

**A Mini Project Report**  
**On**  
**PRICE COMPARISON ON E-COMMERCE WEBSITE USING PYTHON**

**SIDDHARTHA INSTITUTE OF TECHNOLOGY & SCIENCES**  
**(UGC – AUTONOMOUS)**

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A+' Grade.

**Narapally, Korremula Road, Ghatkesar, Medchal- Malkajgiri (Dist.)-500088**



(Submitted in partial fulfilment of the academic requirements of B. Tech)

**In**  
**Department Of Computer Science and Engineering (CSD)**

**By**  
**G. Pranav(20TQ1A6743)**  
**G. Akanksha(20TQ1A6707)**  
**M. Sravani(20TQ1A6719)**  
**B. Sai Kiran(20TQ1A6705)**  
**K. Abhinay Reddy(20TQ1A6741)**  
**N. Sravan kumar (20TQ1A6727)**

Under the Esteemed Guidance of  
**Mrs.V.N.S. Manaswini**



**SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NAAC(A+))**

Korremula Road, Narapally(V), Ghatkesar Mandal, Medchal-Dist:-500088



## **CERTIFICATE**

This is to certify that the project report entitled

**PRICE COMPARISON ON E-COMMERCE WEBSITE USING PYTHON**

being submitted

by

**G. Pranav(20TQ1A6743)**

**G. Akanksha(20TQ1A6707)**

**M. Sravani(20TQ1A6719)**

**B. Sai Kiran(20TQ1A6705)**

**K. Abhinay Reddy(20TQ1A6741)**

**N. Sravan kumar(20TQ1A6727)**

In partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

**Guide**

**Mrs.V.N.S. Manaswini**

Department of CSD

**Head of the department**

**Mr. S. V. Satya Krishna**

Department of CSD

**Internal Examiner**

**External Examiner**

## **DECLARATION**

We declare that this project report titled **PRICE COMPARISION ON E-COMMERCE WEBSITE USING PYTHON** submitted in partial fulfilment of the degree of **B. Tech in Computer Science and Engineering (CSD)** is a record of original work carried out by me under the supervision of **Mrs. Manaswini** and has not formed the basis for the award of any other degree or diploma, in this or any other Institute or University. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

**G. Pranav(20TQ1A6743)**

**G. Akanksha(20TQ1A6707)**

**M. Sravani(20TQ1A6719)**

**B. Sai Kiran(20TQ1A6705)**

**K. Abhinay Reddy(20TQ1A6741)**

**N. Sravan kumar (20TQ1A6727)**



## ACKNOWLEDGMENT

Any endeavor in the field of development is a person's intensive activity. A successful project is a fruitful culmination of efforts by many people, some directly involved and some others who have quietly encouraged and supported.

Salutation to be beloved and highly esteemed institute **SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES** for grooming us into Computer Science and Engineering graduate, We wish to thank **Principal Dr. M. Janardhan** for providing a great learning environment.

We wish to express profound gratitude to **Mr. S. V. Satya Krishna**, Associate Professor and **Head of Department**, Computer Science and Engineering (DS), for his continuous encouragement to ensure successful results in all my endeavors.

We would like to thank **Ms. Manaswini**, Department of Computer Science and Engineering, who patiently guided and helped us throughout our project.

We take this opportunity to thank the department's Project Review Co- Ordinator **Dr. A. Satyanarayana** for all the review meetings, suggestions, and support throughout the project development.

By

**G. Pranav(20TQ1A6743)**

**G. Akanksha(20TQ1A6707)**

**M. Sravani(20TQ1A6719)**

**B. Sai Kiran(20TQ1A6705)**

**K. Abhinay Reddy(20TQ1A6741)**

**N. Sravan kumar (20TQ1A6727)**



## SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NAAC(A+))

Korremula Road, Narapally(V), Ghatkesar Mandal, Medchal-Dist:-500088



**Vision of the Department:** To be a Recognized Center of Computer Science Education with values and quality research.

### **Mission of the Department:**

MISSION	STATEMENT
DM1	Import High Quality Professional Training With An Emphasis On Basic principles Of Computer Science And Allied Engineering
DM2	Imbibe Social Awareness And Responsibility To Serve The Society.
DM3	Provide Academic Facilitates Organize Collaborated Activities To enable Overall Development Of Stakeholders

### **Programme Educational Objectives (PEO)**

- **PEO1:** Graduates will be able to synthesize mathematics, science, engineering fundamentals, laboratory and work – based experiences to formulate and to solve problems proficiently in Computer science and Engineering and related domains.
- **PEO2:** Graduates will be prepared to communicate effectively and work in multidisciplinary engineering projects following the ethics in their profession.
- **PEO3:** Graduates will recognize the importance of and acquire the skill of independent learning to shine as experts in the field with a sound knowledge.

# TABLE OF CONTENTS

DESCRIPTION	PAGE NUMBER
CERTIFICATE	2
DECLARATION	3
ACKNOWLEDGEMENTS	III
ABSTRACT	7
CHAPTER 1: INTRODUCTION	8-10
1.1 Objective	
CHAPTER 2: LITERATURE SURVEY	11-14
CHAPTER 3: EXISTING SYSTEM	15-18
CHAPTER 4: PROPOSED SYSTEM	19-20
CHAPTER 5: PROCESS MODEL USED WITH	21-30
JUSTIFICATION	
5.1 SDLC (Umbrella Model)	
CHAPTER 6: SOFTWARE REQUIREMENT SPECIFICATION	31-33
CHAPTER 7: EXTERNAL INTERFACE REQUIREMENTS	34-35
CHAPTER 8: SYSTEM DESIGN	36-43
8.1 UML DIAGRAMS	
CHAPTER9: IMPLEMENTATION	44-58
CHAPTER 10: TESTING	59-62
CHAPTER 11: SCREENSHOTS	63-66
CHAPTER 12: CONCLUSION	67-68
CHAPTER 12: REFERENCES	69-70

## LIST OF FIGURES

Figure No	Name Of The Figure	Page No
5.1	SDLC (umbrella model)	22
5.2	Requirement Gathering	23
5.3	Analysis Stage	25
5.4	Designing Stage	26
5.5	Development Stage	27
5.6	Integration and Test stage	28
5.7	Installation and acceptance stage	29
8.1	Class Diagram	38
8.3	Sequence Diagram	39
8.4	Collaboration Diagram	40
8.5	Component Diagram	40
8.6	Deployment Diagram	41
8.7	Activity Diagram	42
8.8	Data Flow Diagram	43



## **ABSTRACT**

In the current era of online business, E-commerce has become a huge market for the people to buy goods online. Increasing use of smart devices and other mediums has paved the way for users to buy products almost from anywhere. This has increased involvement of online buyers evolving E-commerce business. These large numbers of E-commerce websites put users in turmoil to search and choose to buy a single product from multiple E-commerce websites. E-commerce websites nowadays have become one of the most important sources for buying all kinds of products. As there are many E-commerce websites available it becomes difficult for users to choose the best deal for desired product amongst these websites. The proposed solution helps online users to grab the best deal for their product from multiple E-commerce websites on a single web interface. The given application will contain a user-based system, which retrieves the costs of the same product across various different websites, providing the best choice to the user, along with keeping track of the user's purchase history, to recommend products the user is most likely to buy. Hence, saving time and, at the same time, providing a good and friendly user interface. Users can also select multiple products that belong to the same category for comparing its features. To obtain best deals from ecommerce websites web scraping techniques are used to fetch detailed information. The price of the same product from all the sources is displayed on the console window, so that buyers can see the prices and make the decision to buy from the platform which offers the lowest price

# **CHAPTER 1**

## **INTRODUCTION**

# **1.INTRODUCTION**

A price comparison interface acts as a platform or medium between the shoppers and also the sellers. It permits shoppers to ascertain completely different lists of costs for the product chosen by user and it helps shoppers to form a wise to call concerning that to decide on so as to avoid wasting cash. On the opposite words, the value comparison web site additionally has the role to market the retailer/shop/hypermarket/supermarket to the purchasers. On the opposite words, the value comparison web site additionally has the role to market the retailer/shop/hypermarket/supermarket to the purchasers. The pressures on time and cash particularly within the current economic state of affairs wherever the living value will increase and there's solely very little time to try to some buying house, a value comparison web site like value comparison web site can completely become an excellent facilitate towards shoppers. Besides, users these days' square measure terribly snug with the web that it's big a wider sort of applications from networking and currently offer numerous references for the users. it's necessary for an internet comparison web site to come results with the low costs as what the purchasers need however correct results additionally necessary in order that customers will get what they actually need. It additionally depends on however regular the info is being updated alternative wise customers are going to be confused once they compared it from other web site. The planned visual perception enabled value comparison interface in its initial kind. The meant audience can embrace people who square measure a lot of dependent and attentive to getting product from Ecommerce sites. the online application can ease their effort of finding a product at a decent value. the most practicality involves providing the user, the simplest on-line value of the product from AN input image of the product. This net application can therefore alter tons of users to appear for a decent value by taking its image which cannot be in their possession. the online application also will function AN advertising agent for E-commerce sites in addition.

## **1.1 OBJECTIVE**

With the increasing technology and rapid growth in e-commerce nowadays, a number of online shopping websites are seen, which provide door-to-door service to the end user. But, due to the vast number of websites, the user has to spend a lot of time to find quality products at a cheap price. The given application will contain a user-based system, which retrieves the costs of the same product across various different websites, providing the best choice to the user, along with keeping

track of the user's purchase history, to recommend products the user is most likely to buy. Hence, saving time and, at the same time, providing a good and friendly user interface.

# **CHAPTER 2**

## **LITERATURE SURVEY**

## **2. LITERATURE SURVEY**

### **Recommendation system in E-commerce websites: A Graph Based Approach**

This paper is regarding the lack of semantic factor in recommendation systems and describes the different recommendation techniques that are being employed in the current e-commerce website. Recommendation system can be broadly classified into three categories: content-based, collaborative, and hybrid recommendation approaches. Content based systems consider the properties of the items to be recommended. For instance, if a Amazon user has purchased many romantic novels, then content based recommendation system recommends novels in the database as having the "romantic" genre. Collaborative filtering systems recommend items based on similarity measures between like minded users and/or items. The items recommended to a user are those preferred by similar users. This paper also emphasizes the need for semantics in current recommendation system to recommend products accurately. This also describes various limitations that are present in the current recommendation methods and suggests possible solutions that can improve current recommendation system used in e-commerce websites. It also includes a survey on popular e-commerce websites such as Amazon, Ebay, Flipkart Snapdeal and Paytm by rating them on different parameters and doing their comparative analyses This paper also focuses on how graph algorithm can be used to improve recommendation in ecommerce websites. The proposed system compares flickr.com recommendation of images with the proposed method. The method incorporates semantic recommendation using overlap technique based in graph

### **"E-commerce Personalized Recommendation System Based on Multi-Agent",**

This paper applied Multi-Agent to E-commerce personalized Recommender System, and designed E-commerce personalized Recommender System based on Multi-Agent, namely, MAPRS. Off-line recommendation and on-line hybrid recommendation are used to construct the core recommender model under the intelligent control. The paper presents the function and design ideas of various components of the system.

### **"Chatbot with a Discourse Structure-Driven Dialogue Management"**

We build a chat bot with iterative content exploration that leads a user through a personalized knowledge acquisition session. The chat bot is designed as an automated customer support or

product recommendation agent assisting a user in learning product features, product usability, suitability, troubleshooting and other related tasks. To control the user navigation through content, we extend the notion of a linguistic discourse tree (DT) towards a set of documents with multiple sections covering a topic. For a given paragraph, a DT is built by DT parsers. We then combine DTs for the paragraphs of documents to form what we call extended DT, which is a basis for interactive content exploration facilitated by the chat bot. To provide cohesive answers, we use a measure of rhetoric agreement between a question and an answer by tree kernel learning of their DTs.

### **“The use of web scraping in computer parts and assembly price comparison”**

If originally computers were used only as a tool to perform some calculations, nowadays computers has a lot of functions to help people finish their tasks in almost every aspect of human life. As a lot of various functions computers have, they also need different specifications for each computer so they can do their tasks according to their functionalities. Therefore this application was build with a purpose to recommend a solution to its users in assembling computes suited to their needs. This application also has a price comparison feature based on data sources retrieved from five computer shops so the users can save the costs of purchasing PC parts and assembling the computer easier. This comparison feature is based on a basic consumer's principal which are basically they wanting to buy items not only with the lowest price but also expect the best quality as possible. The research starts with the deployment of questionnaires to some respondents who had bought computer parts or assembled a computer online. This questionnaire is made to assure that all features which previously has been specified by the author is appropriate to user needs. Then, in order to obtain required data from five computer shops, the author use Pentaho Software as a tool to do web scraping and web grabbing method. These methods allow the application to obtain data from those five computer shops. The result of this research is a web-based application built in PHP and javascript with MySQL as its database.

### **“Exploiting Filtering approach with Web Scrapping for Smart Online Shopping”,**

With the advancement in technology and popularity of e-commerce, the number of online shopping websites have been increased rapidly in the cyber world. This made people's life easy

because it is easy to shop through internet. But this also bring effort for people as they spend a lot of time and efforts to search best product deals and offers on e-commerce websites. They have to filter and compare data by themselves. It takes a lot of time and still there are chances of ambiguous results. This paper is based on web crawling and scraping methods applied for identifying best deals from five e-commerce websites. The framework is designed using HTML (Hypertext markup language) and CSS (Cascading style sheet) as front-end and PHP: Hypertext preprocessor language as back-end support. The scrapping scripts are written using python libraries and web crawling works on HTML labels. The novelty in this framework is that we are not storing scrapped data on local database. Instead the results are dynamically fetched and showed every time the user input the query. It will help to improvise the storage and processing ability. Furthermore, the data retrieval process accuracy is 93% with minimum computation and less time.



# CHAPTER 3

## EXISTING SYSTEM

### **3. EXISTING SYSTEM**

In olden days when you want to buy products from online shopping apps like Amazon, Flipkart. We need to search that product by clicking those products. We check product price individually by going their apps. There is no price comparison in single. we can check price individual apps then compare prices and buy product which is low price.

There is no unified system for price comparison of e-commerce websites. The basic problems with the existing systems are the non-interactive environment they provide to the users. The user has to search for a specific product from various e-commerce websites in order to check the price and buy a product of his/her desire based on price and reviews.

The objective of this project is to develop a price comparison website that will have the following functions:

- 1.To provide customers with a list of price comparison and highlight the cheapest price.
- 2.To increase price consciousness among consumers.
3. To provide service for users to find the product's price.

#### **Disadvantage of existing system**

Online comparison websites have become a very popular way of helping customers compare the market for energy suppliers, insurance products and many other products. However there are disadvantages to using to using online price comparison sites.

#### **Not all providers use comparison websites**

Not all providers use price comparison websites and prefer to be contacted for competitive prices, e.g., Hughes insurance. This means that the consumers may be missing out another competitive prices

## **The cheap prices can be misleading**

Sometimes the price comparison sites show a very cheap, competitive price that appears at the top of the page. However usually this is a price for basic cover, so by the time the customer has added all the additional charges needed to meet their needs, there may not be a saving at all.

### **Comparison sites charge fees**

The companies shown on these price comparison sites often pay a fee when they are selected by a customer. How they get paid depends on the product.

# CHAPTER 4

## **PROPOSED SYSTEM**

## 4. PROPOSED SYSTEM

The proposed system consists of various modules:

A price comparison engine allows you to compare prices for products sold by various E Commerce websites. These price comparison tools allow you to compare prices across E commerce websites by typing in the product name to find your product. The price comparison tools then show you a list of retailers who are selling the same product, and how much they're selling it for. Price comparison engines skim through multiple e-commerce sites to gather data about products and services like prices etc. This information is then combined on a price comparison website and results are tailored according to a visitor's search request. This way when a buyer searches for a product on a price comparison website, the site compares and displays listings of the same product from multiple E-Commerce websites. The buyer can then compare the listings based on price deals, features, shipping costs, etc. to find the best deal.

### ADVANTAGES OF PROPOSED SYSTEM

Comparison shopping online allows shoppers to use a Price comparison engine to quickly locate their product, find the lowest price,

**1.Price comparison.** This is a big one! Here, consumers can quickly find what seller has the best price, rather than calling each and every physical store, waiting on the line for an associate, inquiring about the item, waiting for the employee to check...only to find out that the store is out of this product.

**2.Save on driving.** Of course, those who use a CSE to get exactly what they want are also saving on driving too. No need to waste gas and time going to three different retailers trying to find the right product when one hour on the computer can save a shopper a whole afternoon.

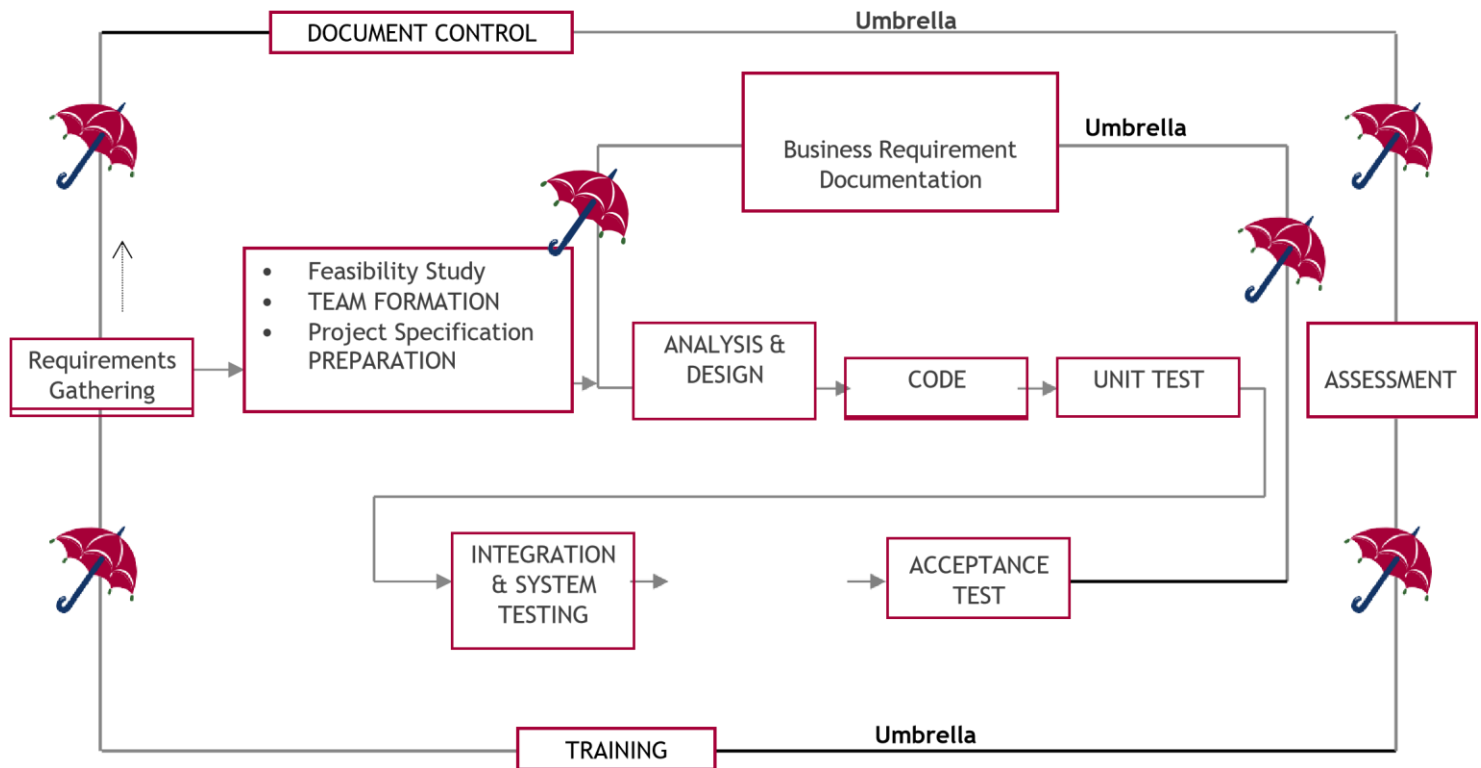
**3.Less Time Taking**

# **CHAPTER 5**

## **PROCESSSS MODEL USED**

### **WITH JUSTIFICATION**

## 5. PROCESS MODEL USED WITH JUSTIFICATION



### 5.1 SDLC(umbrella model)

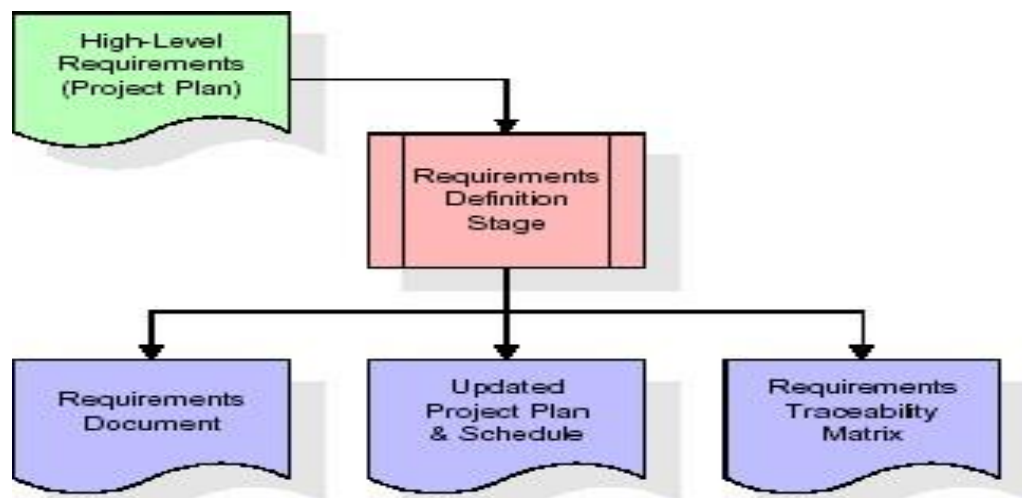
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

### Stages in SDLC

- ☐ Requirement Gathering
- ☐ Analysis
- ☐ Designing
- ☐ Coding
- ☐ Testing
- ☐ Maintenance

### Requirements Gathering stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



**Fig 5.2 Requirement gathering**

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this



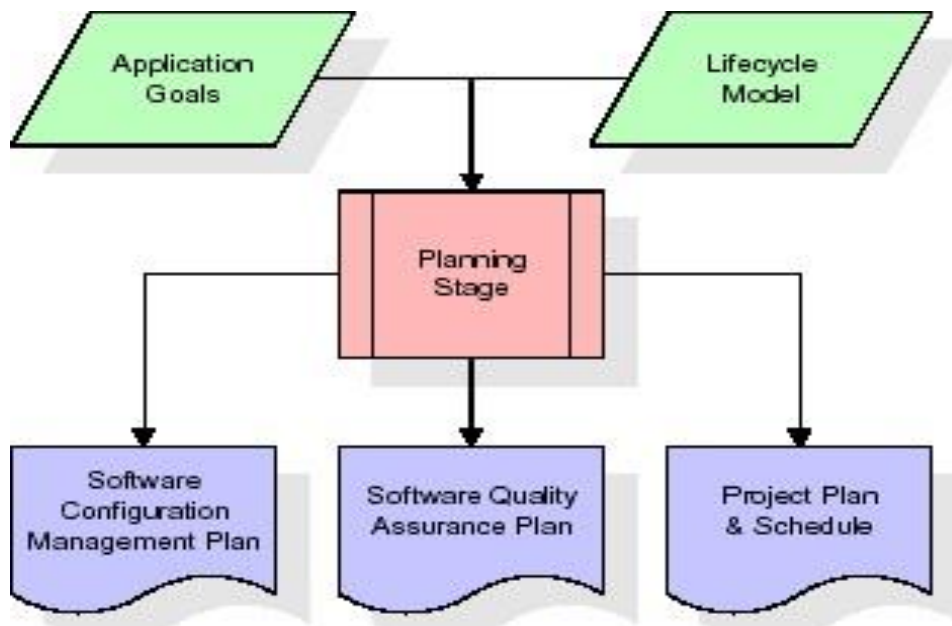
hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- Feasibility study is all about identification of problems in a project.
- No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
- Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

## **Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



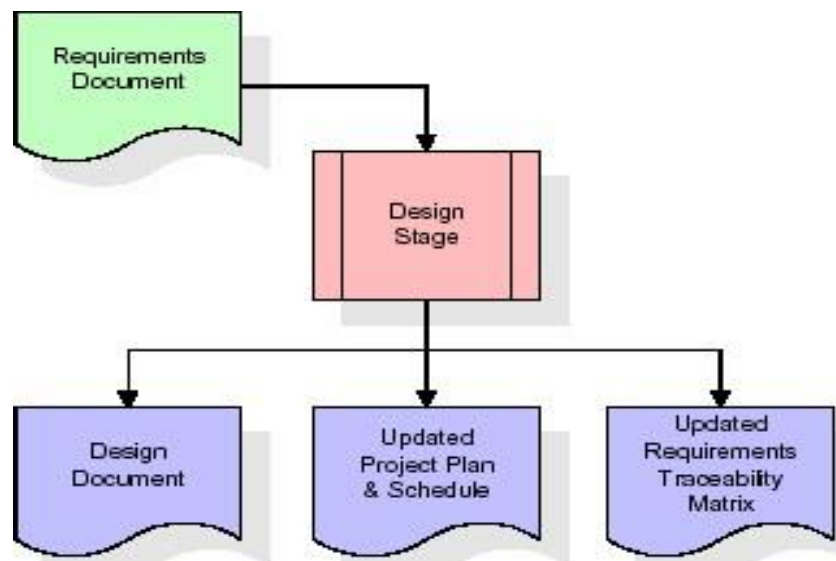
**Fig 5.3 Analysis stage**

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

## **Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the

software in sufficient detail that skilled programmers may develop the software with minimal additional input.



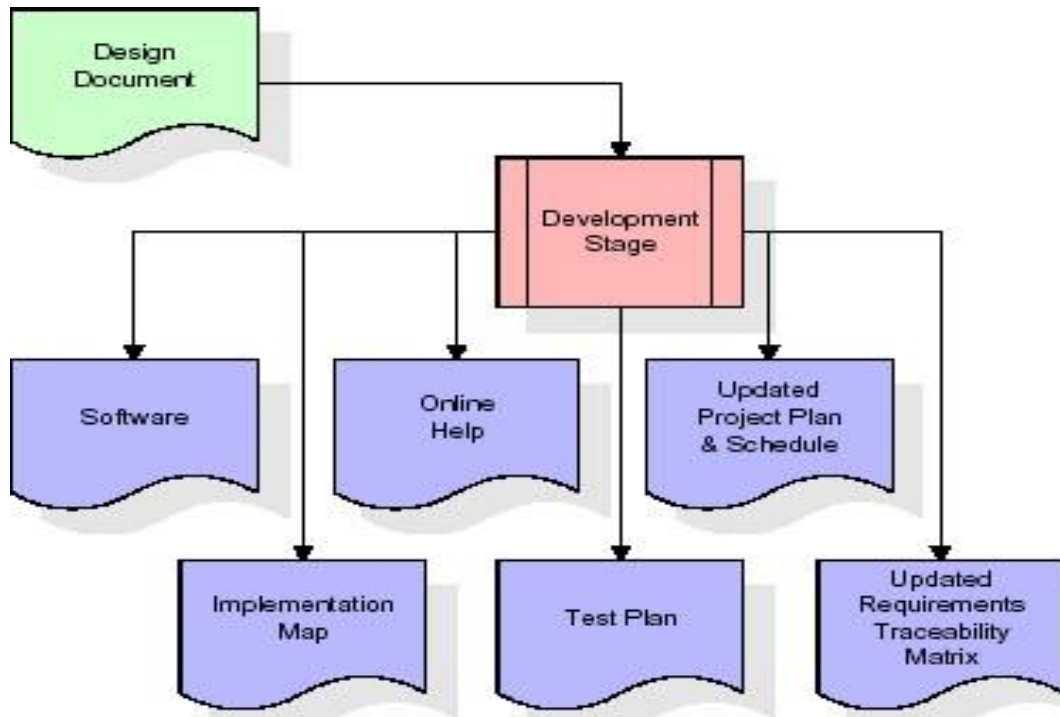
**Fig 5.4 Designing stage**

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

## **Development (Coding) Stage**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be

produced. Software artefacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artefacts, and an online help system will be developed to guide users in their interactions with the software.

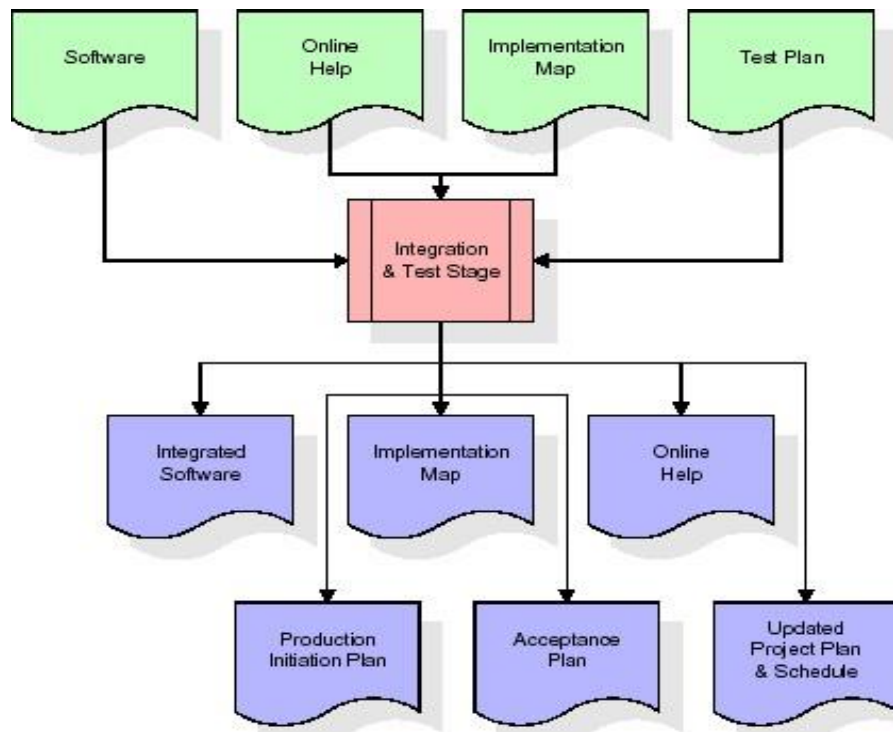


**Fig 5.5 Development stage**

The RTM will be updated to show that each developed artefact is linked to a specific design element, and that each developed artefact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

## **Integration & Test Stage**

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



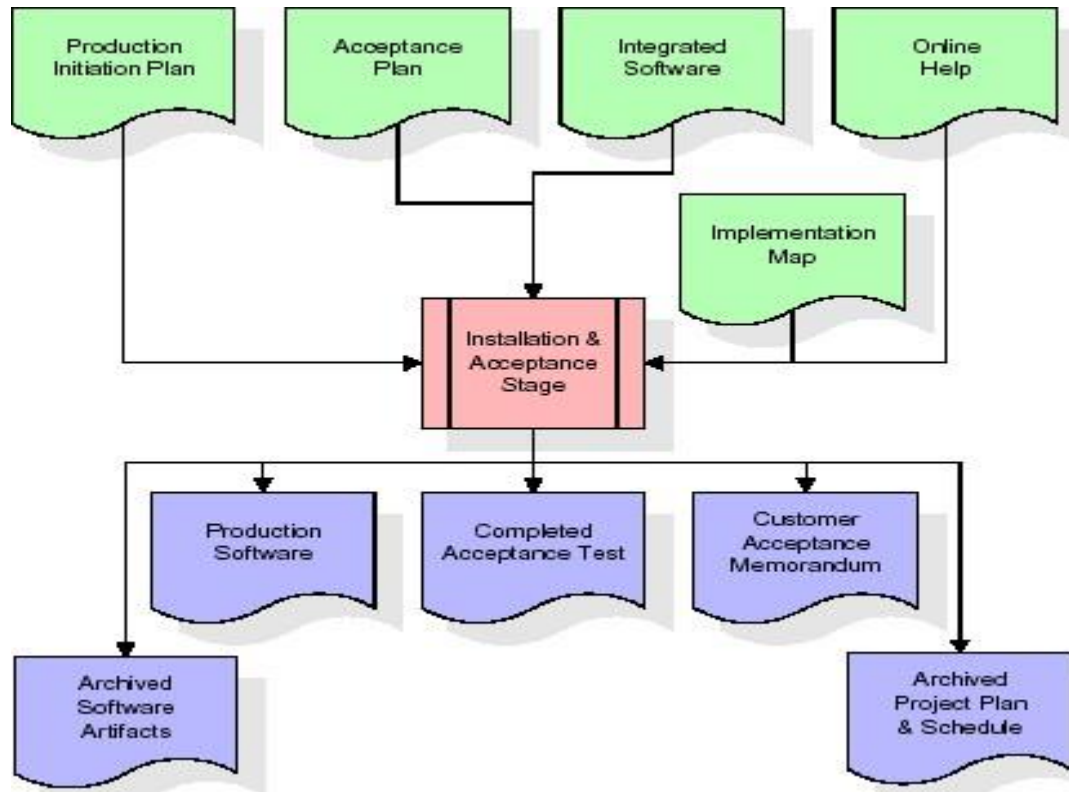
**Fig 5.6 Integration and test stage**

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

### □ **Installation & Acceptance Test**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



**Fig 5.7 Installation and Acceptance Stage**

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

## Maintenance

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**CHAPTER 6**

**SOFTWARE REQUIREMENT**

**SPECIFICATION**



## **6. SOFTWARE REQUIREMENT SPECIFICATION**

### **Overall Description**

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms what must be delivered or accomplished to provide value.
- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

#### **• ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the

system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

- **OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

- **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

# **CHAPTER 7**

## **EXTERNAL INTERFACE**

### **REQUIREMENTS**

## **7. EXTERNAL INTERFACE REQUIREMENTS**

### **User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

### **Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

### **Software Interfaces**

The required software is python.

### **HARDWARE REQUIREMENTS:**

- |             |   |                           |
|-------------|---|---------------------------|
| • Processor | - | Intel I3(min)             |
| • Speed     | - | 1.1 Ghz                   |
| • RAM       | - | 4GB(min)                  |
| • Hard Disk | - | 500GB(min)                |
| • Key Board | - | Standard Windows Keyboard |
| • Mouse     | - | Two or Three Button Mouse |
| • Monitor   | - | SVGA                      |

### **SOFTWARE REQUIREMENTS:**

- |                        |   |                  |
|------------------------|---|------------------|
| • Operating System     | - | Windows10/above  |
| • Programming Language | - | Python 3.7/above |

# **CHAPTER 8**

# **SYSTEM DESIGN**

## **8. SYSTEM DESIGN**

### **8.1 UML Diagram**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

- **User Model View**

- i. This view represents the system from the users perspective.
- ii. The analysis representation describes a usage scenario from the end-users perspective.

- **Structural Model view**

- i. In this model the data and functionality are arrived from inside the system.
- ii. This model view models the static structures.

- **Behavioural Model View**

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are to be built.

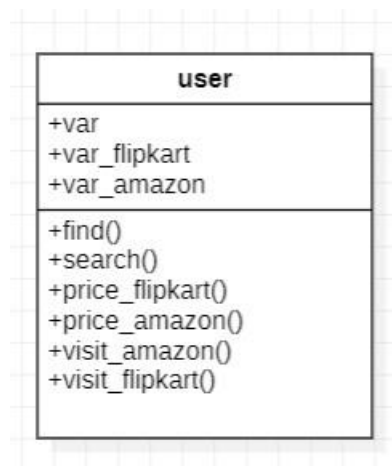
- **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

## Class Diagram:

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake



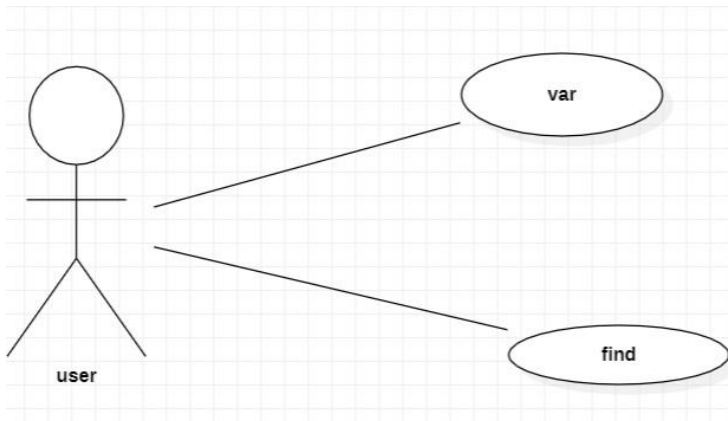
### 8.1 Class Diagram

## Use case Diagram:

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram

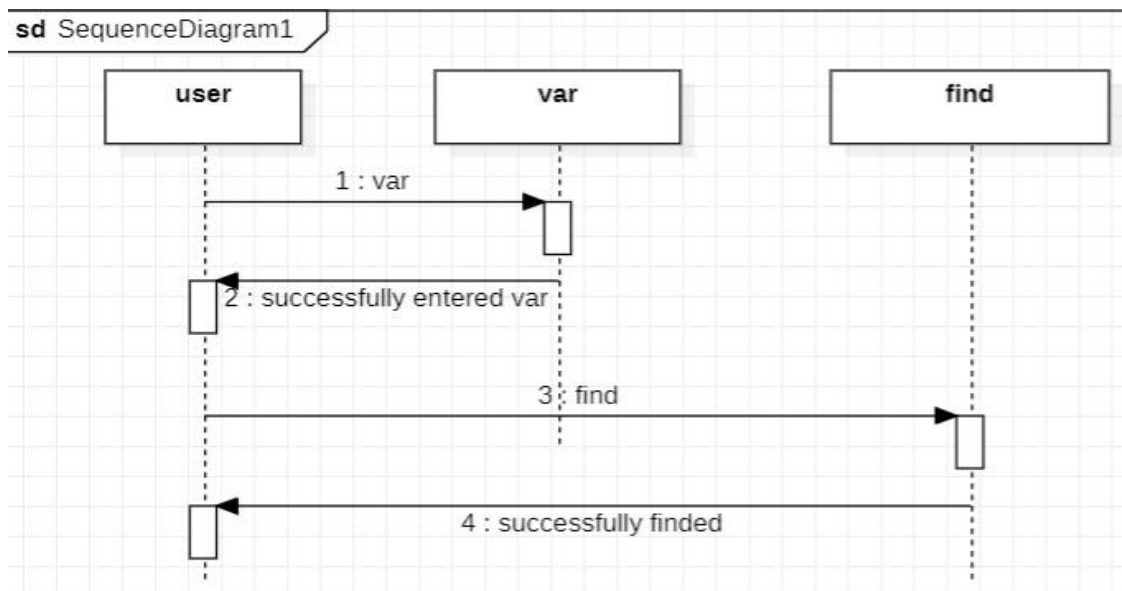


is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



## Sequence diagram:

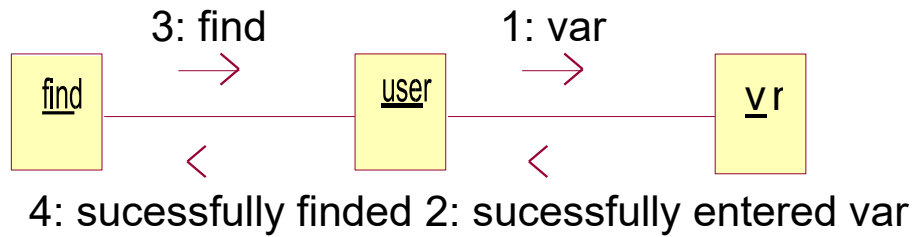
A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and **timing diagrams**.



### 8.3 Sequence diagram

#### Collaboration diagram:

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.

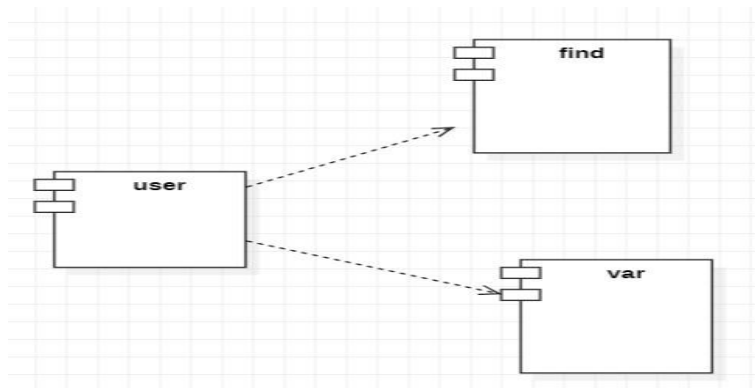


### 8.4 Collaboration Diagram

#### Component Diagram:

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

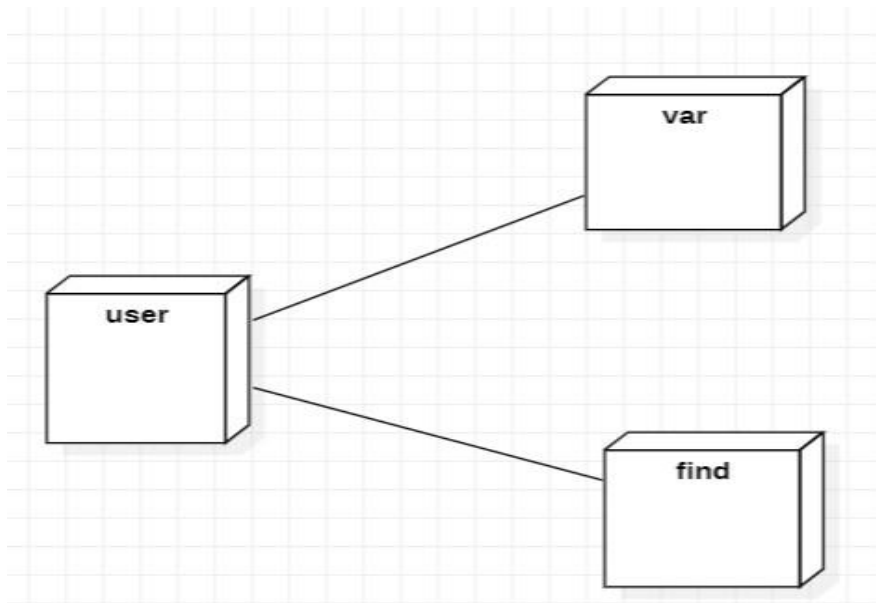


### 8.5 component Diagram

#### Deployment Diagram:

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

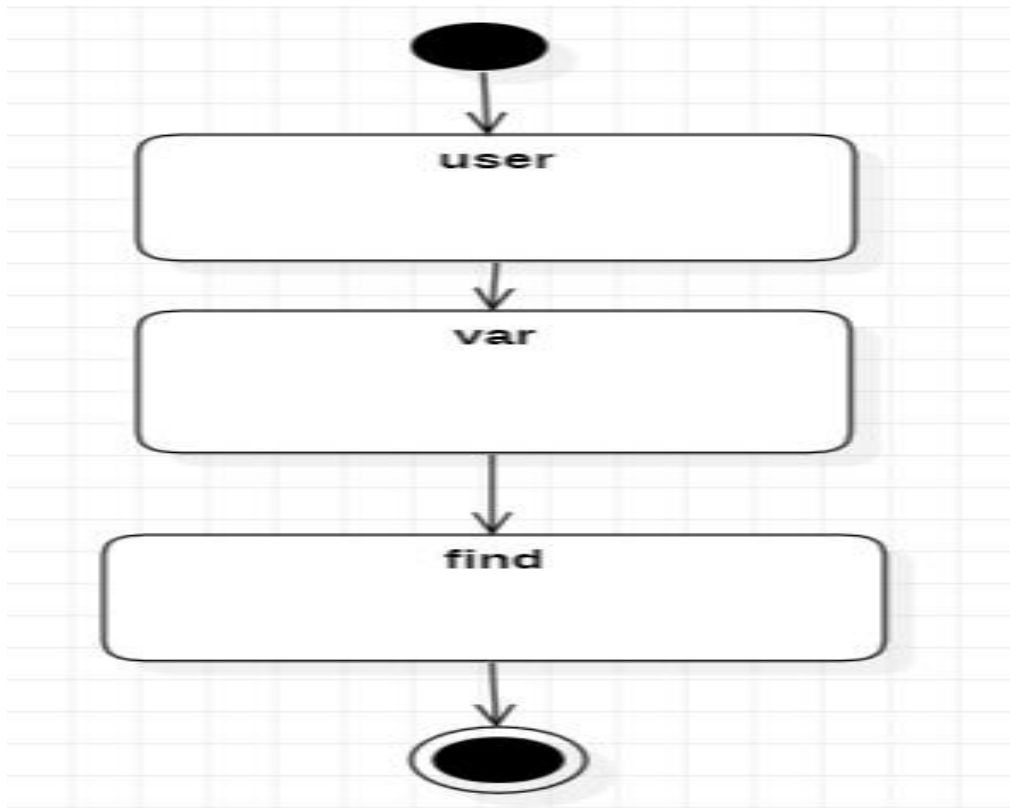
The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



### 8.6 Deployment Diagram

#### Activity Diagram:

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.



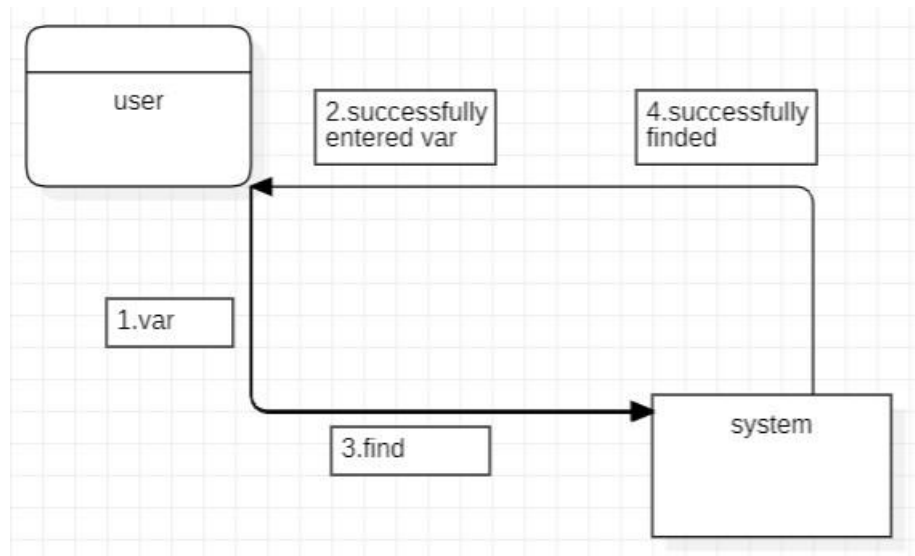
### 8.7 Activity Diagram

## Data Flow Diagram

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information

processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**8.8 data flow Diagram**

# **CHAPTER 9**

## **IMPLEMENTATION**

## **9. IMPLEMENTATION**

### **Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

### **History of Python**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

### **Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

### **Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

### **Features of Python**

#### **A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you



to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

## **Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

## **Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

## **Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

## **A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

## **Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using `import MySQLdb` .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

## **Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

## **Applications of Python**

### **1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

```
a = 2
b = 3
sum = a + b
print(sum)
```

### **2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

### 3. Expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

### 4. Great Community and Support

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

#### Sample Code:

```
from tkinter import *
from bs4 import BeautifulSoup
import requests
from difflib import
get_close_matches
import webbrowser
from collections import
defaultdict
import random

root = Tk()
root.geometry(
    "320x150")

class Price_compare:
```

```

def _init_(self, master):

    self.var      =      StringVar()

    self.var_ebay = StringVar()

    self.var_flipkart = StringVar()

    self.var_amzn = StringVar()

    label      =      Label(master,      text='Enter      the      product')
label.grid(row=0, column=0,padx=(30,10),pady=30)

    entry = Entry(master, textvariable=self.var)
entry.grid(row=0, column=1)

    button_find = Button(master, text='Find', bd=4, command=self.find)
button_find.grid(row=1, column=1, sticky=W, pady=8)

    def find(self):
self.product = self.var.get() self.product_arr =

        self.product.split() self.n = 1 self.key

        = ""

self.title_flip_var      =      StringVar()

    self.title_amzn_var = StringVar()

    self.variable_amzn = StringVar()

    self.variable_flip = StringVar()

for word in self.product_arr:

```

```

        if self.n == 1: self.key = self.key

            + str(word) self.n += 1

else:

    self.key = self.key + '+' + str(word)

self.window = Toplevel(root) self.window.title('Price
Comparison Engine') label_title_flip =
Label(self.window, text='Flipkart Title:')
label_title_flip.grid(row=0, column=0,
sticky=W)

label_flipkart = Label(self.window, text='Flipkart price (Rs):')
label_flipkart.grid(row=1, column=0, sticky=W)

entry_flipkart = Entry(self.window, textvariable=self.var_flipkart)
entry_flipkart.grid(row=1, column=1, sticky=W)

label_title_amzn = Label(self.window, text='Amazon Title:')
label_title_amzn.grid(row=3, column=0, sticky=W)

label_amzn = Label(self.window, text='Amazon price (Rs):')
label_amzn.grid(row=4, column=0, sticky=W)

entry_amzn = Entry(self.window, textvariable=self.var_amzn)
entry_amzn.grid(row=4, column=1, sticky=W)

self.price_flipkart(self.key)
self.price_amzn(self.key)

try:

```

```

        self.variable_amzn.set(self.matches_amzn[0])

except:

    self.variable_amzn.set('Product not available') try:

        self.variable_flip.set(self.matches_flip[0])

except:

    self.variable_flip.set('Product not available')


option_amzn = OptionMenu(self.window, self.variable_amzn, *self.matches_amzn)

option_amzn.grid(row=3, column=1, sticky=W)


lab_amz = Label(self.window, text='Not this? Try out suggestions by clicking on the title')
lab_amz.grid(row=3, column=2, padx=4)


option_flip = OptionMenu(self.window, self.variable_flip, *self.matches_flip)
option_flip.grid(row=0, column=1, sticky=W)


lab_flip = Label(self.window, text='Not this? Try out suggestions by clicking on the title')
lab_flip.grid(row=0, column=2, padx=4)


button_search = Button(self.window, text='Search', command=self.search, bd=4)

button_search.grid(row=2, column=2, sticky=E, padx=10, pady=4)


button_amzn_visit = Button(self.window, text='Visit Site', command=self.visit_amzn, bd=4)
button_amzn_visit.grid(row=4, column=2, sticky=W)

```

```

button_flip_visit = Button(self.window, text='Visit Site', command=self.visit_flip, bd=4)
button_flip_visit.grid(row=1, column=2, sticky=W)

```

```

def price_flipkart(self, key):

```

```

    url_flip = 'https://www.flipkart.com/search?q=' + str(key) +
    '&marketplace=FLIPKART&otracker=start&as-show=on&as=off'

```

```

    #url_flip =
    "https://www.flipkart.com/search?q="+key+"&otracker=search&otracker1=search&marketplace
    =FLIPKART&as-show=on&as=off"

```

```

    #print(key)

```

```

    map = defaultdict(list)

```

```

    self.headers = {
        'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36
        (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36'}

```

```

    source_code = requests.get(url_flip, headers=self.headers)

```

```

    soup = BeautifulSoup(source_code.text, "html.parser")

```

```

    self.opt_title_flip = StringVar()
    home =

```

```

    'https://www.flipkart.com'

```

```

    for block in soup.find_all('div', {'class': '_2kHMtA'}):

```

```

        title, price, link = None, 'Currently Unavailable', None

```

```

        for heading in block.find_all('div', {'class': '_4rR01T'}):

```

```

            title = heading.text

```

```
for p in block.find_all('div', {'class': '_30jeq3 _1_WHN1'}):
```

```
    price = p.text[1:] for l in block.find_all('a',  
  
{'class': '_1fQZEK'}):
```

```
        link = home + l.get('href')
```

```
    map[title] = [price, link]
```

```
user_input = self.var.get().title()
```

```
self.matches_flip = get_close_matches(user_input, map.keys(), 20, 0.1)
```

```
self.looktable_flip = {} for title in self.matches_flip:
```

```
    self.looktable_flip[title] = map[title]
```

```
try:
```

```
    self.opt_title_flip.set(self.matches_flip[0])
```

```
self.var_flipkart.set(self.looktable_flip[self.matches_flip[0]][0] + '.00')
```

```
self.link_flip = self.looktable_flip[self.matches_flip[0]][1] except
```

```
IndexError:
```

```
    self.opt_title_flip.set('Product not found')
```

```
def price_amzn(self, key):
```

```
    url_amzn = 'https://www.amazon.in/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-  
keywords=' + str(key)
```



```

# Faking the visit from a browser headers

= {

    'authority': 'www.amazon.com',

    'pragma': 'no-cache',

    'cache-control': 'no-cache',

    'dnt': '1',

    'upgrade-insecure-requests': '1',
    'user-agent': 'Mozilla/5.0 (X11; CrOS x86_64 8172.45.0) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/51.0.2704.64 Safari/537.36',

    'accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,;q=0.8,applicat
ion/signed-exchange;v=b3;q=0.9',

    'sec-fetch-site': 'none',

    'sec-fetch-mode': 'navigate',

    'sec-fetch-dest': 'document',

    'accept-language': 'en-GB,en-US;q=0.9,en;q=0.8',

}

map = defaultdict(list)

home = 'https://www.amazon.in'

```

```
proxies_list = ["128.199.109.241:8080", "113.53.230.195:3128", "125.141.200.53:80",
"125.141.200.14:80",

                "128.199.200.112:138",      "149.56.123.99:3128",      "128.199.200.112:80",
"125.141.200.39:80",

                "134.213.29.202:4444"]
```

```
proxies    =    {'https':    random.choice(proxies_list)}

source_code = requests.get(url_amzn, headers=headers)

plain_text    =    source_code.text    self.opt_title    =

StringVar()

self.soup = BeautifulSoup(plain_text, "html.parser")

# print(self.soup)

# print(self.soup.find_all('div', {'class': 'sg-col-inner'}))

for html in self.soup.find_all('div', {'class': 'sg-col-inner'}):

    title, link, price = None, None, None for heading in html.find_all('span', {'class': 'a-size-
medium a-color-base a-text-normal'}):

        title = heading.text for p in html.find_all('span',
{'class': 'a-price-whole'}):

            price = p.text

            for l in html.find_all('a', {'class': 'a-link-normal s-underline-text s-underline-link-text s-
link-style a-text-normal'}):
```

```
link = home + l.get('href') if
```

```
title and link:
```

```
map[title] = [price, link] user_input = self.var.get().title()
```

```
self.matches_amzn = get_close_matches(user_input, list(map.keys()), 20, 0.01)
```

```
self.looktable = {} for title in self.matches_amzn:
```

```
    self.looktable[title] = map[title]
```

```
self.opt_title.set(self.matches_amzn[0])
```

```
self.var_amzn.set(self.looktable[self.matches_amzn[0]][0] + '.00')
```

```
self.product_link = self.looktable[self.matches_amzn[0]][1]
```

```
def search(self):
```

```
    amzn_get =
```

```
    self.variable_amzn.get()
```

```
    self.opt_title.set(amzn_get) product
```

```
    = self.opt_title.get()
```

```
    price, self.product_link = self.looktable[product][0], self.looktable[product][1]
```

```
    self.var_amzn.set(price + '.00') flip_get = self.variable_flip.get()
```

```
    flip_price, self.link_flip = self.looktable_flip[flip_get][0], self.looktable_flip[flip_get][1]
```

```
    self.var_flipkart.set(flip_price + '.00') def visit_amzn(self):
```

```
    webbrowser.open(self.product_link) def visit_flip(self): webbrowser.open(self.link_flip)
```

```
if __name__ == "__main__":  
  
    c = Price_compare(root)  
  
    root.title('Price Comparison  
Engine') root.mainloop()
```

# **CHAPTER 10**

## **TESTING**

## **10. TESTING**

### **Implementation and Testing**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

### **Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

### **Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

## **System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

## **Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

## **Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

## **Acceptance Testing**

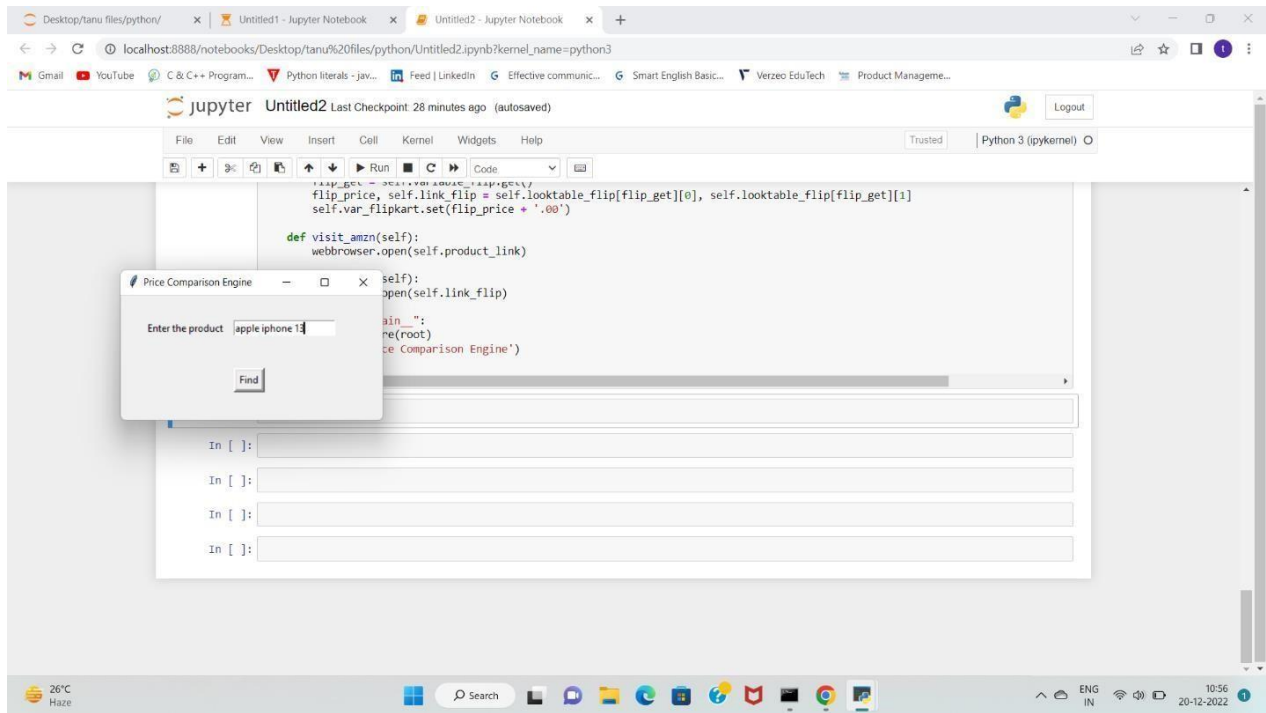
When that user find no major problems with its accuracy, the system passes through a final acceptance test. This test confirms that the system meets the original goals, objectives and requirements established during analysis without actual execution which eliminates wastage of time and money. Acceptance tests are on the shoulders of users and management, it is finally acceptable and ready for the operation.

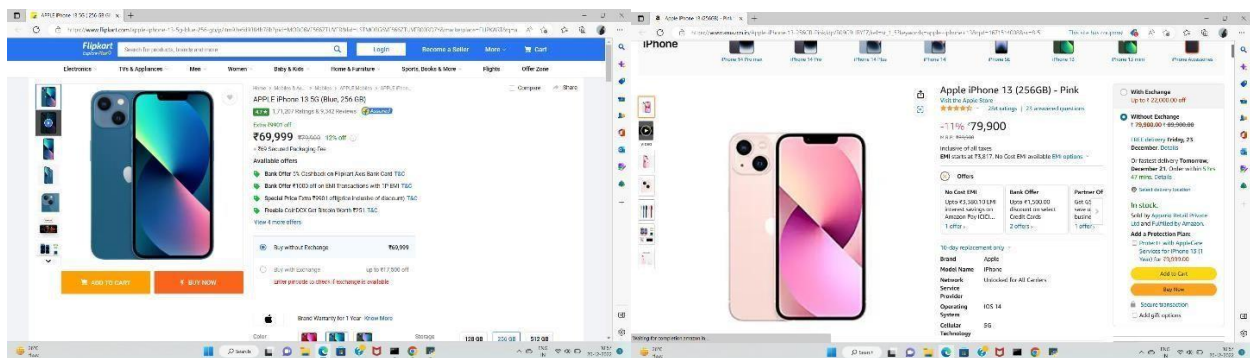
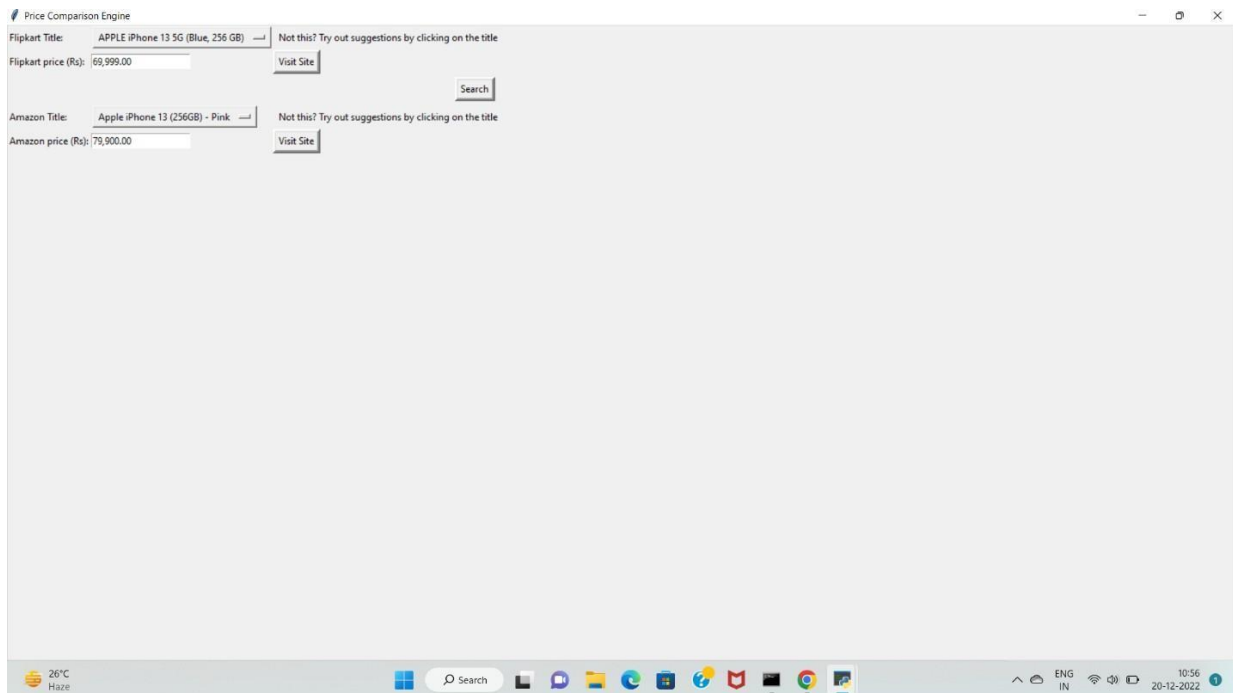


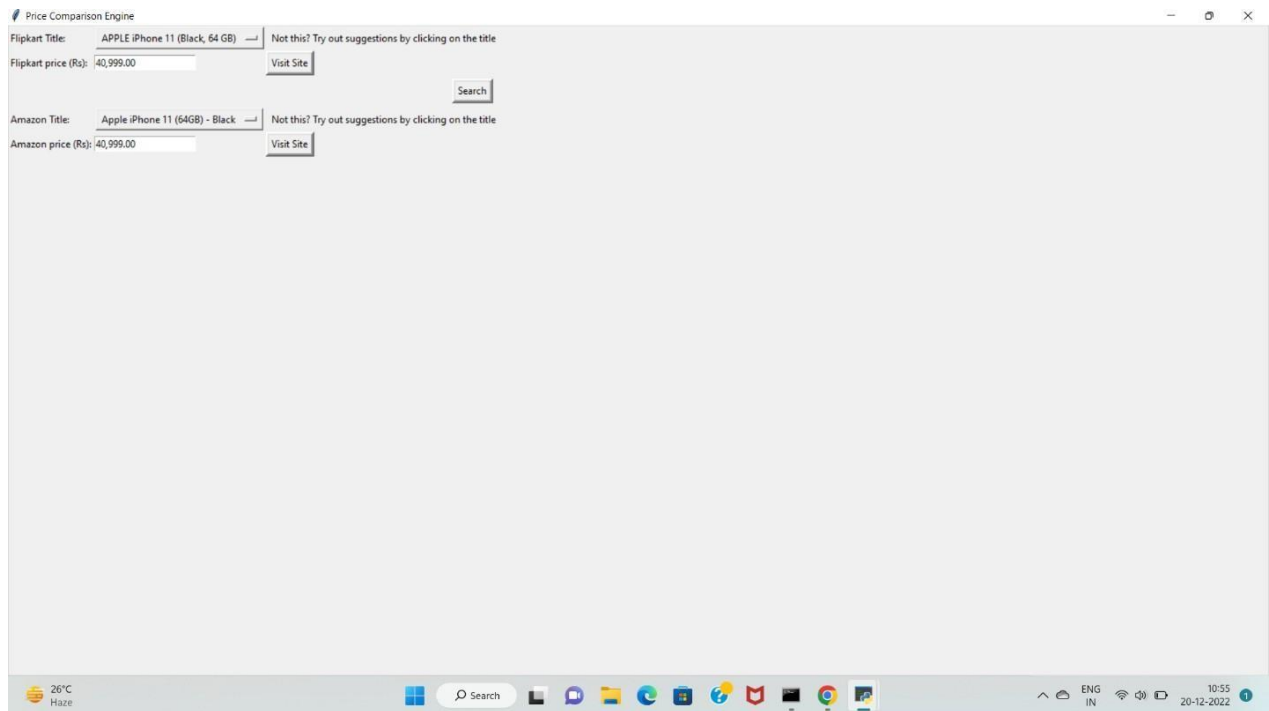
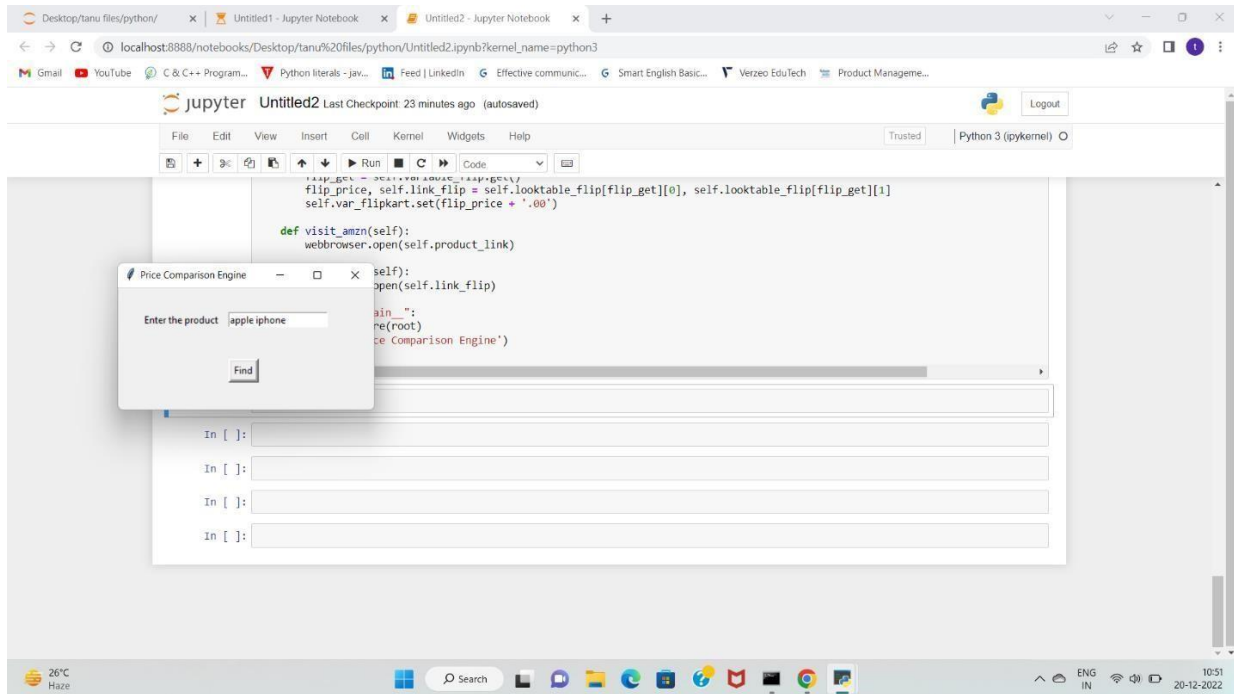
# **CHAPTER 11**

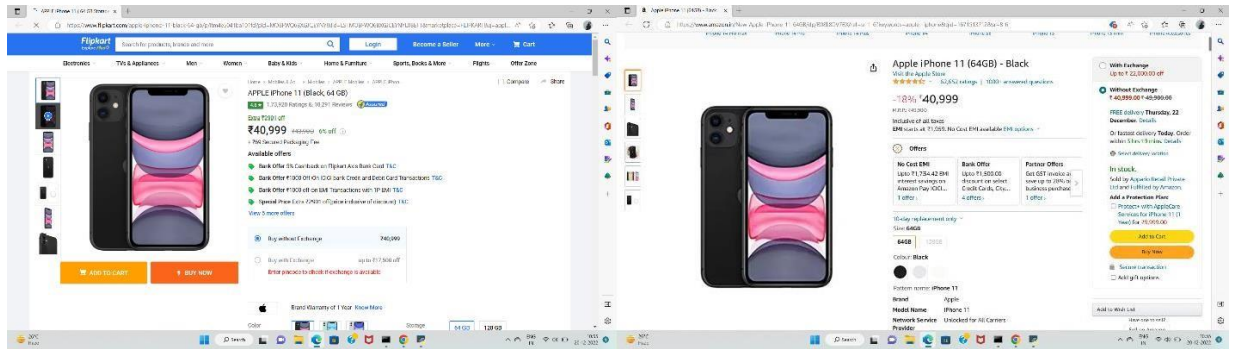
## **SCREENSHOTS**

## 11. SCREENSHOTS









# **CHAPTER 12**

# **CONCLUSION**

## 12. CONCLUSION

The Online Shopping Platform enables the user to find the best price available of a product on the internet with ease. As there are many available e-commerce websites with thousands of products, our system uses powerful web scrapping technologies and analytics to determine the best price of a product. The consistent application interface helps user to avoid unnecessary hassle to navigate through different websites while looking for best available price. Due to unification of search results from various platforms at a single destination, it becomes incredibly easy to use this platform. The integrated Notification System in our application provides a much coherent point of access and enables user to keep track of the prices of their favorite products. The Notification System communicates the user about the price drop whenever it occurs. Henceforth, reducing the inconvenience of signing up into different websites just to get notified for a price drop. The Online Shopping Platform not only determines the best product price but it also allows for an improved personal shopping experience by suggesting different products to users using the Personalized Recommendation system. The Analytics keeps track of a user purchasing history and provide a detailed analysis which helps user in maintaining their shopping budget and assess their shopping habits. Navigation through multiple platforms while shopping online is many times unintuitive and time-consuming. So, in our application navigation is made seamless by the Chatbot developed using Machine Learning algorithms. It assists the user in navigating through the platform and provides a smooth user experience while shopping. So, our system is able to reduce time and efforts in searching for the best price for a product from different websites. When it comes to comparing prices of a product, product recommendation and price tracking of a product, with the vast numbers of features available in our system, it becomes a one-stop solution for frequent online shoppers.

# **CHAPTER 13**

## **REFERENCES**



### 13. REFERENCES

- [1] Ms. Shakila Shaikh , Dr. Sheetal Rathi, "Recommendation system in E-commerce websites: A Graph Based Approached", 2017 IEEE 7th.
- [2] GUO Yan-yan, LIU Qi-cheng, "E-commerce Personalized Recommendation System Based on Multi-Agent", 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010).
- [3] Boris Galitsky, Dmitry Ilvovsky, "Chatbot with a Discourse Structure-Driven Dialogue Management", Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017, pages 87–90.
- [4] Leo Rizky Julian, Friska Natalia, "The use of web scraping in computer parts and assembly price comparison", 2015 3rd International Conference on New Media.
- [5] Shakra Mehak, Rabia Zafar, Sharaz Aslam, Sohail Masood Bhatti, "Exploiting Filtering approach with Web Scrapping for Smart Online Shopping", 2019 International Conference on Computing, Mathematics and Engineering Technologies – comet 2019.
- [6] Google shopping, wekepedia, [https://en.wikipedia.org/wiki/Google Shopping](https://en.wikipedia.org/wiki/Google_Shopping)
- [7] Google Shopping, Google, Accessed December 2002  
<https://support.google.com/faqs/answer/2987537?hl=en>
- [8] Julian McAuley, UCSD, "Amazon product data" 2018,  
<http://jmcauley.ucsd.edu/data/amazon/links.html>
- [9] Jianmo Ni, Jiacheng Li, Julian McAuley, "Justifying recommendations using distantly- labeled reviews and fined-grained aspects", Empirical Methods in Natural Language Processing (EMNLP), 2019.
- [10] Yuefeng Zhang, PhD, Published on 5 August 2020, <https://towardsdatascience.com/machine-learning-forbuildingrecommender-system-in-python9e4922dd7e97>