having the knowledge of the source code is beneficial to thorough testing.
2) Optimization of code becomes easy as inconspicuous bottlenecks are exposed.
3) Gives the programmer introspection because developers carefully describe any new implementation.
4) Provides traceability of tests from the source, thereby allowing future changes to the source to be easily captured in the newly added or modifies tests.
5) Easy to automate.
6) Provides clear, engineering-based rules for when to stop testing.

Disadvantages

1) White-box testing brings complexity to testing because the tester must have knowledge of the program, including being a programmer.
2) On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.
3) The tests focus on the software as it exists, and missing functionality may not be discovered.

The resulting test can be fragile because they are tightly coupled to the specific implementation of the thing being tested. The code under test could be written to implement the same functionality in a different way that invalidates the assumptions baked into the test. This could result in tests that fail unnecessarily or, in the worst case, tests that now give false positives and mask errors in the code.



TEST REPORT

CUSTOMER SIGNUP

Sr. No	Test Case	Action Performed	Input	Excepted Output	Actual output	Remark
1	Enter Name	Name entered in textbox	Satish	It should accept	It accepted	Pass
2	Enter Name	Invalid Name entered in textbox	satish@123	It should not accept	It did not accepted	Pass
3	Enter Email Id	EmailID entered in textbox	Satishvishwakarma432@gmailcom	It should accept	It accepted	Pass
4	Enter Email Id	Invalid EmailID entered in textbox	satishvishwakrmagmail.com	It should not accept	It did not accepted	Pass
5	Enter Phone number	Invalid Phone Number entered in textbox	RGE@123	It should not accept	It did not accepted	Pass
6	Enter Phone number	Phone Number entered in textbox	8286782900	It should accept	It accepted	Pass

CUSTOMER

Sr. No	Test Case	Action Performed	Input	Excepted Output	Actual output	Remarkt
1	Sign Up	Register button clicked	First name, Last name, Password, , Email id, Phone number.	Checks and validates the details and doctor is registered if details are correct	Student is Registered	Pass
2	Login	Login button clicked	User name and password	Checks with database and Logs in	Logs in	Pass

3	Select product	Menu Item	product is selected	product Should be displayed	product Displayed	Pass
4	Logout	Logout button clicked	Click on Logout	Checks with database and logs out	Logs out successfully	Pass

FUTURE ENHANCEMENTS

As we all know today's era is of web and internet, and the customer is the soul of any business. Now a days any organization whether it may be small or big is present on the web. It is very important and necessity thing to use the e-media for conducting your business effectively and successfully.

The software developed by me is a single computer based application use only by single person at a time. We can enhance its efficiency by providing more flexibility to run like a web services so that it can be used by more number of users. The client/ server network should be very helpful to achieve this goal.

Other than that the system developed by me can be integrated with the latest technology of telecommunication service to track the customer's records and call details more accurately and easily.

CONCLUSION

It was great opportunity for me as a student to learn and understand various aspects associated with project development. I did undergo from various phases of project development like analysis, design, coding, implementation and testing.

This project work is very helpful to understand the overall processes include in developing a single application. I got the idea about the ups and down taking place during project development. On the other hand I got an opportunity to understand the processes taking place in INTERIOR DESIGNING MANAGEMENT

REFERENCE & BIBLOGRAPHY

www.developer.com

www.15seconds.com

- **FOR SQL**

www.msdn.microsoft.com

- **FOR JSP**

www.msdn.microsoft.com/net/quickstart/aspplus/default.com

www.4guysfromrolla.com/index.aspx

www.w3school.com