



SARASWATI Education Society's
SARASWATI College of Engineering
Learn Live Achieve and Contribute
Kharghar, Navi Mumbai - 410 210.
NAAC Accredited

Department of Computer Engineering

Name: Shaikh Adnan Shaukat Ali

Roll No: 68

Subject: SOFTWARE ENGINEERING (SE)

Class/Sem: T.E. / SEM-V



Department of Computer Engineering

Subject :Software Engineering

Class/Sem:TE/V

Name of the Laboratory: Software Engineering Lab

Year: 2021-2022

LIST OF EXPERIMENTS

Expt. No.	Date	Name of the Experiment	Page No.
1.	30-07	Application of at least two traditional process models.	3-10
2.	30-07	Application of the Agile process models.	11-12
3.	06-08	Preparation of software requirement specification (SRS) document in IEEE format	13-28
4.	13-08	Structured data flow analysis.	29-35
5.	03-09	Use of metrics to estimate the cost	36-40
6.	16-09	Scheduling & tracking of the project.	41-47
7.	01-10	Write test cases for black box testing.	48-55
8.	01-10	Write test cases for white box testing.	56-62
9	06-10	Preparation of Risk Mitigation, Monitoring and Management Plan (RMMM).	63-66
10	13-10	Version controlling of the project	67-74
1	09-09	Assignment-1	75-88
2	04-10	Assignment-2	89-101

Prof. Diksha Bhalerao

Subject In-charge

Prof. Sujata Bhairnallykar

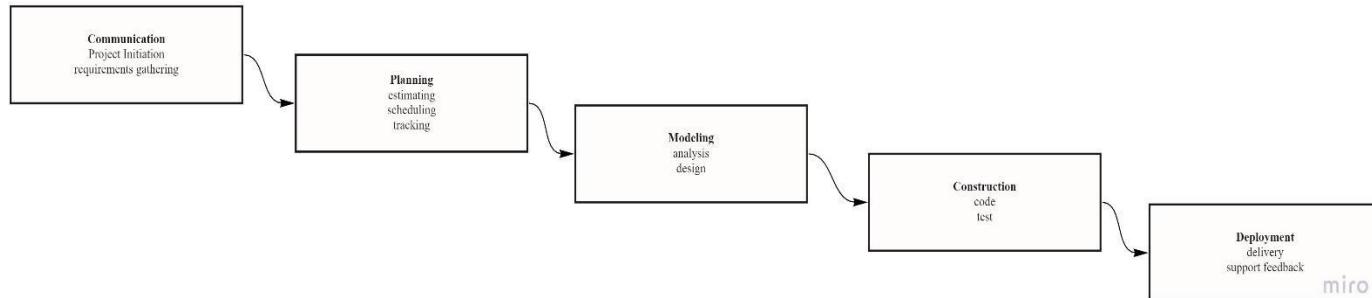
HOD

Experiment no. 1

Aim: Application of two traditional process models.

Theory:

Waterfall model: The waterfall model, sometimes called the classic life cycle, suggests a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through planning, modelling, construction, and deployment, resulting in completed software



Problem Statement: To apply waterfall model of E-commerce website by following path mentioned in above figure.

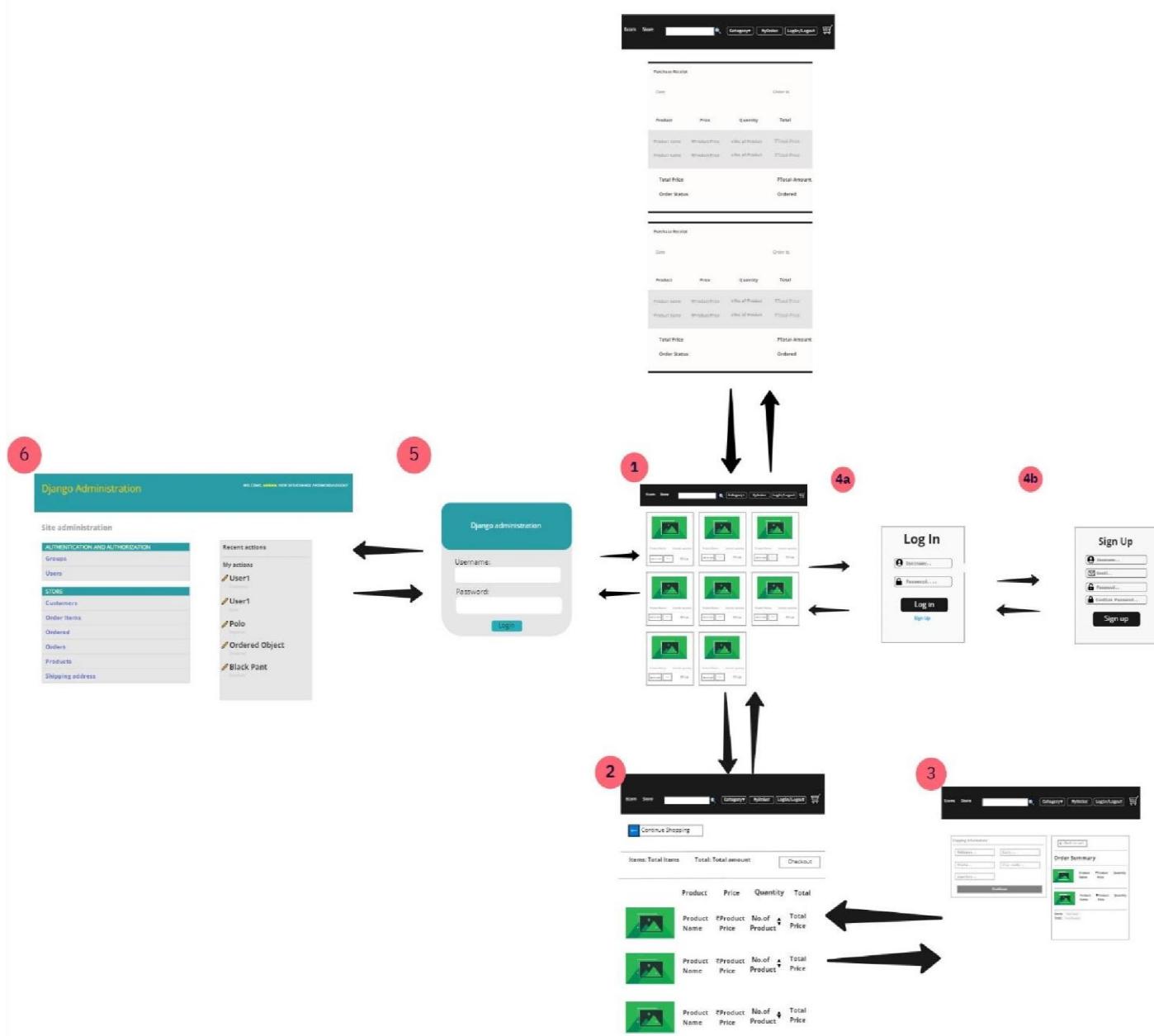
Requirement Gathering:

1. Add to cart facility.
2. Cash On Delivery (COD) as well as online payment option.
3. Purchased Product feedback facility.
4. Filtering, Searching product options on homepage.
5. Wish list option.
6. Ordered product status.
7. Multiple shipping address option.
8. Viewing option where more detail of the product.
9. Enquiry for product facility.
10. Login and Sign up option, Account recovery option, Credential security, Using App without Login.
11. Update through E-mail.

Planning:

1. Work products obtained each month should be reported in monthly discussion.
2. OS should be Windows 10 / Ubuntu >=18.04 with compatible Hardware.
3. Each Team should work at least 5 hours a day 5 days in a week.
4. Documentation should be prepare in multiple languages (English, Hindi, Kanji,...).
5. Model should be prepare under 2 months.
6. Each feature construction should be done under 20 days.
7. Cost of Design and Analysis – 10,000, Cost of Construction – 20,000,
Cost of Deployment - 8,000, Additional Charges – 2,000, Cumulated Cost – 40,000
8. Final work product delivery in 1 and half year (Date decision).

Modeling:



Construction:

Coding:

1. Languages to be used are Python >= 3.6, Django (Python Framework) >= 2.2, JavaScript ES6, React-Native (JS framework) >= 0.6, HTML5, CSS4, Bootstrap4 and Database connectivity with MySQL.
2. Flow of programming: Home page → Adding Features at home page (Filtration, Searching and Viewing) → Add to Cart page → Login and Signup page → Profile Page → Transaction Page → Product Status page → Admin Page → Adding remaining small features.

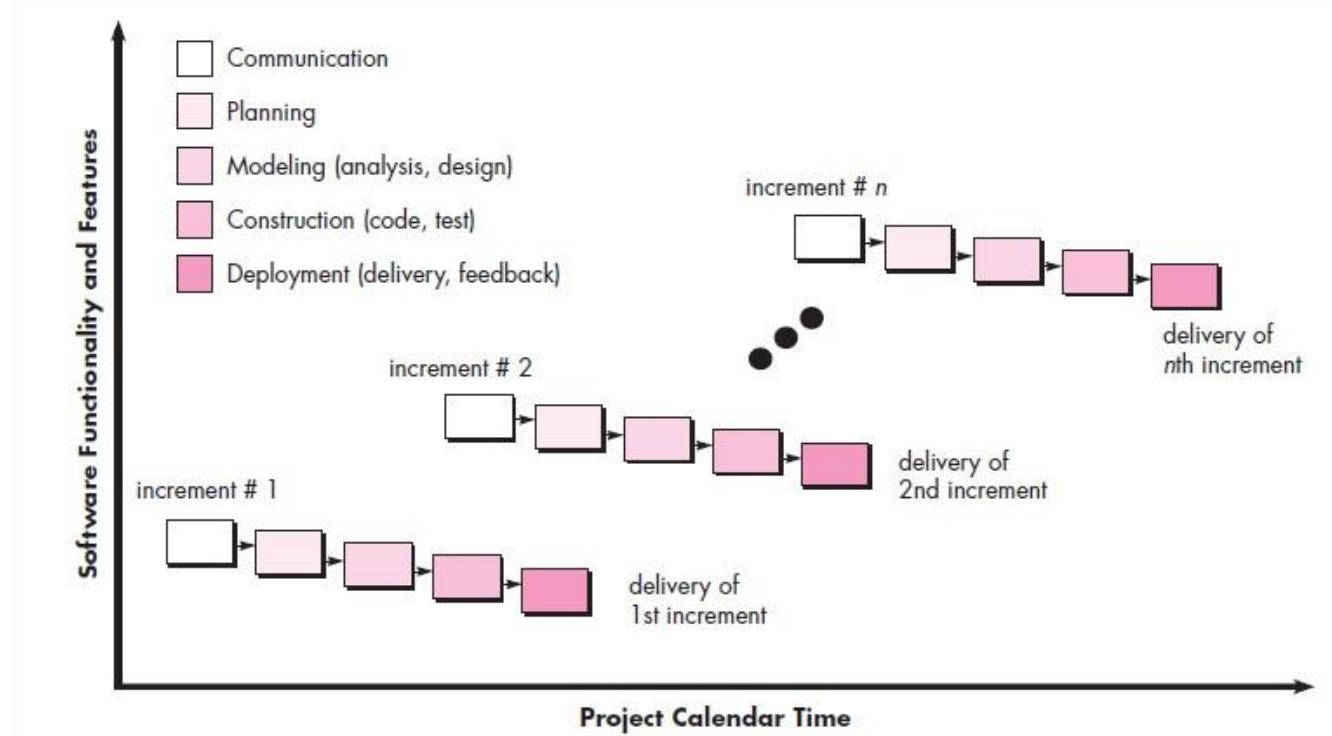
Testing: Unit Testing → Integration Testing → System Testing → Acceptance Testing.

Deployment:

1. Purchasing suitable domain name.
2. Selecting hosting server.
3. Releasing product on decided date.

Incremental model:

Incremental model is combination of systemic and linear sequential flow of Water fall model and parallel process flows. The incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces deliverable “increments” of software.



Problem Statement: To apply Incremental model on E-commerce website by following path mentioned in above figure.

Increment 1: Core product delivery

Requirement Gathering:

1. Add to cart Functionality.
2. Cash On delivery option.
3. Login and Sign Up Options.

Planning:

1. Model should be prepared under 20 days.
2. Each Team member should work at least 7 hours a day, 5 days in a week.
3. OS should be Windows 10 / Ubuntu ≥ 18.04 with compatible Hardware.
4. I) What tasks accomplished since last meeting?

II) What issues are encountered?

III) What task will be accomplished by the next meeting?

These questions will be answered by every team members in daily short meeting of 15 minutes.

5. Each requirements constructions should be given at most 40 days.
6. Cumulated cost – 30,000.
7. Final core product should be delivered in 6 months.

Modeling:



Construction:Coding:

1. Languages to be used are Python >= 3.6, Django (Python Framework) >= 2.2, HTML5, CSS4, Bootstrap4 and Database connectivity with SQLite3.
 2. Flow of programming: Homepage → Cart Page → Check out page → Login page → Sign Up page.
- Testing: Unit Testing → Integration Testing → System Testing → Acceptance Testing.

Deployment:

1. Purchasing suitable domain name.
2. Selecting hosting server.
3. Releasing product on decided date.

Increment 2: Fixing hidden bugs and adding important new features.

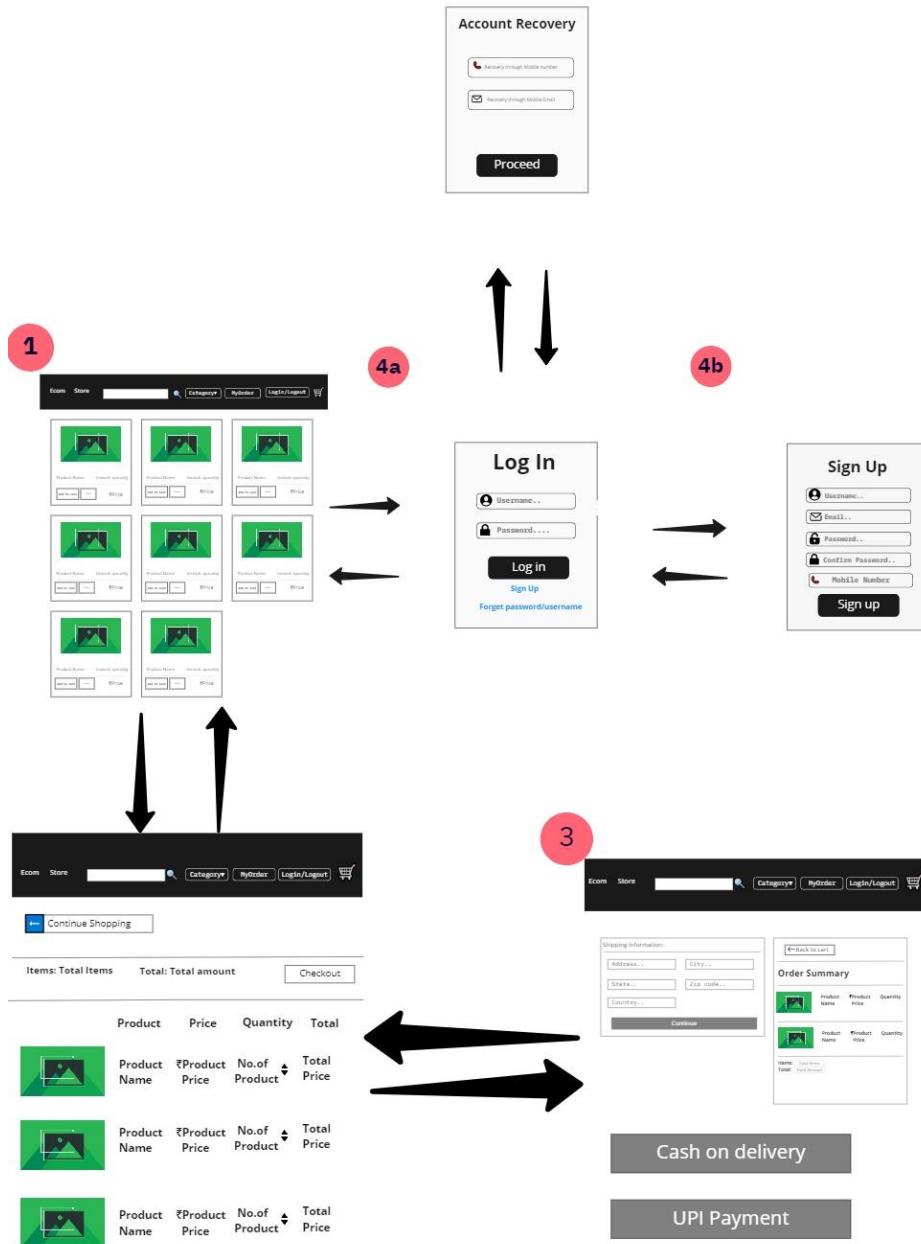
Requirement gathering:

1. Adding Search and filter features.
2. Account recovery option, Credential security and using app without login.
3. Increasing product quantity in cart page
4. Order summary in Check out page.
5. Online transaction facility.

Planning:

1. Model modification in 14-20 days.
2. Each Team member should work at least 6 hours a day, 5 days in a week.
3. Each requirements constructions should be given at most 40 days.
4. Simple documentation should be prepared in 20 days.
5. No change in team meeting.
6. Cumulated Cost - 30,000.
7. Release of increment 2 in 8 months

Modeling: Modified previous model.



Construction:

Coding:

1. Java Script ES6 integrating with languages used in Increment 1 and Database changed to MySQL.
 2. Flow of programming: Adding filtration and search options → Use of app without logging using cookies → Adding online transaction option → Account recovery option → Enhancing Security.
- Testing: Unit Testing → Integration Testing → System Testing → Acceptance Testing.

Deployment:

1. Uploading code to server.
2. Releasing Increment 2.

Increment 3: Adding new features for ease of customers.

Requirement gathering:

1. Purchased Product feedback facility.
2. Wish list option.
3. Ordered product status.
4. Multiple shipping address option.
5. Viewing option where more detail of the product.
6. Enquiry for product facility.
7. Update through E-mail and Phone Number.

Planning:

1. Creating Model for Product status page at most 4 days.
2. Each Team member should work at least 5 hours a day, 5 days in a week.
3. Each requirements constructions should be given at most 40 days.
4. Documentation preparation in different languages in 40 days.
5. No change in team meeting.
6. Cumulated Cost - 30,000.
7. Release of increment 3 in 12 months.

Modeling: Model of product status page

Purchase Receipt

Date	Order Id.		
Product name	₹Product Price	x No. of Product	₹Total-Price
Product name	₹Product Price	x No. of Product	₹Total-Price
Total Price		₹Total-Amount	
Order Status		Ordered	

Purchase Receipt

Date	Order Id.		
Product name	₹Product Price	x No. of Product	₹Total-Price
Product name	₹Product Price	x No. of Product	₹Total-Price
Total Price		₹Total-Amount	
Order Status		Ordered	

mirco

Construction:

Coding:

1. Reactive Native + languages used in Increment 2.

2. Flow of programming: Wish list option → Multiple shipping address option → Viewing option → Update through Email and phone number → Product status page → Enquiry for product → Purchased Product feedback facility.

Testing: Unit Testing → Integration Testing → System Testing → Acceptance Testing.

Deployment:

1. Uploading code to server.

2. Releasing Increment 3.

Conclusion: We have successfully applied two traditional process models (Water fall and Incremental model) for development of E-commerce shopping website.

Experiment no. 2

Aim: Application of the Agile process models.

Problem Statement: To apply Kanban model for development of E-commerce shopping website.

Theory:

Kanban Model:

Kanban is a visual system for managing work as it moves through a process. Kanban visualizes both the process (the workflow) and the actual work passing through that process. The goal of Kanban is to identify potential bottlenecks in your process and fix them so work can flow through it cost-effectively at an optimal speed or throughput.

In our Kanban Board work pass flow through three process: To Do, In progress and Done. workflow from To Do to In progress to Done.

- 1) To Do contain all the things that are need to be completed in near future but not started yet.
- 2) In progress (Doing) name implies itself, work in progress limit (WIP Limit) which define maximum number of work can be there at a time and it also help us to increase our productivity, our WIP Limit is 6.
- 3) Done contain all the work which is completed, it track progress of our project.

Kanban Board:

To Do 8	In progress 6 WIP Limit: 6	Done 4
Testing CartPage and SignUp page	Testing Homepage	Adding features at homepage (Filtration, Searching and Viewing)
Coding Product Status	Coding Cart page	Coding Home page
Testing Profile Profile, Transaction and Product Status page	Coding Login and Signup page	Model Design for HomePage, Cart page and Authentication page
Enhancement in Modeling Design adding new features: Comparing Product and Feedback features	Security check for Authentication	Necessary Environment setup
Coding Comparing Product and Feedback features	Enhancement in Model Design adding new features: Profile page, Transaction Facility, Product status Facility.	
Completing Unit Testing, Integration testing , System Testing and Acceptance Testing	Coding Profile page and Transaction page in pair	
Deciding Domain name and Hosting Server.		
Releasing Software		

Conclusion: We have successfully applied Agile process model i.e. Kanban model for development of E-commerce shopping website.

Experiment No. 3

Aim: To prepare SRS document in IEEE format

Problem Statement: To prepare SRS document for E-commerce shopping website in IEEE format.

Theory:

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

A typical SRS includes:

- A purpose
- An overall description
- Specific requirements

The best SRS documents define how the software will interact when embedded in hardware — or when connected to other software. Good SRS documents also account for real-life users.

Why Use an SRS Document?

A software requirements specification is the basis for our entire project. It lays the framework that every team involved in development will follow.

It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

Using the SRS helps to ensure requirements are fulfilled. And it can also help us make decisions about our product's lifecycle — for instance, when to retire a feature.

Writing an SRS can also minimize overall development time and costs. Embedded development teams especially benefit from using an SRS.

Software Requirements Specification

**E-COMMERCE SHOPPING
WEBSITE**

Version 1.0 approved

Prepared by

ADNAN SHAIKH

SARASWATI COLLEGE OF ENGINEERING

29-08-2021

Table of Contents

Table of Contents

1.	Introduction	1
1.1	Purpose.....	1
1.2	Document Conventions	1
1.3	Intended Audience and Reading Suggestions	2
1.4	Product Scope.....	2
1.5	References	2
2.	Overall Description.....	3
2.1	Product Perspective.....	3
2.2	Product Functions.....	4
2.3	User Classes and Characteristics.....	5
2.4	Operating Environment	5
2.5	Design and Implementation Constraints	6
2.6	Assumptions and Dependencies.....	8
3.	External Interface Requirements	9
3.1	User Interfaces.....	9
3.2	Hardware Interfaces	9
3.3	Software Interfaces.....	9
3.4	Communications Interfaces.....	9
4.	System Features	10
4.1	System Architecture	10
4.1.1	Description and Priority	10
4.1.2	Action	11
4.1.3	Functional requirements.....	12
5.	Other Nonfunctional Requirements.....	13
5.1	Performance Requirements	13
5.2	Security Requirements	13
5.3	Software Quality Attributes	13

1. Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality to purchase goods. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a general-purpose e-commerce store where product like clothes can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online shopping for clothes. An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a shipping address, and payment information (COD). Customer Orders will be visible in my order page where customer can see the status of its orders

1.1 Purpose

- The software is for shopping products online.
- It maintains three levels of users
 1. Administrator level (Admin + User privileges).
 2. User level.
 3. Cookie User.
- The software includes home page for checking product.
- Product view to give feedback and for product query.
- Cart page to add desire product.
- Checkout page to purchase product (User level).

1.2 Document Conventions

- E-com.: E-commerce shopping website.
- HP: Home Page.
- PP: Product Preview.
- PF: Product Feedback.
- LP: Login Page.
- SP: Signup Page.
- ACP: Add To Cart Page.
- CP: Checkout Page.
- COD: Cash on Delivery.
- AP: Administrator Page.
- MOP: My Order Page

1.3 Intended Audience and Reading Suggestions

This document is to be read by the development team, the project managers, marketing staff, testers and documentation writers. Our stakeholders, company manufacturing associated hardware, company providing embedded operating system, shareholders, and distributors who markets the finished product, may review the document to learn about the project and to understand the requirements. The SRS has been organized approximately in order of increasing specificity. The developers and project managers need to become intimately familiar with the SRS.

1.4 Product Scope

E-com. is a web application that is designed to allow users to easily checkout products of their choice and store the desire product in ACP where total price and quantity of products can be view. For purchasing product user can move further to CP for online transaction and COD. Purchased product status and receipt will available in MOP. LP and SP can be used for Login, Sign up and recovery of account.

1.5 References

- Django: <https://docs.djangoproject.com/en/3.2/>
- Bootstrap: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- Vue JS: <https://vuejs.org/v2/guide/>
- Flow Chart: <https://app.diagrams.net/>
- Modelling: <https://miro.com/welcomeonboard/WGpZT2FEZDFRTW91QnpaUTJVTVVViYjJVaFlREdOcXVhYjltQjBKdWFjeFZvbVRNbThqYXZnRTZFRIBmdzR1SnwzMDe0NDU3MzU3ODAwNTI0NDYx>

2. Overall Description

2.1 Product Perspective

The various system tools that have been used in developing both the front end, back end and other tools of the project are being discussed in this section.

1) FRONT END:

HTML, CSS, JAVA SCRIPT, BOOTSTRAP, VUE JS, JINJA are utilized to implement the frontend.

- **HTML (Hyper Text Mark-up Language):**

HTML is a syntax used to format a text document on the web.

- **CSS (Cascading Style Sheet):**

CSS is a style sheet language used for describing the look and formatting of a document written in a mark-up language.

- **Java Script:**

JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

- **Bootstrap:**

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

- **Vue JS:**

Vue.js is an open-source progressive JavaScript framework for building user interfaces (UIs) and single-page applications; it is commonly referred to as Vue. This framework uses “high decoupling”, allowing developers to progressively create user interfaces (UIs).

- **Jinja:**

Jinja is a fast, expressive, extensible templating engine. Special placeholders in the template allow writing code similar to Python syntax. Then the template is passed data to render the final document.

2) BACK END:

The back end is implemented using MYSQL which is used to design the databases.

- **MYSQL:**

MySQL is the world’s second most widely used open source relational database management system (RDMS). The SQL phrase stands for structured query.

3) API:

To connect frontend and backend and writing logical operation Django framework is used.

- **Django:**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so we can focus on writing our app without needing to reinvent the wheel.

2.2 Product Functions

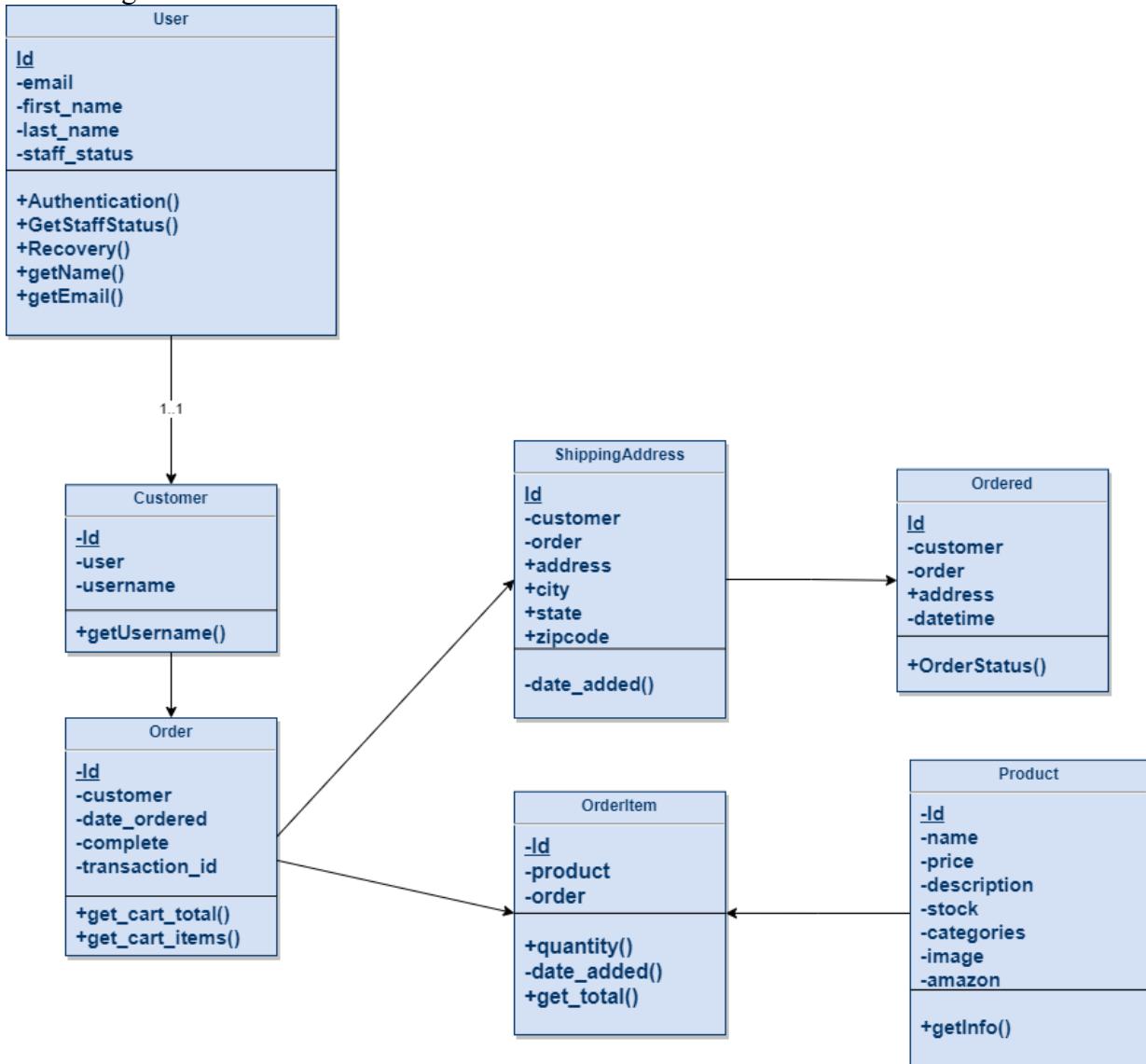
- The system will allow user to use the app without logging.
- Cookie data is used to remember activity of anonymous user.
- To purchase any product user need to login and provide address and some other necessary requirements.
- User can always check what products they purchased and what is the status of product in MOP section
- If user forget their account recovery functionality is also available.
- If user have any query regarding product they can ask in PF section.
- Admin have all privilege of user as well as additional administrator functionality which let them add, update and delete products, accounts, feedbacks and orders.

2.3 User Classes and Characteristics

1. Use Case Diagram:



2. Class Diagram:

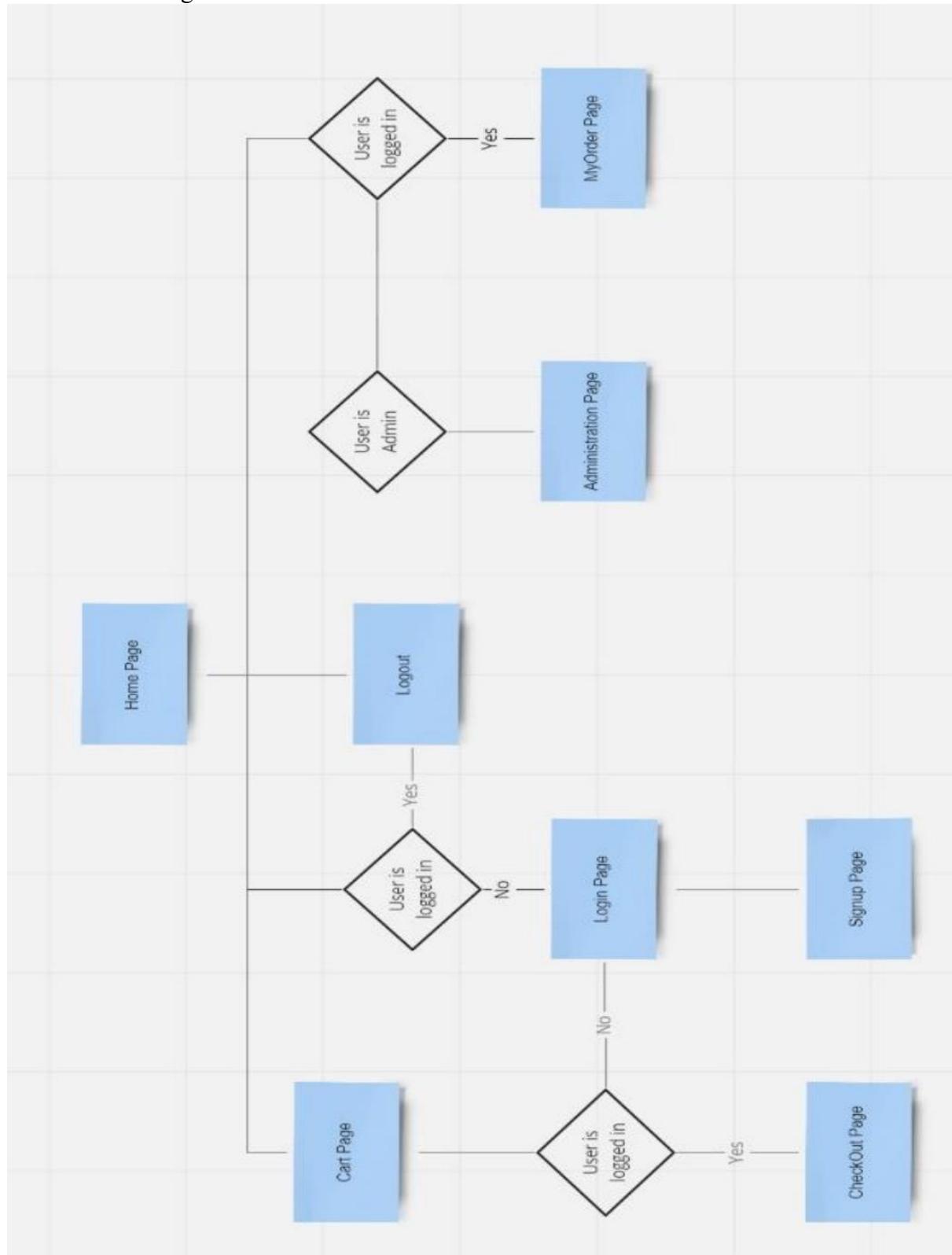


2.4 Operating Environment

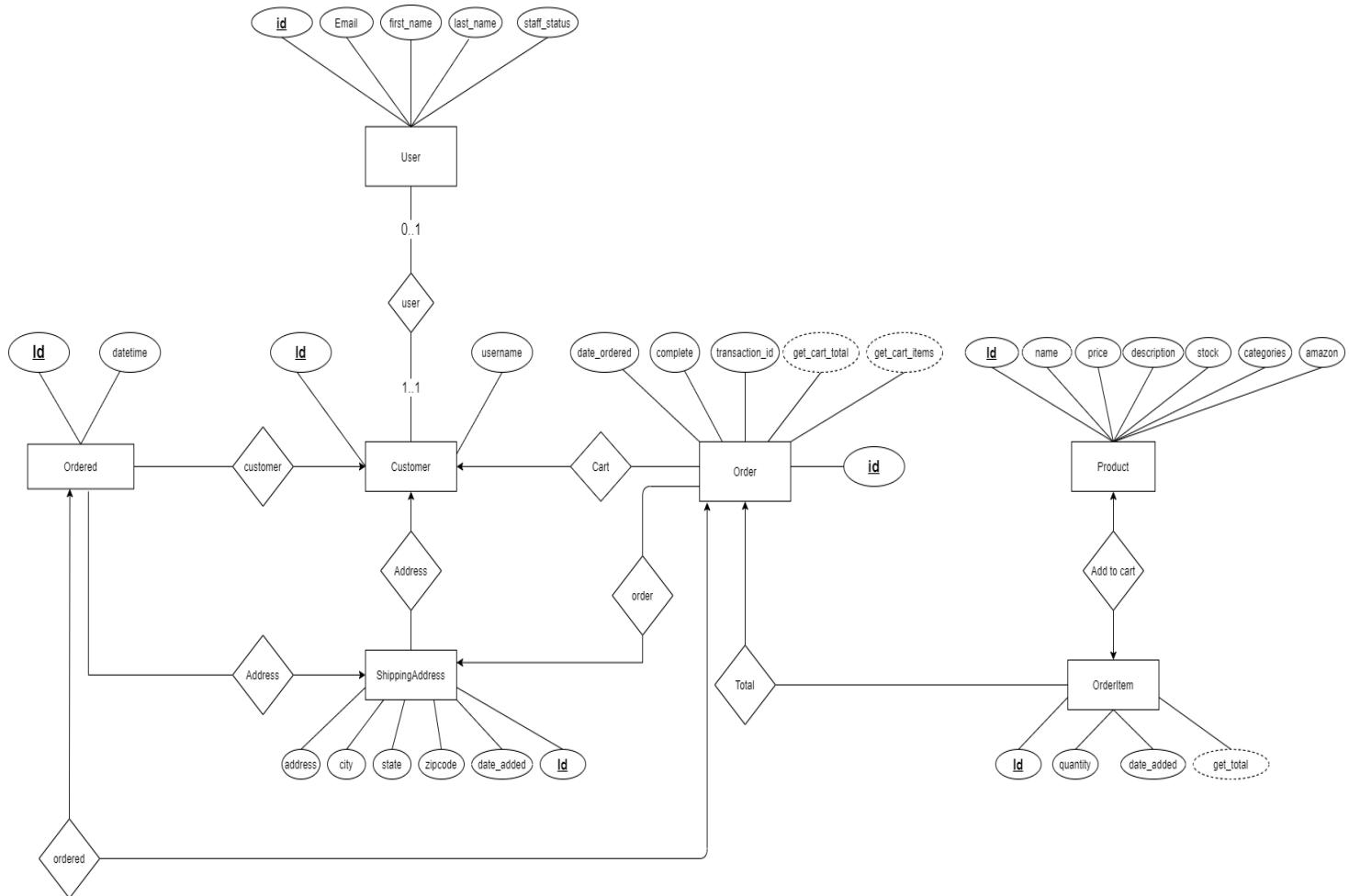
The system will be accessible through any web browser with JavaScript enabled, user should have good internet connectivity and proper version of web browser in Mobile/P.C, O.S is not a barrier as long as it can handle web easily. Some of the best browsers to access: Chrome, Firefox, Brave, Safari and Opera.

2.5 Design and Implementation Constraints

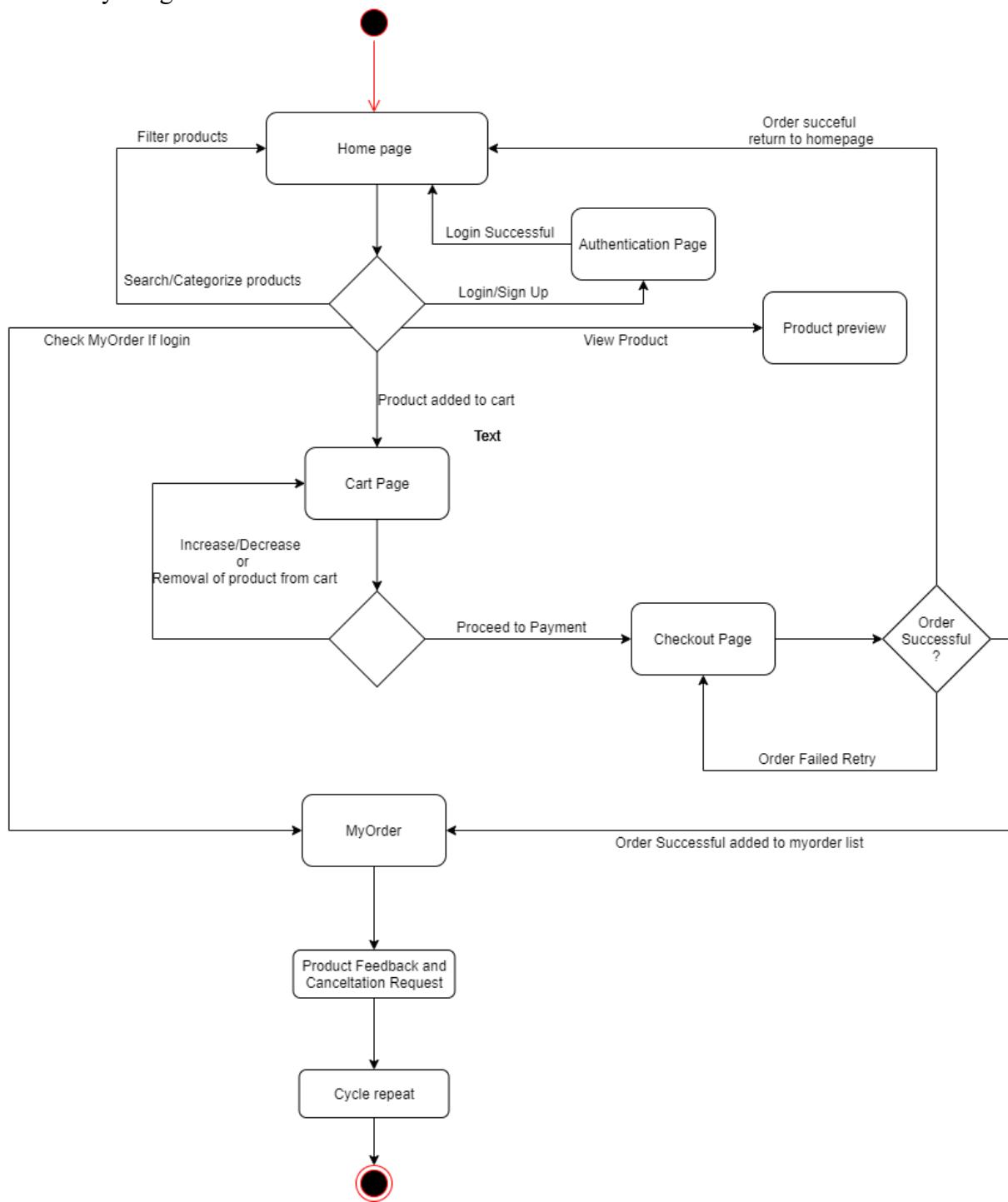
1. Data Flow Diagram:



2. Entity Relationship Diagram:



3. Activity Diagram:



2.6 Assumptions and Dependencies

- Server can take load of 10,000 users at any instance.
- User is familiar with browser interface.
- Minimum display aspect ratio is 5:3.

3. External Interface Requirements

3.1 User Interfaces (GUI Design)

User will be able to interact with the system using Mobile (Smartphone), Laptop or P.C. User can use touch or Keyboard mouse to interact with system functions.

3.2 Hardware Interfaces

All components able to be executed on any device which handle web browser and have minimum display aspect ratio 5:3.

- O.S: Ubuntu, Debian, Windows, Android, MAC OS, IOS, etc.
- Hard Disk: 1Gb able to handle web browser.
- RAM: 100 MB for E-com. + 250 MB for browser.
- Processor: Dual Core up to 1.5-2GHZ.

3.3 Software Interfaces

The System will be hosted using AWS (Amazon Web Server) and system uses MYSQL to store data in database and code store in GitHub Repository.

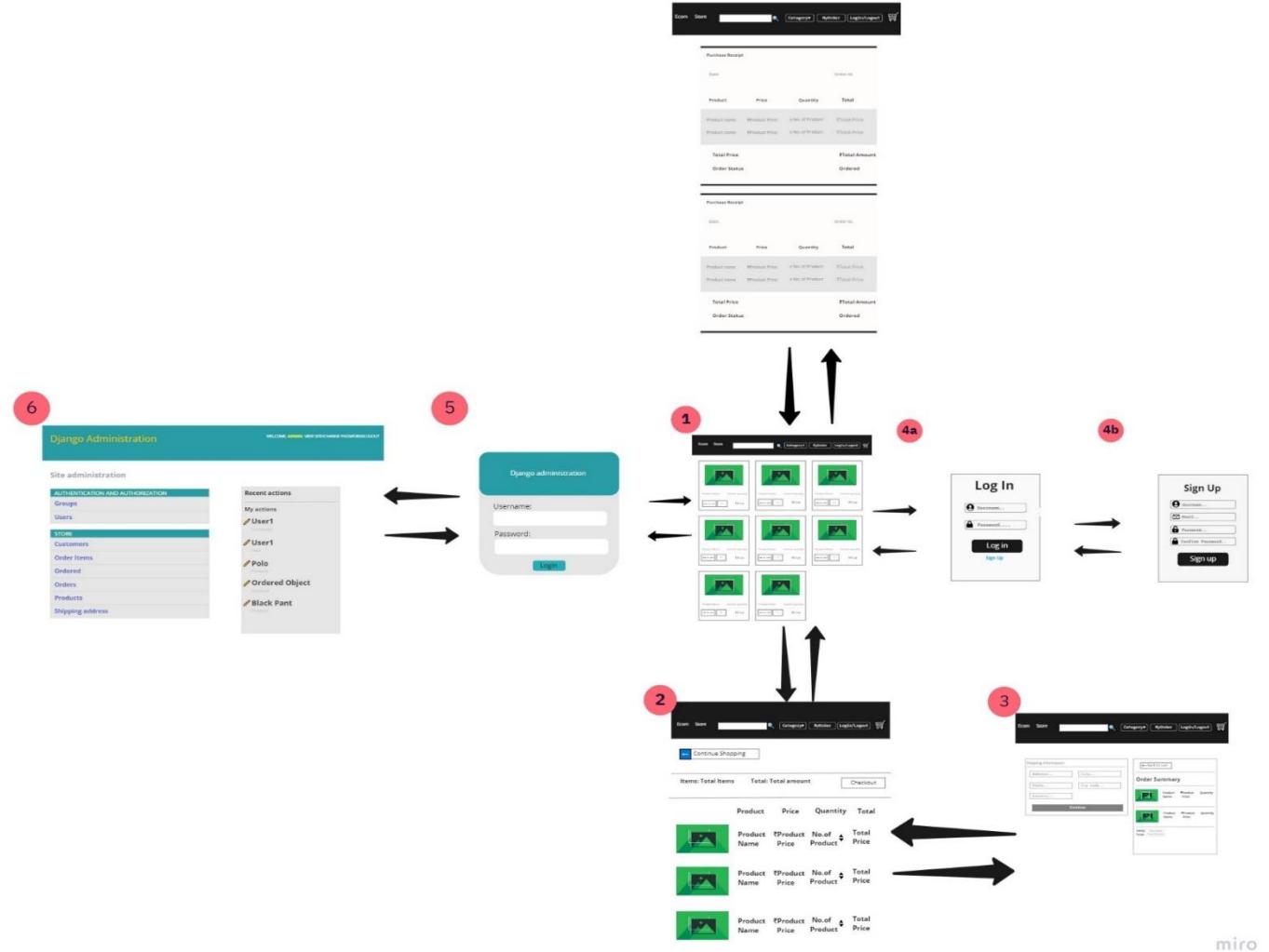
- Python and Cython Language
- MySQL DB
- AWS for cloud hosting.
- GitHub for cloud repository.

3.4 Communications Interfaces

System uses DRF (Django Rest Framework) which provide JSON (JavaScript Object Notation) response through HTTP which comes under TCP/IP protocol, Naïve user of the E-com. doesn't need to know about these things, they just need to know Django uses CSRF token which provides high security and reliability so their data is safe while interacting with website using internet through any web browser.

4. System Features

4.1 System Architecture



The project is divided into 6 major sections:

1. Home Page
2. Cart Page
3. Check out Page (No access to Anonymous user and web crawlers).
4. Login and Sign Up Page.
5. My Order Page (No access to Anonymous user and web crawlers).
6. Admin Page (Only Accessible by Staff level authority account).

4.1.1 Description and Priority

Admin: Admin account have highest priority it can access and modify every single function provided by project, it can also increase authority level of user account to staff account.

Staff user: Can maintain order status and feedback of user, priority below admin level.
Access permitted to administrator page

User: User can surf through website provide feedback for the product they purchased and request return or cancelation of product they purchased. No access to administrator page.

Cookie user and Web Crawler: Can only surf through home page and cart page can't purchase any product, lowest level priority, No access to Payment and My order page.

4.1.2 Action

1) Admin Page:

- Admin can add, delete and modify products, accounts, ordered products and address.
- Staff user can add, delete and modify products and ordered products.
- Each schema object is available to handle.

2) Home Page:

- Available product can be view here and product preview provide detail description of product.
- User and anonymous user can add product to cart from her.
- Search and categorized options are provided to quickly check whether or not the product is available.
- Most of the page are accessible from here.

3) Authentication:

- Login and Signup page is provided for authentication.
- If user forget the account recovery option is also there, user can recover their account by providing their corresponding email or phone number which they provided when signup.

4) Add to Cart Page:

- Product added from homepage to cart will be visible here.
- Added product can be increase or decrease in quantity or removed from cart.
- Summary of added products will be shown here i.e. total price of each product, total price of overall products and the total quantity.
- Login user can move to check out page from here.

5) Checkout Page:

- Login user is only allow on this page.
- Product payment can be done here (Online or COD).
- Address is needed for deliverable product.
- Once the order payment is successful user is redirect to homepage and receipt is generated in MyOrder section.
- Receipt is send on mail provided by user.

6) MyOrder Page:

- User order product receipt will be visible here.
- Delivery status of product will be shown in receipt.
- User can give product feedback once it is delivered.

- User can request for order cancelation and return from this page.
- Customer service number is also given here.

4.1.3 Functional Requirements

The software provides good graphical interface for the user any administrator can operate on the system, performing the required task such as create, update, and modify the details of products, orders and accounts.

Allows user and anonymous user to view products and add desire product to cart and purchase product option to user.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance of system is at its best because of Django we don't need to think about speed and performance, we just need to look for any bugs in the code and update the software time to time

5.2 Security Requirements

Django uses CSRF token and hashed user credential data, password of user is not even visible to Admin so user account is well secured and time to time updated software is mostly free of attack, for maintaining security we need to keep software update.

5.3 Software Quality Attributes

The Quality of the system is maintained in such a way so that it can be very user-friendly. The software quality attributes are assumed as under:

- Accurate and hence reliable.
- Secured.
- Fast Speed.
- Well maintained.

Conclusion: We have successfully prepared SRS document for E-commerce Shopping website.

Experiment No. 4

Aim: Structured Data Flow Analysis of E-commerce website

Problem Statement: To prepare Data Flow Diagram (DFD) for E-commerce shopping website

Theory:

Analysts use various tools to understand and describe the information system. One of the ways is using structured analysis.

What is Structured Analysis?

Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyse and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user.

It has following attributes –

- It is graphic which specifies the presentation of application.
- It divides the processes so that it gives a clear picture of system flow.
- It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
- It is an approach that works from high-level overviews to lower-level details.

Data Flow Diagrams (DFD) or Bubble Chart

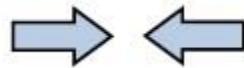
It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

- It shows the flow of data between various functions of system and specifies how the current system is implemented.
- It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
- Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
- It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.

Basic Elements of DFD

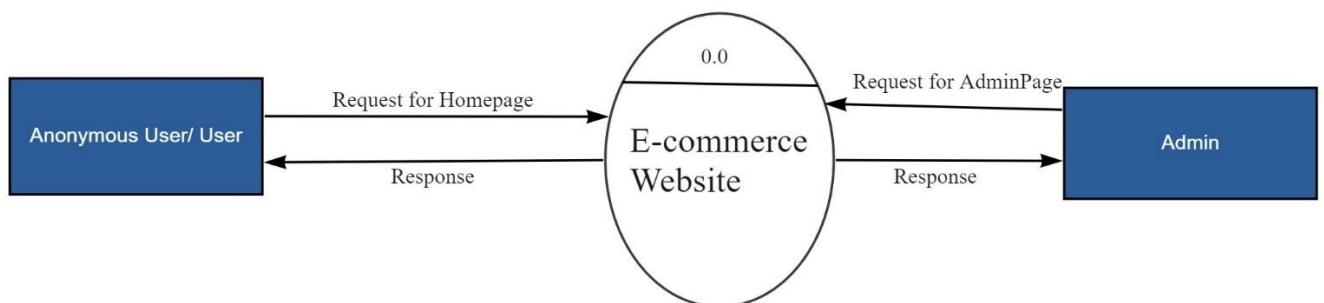
DFD is easy to understand and quite effective when the required design is not clear and the user wants a notational language for communication. However, it requires a large number of iterations for obtaining the most accurate and complete solution.

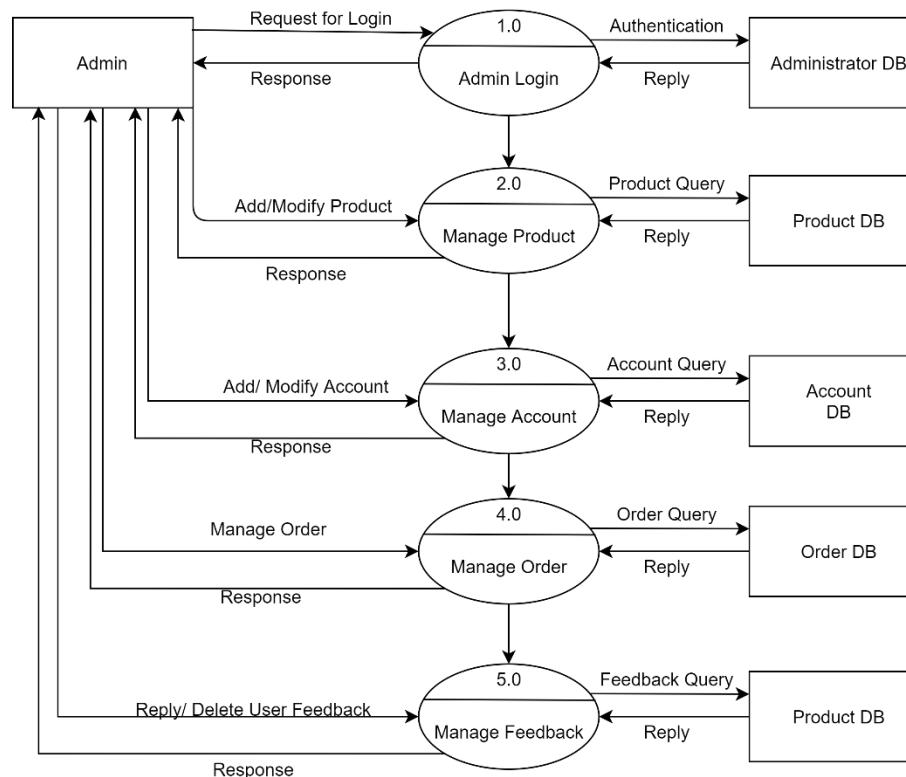
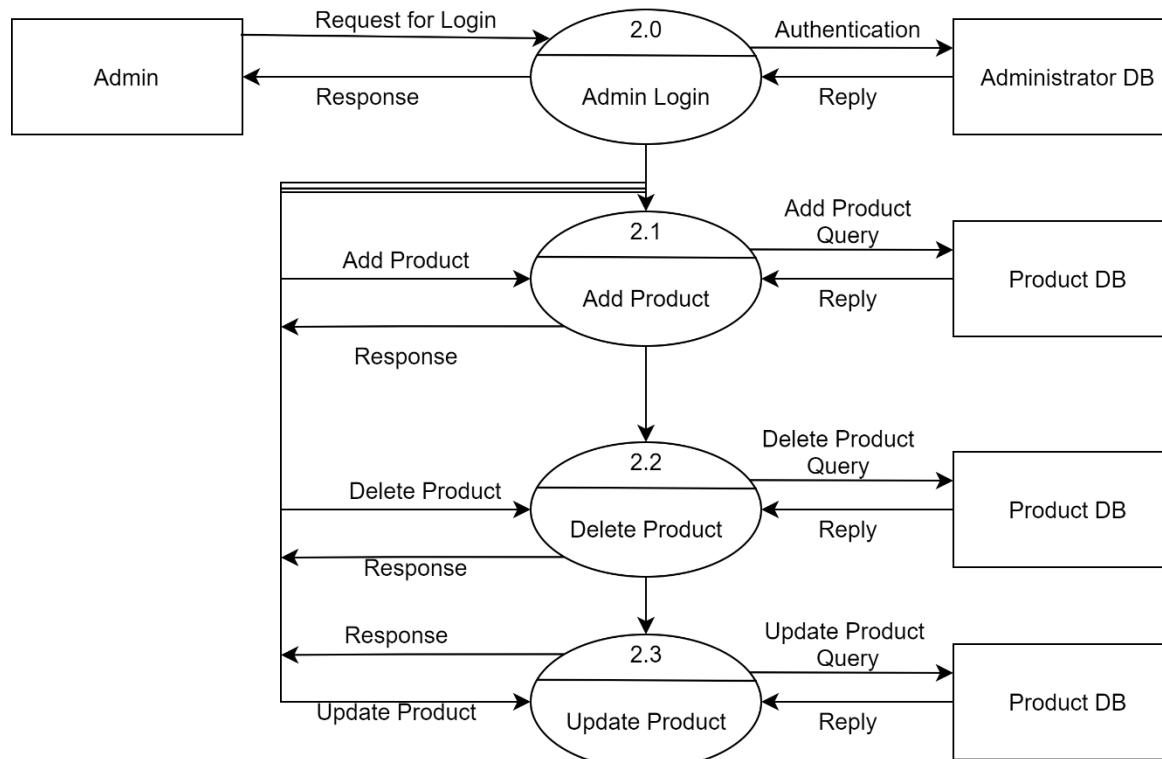
The following table shows the symbols used in designing a DFD and their significance –

Symbol Name	Symbol	Meaning
Square		Source or Destination of Data
Arrow		Data flow
Circle		Process transforming data flow
Open Rectangle		Data Store

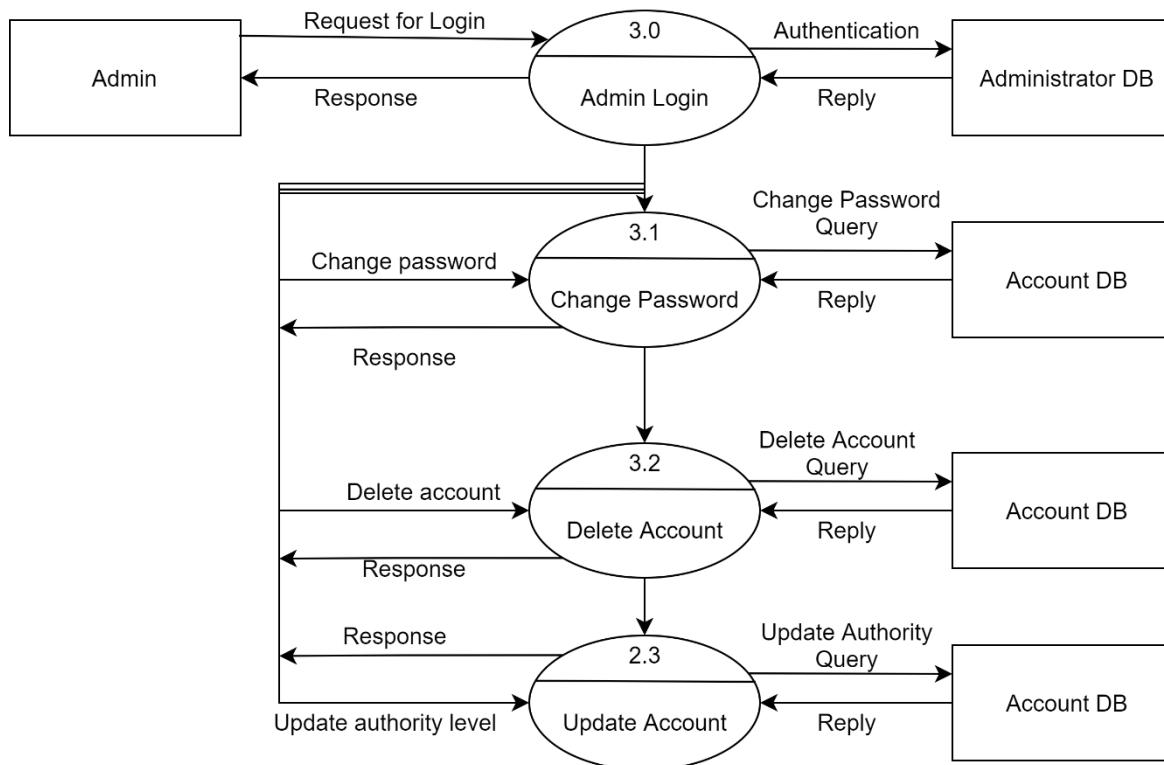
E-Commerce DFD:

Level 0:

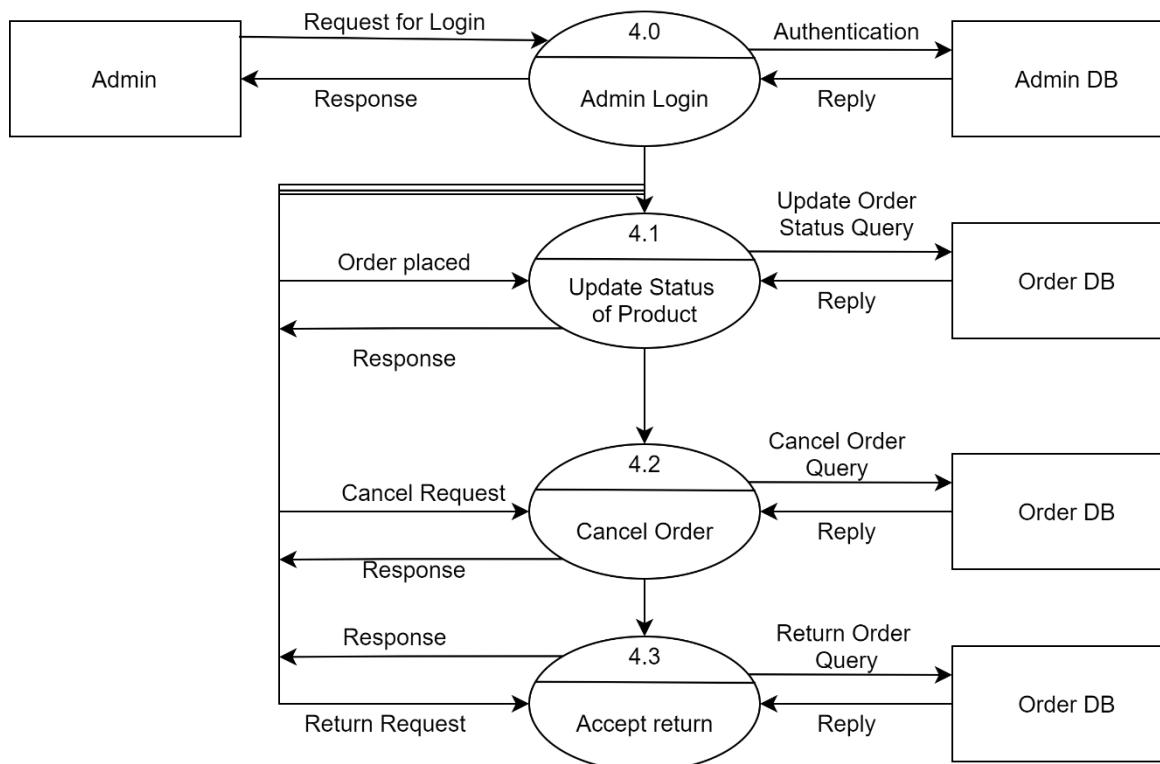


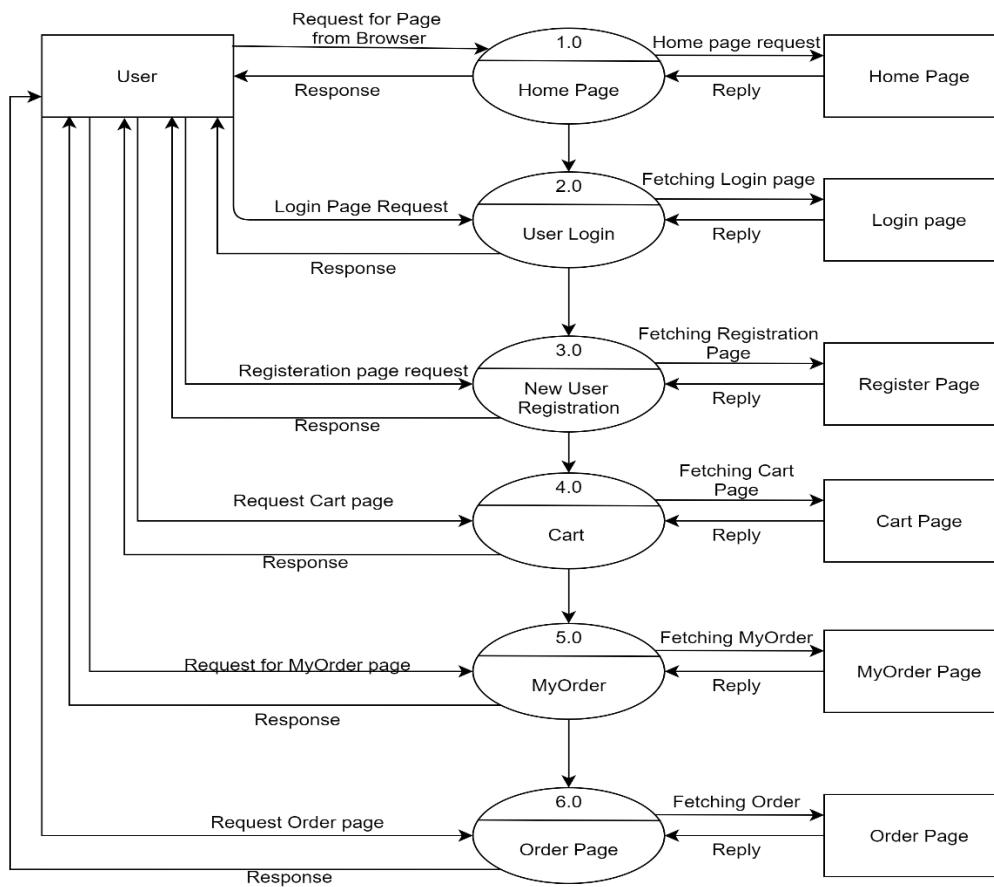
Admin 1st level DFD:**Admin DFD 2.0:**

Admin level 3.0:

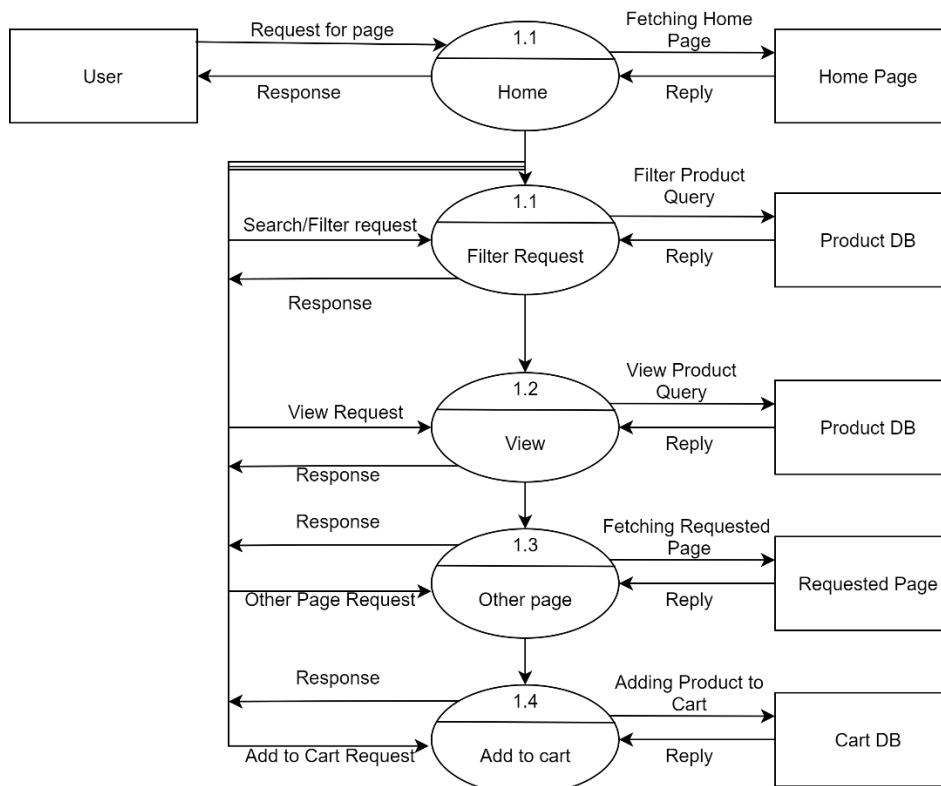


Admin level 4.0:

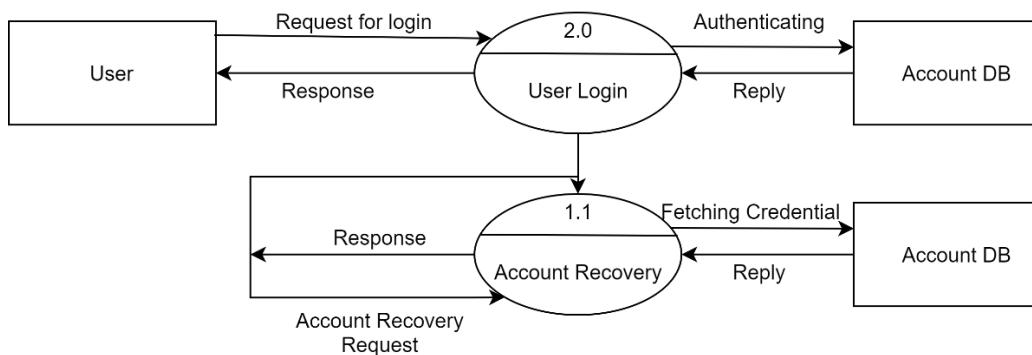


User 1st level DFD:

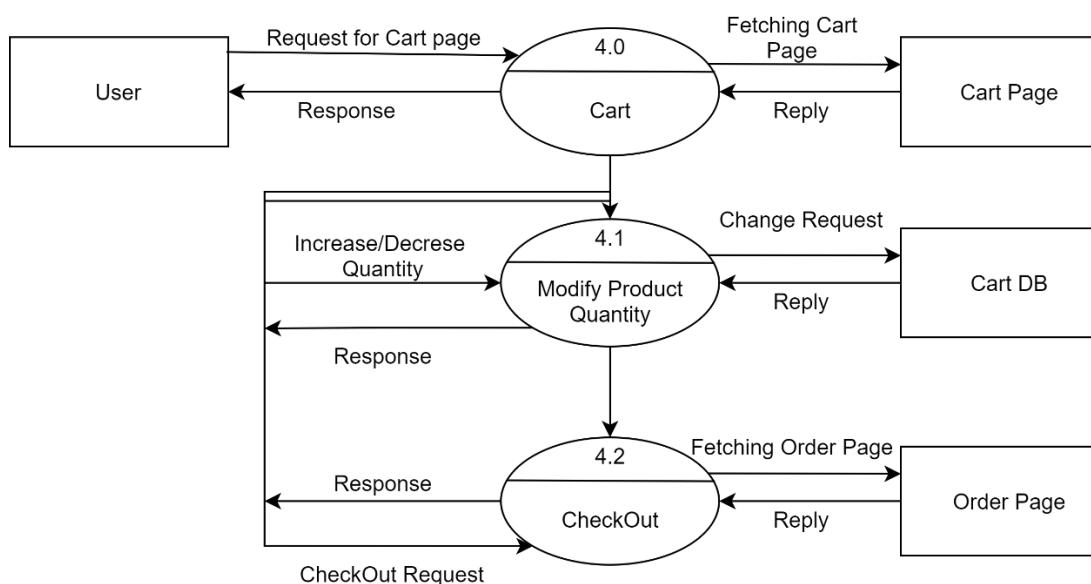
User level 1.0:



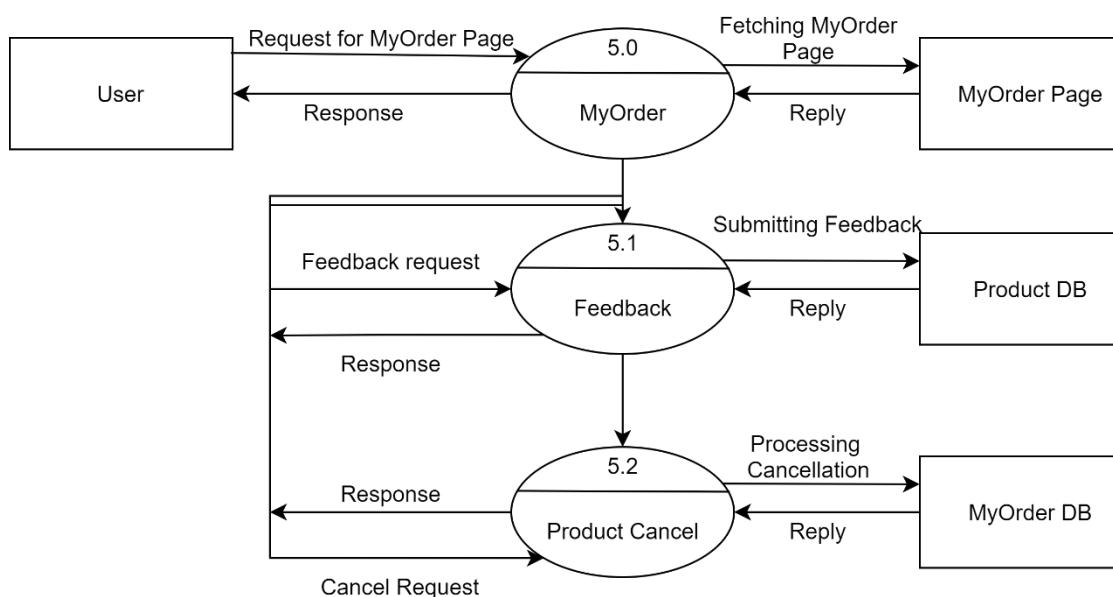
User level 2.0:



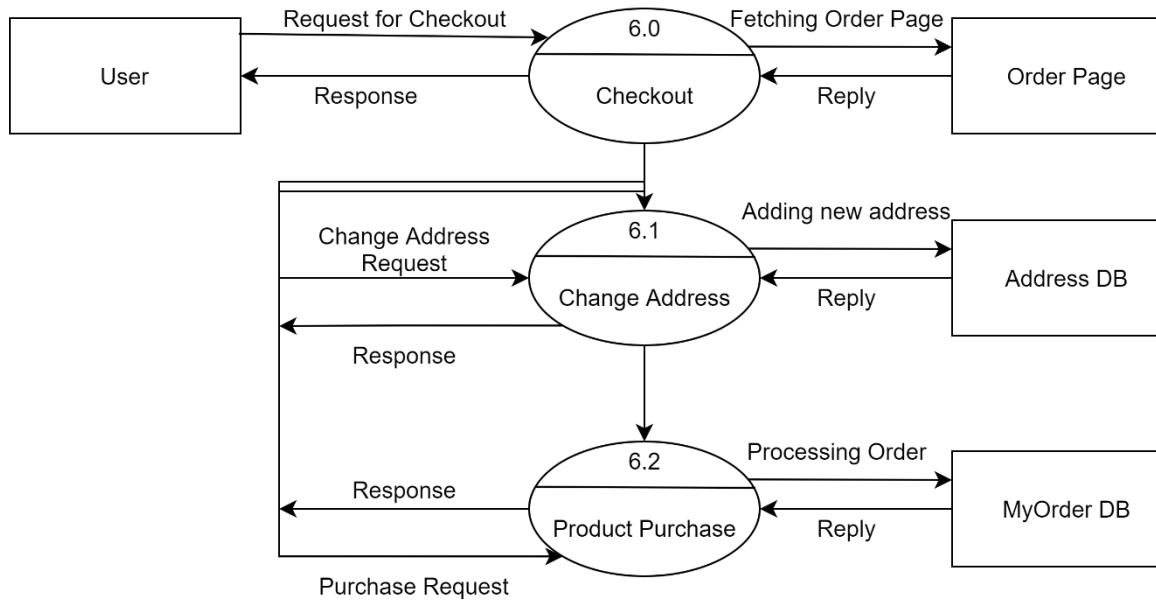
User level 4.0:



User level 5.0:



User level 6.0:



Conclusion: We have successfully learnt the concept of Structure Data Flow Analysis and how to implement it using Data Flow Diagram on E-commerce Shopping Website.

Experiment No. 5

Aim: Use of metrics to estimate the cost

Problem Statement: To estimate the cost using Function Point.

Theory:

Software Cost Estimation:

- Software cost estimation is the process of predicting the amount of effort required to build a software system and time develop it.
- Models provide mathematical algorithms to compute cost as a function of a number of variables such as size (using lines of code, function points, etc.) and/or complexity (using cyclomatic complexity, etc.).

Function Point:

Allan J. Albrecht initially developed Function Point Analysis in 1979 at IBM and it has been further modified by the International Function Point Users Group (IFPUG). FPA is used to make estimate of the software project, including its testing in terms of functionality or function size of the software product. However, functional point analysis may be used for the test estimation of the product. The functional size of the product is measured in terms of the function point, which is a standard of measurement to measure the software application.

Objectives of FPA

The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request. Further, it is used to measure the software project development along with its maintenance, consistently throughout the project irrespective of the tools and the technologies.

Following are the points regarding FPs

FPs of an application is found out by counting the number and types of functions used in the applications. Various functions used in an application can be put under five types, as shown in Table:

Types of FP Attributes

Measurements Parameters	Examples
1.Number of External Inputs(EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.
4. Number of internal files (ILF)	Databases and directories

5. Number of external interfaces (EIF)	Shared databases and shared routines.
--	---------------------------------------

All these parameters are then individually assessed for complexity.

Value Adjustment Factor:

The value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. Each characteristic has associated descriptions to determine the degrees of influence.

Rating:

The degrees of influence range on a scale of zero to five, from no influence to strong influence. Each characteristic is assigned the rating based upon detail descriptions provided by the IFPUG 4.1 Manual. Their ratings are:

0 - Not present, or no influence

1 - Incidental influence

2 - Moderate influence

3 - Average influence

4 - Significant influence

5 - Strong influence throughout

Value adjustment equation

$$VAF = 0.65 + [(\sum_{i=1}^{14} Ci)/100]$$

GSC's at a Glance:

General System Characteristic		Brief Description
1	Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?
2	Distributed data processing	How are distributed data and processing functions handled?

3	Performance	Did the user require response time or throughput?
4	Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?
5	Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?
6	On-Line data entry	What percentage of the information is entered On-Line?
7	End-user efficiency	Was the application designed for end-user efficiency?
8	On-Line update	How many ILF's are updated by On-Line transaction?
9	Complex processing	Does the application have extensive logical or mathematical processing?
10	Reusability	Was the application developed to meet one or many user's needs?
11	Installation ease	How difficult is conversion and installation?
12	Operational ease	How effective and/or automated are start-up, back up, and recovery procedures?
13	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?
14	Facilitate change	Was the application specifically designed, developed, and supported to facilitate change?

Calculating FP:

Computing Function points (FPs):-					
Information Domain value	Count	Weight factors			=
		Simple	Average	Complex	
1. External Inputs(EI)	20	3	4	6 =	80
2. External Outputs(EO)	20	4	5	7 =	100
3. External Inquiries(EQ)	13	3	4	6 =	52
4. Internal Logical Files(IF)	19	7	10	15 =	190
5. External Interface Files(EIF)	75	5	7	10 =	21
Count Total					→ 443
Value adjustment factors:-					
Factor	Value				
1) Backup & Recovery	5				
2) Data Communications	3				
3) Distributed Processing	0				
4) Performance Critical	4				
5) Existing operating Environment	4				
6) Online data entry	5				
7) Input transaction over Multiple screen	5				
8) Master files updated online	4				
9) Information domain values complex	3				
10) Internal processing Complex	4				
11) Code designed for reuse	2				

Factor	Value
12) Conversion installation in design	2
13) Multiple installations	5
14) Application design for change	5
ΣF_i	51
$F.P = \text{Count-total} * \text{Value-Adjustment-Factor (CAF)}$	
C.A.F also known as Complexity-Adjustment Factor	
$C.A.F = 0.06 + 0.65 + 0.01 * (\Sigma F_i)$	
$C.A.F = 1.16$	
$\therefore F.P = 443 * 1.16 = 513.88$	

Conclusion: We have successfully estimated the cost of project using Function Point Metrics

EXPERIMENT NO. 6

Aim: Scheduling and tracking the project

Problem Statement: To schedule and track the E-commerce website project.

Theory:

Project Scheduling:

Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software project manager wants to do the following:

1. Identify all the functions required to complete the project.
2. Break down large functions into small activities.
3. Determine the dependency among various activities.
4. Establish the most likely size for the time duration required to complete the activities.
5. Allocate resources to activities.
6. Plan the beginning and ending dates for different activities.
7. Determine the critical path. A critical way is the group of activities that decide the duration of the project.

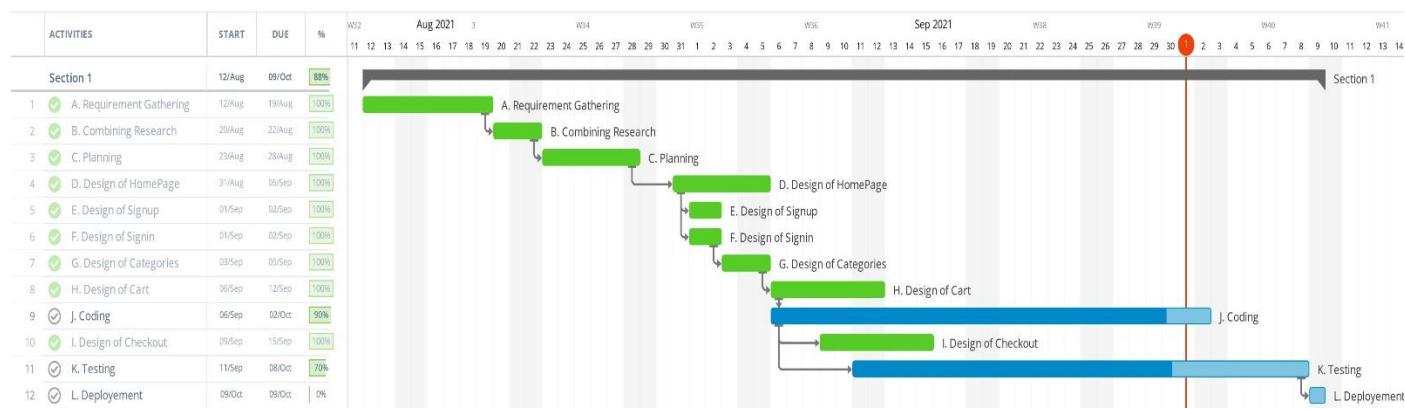
The first method in scheduling a software plan involves identifying all the functions required to complete the project. A good judgment of the intricacies of the project and the development process helps the supervisor to identify the critical role of the project effectively. Next, the large functions are broken down into a valid set of small activities which would be assigned to various engineers. The work breakdown structure formalism supports the manager to breakdown the function systematically after the project manager has broken down the purpose and constructs the work breakdown structure; he has to find the dependency among the activities. Dependency among the various activities determines the order in which the various events would be carried out. If an activity A necessary the results of another activity B, then activity A must be scheduled after activity B. In general, the function dependencies describe a partial ordering among functions, i.e., each service may precede a subset of other functions, but some functions might not have any precedence ordering describe between them (called concurrent function). The dependency among the activities is defined in the pattern of an activity network.

Once the activity network representation has been processed out, resources are allocated to every activity. Resource allocation is usually done using a Gantt chart. After resource allocation is completed, a PERT chart representation is developed. The PERT chart representation is useful for program monitoring and control. For task scheduling, the project plan needs to decompose the project functions into a set of activities. The time frame when every activity is to be performed is to be determined. The end of every action is called a milestone. The project manager tracks the function of a project by audit the timely completion of the milestones. If he examines that the milestones start getting delayed, then he has to handle the activities carefully so that the complete deadline can still be met.

Gantt chart:

E-Commerce site
Read-only view, generated on 01 Oct 2021

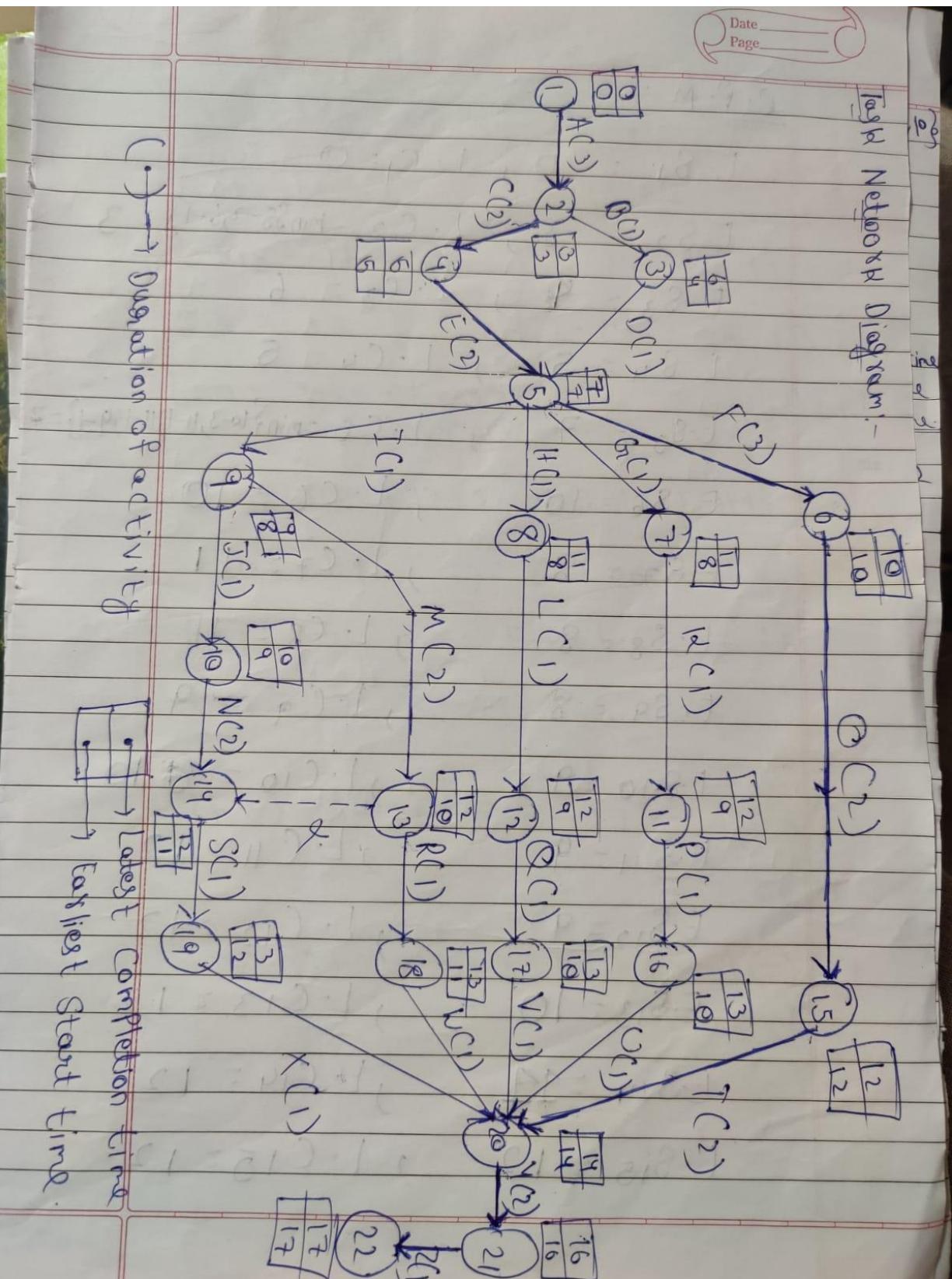
Instagantt



CPM and PERT:

Task table:		
Activity	Immediate Predecessor(s)	Duration (in Month)
Req. Gathering (A)	-	3
Comb. Research (B)	A	1
Planning (C)	A	2
Analysis (D)	B	1
Design H.P (E)	C	2
Coding H.P (F)	D, E	3
Design L.P (G)	D, E	1
Design S.P (H)	D, E	1
Design C.P (I)	D, E	1
Design O.P (J)	I	1
Coding L.P (K)	G	1
Coding S.P (L)	H	1
Coding C.P (M)	I	2
Coding O.P (N)	J	2
INPUT H.P (O)	F	2
U.T L.P (P)	K	1
U.T S.P (Q)	L	1
U.T C.P (R)	M	1
U.T O.P (S)	N	1
T.T H.P with other (T)	O	2
T.T L.P (U)	P	1
I.T S.P (V)	Q	1
I.T C.P (W)	R	1
I.T O.P (X)	S	1
System test (Y)	T, U, V, W, X	2
Deployment (Z)	Y	1

Task Network Diagram:-



C.P.M

$$E.S_1 = 0, L.C_1 = 0$$

$$E.S_2 = 3, L.C_2 = \min\{5-2, 6-1\} = 3$$

$$E.S_3 = 4, L.C_3 = 6$$

$$E.S_4 = 5, L.C_4 = 5$$

$$E.S_5 = 7, L.C_5 = \min\{10-3, 11-1, 11-1, 9-1\} = 7$$

$$E.S_6 = 10, L.C_6 = 10$$

$$E.S_7 = 8, L.C_7 = 11$$

$$E.S_8 = 8, L.C_8 = 11$$

$$E.S_9 = 8, L.C_9 = 9$$

$$E.S_{10} = 9, L.C_{10} = 10$$

$$E.S_{11} = 9, L.C_{11} = 12$$

$$E.S_{12} = 9, L.C_{12} = 12$$

$$E.S_{13} = 10, L.C_{13} = 12$$

$$E.S_{14} = 11, L.C_{14} = 12$$

$$E.S_{15} = 12, L.C_{15} = 12$$

$$E.S_{16} = 10$$

$$L.C_{16} = 13$$

$$E.S_{17} = 10$$

$$L.C_{17} = 13$$

$$E.S_{18} = 11$$

$$L.C_{18} = 13$$

$$E.I_9 = 12$$

$$L.C_{19} = \cancel{13}$$

$$E_{20} = \max \{ E_{15} + 2, E_{16} + 1, E_{17} + 1, E_{18} + 1, \\ E_{19} + 1 \}$$

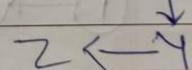
$$E_{20} = \max \{ 12 + 2, 10 + 1, 10 + 1, 11 + 1, 12 + 1 \}$$

$$E_{20} = 14. \quad , \quad L.C_{20} = 14$$

$$E_{21} = 16 \quad , \quad E_2 + L.C_{21} = 16$$

$$E_{22} = 17 \quad \leftarrow L.C_{22} = 17$$

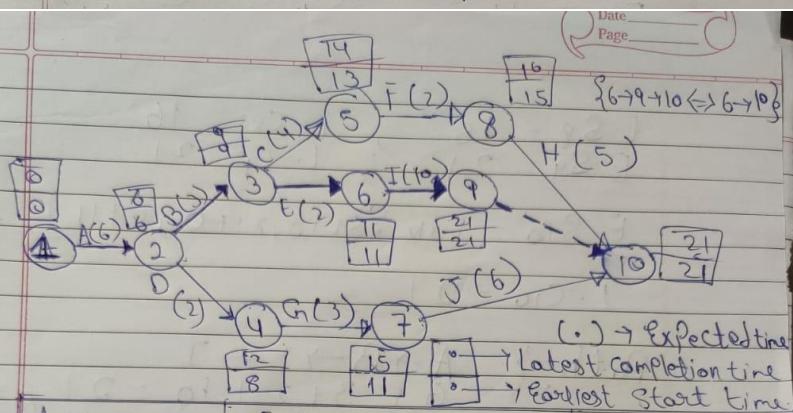
Path $\Rightarrow A \rightarrow C \rightarrow E \rightarrow F \rightarrow O \rightarrow T$



$$\Rightarrow 3 + 2 + 2 + 3 + 2 + 2 + 2 + 1 \leftarrow 17.$$

PERT:-

Activity	Predecessor	Time (in Days)		
		O	M	P
A	-	5	6	7
B	A	11	13	15
C	B	1	4	7
D	A	1	2	3
E	B	1	2	3
F	C	1	2	3
G	D	2	2	5
H	F	3	4	6
I	E, F	5	10	13
J	G	2	6	8



Activity	Expected time	Variance
A	$0 + 4M + P$ 6	$\sigma^2 = [P - O]^2$ $\frac{6}{6}$
B	3	0.11111
C	4	1
D	2	0.11111
E	2	0.11111
F	2	0.11111
G	$2.5 \approx 3$	0.25
H	$\frac{14.6667}{7} = 2$	0.25
I	$\frac{19.6667}{7} = 3$	1.77778
J	$\frac{25.6667}{7} = 4$	1

$$\begin{aligned}
 E.S_1 &= 0 & L.C_1 &= 0 \\
 E.S_2 &= 6 & L.C_2 &= \min\{8-2, 9-3\} = 6 \\
 E.S_3 &= 9 & L.C_3 &= \min\{11-2, 19-4\} = 9 \\
 E.S_4 &= 8 & L.C_4 &= 12 \\
 E.S_5 &= 13 & L.C_5 &= 14 \\
 E.S_6 &= 11 & L.C_6 &= 11 \\
 E.S_7 &= 11 & L.C_7 &= 15
 \end{aligned}$$

$$E.S_8 = 15 ; L.C_8 = 16$$

$$E.S_9 = 21 ; L.C_9 = 21$$

$$E.S_{10} = \max \{ S_5 + 15, 21 + 0, 6 + 11 \} = 21$$

$$L.C_{10} = 21$$

Path $\Rightarrow A \rightarrow B \rightarrow E \rightarrow I$

$$\Rightarrow 6 + 3 + 2 + 10 = 21 //$$

Conclusion: We have successfully schedule and track the E-commerce shopping website project using Gantt chart, CPM and PERT method.

EXPERIMENT NO. 7

Aim: To write test cases for black box testing

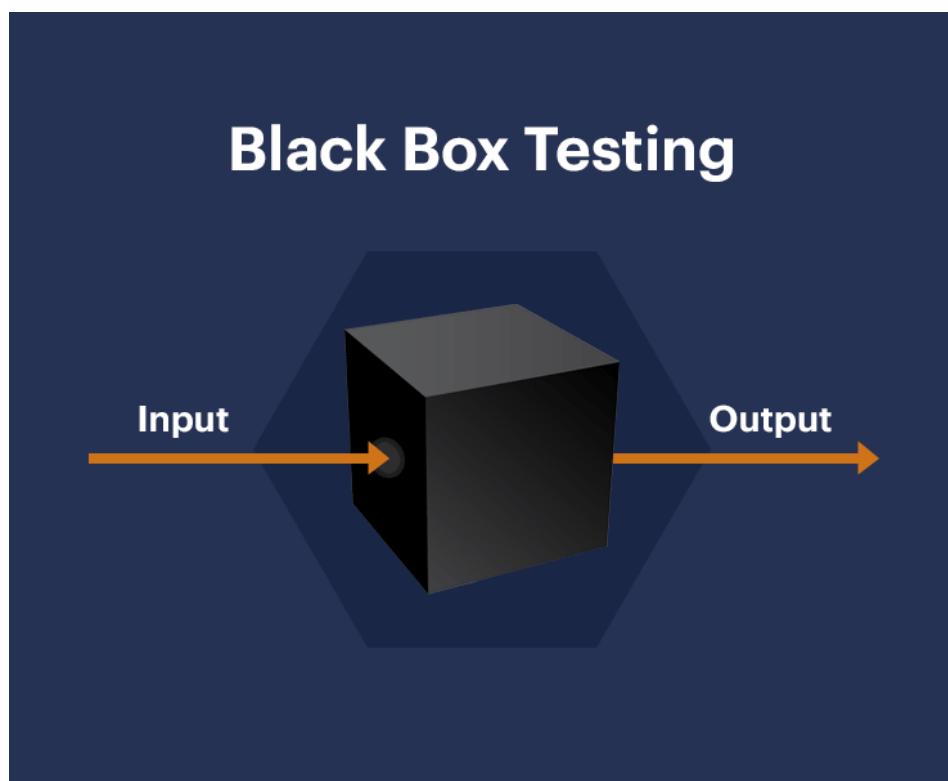
Requirements: Windows/MAC/LINUX O.S, Compatible version of JDK, Eclipse and Selenium

Problem statement: To design test cases for black box testing and implement using Eclipse and Selenium.

Theory:

What is Black Box Testing?

Black box testing is also known as opaque technique, behavioral testing, functional testing, and closed-box testing is a type of **software testing**. When we enter a topic to search on the search engine, we type out the topic and enter search. The result is obtained thereafter without looking at the internal structure or working. This is an appropriate example of black-box testing.



How is Black Box Testing done?

The steps for carrying out Black Box Testing are as follows:

- At first, the application to be tested is studied to find out the requirements and specifications. The SRS (Software Requirement Specification) document should be maintained with accuracy.
- The inputs and test scenarios are evaluated. Efficient and time-saving techniques are incorporated.

- Test cases are generated. These test cases are made in such a way that the input range is maximum.
- The test cases are then processed to obtain the output. The generated output is compared with the expected output to understand the success of the result.
- If there are unsuccessful steps, they are sent to the software development teams for fixing.
- The defects are fixed.
- Run the tests again for confirmation.

Techniques of Black Box Testing

Equivalence Partitioning:

As the name suggests, the inputs are partitioned into groups or more literally partitions. Only one input from every group is tested to find the results. The inputs are usually numeric values or a set of values or Boolean conditions. For example, if the field accepts an integer in the range 1 and 20 then:

Valid Equivalence Class Partition: 1 to 20 inclusive.

Invalid Equivalence Class Partition: Less than 1 or more than 20, decimal numbers or alphabets and other non-numeric characters.

Boundary Value Testing:

In boundary value analysis the answers are within specific boundaries. The two ends, the inner and the outer limits are considered in this type of testing. For example, an offer is valid for customers between the ages of 18 and 30 only. Therefore other values such as 17, 18, 30 or 31 can be tested to check whether the inputs are accepted.

Test Procedure:

The test procedure of black box testing is a kind of process in which the tester has specific knowledge about the software's work, and it develops test cases to check the accuracy of the software's functionality.

It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function. A tester knows about the definite output of a particular input, but not about how the result is arising. There are various techniques used in black box testing for testing like decision table technique, boundary value analysis technique, state transition, All-pair testing, cause-effect graph technique, equivalence partitioning technique, error guessing technique, use case technique and user story technique.

Test Cases:

Test cases are created considering the specification of the requirements. These test cases are generally created from working descriptions of the software including requirements, design parameters, and other specifications. For the testing, the test designer selects both positive test scenario by taking valid input values and adverse test scenario by taking invalid input values

to determine the correct output. Test cases are mainly designed for functional testing but can also be used for non-functional testing. Test cases are designed by the testing team, there is not any involvement of the development team of software.

A basic example of test case design:

Title: Login to the website or app

Description: User should be able to successfully log in to their account on the website/app

Preconditions: User must already be registered and use their correct login details

Assumptions: They are using a supported device or browser to log in

Test Steps:

1. Open website or app
2. Enter the username and password in the appropriate fields
3. Click “login”

Expected Result: The user should log in successfully.

To execute following test case we need a tool which can perform the black box testing. There is variety of tool which are available online. But for our practical Selenium tool has been used.

ECLIPSE:

Eclipse is an integrated development environment that is used in computer programming.

It is the mostly widely use Java IDE and contains a base workspace and an extensible plug-in system for customising the environment.

The platform has been designed to build integrated web and application development tooling.

It is designed to not offer a huge amount of end user functionality but the value of the platform comes with its ability to encourage the rapid development of integrated features based on a plug-in model.

Eclipse provides a common user interface model for working with tools and is designed to run on multiple operating systems.

JUnit is a Java unit testing framework that is useful for creating scalable and repeatable tests. It is provided in Eclipse and can be use with Selenium web driver.

Selenium:

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others.

A browser-driver then executes these scripts on a browser-instance on your device (more on this in a moment).

Selenium WebDriver also known as Selenium 2.0, WebDriver executes test scripts through browser-specific drivers.

Selenium vs. its Counterparts

Features	HP QTP	IBM RFT	TestComplete	Selenium
License	Required	Required	Required	Open Source
Cost	High	High	High	Free
Customer support	Yes	Yes	Yes	Yes; Open source community
Coding skills	Low	Low	High	Very High
Environment support	Only Windows	Only Windows	Windows only (7, Vista, Server 2008 or later OS)	Windows, Linux, Mac
Language support	VB Script	Java and C#	VB Script, JS Script, Delphi Script, C++ & C#	Java, C#, Ruby, Python, Perl & PHP

There are several tools to test other types of applications, but testing of dynamic web applications is done best with Selenium.

Versions and Suite of Tools

As of April – 2018, the latest release of the tool is Selenium Version 3.12.0.

The Selenium suite comprised the following four components:



Selenium RC, however, was merged with WebDriver and launched anew as Selenium WebDriver with better functionalities.

Output:

How to Set-up Selenium?

The following three software are prerequisite to begin using Selenium.

I. Java → Programming Language to write scripts



II. Eclipse → Environment to compile and run scripts



III. Selenium → Framework for testing web applications

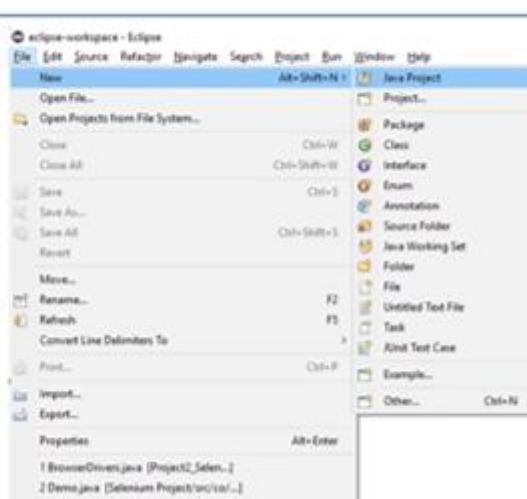


The screenshot shows the SeleniumHQ website's 'Downloads' page. On the left, there are links for 'Selenium Downloads', 'Previous Releases', 'Source Code', and 'Maven Information'. Below these are sections for 'Donate to Selenium' (with PayPal and a 'Donate' button) and 'Through sponsorship' (listing 'Selenium Sponsors' and 'See who supports the Selenium project'). A 'BrowserStack' logo is also present. The main content area is titled 'Downloads' and contains information about the Selenium Standalone Server, including download links for version 3.2.2.2 and a note about running it through emulation. It also mentions the Internet Explorer Driver Server and Selenium Client & WebDriver Language Bindings.

How to Set-up Selenium?

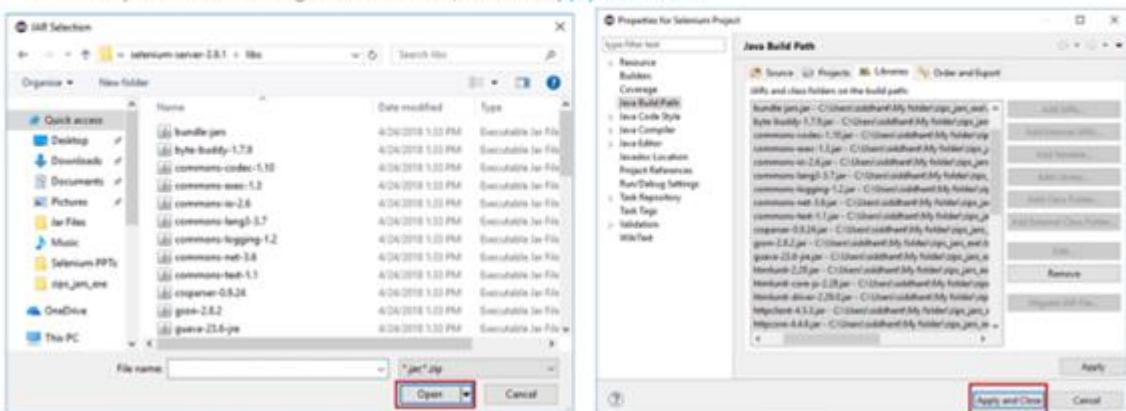
- After Eclipse is launched, go to

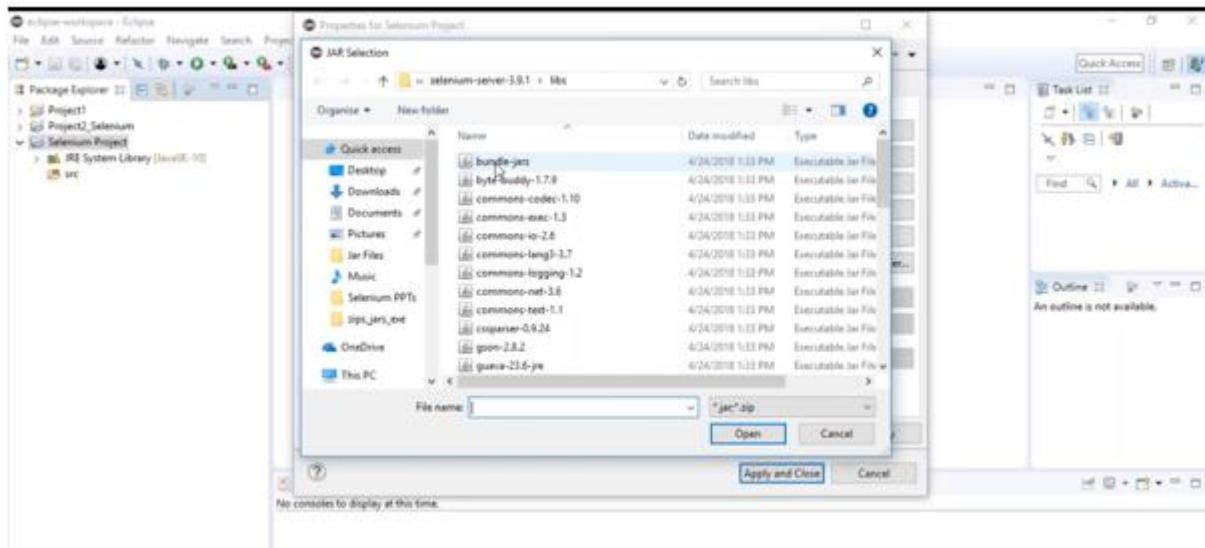
File → New → Java Project.



How to Set-up Selenium?

- Find all the JAR files that we downloaded, select them and click on **Open**.
- After you're done adding all the JAR files, Click on **Apply and Close**.



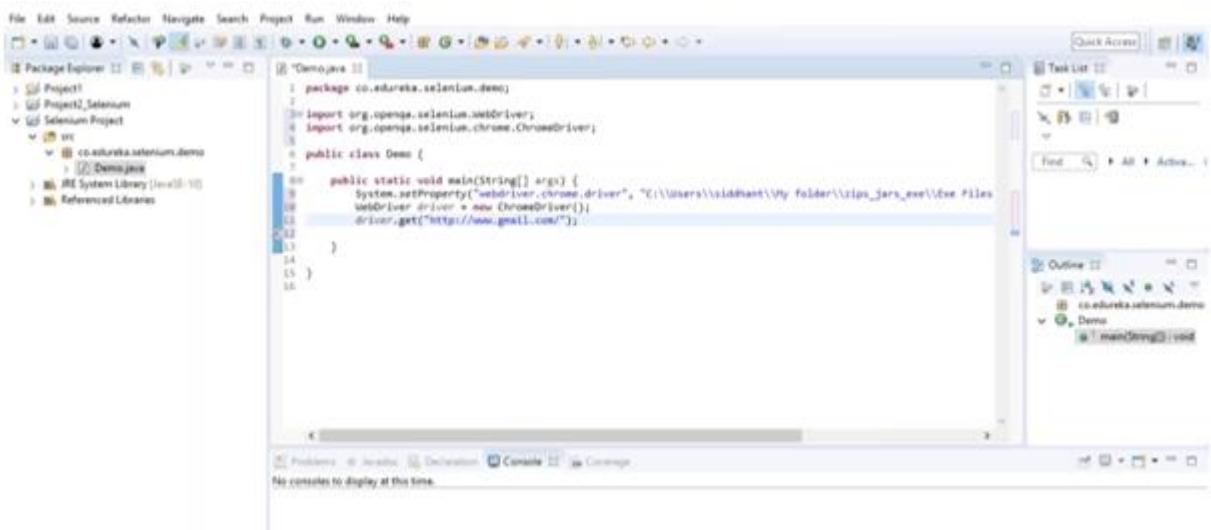


Hands-on in Selenium

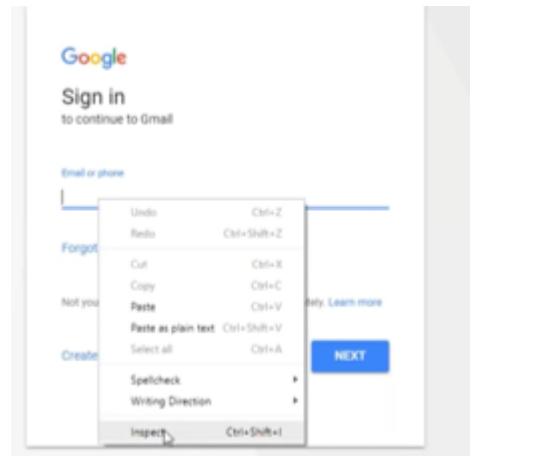
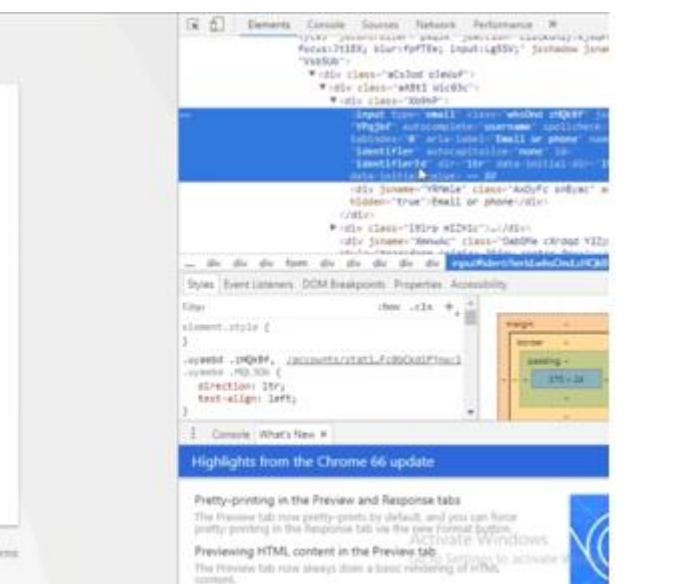
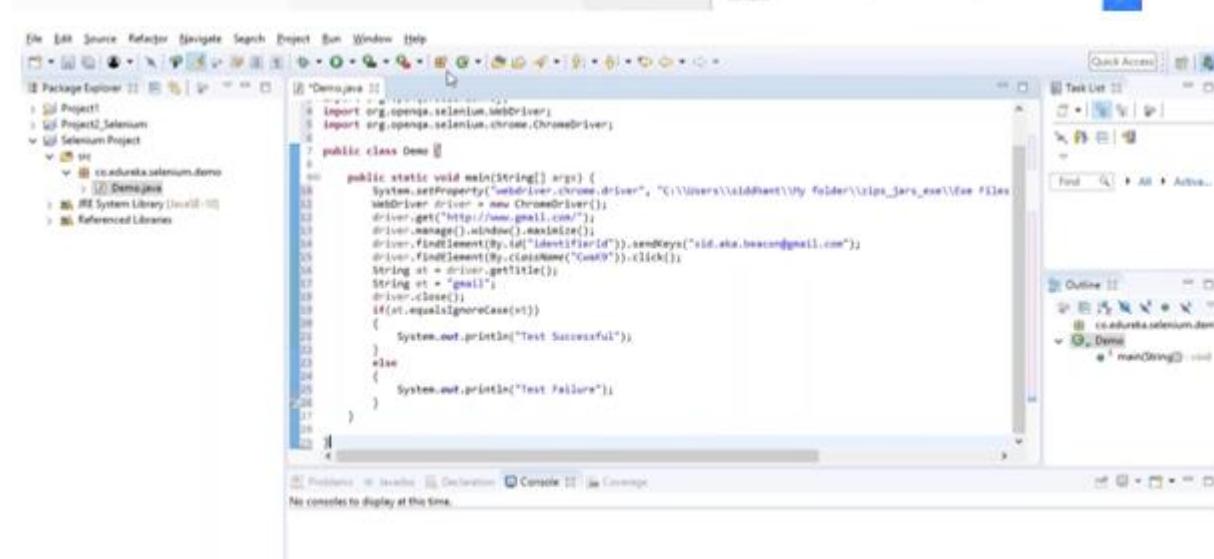
Now that the prerequisites have all been installed and Selenium has been configured for the Eclipse IDE, how about we automate and test Gmail using Google Chrome?

```

1 package co.edureka.selenium.demo;
2
3 import org.openqa.selenium.By;
4
5 public class Demo {
6
7     public static void main(String[] args) throws InterruptedException {
8         System.setProperty("webdriver.chrome.driver", "C:\\users\\siddhant\\My folder\\selenium\\exe\\bin\\chromedriver.exe");
9         WebDriver driver = new ChromeDriver();
10        driver.get("https://www.gmail.com");
11        driver.findElement(By.id("username")).sendKeys("siddhant.siddhant@gmail.com");
12        driver.findElement(By.id("password")).sendKeys("password123");
13        driver.findElement(By.id("signIn")).click();
14        Thread.sleep(3000);
15        driver.findElement(By.className("GmailLogo")).click();
16        Thread.sleep(1000);
17        String et = driver.getTitle();
18        String et1 = "gmail";
19        System.out.println(et);
20        driver.close();
21        if(et.equalsIgnoreCase(et1))
22        {
23            System.out.println("Test successful");
24        }
25        else
26        {
27            System.out.println("Test failure");
28        }
29    }
30 }
```



The following code checks whether Gmail account recognises the registered user id and password. If the registered user id and password is typed then result gives the test is successful otherwise it is unsuccessful.

```

public class Demo {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\siddhant\\My Folder\\zips_jars_exec\\bin Files\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.gmail.com");
        driver.manage().window().maximize();
        driver.findElement(By.id("identifierId")).sendKeys("sidd.saka.bear@gmail.com");
        driver.findElement(By.className("Continue")).click();
        String st = driver.getTitle();
        String st1 = "gmail";
        driver.close();
        if(st.equalsIgnoreCase(st1))
        {
            System.out.println("Test Successful");
        }
        else
        {
            System.out.println("Test Failure");
        }
    }
}

```

```

1 package co.edureka.selenium.demo;
2
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5
6 public class Demo {
7
8     public static void main(String[] args) {
9         System.setProperty("webdriver.chrome.driver", "C:\\Users\\vishwanath\\My Folder\\selenium_jars\\exe\\bin\\chromedriver.exe");
10        WebDriver driver = new ChromeDriver();
11        driver.get("http://www.gmail.com");
12        driver.manage().window().maximize();
13        driver.findElement(By.id("identifierId")).sendKeys("sidd.sha.bear@gmail.com");
14        driver.findElement(By.className("GmailLogo")).click();
15        String title = driver.getTitle();
16        String et = "gmail";
17        driver.close();
18        if(et.equalsIgnoreCase(title))
19        {
20            System.out.println("Test Successful");
21        }
22        else
23        {
24            System.out.println("Test Failure");
25        }
26    }
27 }
28

```

Terminal Output:

```

terminated: Demo (D:\Java Application\Java\Program Files\Java\java-10\bin\java.exe) [May 11, 2018, 5:08:30 PM]
Starting ChromeDriver 2.36.548479 (e522004894c7bba4d82172d9ef4f5d688c91) on port 42179
Only local connections are allowed.
Mon May 14, 2018 5:08:34 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Test Successful

```

Conclusion: We have successfully design test cases for black box testing and tested them using Eclipse and Selenium in Java.

EXPERIMENT NO. 8

Aim: To write test cases for white box testing.

Requirements: Windows/MAC/Linux O.S, Compatible version of JDK and Eclipse

Problem Statement: To design test cases for white box testing and test it using Eclipse.

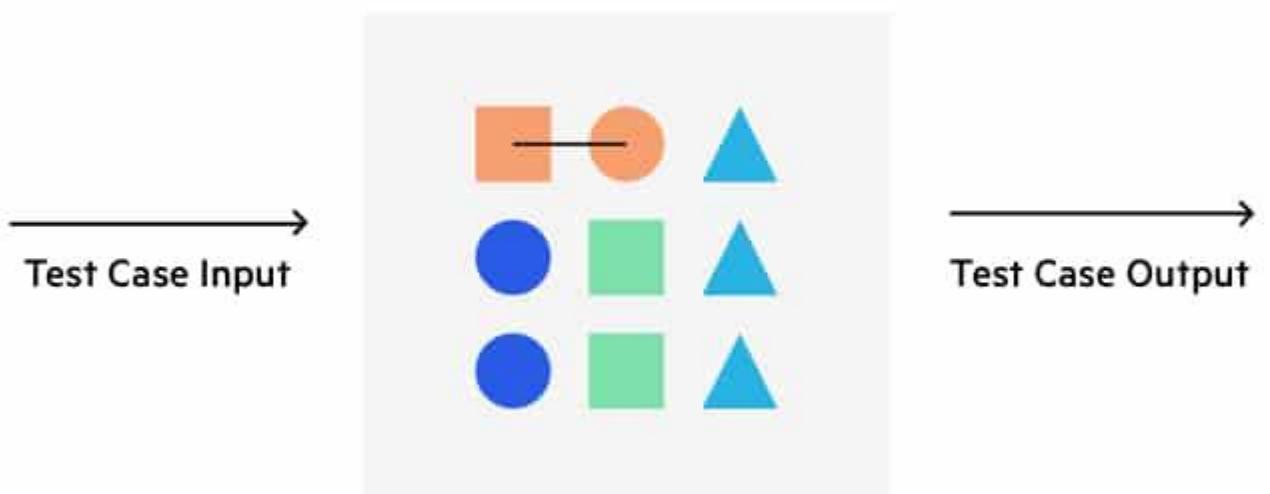
Theory:

What is White Box Testing?

White box testing is an approach that allows testers to inspect and verify the inner workings of a software system—its code, infrastructure, and integrations with external systems. White box testing is an essential part of automated build processes in a modern Continuous Integration/Continuous Delivery (CI/CD) development pipeline.

White box testing is often referenced in the context of Static Application Security Testing (SAST), an approach that checks source code or binaries automatically and provides feedback on bugs and possible vulnerabilities.

Application Code



White box testing provides inputs and examines outputs, considering the inner workings (control structure) of the code.

What Does White Box Testing Focus On?

White box tests can focus on discovering any of the following problems with an application's code:

- **Security gaps and vulnerabilities** — checking to see if security best practices were applied when coding the application, and if the code is vulnerable to known security threats and exploits.
- **Broken or poorly structured paths** — identifying conditional logic that is redundant, broken or inefficient.
- **Expected output** — executing all possible inputs to a function to see if it always returns the expected result.
- **Loop testing** — checking single loops, concatenated loops and nested loops for efficiency, conditional logic, and correct handling of local and global variables.
- **Data Flow Testing (DFT)** — tracking variables and their values as they pass through the code to find variables that are not correctly initialized, declared but never used, or incorrectly manipulated.

Testing Techniques and Code Coverage

One of the main goals of white box testing is to cover the source code as comprehensively as possible. Code coverage is a metric that shows how much of an application's code has unit tests checking its functionality.

Within code coverage, it is possible to verify how much of an application's logic is actually executed and tested by the unit test suite, using concepts like statement coverage, branch coverage, and path coverage. These concepts are discussed in more detail below.

Statement Coverage

Statement coverage is a white box testing technique that ensures all executable statements in the code are run and tested at least once. For example, if there are several conditions in a block of code, each of which is used for a certain range of inputs, the test should execute each and every range of inputs, to ensure all lines of code are actually executed.

Statement coverage helps uncover unused statements, unused branches, missing statement that are referenced by part of the code, and dead code left over from previous versions.

Branch Coverage

Branch coverage maps the code into branches of conditional logic, and ensures that each and every branch is covered by unit tests. For example, if there are several nested conditional statements:

if X then..

 if Y then..

 A

 B

 else

 if Z then..

C

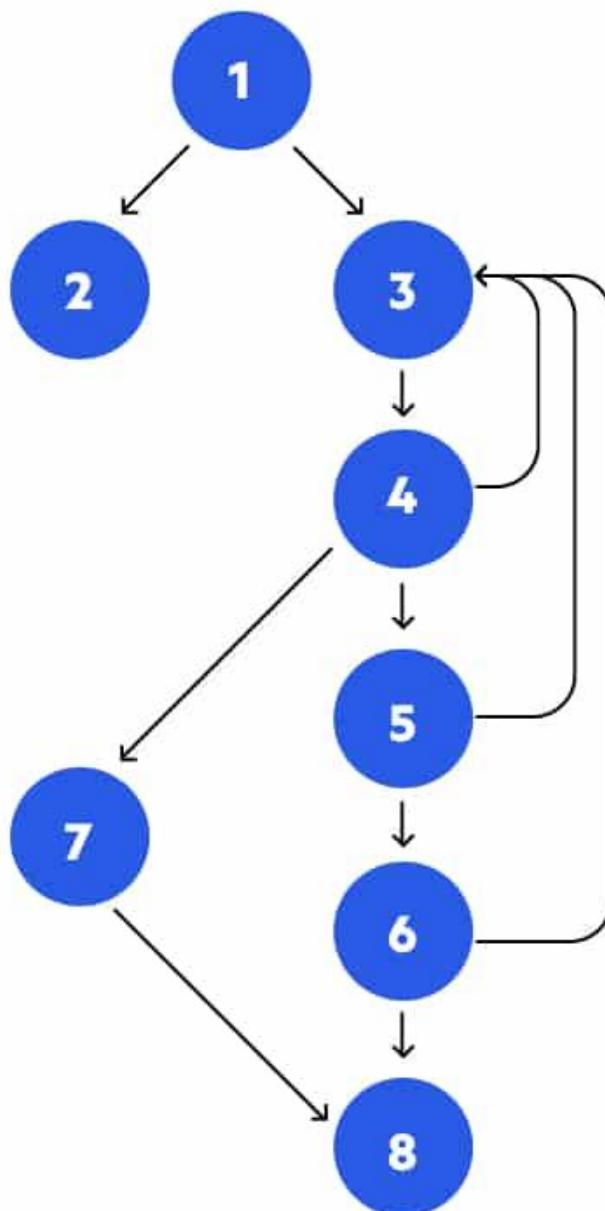
else..

D

A, C, and D are **conditional branches**, because they occur only if a condition is satisfied. B is an **unconditional branch**, because it is always executed after A. In a branch coverage approach, the tester identifies all conditional and unconditional branches and writes code to execute as many branches as possible.

Path Coverage

Path coverage is concerned with linearly independent paths through the code. Testers draw a control flow diagram of the code, such as the example below.



Control flow diagram used to design tests in a path coverage approach

In this example, there are several possible paths through the code:

- 1, 2
- 1, 3, 4, 5, 6, 8
- 1, 3, 4, 7, 6, 8
- etc.

In a path coverage approach, the tester writer's unit tests to execute as many as possible of the paths through the program's control flow. The objective is to identify paths that are broken, redundant, or inefficient.

ECLIPSE:

Eclipse is an integrated development environment that is used in computer programming.

It is the mostly widely use Java IDE and contains a base workspace and an extensible plug-in system for customising the environment.

The platform has been designed to build integrated web and application development tooling.

It is designed to not offer a huge amount of end user functionality but the value of the platform comes with its ability to encourage the rapid development of integrated features based on a plug-in model.

Eclipse provides a common user interface model for working with tools and is designed to run on multiple operating systems.

JUNIT:

JUnit is a unit testing framework for Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit, that originated with JUnit.

edureka! www.edureka.co

What is JUnit? Advantages and uses

01 The best testing frameworks that can be selected for an efficient testing process

02 More Application Developer IDEs includes JUnit

03 It also provides an AWT and Swing-based graphical test reporting mechanism

04 JUnit provides a text-based command line

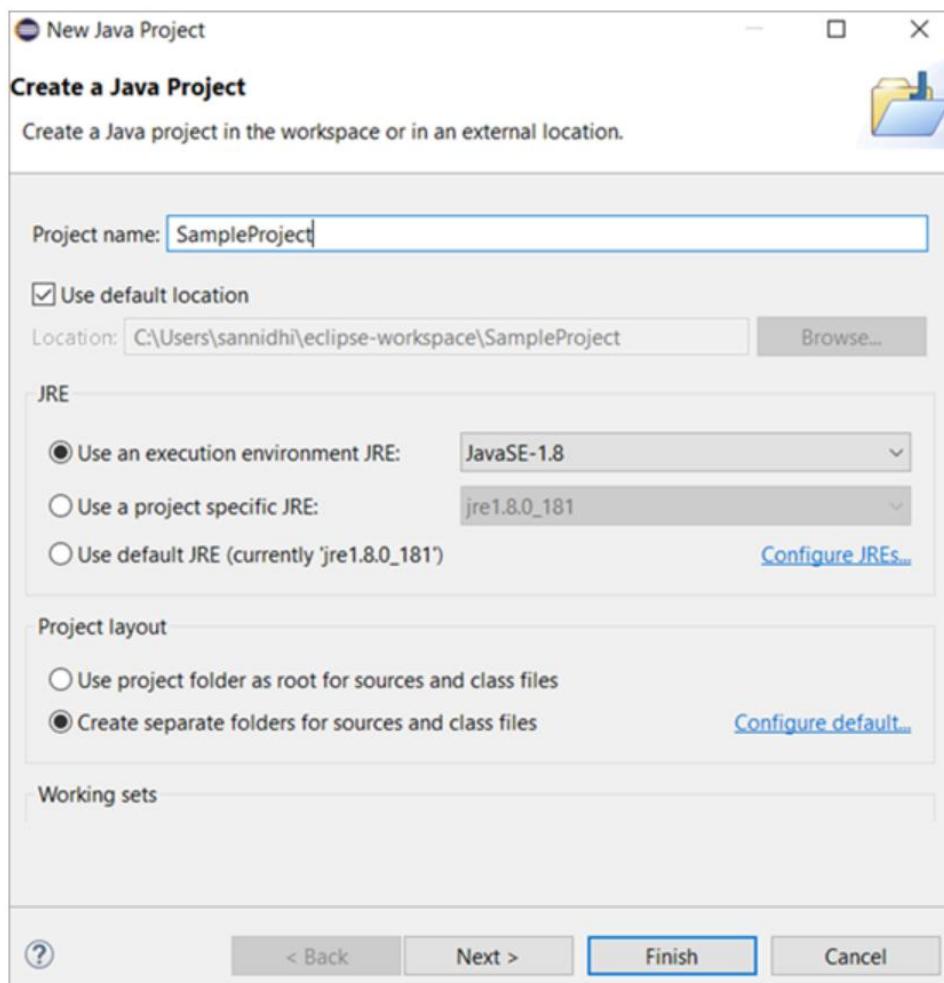
05 JUnitEE test framework that enables it to test within the application server's container

06 Widely adopted by many organizations around the world

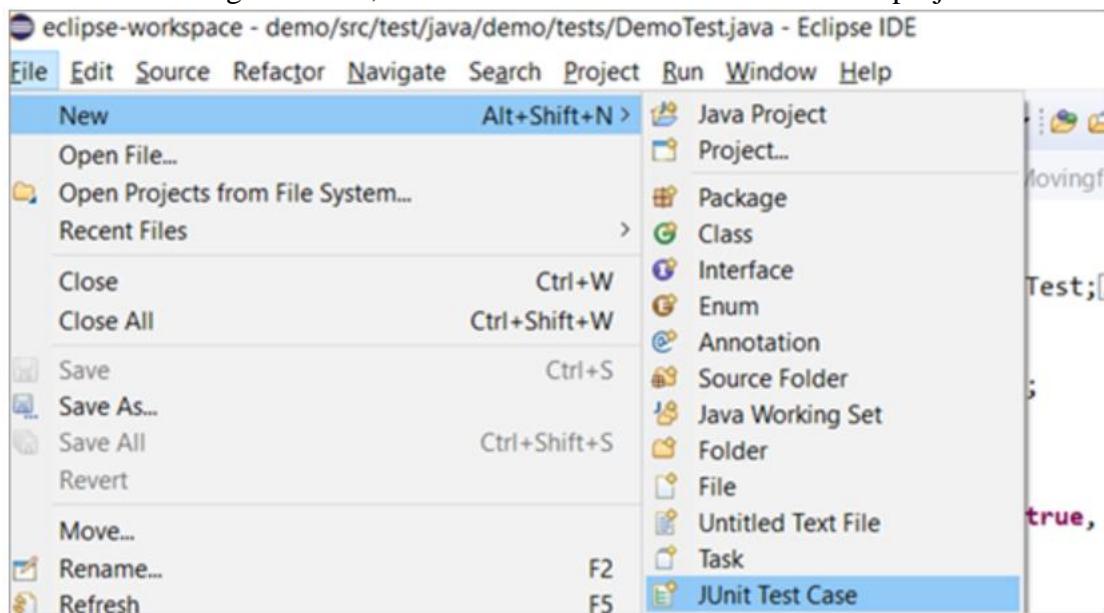
07 A benchmark for testing in Java programming language

Looking for Online Training Classes? Call us at IN: 9606058406 / US: 18338555775 or visit www.edureka.co

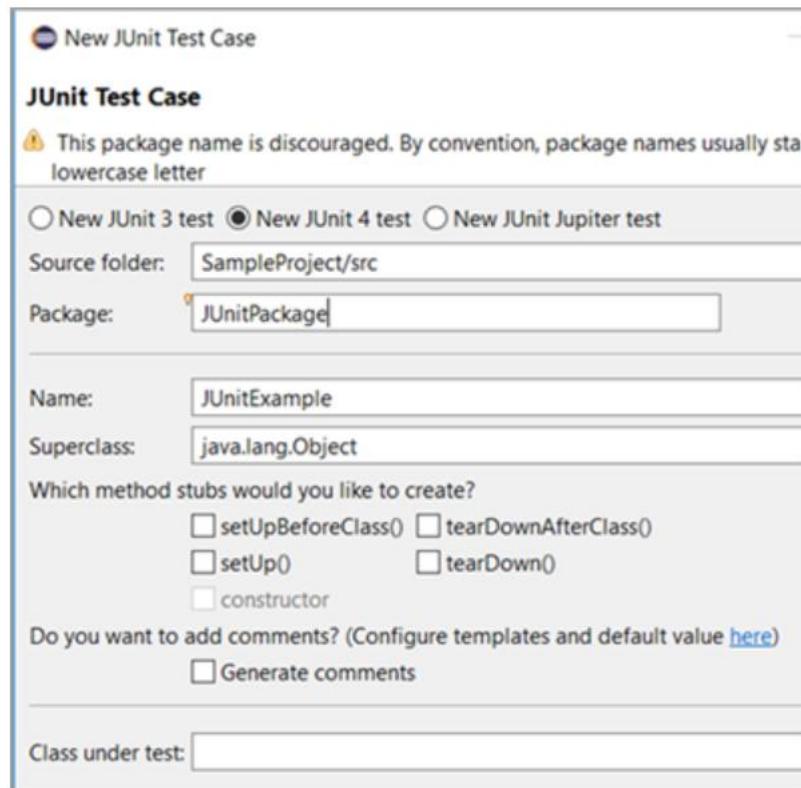
SUBSCRIBE

Output:

After installing JUnit tool, we have to add new JUnit test case into project folder.



Then we have to select source folder path in source folder package name, stub method if any and then click on finish.



Now you will get default template.

```

1 package demo.tests;
2
3+import static org.junit.Assert.*;
4
5 public class JUnitProgram {
6
7     @Before
8     public void setUp() throws Exception {
9         }
10
11     @After
12     public void tearDown() throws Exception {
13         }
14
15     @Test
16     public void test() {
17         fail("Not yet implemented");
18     }
19 }
```

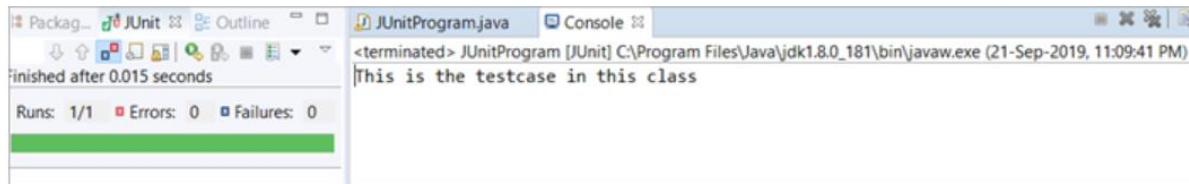
Test Case:

The test case should verify the string statements, if the string typed by user matches with the given string statement then we can say that the test case is successfully completed and if the user enter another statement then we can say that test case is unsuccessful.

```
package demo.tests;
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
public class JUnitProgram {
    @Test
    public void test_JUnit() {
        System.out.println("This is the testcase in this class");
        String str1="This is the testcase in this class";
        assertEquals("This is the testcase in this class", str1);
    }
}
```

Here, it verifies if str1 variable and string passed in the condition are both equal.

The comparison of expected condition has been performed by **assertEquals()** method which is JUNIT specific method.



The JUNIT result tab displays mainly number of test cases run, number of errors and number of failures encountered.

Conclusion: We have successfully design test cases for white box testing and implemented them using JUnit with Eclipse in Java.

EXPERIMENT NO. 9

Aim: To prepare Risk Mitigation, Monitoring and Management Plan (RMMM).

Problem Statement: To prepare RMMM plan for E-Commerce Shopping Website.

Theory:

R.M.M.M stands for risk mitigation, monitoring and management. There are three issues in strategy for handling the risk is

- Risk Avoidance
- Risk monitoring
- Risk management

Risk Mitigation:

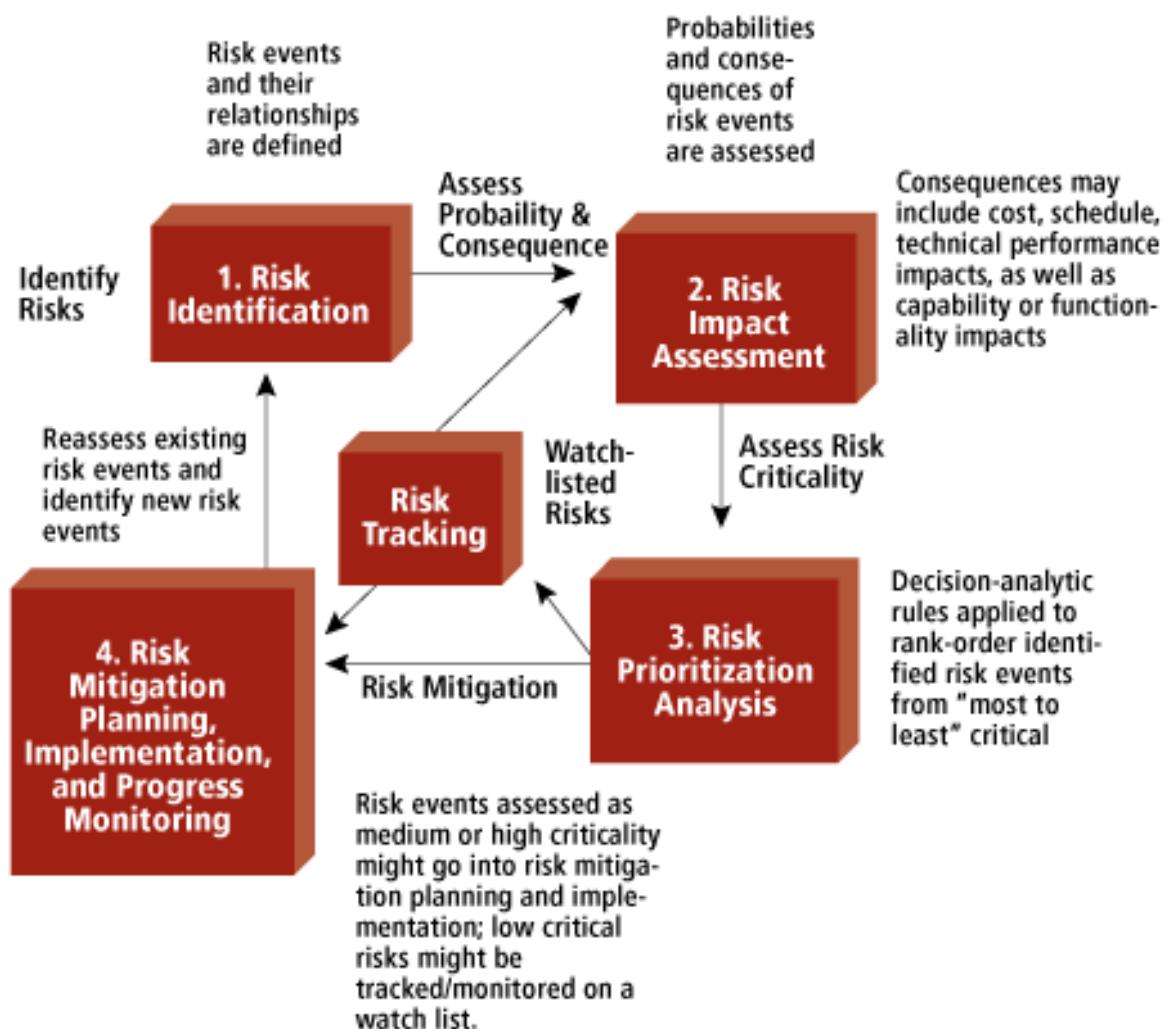
- Risk mitigation means preventing the risk to occur (Risk Avoidance). Following are the steps to be taken for mitigating the risks.
- Communicate with the concerned staff to find of probable risk.
- Find out and eliminate all those causes that can create rick before the project starts.
- Develop a policy in an organization which will help to continue the project even though some staff leaves the organization.
- Everybody in the project team should be acquainted with the current development activity.
- Maintain the corresponding documents in timely manner. This documentation should be strictly as per the standards set by the organization.
- Conduct timely reviews in order to speed up the work.
- For conducting every critical activity during software development, provide the additional staff if required.

Risk Monitoring:

- In risk monitoring process following things must be monitored by the project manager..
- The approach of the team members as pressure of project varies.
- The degree in which the team performs with the spirit of "team-work".
- The type of co-operation among the team members. The types of problems that are occurring. Availability of jobs within and outside the organization.
- The project manager should monitor certain mitigation steps. For example: If the current development activity is monitored continuously then everybody in the team will get acquainted with current development activity.

- The objective of risk monitoring is to check whether the predicted risks really occur or not. To ensure the step defined to avoid the risk are applied properly or not. To gather the information which can be useful for analysing the risk.

Risk Management:



- Project manager perform this task when risk becomes a reality. If project manager is successful in applying the project mitigation effectively then it becomes very much easy to manage the risks.
- For example, Consider a scenario that many people are leaving the organization then if sufficient additional staff is available, if current development activity is known to everybody in the team, if latest and systematic documentation is available then any 'new comer' can easily understand current development activity. This will ultimately help in continuing the work without any interval.

Risk Plan:

The R.M.M.M plan is a document in which all the risk analysis activities are described. Sometimes project manager includes this document as a part of overall project plan. Sometimes

specific R.M.M.M plan is not created, however each risk can be described individually using risk information sheet.

RMMM Plan for E-commerce website:

E-Online Security:

Risk Mitigation:

1. Phishing
2. Data error
3. Credit card fraud
4. Hacking

Risk Management:

1. Multi-Layer security
2. Monitor all transaction
3. Deploy PCI scan
4. Get trust marked with SSL certificate

Customer disputes and chargebacks:

There are several customer disputes which arises time to time from ordering delivery.

Risk Mitigation:

1. Product never arrived
2. Didn't meet expectation
3. Didn't match as website describe.
4. Delivery of wrong product.

Risk Management:

1. Keeping track of product
2. Not to charge until shipped
3. Clear refund policy
4. Full authorization for order before shipping.

Product Warehouse & Logistic:

Risk Mitigation:

Running high performance app warehouse operation and logistics can be tricky and if it is not done right, it can result in time, money and customer satisfaction going down the drain.

1. Run out of stock in warehouse when orders come in.
2. Product shipment delayed.
3. Misdelivery.

Risk Management:

1. Stock management: Use inventory system
2. Barcode scanner to speed up process.
3. Ensure proper order tracking to customer.
4. Cross check whether the right delivery details are entered.
5. If package was misdelivered, check the receipt's sign and address and ensure it is delivered to right location, otherwise process the refund.

Customer Service:

Risk Mitigation:

- Reviews, services, culture, social media and negative PR.

Risk Management:

- Provide chat box, personalization, respond to customer complaints quickly and gracefully.

Conclusion: We have successfully understand the concept of RMMM plan and prepared RMMM plan for E-Com. Website.

EXPERIMENT NO. 10

Aim: Version controlling of the project.

Requirements: Windows O.S and Tortoise SVN.

Problem Statement: To version control project using Tortoise SVN.

Theory:

What is version control?

How version control helps high performing development and DevOps teams prosper

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time. As development environments have accelerated, version control systems help software team work faster and smarter. They are especially useful for DevOps teams since they help them to reduce development time and increase successful deployments.

Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

For almost all software projects, the source code is like the crown jewels - a precious asset whose value must be protected. For most software teams, the source code is a repository of the invaluable knowledge and understanding about the problem domain that the developers have collected and refined through careful effort. Version control protects source code from both catastrophe and the casual degradation of human error and unintended consequences.

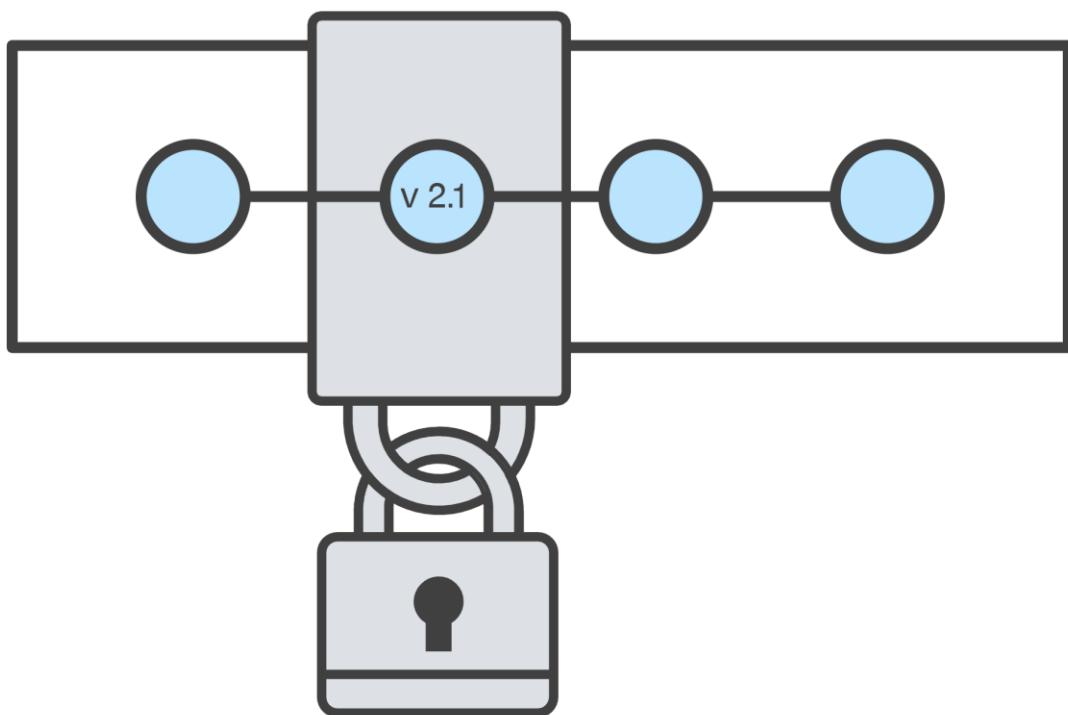
Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.

Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatible with those made by another developer working at the same time. This problem should be discovered and solved in an orderly manner without blocking the work of the rest of the team. Further, in all software development, any change can introduce new bugs on its own and new software can't be trusted until it's tested. So testing and development proceed together until a new version is ready.

Good version control software supports a developer's preferred workflow without imposing one particular way of working. Ideally it also works on any platform, rather than dictate what operating system or tool chain developers must use. Great version control systems facilitate a smooth and continuous flow of changes to the code rather than the frustrating and clumsy mechanism of file locking - giving the green light to one developer at the expense of blocking the progress of others.

Software teams that do not use any form of version control often run into problems like not knowing which changes that have been made are available to users or the creation of incompatible changes between two unrelated pieces of work that must then be painstakingly untangled and reworked. If you're a developer who has never used version control you may have added versions to your files, perhaps with suffixes like "final" or "latest" and then had to later deal with a new final version. Perhaps you've commented out code blocks because you want to disable certain functionality without deleting the code, fearing that there may be a use for it later. Version control is a way out of these problems.

Version control software is an essential part of the every-day of the modern software team's professional practices. Individual software developers who are accustomed to working with a capable version control system in their teams typically recognize the incredible value version control also gives them even on small solo projects. Once accustomed to the powerful benefits of version control systems, many developers wouldn't consider working without it even for non-software projects.



Benefits of version control systems:

Using version control software is a best practice for high performing software and DevOps teams. Version control also helps developers move faster and allows software teams to preserve efficiency and agility as the team scales to include more developers.

Version Control Systems (VCS) have seen great improvements over the past few decades and some are better than others. VCS are sometimes known as SCM (Source Code Management) tools or RCS (Revision Control System). One of the most popular VCS tools in use today is called Git. Git is a *Distributed VCS*, a category known as DVCS, more on that later. Like many

of the most popular VCS systems available today, Git is free and open source. Regardless of what they are called, or which system is used, the primary benefits you should expect from version control are as follows.

1. A complete long-term change history of every file. This means every change made by many individuals over the years. Changes include the creation and deletion of files as well as edits to their contents. Different VCS tools differ on how well they handle renaming and moving of files. This history should also include the author, date and written notes on the purpose of each change. Having the complete history enables going back to previous versions to help in root cause analysis for bugs and it is crucial when needing to fix problems in older versions of software. If the software is being actively worked on, almost everything can be considered an "older version" of the software.
2. Branching and merging. Having team members' work concurrently is a no-brainer, but even individuals working on their own can benefit from the ability to work on independent streams of changes. Creating a "branch" in VCS tools keeps multiple streams of work independent from each other while also providing the facility to merge that work back together, enabling developers to verify that the changes on each branch do not conflict. Many software teams adopt a practice of branching for each feature or perhaps branching for each release, or both. There are many different workflows that teams can choose from when they decide how to make use of branching and merging facilities in VCS.
3. Traceability. Being able to trace each change made to the software and connect it to project management and bug tracking software such as Jira, and being able to annotate each change with a message describing the purpose and intent of the change can help not only with root cause analysis and other forensics. Having the annotated history of the code at your fingertips when you are reading the code, trying to understand what it is doing and why it is so designed can enable developers to make correct and harmonious changes that are in accord with the intended long-term design of the system. This can be especially important for working effectively with legacy code and is crucial in enabling developers to estimate future work with any accuracy.

While it is possible to develop software without using any version control, doing so subjects the project to a huge risk that no professional team would be advised to accept. So the question is not whether to use version control but which version control system to use.

Output:

TortoiseSVN *the coolest interface to (Sub)version control*

Home About Downloads Translations Support/Docs Other tools Support this project



Tortoise SVN

Info

- About
- About TortoiseSVN
- Screenshots
- Screenshots of various dialogs
- Testimonials
- What users say about TortoiseSVN
- News Archive
- News archive

Support

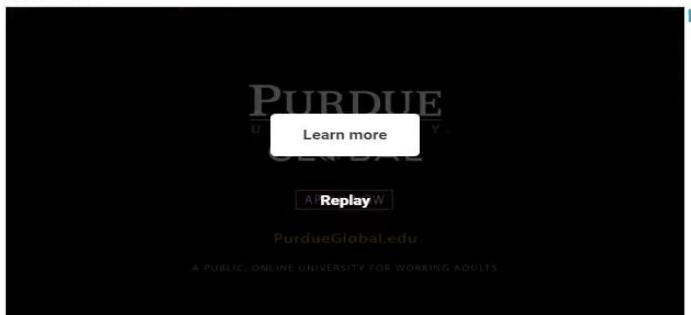
- FAQ
- Frequently asked questions
- Help files
- The complete documentation
- Useful tips
- Tips about not well known features
- Mailing lists
- Where to find the mailing lists
- Report bugs
- How and where to report a bug
- Translations
- Help translate

Downloads

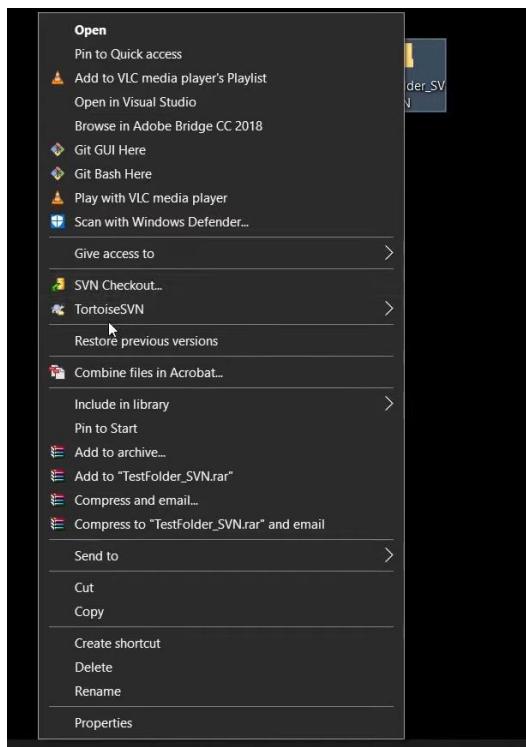
The current version is 1.14.0

For detailed info on what's new, read the [changelog](#) and the [release notes](#).

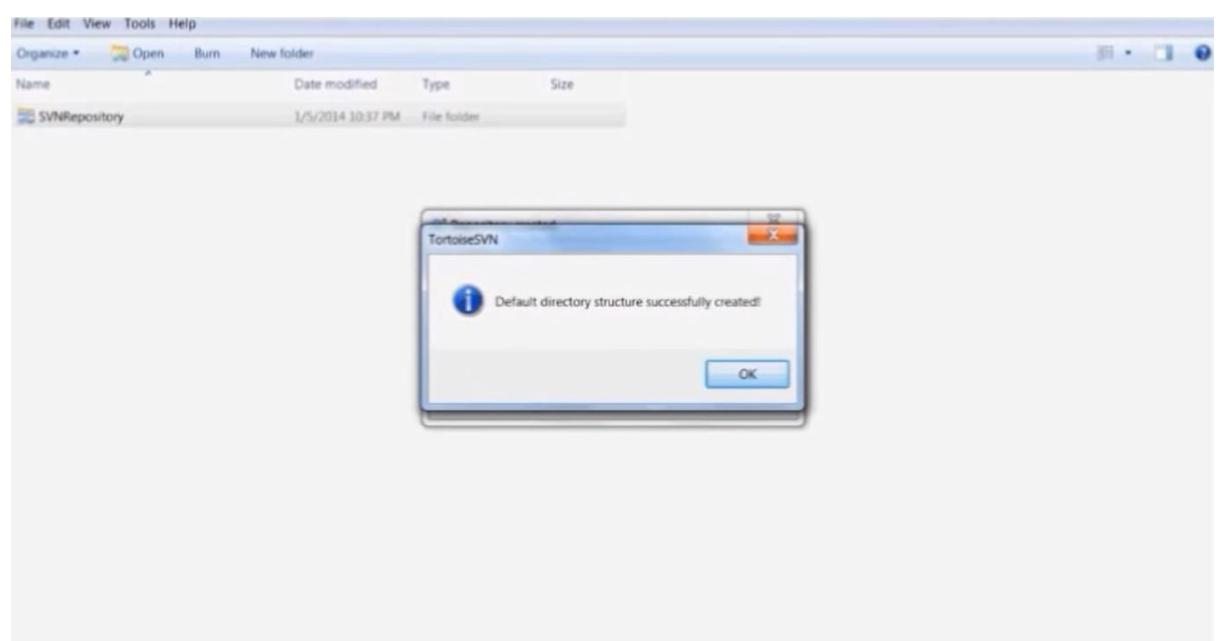
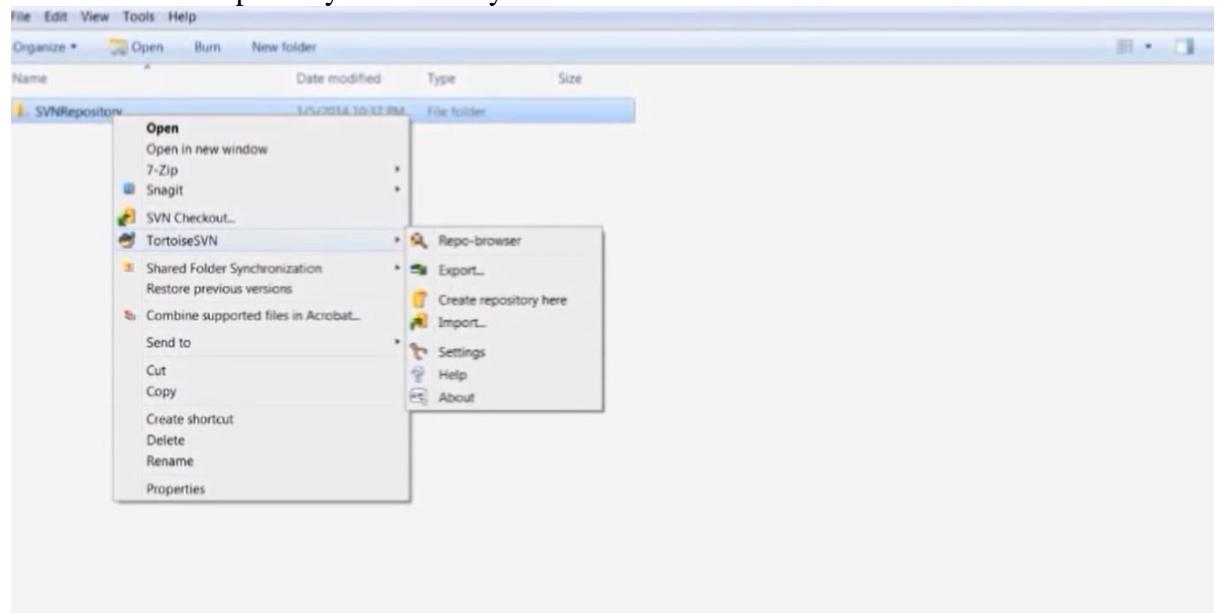
Advertisement



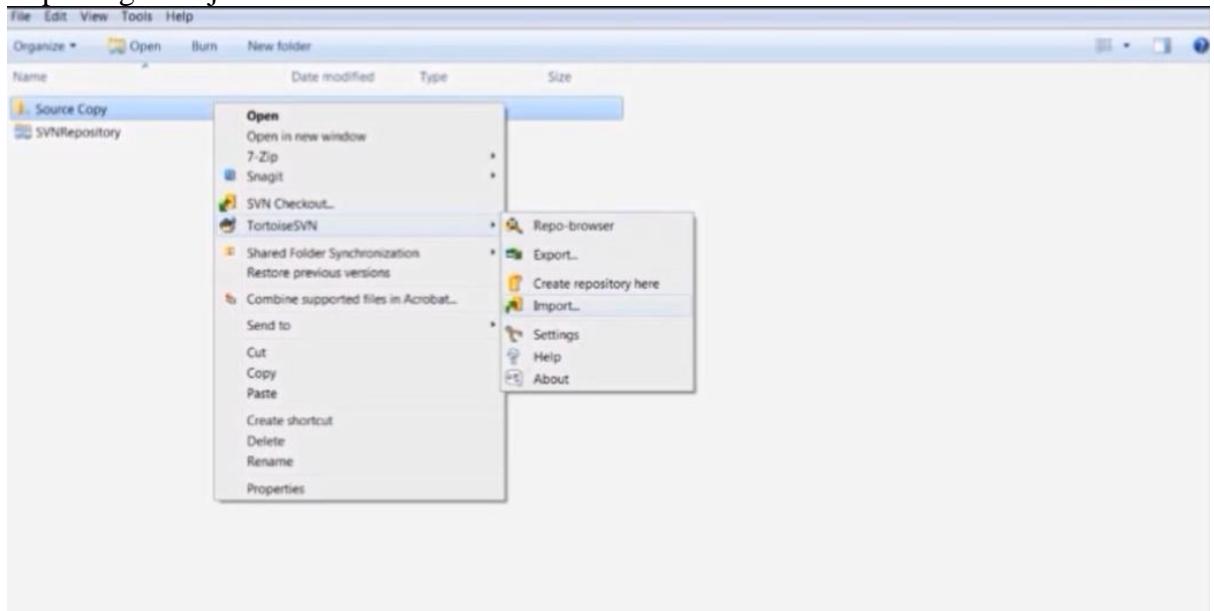
The current version 1.14.0 is linked against the Subversion library 1.14.0.



1. To create SVN repository- to centrally store data.

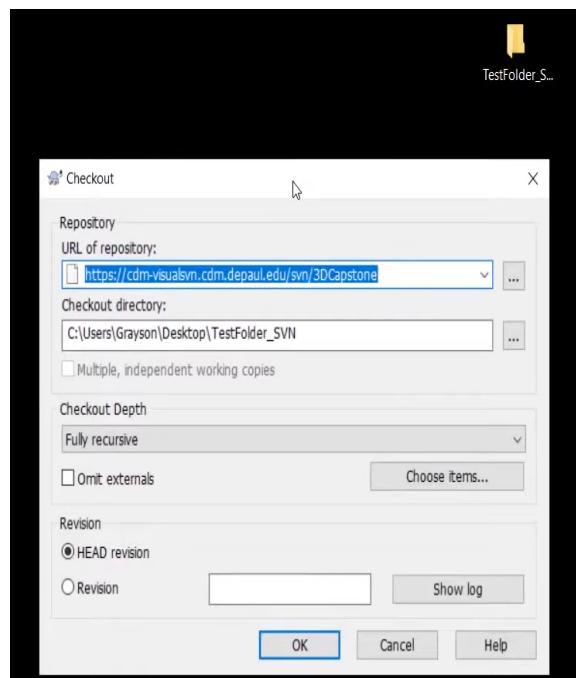


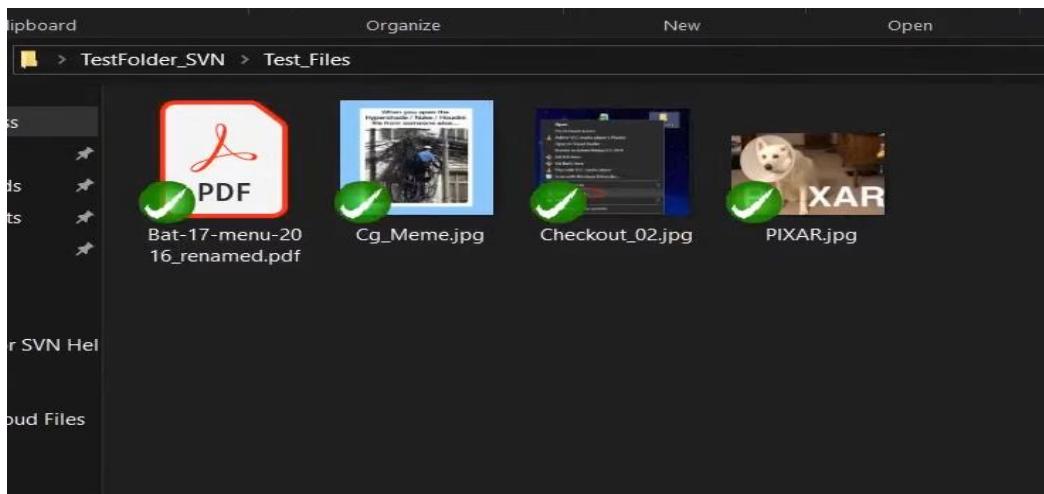
2. Importing a Project



In this way we can import the files on which we have to control it.

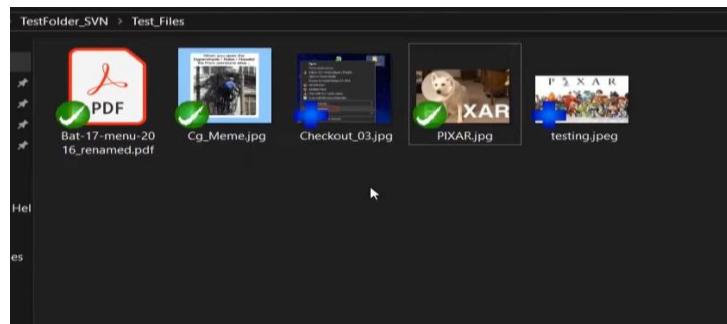
Now to perform operations or we can say that to version control some files we need to have client side and server-side directory. Test folder will be having the files which are working properly from server side.





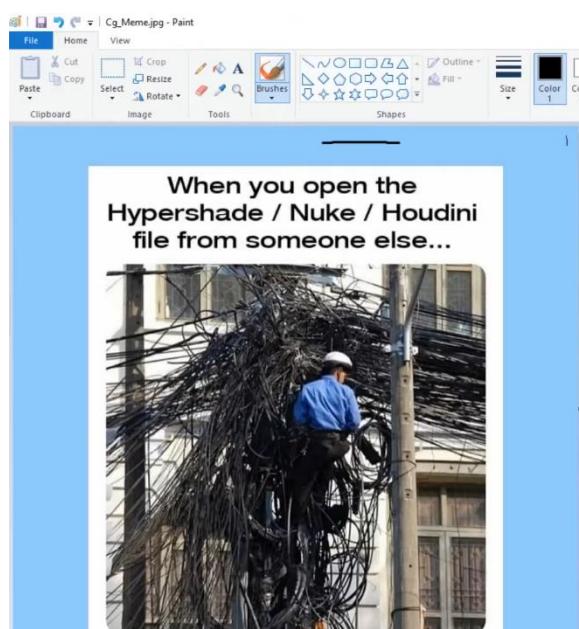
The green ticks show that the server is working correctly. And all the files are in updated version.

But if we rename any file here, it gives a blue mark. Showing that we are working on a copy not on original file.

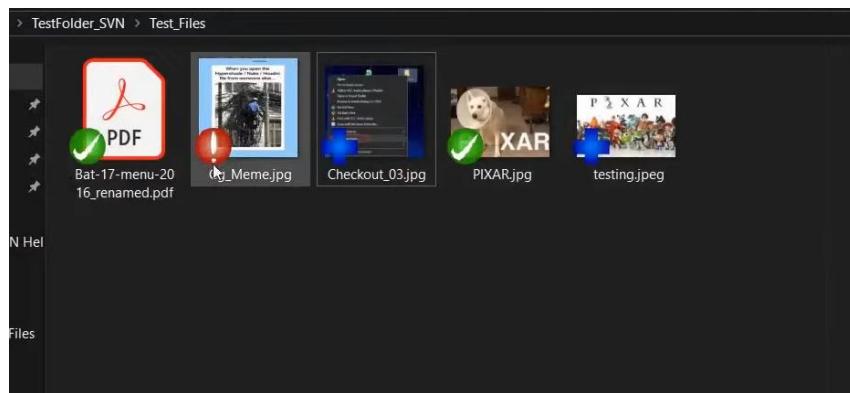


You can't edit rename any file using sync tortoise. You have to rename any file using server side only. Means in repository you should make changes.

3. When you edit any file,



Check what happens,



It gives exclamation mark. Meaning that the edited file is different than the previous one.

This way you can version control your software.

Conclusion: We have successfully understand the concept of version controlling and version control our project using SVN Tortoise.

Assignment 1

<1> What are agile Process and explain any one of them.

Ans) Agile Software engineering combines a philosophy and a set of development guidelines.

The Philosophy encourages customer satisfaction, and in early incremental delivery of software; small, highly motivated project teams; informal methods; minimal software engineering work products; and overall development simplicity.

The development guidelines stress delivery over analysis and design and active and continuous communication between developers and customers.

(2) Any agile Software Process is characterized in a manner that address a number of key assumptions about the majority of software projects:

i) It is difficult to predict in advance which software requirements will persist and which will change. It is equally difficult to predict how customer priorities will change as the project proceeds.

ii) For many types of software, design and construction are interleaved. That is, both activities should be performed in tandem so that design models are proven as they created. It is difficult to predict how much design is necessary before construction is used to prove the design.

(iii) Analysis, design, construction, and testing are not as predictable from a planning point of view as we might like.

③ Given these three assumptions, an important question arises: How do we create a process that can manage unpredictability? Answer is adaptability. An agile process, therefore, must be adaptable.

④ An agile software process must adapt incrementally for forward progress.

⑤ To accomplish incremental adaptation, an agile team requires customer feedback.

⑥ An effective catalyst for customer feedback is an operational prototype or a portion of an operational system.

⑦ Iterative approach enables the customer to evaluate the software increment regularly, provide necessary feed-back to the software team, and influence the process adaptations that are made to accommodate the feed back.

⑧ Common Agile Process models: XP, TXP, Scrum, Agile modeling and Agile unified process.

⑨ XP Process:-

① Extreme Programming (XP) uses an object-oriented approach as its preferred development paradigm & encompasses a set of rules & practices that occur within the context of four frame-work activities: Planning, design, coding and

testing. Idey XP activities.

- ① Planning:- ① The planning activity begins with listening - a requirements gathering activity that enables the technical members of the XP team to understand the business context for the software and to get a broad feel for required output and major features and functionality. Requirement in XP is known as user story.
② The highest ^{value} and riskiest stories will be released in early increment.
③ After delivery of 1st increment project velocity is calculated which helps for production & delivery of future increment.

- ② Design:- ① XP design rigorously follows the KIS (Keep It Simple) principle.
② A simple design is always preferred over a more complex representation.
③ The design provides implementation guidance for a Story as it is written - nothing less, nothing more.
④ The design of extra functionality is discouraged.

- ③ Coding:- ① Before coding, unit tests are developed that will exercise each of the Stories that is to be included in current release.
② Depending on unit test developer focuses on what is to be implemented to pass the test.

- ③ Once the code is complete, it can be unit-tested immediately, thereby providing instantaneous feed back to the developers.
- ④ XP encourages pair programming.

(iv) Testing :- Since, individual unit tests are organized into a "universal testing suite", integration and validation testing of the system can occur on a daily basis.

⑤ XP acceptance tests, also called customer tests, are specified by the customer and focus on overall system features and functionality that are visible and reviewable by the customer.

2) Difference between waterfall and spiral model.

Ans) Waterfall model

① Waterfall model works in sequential method.

② In waterfall model errors or risk are identified and rectified after the completion of stages.

③ Waterfall model is applicable for small and simple project.

④ No concept of incremental delivery.

⑤ Flexibility to adapt is difficult & once the same activity is completed there is no going back.

⑥ High risk of failure.

⑦ Cost of applying is comparatively low.

⑧ Needs a thorough understanding of requirements from the beginning.

Spiral model

① Spiral model works in combination of iteration and sequential method i.e. Evolutionary method.

② In spiral model errors or risks are identified and rectified earlier due to its iterative pattern.

③ Spiral model is applicable for large and complex project.

④ Encourage incremental delivery after one circuit is completed.

⑤ Flexibility to adapt is easy due to its iterative nature and same work activity can be repeated.

⑥ Risk of failure is low.

⑦ Spiral model is very expensive and not applicable for tight budget.

⑧ Requirements can be added up in the new iterations if required.

b) What are current trends in IP and explain any two of them.

- Ans) Current Trends in I.P. :-
- 1) Internet of Things (I.O.T)
 - 2) Blockchain
 - 3) Artificial Intelligence (A.I)
 - 4) Virtual Reality
 - 5) Augmented Reality
 - 6) Low-Code development
 - 7) Human Augmentation
 - 8) Mixed Reality (M.R)
 - 9) Cloud Computing

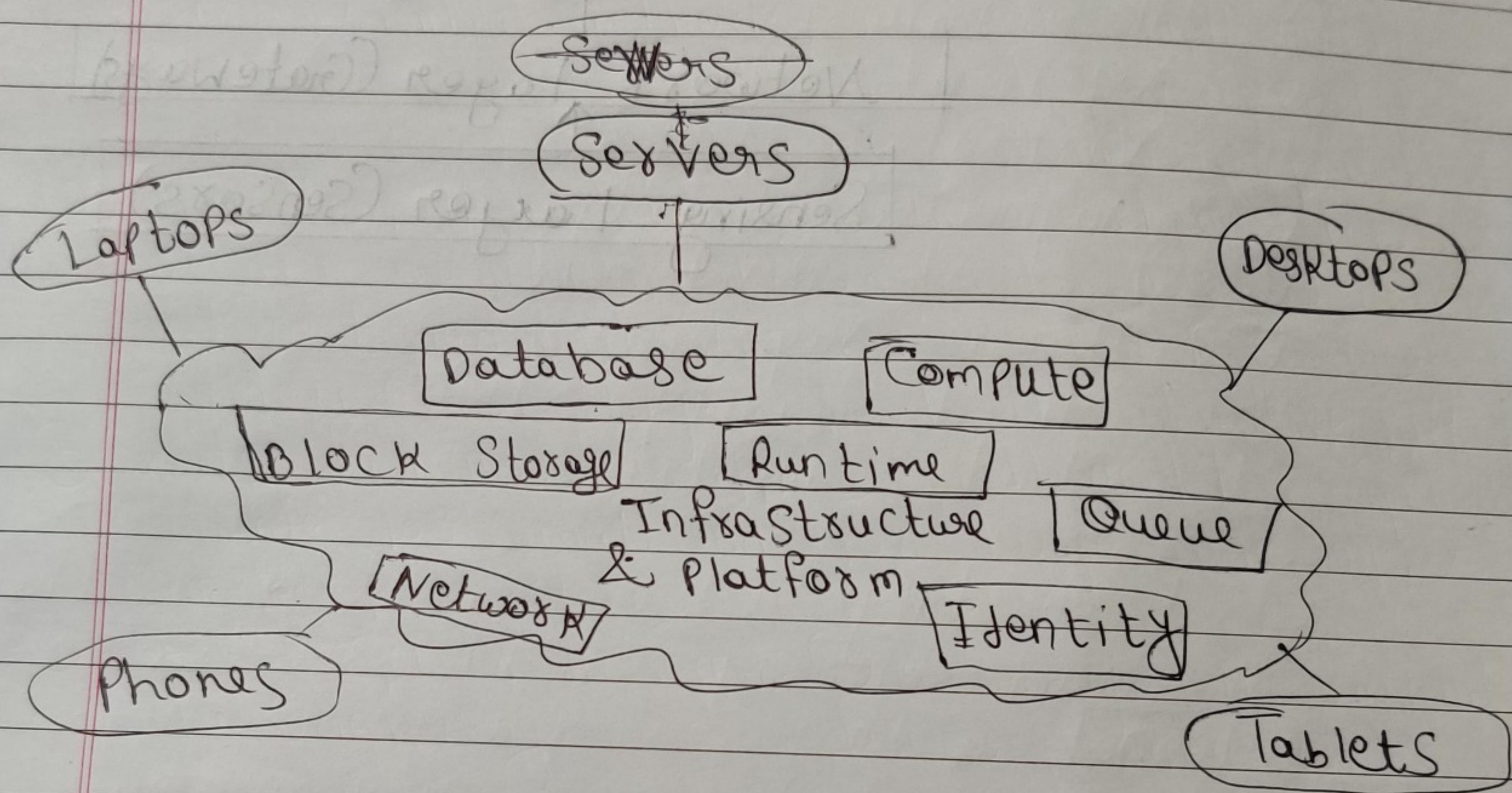
Cloud Computing:- Cloud Computing is the on-demand delivery of IT resources over the internet with Pay-as-you-go pricing.

(i) Instead of buying, owning and maintaining physical data centers and servers, you can access technology services such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

(ii) Application:- Organizations of every type, size and industry are using the cloud for a wide variety of use cases, such as data backup, disaster recovery, email, virtual desktops, software development and testing, big data analytics and customer-facing web applications.

ain
 (iv) Characteristics:- Agility, Elasticity (amount of resources can be scaled), cost savings and Deploy globally in minutes.

(v) Architecture:-



Internet of Things (IoT):- i) IoT is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.

ii) Applications:- RFID to track objects, wearables like smart watch to monitor calorie expenditure and heartbeats, GPS tracking belts, Smart grid and energy saving, etc.

iii) Characteristics:- Connectivity, Intelligence and Identity, Scalability, Dynamic and self adapting, Architecture & Safety.

(iv)

Architecture:-Application Layer (Application)Data Processing Layer (Processing unit)Network Layer (Gateways)Sensing Layer (Sensors)

Q.4) Develop a Software requirement Specification (SRS) for developing a software for Hospital Management System.

1.) Introduction:-

<<1.1>> Purpose :- The main purpose of our system is to make hospital task easy and is to develop Software that replaces the manual hospital System into automated hospital management System. This document serves as the unambiguous guide for the developers of this Software System.

1.2) Document Conventions:-

- i) HMS - Hospital Management System
- ii) GUI - Graphical User Interface
- iii) PHID - Patient Hospital Identification Number

1.3) Scope of The Project:-

- i) The purpose of this specification is to document requirements for a system to manage the hospital.
- ii) The specification identifies what such a system is required to do.
- iii) The H.M.S will manage a waiting list of patients required different treatments.
- iv) The availability of beds will be determined and if beds are available the next appropriate patient on the list will be notified.
- v) Nurses will be allocated to wards depending on ward size, what type of nursing is needed, operating schedules, etc.

- (vi) The HMS will allow hospital administrative staff to access relevant information efficiently and effectively.
- (vii) The goal of HMS is to manage nurses, patients, beds, and patients medical information in an cost-effective manner.

2) Overall Description:-

2.1) Product Perspective:-

The various system tools that have been used in developing both the frontend, backend and other tools of the project are being discussed in this section.

- i) Frontend:- JSP, HTML, CSS, Java Script are utilized to implement the frontend.
- ii) Backend:- Database is handle using MySQL
- iii) API : JAVA, PHP are used for connecting Frontend & backend.

2.2) Product Functions:-

- i) The HMS will allow the user to manage information about Patients, nurses, & beds. Patient management will include the checking-in and checking-out of patients to & from the hospital.
- ii) The HMS will also support the automatic backup & protection of data.

2.3) Operating Environment:-

Minimum requirements for smooth functioning of Software:-

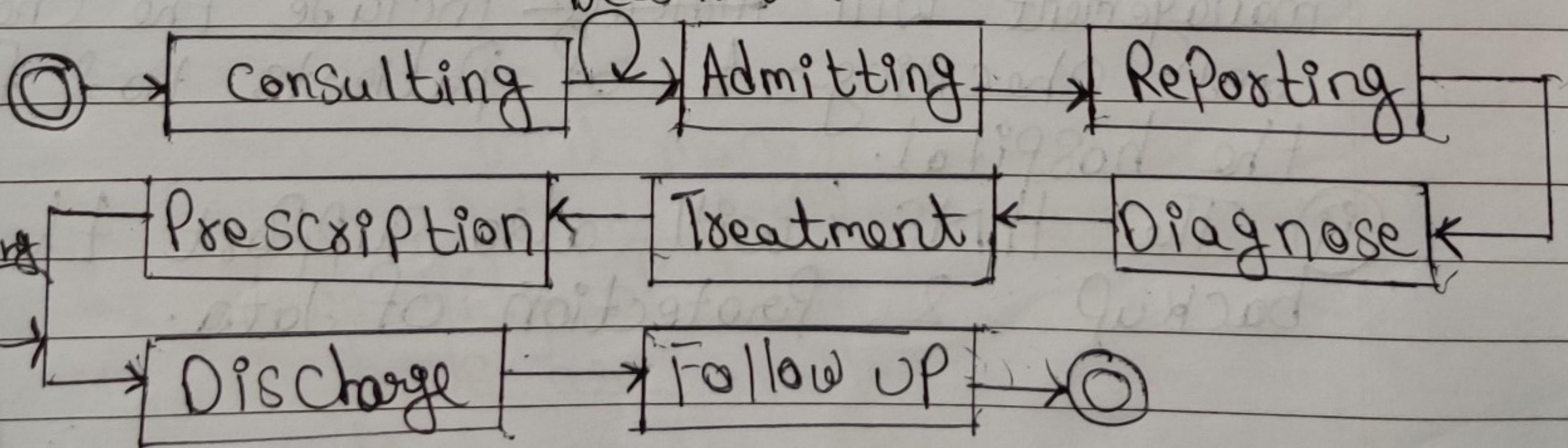
- i Processor: i3 or higher (2 GHz+)
- ii RAM: 4 GB+
- iii Disk Space: 20 GB+
- iv OS: Windows 10/8 or MAC OS.

2.4) Design & Implementation Constraints:-

- i GUI only in English.
- ii No guest users are allowed.
- iii Minimum information ^{limit} should be enough for creating account.

2.5) Assumption and Dependencies:-

- i Staff is familiar with modern technology.
- ii Minimum 100 computers are available.
- iii Maximum no. of users will always be less than 10,000.



not
feeling
well

3.1) External Interface Requirements:-

3.1.1) User Interfaces:

System is GUI based. Keyboard & mouse can be used to interact with software.

3.2) and 3.3) Hardware & Software Interface already mentioned in operating environment & product perspective.

4.) System Features:

4.1) System Features:-

i) Working Schedule: Assigning nurses to doctors & doctors to patients.

ii) Admissions: Admitting Patients, assigning the patients to appropriate wards.

iii) Patient Care: Monitoring Patients while they are in the Hospital.

iv) Surgery Management: Planning and organizing the work that Surgeons & Nurses perform in the operating rooms.

v) Ward Management: Planning and coordinating the management of wards & rooms.

vi) Waiting List: Monitoring to see if there are any patients waiting for available beds, assigning them to doctors and beds once these become available.

5) Other Non-Functional Requirements:-

5.1) Performance Requirements:-

The Performance of Software is at its best when the following regularly are done:-

- i) Password Management.
- ii) Regular Database Archiving.
- iii) Virus Protection.

5.2) Safety Requirements:-

Human are error-prone, but the negative effects of common errors should be limit. e.g.: - users should realize that a given command will delete data, and be asked to confirm their intent or have option to undo.

5.3) Software Requirements:-

i) Each member is required to enter an individual Username and Password when accessing the software. Administrators have the option of increasing level of password security their members use.

ii) The data in the database is secured through multiple layers of protection.

iii) One of those security layers involves member Passwords. For maximum security of your software, each member must protect their password.

5.4 Software Quality Attributes

The Quality of the system is mentioned maintained in such a way that it can be very user-friendly. The Software Quality attributes are assumed as follows:-

- i) Accurate and hence reliable.
- ii) Secured.
- iii) Fast Speed.
- iv) Compatibility.

Q.1) Explain coupling and cohesion and explain different types of cohesion.

Ans) ① Independence is assessed using two Qualitative Criteria: Cohesion and Coupling.

② Cohesion is an indication of the relative functional strength of a module.

③ Coupling is an indication of the relative interdependence among modules.

④ Cohesion is a natural extension of the information-hiding. A cohesive model performing a single task, requiring little interaction with other components in other parts of a program. Stated simply, a Cohesive model should do just one thing.

⑤ Coupling depends on the interface complexity between modules, the point at which entry or reference is made to a module, and what data pass across the interface.

⑥ In software design, you should strive for the lowest coupling and highest cohesion.

⑦ Types of cohesion: Below we stated 7 types of cohesion from best to worst.

(i) Functional cohesion: Functional Cohesion is said to exist if the different elements of a module, cooperate to achieve a single function.

(ii) Sequential Cohesion: A module is said to have Sequential Cohesion if the elements of a module form the components of the sequence, where the output from one component of the sequence is input to next.

3. Communicational Cohesion: A module is said to have communicational cohesion, if all tasks of the module refer to or update the same data structure.
e.g.: Linear and Quadratic Probing applied to same hashing data structure.
4. Procedural Cohesion: A module is said to be procedural cohesion if the set of purpose of the module are all parts of a procedure in which particular sequence of steps has to be carried out for achieving a goal, e.g.: the algorithm applied for decrypting a message.
5. Temporal Cohesion: When a module includes functions that are associated by the fact that all the methods must be executed in the same time, the module is said to exhibit temporal cohesion.
6. Logical Cohesion: A module is said to be logically cohesive if all the elements of the module perform a similar operation. For example Error handling, data input and data output etc.
7. Coincidental Cohesion: A module is said to have coincidental cohesion if it performs a set of tasks that are associated with each other very loosely, if at all.

Q.) Explain Black Box testing.

- Ans)
- ① Black-box testing, also called behavioral testing or functional testing, focuses on functional requirements of the software.
 - ② Black-box testing attempts to find errors in the following categories:
 - (i) Incorrectness or missing functions,
 - (ii) interface errors,
 - (iii) errors in data structures or external database access
 - (iv) behavior or performance errors, and
 - (v) initialization and termination errors.
 - ③ Black box testing focuses on the information domain and disregards control structure.
 - ④ Tests are designed to answer the following questions:
 - (i) How is functional validity tested?
 - (ii) How are system behaviour and performance tested?
 - (iii) What classes of input will make good test cases?
 - (iv) Is the system particularly sensitive to certain input values?
 - (v) How are the boundaries of a data class isolated?
 - (vi) What data rates and data volume can the system tolerate?
 - (vii) What effect will specific combinations of data have on system operation?

- ⑤ By applying black-box techniques, a set of test cases can be derived that satisfy the following criteria: Reduction in test cases, number of additional test cases that must be designed to achieve reasonable

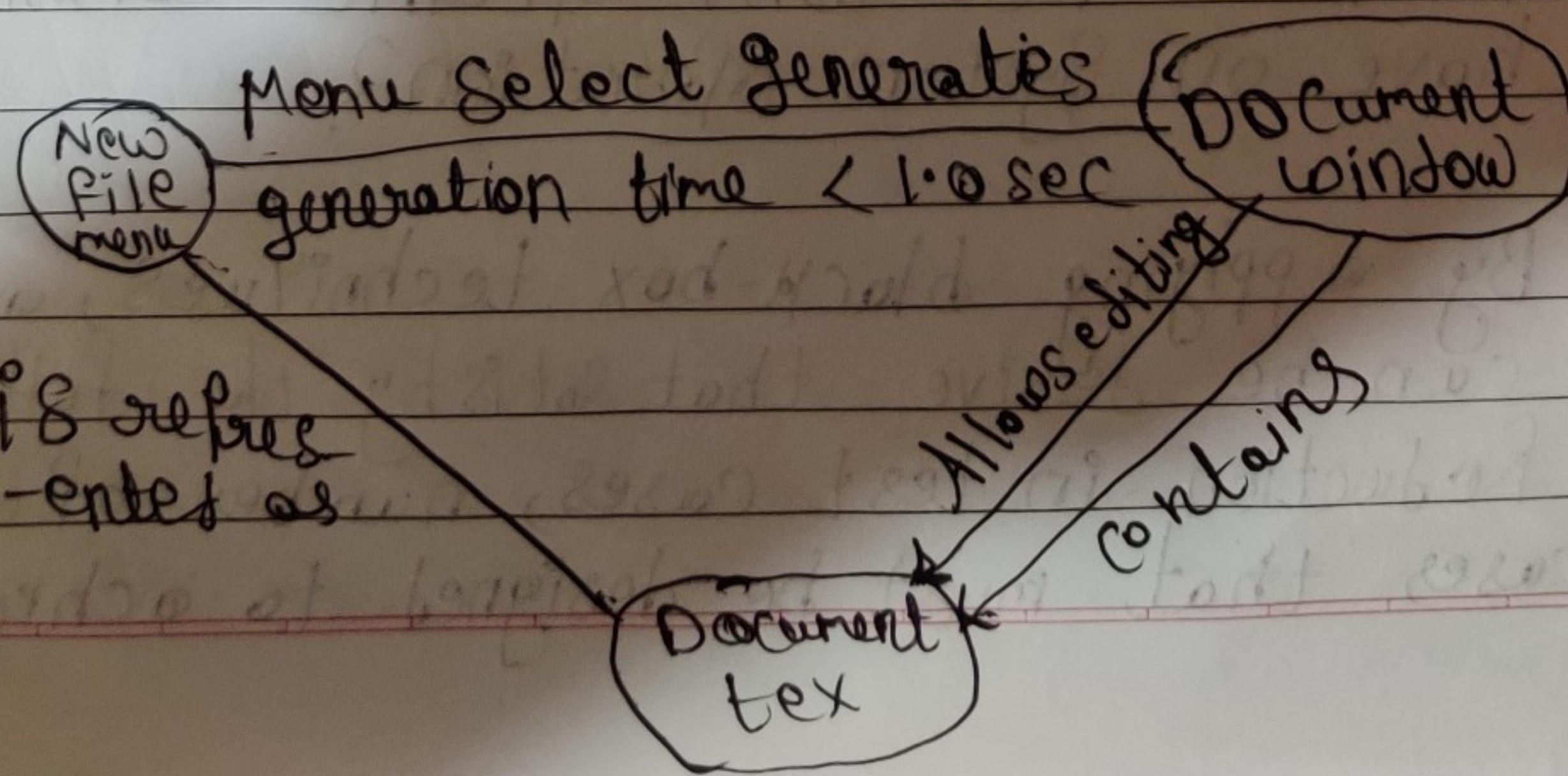
testing, and test cases that tell about presence or absence of classes of errors.

⑥ Methods that can be applied collectively for black box testing:

(7) Graph-based testing Methods: The first step in black-box testing is to understand the objects that are modeled in software and the relationships that connect these objects.

(ii) Once this has been accomplished, the next step is to define a series of tests that verify "all objects have expected relationship to one another". This can be achieved through by creating a graph - a collection of nodes representing objects, links (edges) represents relationship between objects, node weights describe properties of node and link weight describe some characteristic of link. Directed link from tail to head represent relationship in one direction only, Bidirectional or undirected link represents in both direction and parallel link represents more than one relationships among objects.

e.g:-



Boundary value Analysis: A greater number of errors occurs at the boundaries of the input domain rather than in the "center". It is for this reason that boundary value analysis (BVA) has been developed as a testing technique. BVA complements equivalence partitioning by selecting input values at the edges of equivalence classes. This reduces the number of inputs to be tested per test case.

Guidelines for BVA:

- 1) If an input condition specifies a range bounded by values $a & b$, test cases should be designed with values $a & b$ and just above & just below $a & b$.
- 2) If an input condition specifies a number of values, test cases should be designed that exercise the min. & max. numbers and numbers just above and below them.
- 3) Apply guidelines 1 & 2 to output conditions. Input those values which yields max. & min. output values.
- 4) If internal program data structures have prescribed boundaries (e.g. hashing table with 100 entries), be certain to design a test case to exercise the data structure at its boundary.

By applying these guidelines, there is a higher likelihood for error detection.

(iii) Test cases can be derived by traversing the graph. Test cases attempt to finds errors in any of relationships shown.

Different behavioral testing methods that can make use of graphs:

(i) Transaction flow modeling.

(ii) Finite state machine.

(iii) Data flow modeling.

(iv) Timing modeling.

Equivalence Partitioning:- If a set of objects can be linked by relationships that are symmetric, reflexive and transitive, an equivalence class is present.

An equivalence class represents a set of valid or invalid states for input conditions. Typically, an input condition is either a specific numeric value, a range of values, a set of selected values, or a Boolean condition. Equivalence classes may be defined according to the following guidelines:

1. If an input condition specifies a range one valid and two invalid equivalence classes are defined.
2. If an input condition requires a specific value, one valid and two invalid equivalence classes are defined.
3. If an input condition specifies a member of a set, one valid & one invalid equivalence class are defined.
4. If an input condition is Boolean, one valid & one invalid class are defined.

Test cases are selected so that largest number of attributes of an equivalence class are exercised at once.

Q. Write note on RMMM.

Ans) (1) Risk management technique is usually seen in the Software Project plan. This can be divided into Risk Mitigation, Monitoring and Management Plan.

- (2) In this plan, all works are done as part of risk analysis. As part of the overall project plan project manager generally uses this RMMM plan.
- (3) In some software teams, risk is documented with help of a Risk Information Sheet (RIS).
- (4) RIS is controlled by using database system for easier management of information. i.e. Creation, Priority ordering, Searching, and other analysis.
- (5) After documentation of RMM and starts of project, risk mitigation & monitoring commences.
- (6) Risk Mitigation: It is an activity used to avoid problems (Risk avoidance). Steps for mitigating the risks as follows:
- (i) Finding out the risks.
 - (ii) Removing causes that are the reason for risk creation.
 - (iii) Controlling the corresponding documents from time to time.
 - (iv) Conducting timely reviews to speed up the work.

(7)

Risk Monitoring: It is an activity used for project tracking. It has the following primary objectives as follows:

- (i) To check if predicted risks occur or not.
- (ii) To ensure proper application of risk aversion steps defined of risk.
- (iii) To collect data for future risk analysis.
- (iv) To allocate what problems are caused by which risks throughout the project.

(8)

Risk Management and Planning: It assumes that the mitigation activity failed and the risk is a reality. This task is done by project manager when risk becomes reality and causes severe problems. If the project manager effectively uses project mitigation to remove risks successfully then it is easier to manage the risk. This shows that the response that will be taken for each risk by a manager. The main objective of the risk management plan is the risk register. This risk register describes and focuses on the predicted threats to a software project.

MCQ - 2

1) Coupling is a measure of degree of interdependence between modules.

- a) Cohesion
- b) Coupling
- c) Both a & b
- d) None

2) Function oriented design technique start with functional requirement specified in SRS

- a) SDD
- b) SRS
- c) Both a & b
- d) None

3) What DFD notation is represented by Rectangle?

Data Structure

- a) Transform
- b) Data Structure
- c) Function
- d) None

4) What incorporates data, architecture, interface and procedural representation of software?

Design model

- a) Design model
- b) User model
- c) System model
- d) Data flow model

5) Which of the following is a type of architectural model? All of the above (UML, AXML, ADL)

- a) Static Structural Model
- b) Dynamic Process Model
- c) Distribution Model
- d) All of the above

6) What is cyclomatic complexity? White box testing

- a) Black box testing
- b) Yellow box testing
- c) White box testing
- d) Green box testing

7) White Box testing also classified as control structure testing.

- a) Functional testing
- b) Control Structure testing
- c) Error Guessing technique
- d) None

Analysis

- 8) Boundary value belongs to Black box and Complements Equivalence classes
- a) white box & equivalence b) black box & equivalence
 - c) white box & orthogonality d) black box & orthogonality.
- 9) Core of reverse engineering is an activity called
- a) Restructure code b) directionality
 - c) Extract abstraction d) interactivity
- Extract abstraction
- 10) Quality Management is software engineering is also known as SQA
- a) SQA b) SCM c) SQT d) SQR
- 11) Which one is not Risk management activity?
Risk Generation
- a) Risk assessment b) Risk generation
 - c) Risk control d) None
- 12) Risk management is one of important roles for Project Manager
- a) Senior Manager b) Software practitioner.
 - c) Project Manager d) Customer.
- 13) Which of the following doesn't affect SW quality & performance? Market
- a) Market b) Product c) Technology d) People.
- 14) What assess the risk & your plans for risk mitigation & revise these when you learn more about risk? Risk Monitoring
- a) Risk planning b) Risk Analysis
 - c) Risk monitoring d) Risk Management.

- 15) Which of the Following is a collection of component Version that make up System? Baseline
- Version
 - Code line
 - Baseline
 - Milestone

MCQ - I

- 1) CMM Stands For Capability Maturity Model
- Capability Maturity Model
 - Captiability Maturity Model
 - Conservative Maturity Model
 - Capability Management Model.
- 2) Efficiency in software Product doesn't include Licensing
- Responsiveness
 - Licensing
 - Memory utilization
 - Processing time.
- 3) RAD Stands for Rapid Application Development
- Relative application development
 - Rapid Application development
 - Rapid Application documentation
 - None
- 4) RAD model has 5-phases
- 2 phases
 - 3 phases
 - 5 phases
 - 4 Phases
- 5) which model doesn't allow defining requirements early in cycle? Prototyping
- water fall
 - XP
 - Incremental
 - Prototyping
- 6) The user requirements are parts of which document? SRS
- SDD
 - SRS
 - DFD
 - SRD

Q.7) Which level of ERD includes all entities & relationships ? Level 2

- a) Level-1
- b) Level-2
- c) Level-3
- d) Level-4

Q.8) Which model in system modelling depicts the static nature of system ? Structural Model

- a) Behavioral model
- b) Context model
- c) Data Model
- d) Structural model

Q.9) The UML supports event based modelling using Static chart diagrams.

Q.10) Which of the following diagram is not supported by UML considering Data driven modeling ? Data flow Diagram

- a) Activity
- b) State Chart
- c) Data flow
- d) Component

Q.11) Which of the following is not an effective Software project management focus ? Popularity

- a) People
- b) Product
- c) Process
- d) Popularity

Q.12) Which of the following is not Project Manager's activity ? Project design

- a) Project Management
- b) Project control
- c) Project design
- d) Project planning

Q.13) COCOMO Stands for Constructive cost - Model

- a) Constructive cost Model
- b) Comprehensive cost Model
- c) constructive cost estimation model
- d) complete cost estimation model.

Q.14) Sprint concept is used in which Agile model? SCRUM

- a) Agile unified model
- b) SCRUM
- c) XP & IXP
- d) Kanban.

Q.15) Which of the two are cost estimation metrics? LOC & FP

- a) Class Point & LOC
- b) LOC & FP
- c) CRC & FP
- d) CRC & Class Point.