

"Real Time Product Analysis using Data Mining"

Project Report

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering

by

Ansari Mohd Azam Mohd Salim Reshma Bano(11CO53)

Jawahire Nakash Karim Zohra(11CO56)

Shaikh Anas Munawwar Aisha(11CO38)

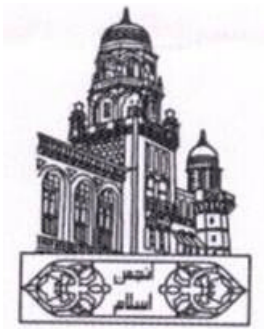
Siddiqi Muzammil Ahmad Mansoor Ahmad Razia Begum(11CO55)

Supervisor

Tabrez Khan

Co-Supervisor

Farhan Meer



Department of Computer Engineering,

School of Engineering and Technology

Anjuman-I-Islam's Kalsekar Technical Campus

Plot No. 2 3, Sector -16, Near Thana Naka, Khanda Gaon,

New Panvel, Navi Mumbai. 410206

Academic Year : 2014-2015

CERTIFICATE



**Department of Computer Engineering,
School of Engineering and Technology,
Anjuman-I-Islam's Kalsekar Technical Campus
Khanda Gaon, New Panvel, Navi Mumbai. 410206**

This is to certify that the project entitled *Real Time Product Analysis using Data Mining* is a bonafide work of **Ansari Mohd Azam Mohd Salim Reshma Bano(11CO53), Jawahire Nakash Karim Zohra(11CO56), Shaikh Anas Munawwar Aisha(11CO38), Siddiqi Muzammil Ahmad Mansoor Ahmad Razia Begum(11CO55)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Department of Computer Engineering**.

Name: Prof. Tabrez Khan

Name: Prof. Farhan Meer

Supervisor/Guide

Co-Supervisor/Guide

Prof. Tabrez Khan

Dr. Abdul Razak Honnutagi

Head of Department

Director

Project Approval for Bachelor of Engineering

This project entitled *Real Time Product Analysis using Data Mining* by *Ansari Mohd Azam Mohd Salim Reshma Bano , Jawahire Nakash Karim Zohra, Shaikh Anas Munawwar Aisha, Siddiqi Muzammil Ahmad Mansoor Ahmad Razia Bagum* is approved for the degree of *Bachelor of Engineering in Department of Computer Engineering*.

Examiners

1.

2.

Supervisors

1.

2.

Chairman

.....

Declaration

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ansari Mohd. Azam Mohd Salim Reshma Bano(11CO53)

Jawahire Nakash Karim Zohra(11CO56)

Shaikh Anas Munawwar Aisha(11CO38)

Siddiqi Muzammil Ahmad Mansoor Ahmad Razia Begum(11CO55)

Abstract

Title: Real Time Product Analysis using Data Mining

E-commerce is widely increasing business in the world with increasing revenues every year by manifold times. This is simple indication that more people each day are moving on-line for shopping. It suggests that the preference of the people are changing. People are preferring to buy products more and more from on-line e-commerce websites. They are successfully able to bring the customer in confidence and overcoming the risk involved in on-line transactions by carefully analyzing the behavior of the buyers/sellers and cultivating confidence among the resellers and buyers by developing strategies to attract more business and participation from people.

Here comes the existence of price comparison engines into picture which enable the buyers to compare products from different E-commerce websites thus facilitating the buyers to purchase the product at the cheapest price. There are many factors that are involved in developing a successful product. The success of the product is defined by the usefulness and the influence the product has on the human life. The Real Time Product analysis using data mining is enables the real time comparison of products on the basis of the prices and other parameters that are obtained though different E-commerce websites. To obtain prices and other parameters of the products from different E-commerce websites , a web Crawler is used that can crawl to different E-commerce websites and fetch the URLs of the products. Once the URL of the products are available, scrapper scrape the details that were abstracted within the URL. This scrapped data is then stored in the database. Then comparison among the products of the different E-commerce websites is being made by using techniques such as inverted indexing. This way the Project aims to provide a solution which grants power in the hands of the users to purchase genuine products at genuine prices and saving user's time , money and efforts.

Ansari Mohd. Azam Mohd Salim Reshma Bano(11CO53)

Jawahire Nakash Karim Zohra(11CO56)

Shaikh Anas Munawwar Aisha(11CO38)

Siddiqi Muzammil Ahmad Mansoor Ahmad Razia Begum(11CO55)

B.E. (Computer Engineering)
University of Mumbai.

Contents

Project Approval for Bachelor of Engineering	ii
Declaration	iii
Abstract	iv
Table of Contents	viii
List of Figures	x
list of Tables	xi
Keywords And Glossary	xii
1 Project Overview	1
1.1 Introduction	1
1.1.1 Motivation	3
1.1.2 Advantages Over Current System	4
1.2 Proposed System Architecture	5
1.2.1 Formulation of Problem With using Technology	5
1.3 Organization of the Project	7
2 Review Of Literature	8
2.1 A Survey of the Comparison Shopping Agents Based Decision Support Systems.	8
2.1.1 Discription	8
2.1.2 Pros	8
2.1.3 Cons	9
2.1.4 How we overcome Those problem in Project	9
2.2 A Multi-Agent System for Electronic Eommerce	9
2.2.1 Discription	9
2.2.2 A System for Electronic Commerce	9
2.2.3 SMACE	10
2.2.4 Pros	10
2.2.5 Cons	11
2.2.6 How we overcome Those problem in Project	11
2.3 Agent Based Mrket Place System for more Fair Efficient Transaction	11
2.3.1 Description	11
2.3.2 Simple Market Place (SMP)	11
2.3.3 Broker-Based Market Place (BBMP)	13

2.3.4	Pros	13
2.3.5	Cons	14
2.3.6	How we overcome Those problem in Project	14
2.4	Building Comparison Shopping Brokers on the Web	14
2.4.1	Description	14
2.4.2	The Online-Shopping Task	15
2.4.3	ShopBot: A Comparison-Shopping Agent	15
2.4.4	Pros	15
2.4.5	Cons	15
2.4.6	How we overcome Those problem in Project	16
2.5	A More Scalable Comparison Shopping Agent	16
2.5.1	Description	16
2.5.2	Pros	16
2.5.3	Cons	16
2.5.4	How we overcome Those problem in Project	16
2.6	Technological Review	17
2.6.1	Java	17
2.6.2	Jsoup	17
2.6.3	MongoDB	17
2.6.4	Elasticsearch	17
3	Requirement Analysis	18
3.1	Platform Requirement :	18
3.1.1	Supportive Operating Systems :	18
3.2	Software Requirement :	18
3.3	Hardware Requirement :	19
3.3.1	Hardware Required For Project Development:	19
4	Project Design	20
4.1	Design Approach	20
4.2	Software Architectural Designs	20
4.2.1	Front End Designs	22
4.2.2	Component Diagram	22
4.2.3	Deployment Diagram	23
4.3	Database Design	24
4.3.1	E-R Diagram	24
4.4	Work-flow Design	25
4.4.1	Flow Diagram	25
5	Implementation Details	27
5.1	Assumptions And Dependencies	27
5.1.1	Assumptions	27

5.1.2	Dependencies	27
5.2	Implementation Methodologies	28
5.2.1	Modular Description of Project	28
5.3	Detailed Analysis and Description of Project	28
5.3.1	Usecase Report	29
5.4	Class Diagram	30
5.4.1	Class Diagram Report	31
6	Results and Discussion	33
6.1	Test cases and Result	33
6.1.1	Unit Testing	33
6.1.2	Functional Testing	35
7	Project Time Line	41
7.1	Project Time Line Matrix	41
7.2	Project Time Line Chart	42
8	Task Distribution	43
8.1	Distribution of Workload	43
8.1.1	Scheduled Working Activities	43
8.1.2	Members activities or task	44
9	Conclusion and Future Scope	48
9.1	Conclusion	48
9.2	Future Scope	48
	References	49
10	Appendix I	51
10.1	What is Elasticsearch?	51
10.1.1	What problems does Elasticsearch solve well?	51
10.2	Flask	52
10.2.1	Features of Flask	52
10.3	Retail E-Commerce sales in India from 2012 to 2017 (in billion U.S. dollars)	53
	Acknowledgment	54

List of Figures

1.1	Webcrawler	3
1.2	System Architecture	6
2.1	Algorithm for direct transaction	12
2.2	Algorithm for brokerbased transaction	12
2.3	Simple Market Place	13
2.4	Broker Market place	14
4.1	Software architecture Design	21
4.2	Front End Design	22
4.3	Component Diagram of Real Time Product Analysis	22
4.4	Deployment Diagram of Real Time Product Analysis system	23
4.5	E-R Diagram of Real Time Product Analysis system	24
4.6	Level 0 DFD of Real Time Product Analysis system	25
4.7	Level 1 DFD of Real Time Product Analysis system	25
4.8	Level 2 DFD of Real Time Product Analysis system	26
4.9	Level 3 DFD of Real Time Product Analysis system	26
5.1	Class Diagram	31
6.1	Url Extraction	34
6.2	Product Details	34
6.3	Data Storage	35
6.4	Indexing Result	36
6.5	product output	37
6.6	Home Page	37
6.7	Searched Product Result	38
6.8	Searched Product Result	38
6.9	Searched Product Result	39
6.10	vandor sites	39
6.11	vandor sites	40
7.1	Time Line Matrix	41
7.2	Time Line Chart	42

7.3 Time Line Chart	42
10.1 E-commerce sales in India	53

List of Tables

5.1	Usecase Report	30
5.2	Class Diagram Report	32
8.1	Scheduled Working Activities	43
8.2	Member Activities and Task	47

Keywords And Glossary

Keywords :

Web Crawler, Scrapper, Data Mining, Product Analysis, Comparison Engine, Business Analytics.

Glossary :

A

Add-on a piece of software that enhances and customizes Mozilla-based applications.

D

Data Mining the practice of examining large pre-existing databases in order to generate new information.

F

Frame Work a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful.

R

Real Time a narrative technique whereby events are depicted as occurring entirely within the span of and at the same rate as the depiction.

S

Shopping Bots Shopping bots are price comparison sites on the World Wide Web that automatically search the inventory of several different online merchants to find the lowest prices for consumers.

U

URLs A URL is one type of Uniform Resource Identifier (URI); the generic term for all types of names and addresses that refer to objects on the World Wide Web. The term "Web address" is a synonym for a URL that uses the HTTP / HTTPS protocol.

W

WebCrawler A web crawler (also known as a web spider or web robot) is a program or automated script which browses the World Wide Web in a methodical, automated manner. This process is called Web crawling or spidering. Many legitimate sites, in particular search engines, use spidering as a means of providing up-to-date data.

Chapter 1

Project Overview

1.1 Introduction

A large number of people nowadays gives the priority to on-line shopping over the Traditional Shopping, for that they are using smart devices such as tablets, mobile phones, laptop and desktop to access E-Commerce websites through the Internet. In addition, they want to get their desire product in best price.

In order to get desire product in minimum price they survey or searches the number of E-commerce sites. To address these challenges, several weird agents-based e-commerce systems and add-on have been proposed. But all those sites and technology do not satisfy the users demand due to restriction, limitation of all these technology limited to its range of domain. The Real time product analysis overcomes these issues. This technology satisfied the user demand. Hereby using this technology user can get their desire product in minimum price apart from this features this technology recommends best to buy product to user over the numbers of e-commerce website.

E-commerce application have a certain model and standards that are followed in the industry. In Real Time Product Analysis using Data Mining, Intelligent agent is used to crawl through to different websites to fetch URLs of different products. The intelligent agent is a web crawler or a Shop-Bot that is an automated program that fetches the URLs from different E-commerce applications. Most E-commerce applications only provide the products that are available with them at a particular rate. In most cases, users before purchasing the products on-line, they need to visit different E-commerce applications to find the particular products at the cheapest price. Real Time product Analysis using Data Mining solves this problem of user by providing user the products from different E-commerce applications at one place with different prices and schemes and offers that are offered by different E-commerce firms. This will provide the user privilege to choose products from different E-commerce applications which they consider is the best for them. The main elements of this technology are as follows:

a) Web Crawler b) Scraper

a) Web Crawler:

Web crawler is one of the main component of the Project. Since the product is price comparison engine, the first thing that is required is to collect large amount of data in terms of products from different E-commerce websites. Manually ,the collection of such large amount of data was not possible. So the best way to get these data is to create a web crawler also known as spider. For crawler to be more effective, it is necessary that the crawler is efficient ,concurrent and multi-threaded. For crawler to be multi-threaded, it is important that the synchronization among the threads are maintained. So use of blocking queue came into picture.

The main purpose of the crawler is to crawl different E-commerce websites and to fetch the URLs of the products from these websites. Every E-commerce website can be considered as a graph consisting of several nodes(Links or URLs)as shown in figure 1. The crawler must traverse to all these nodes and fetch these nodes. Once it has fetched the node, that node must be kept in a set of visited nodes so that no two same URLs are fetched. Threads that are created in the thread-pool must be limited so that they do not eat up the entire memory. And each thread that's been started has to be terminated. The Coordinating thread distributes the crawl job to the processing threads. These processing threads fetches the URLs and returns to the Coordinating threads. Thus the fetched URLs that we have in the set visited nodes are given to the scraper for scraping purpose.

Web crawler is one of the main component of the Project. Since the product is price comparison engine, the first thing that is required is to collect large amount of data in terms of products from different E-commerce websites. Manually ,the collection of such large amount of data was not possible. So the best way to get these data is to create a web crawler also known as spider. For crawler to be more effective, it is necessary that the crawler is efficient ,concurrent and multi-threaded. For crawler to be multi-threaded, it is important that the synchronization among the threads are maintained. So use of blocking queue came into picture. The main purpose of the crawler is to crawl different E-commerce websites and to fetch the URLs of the products from these websites.

Every E-commerce website can be considered as a graph consisting of several nodes(Links or URLs)as shown in figure 1. The crawler must traverse to all these nodes and fetch these nodes .Once it has fetched the node, that node must be kept in a set of visited nodes so that no two same URLs are fetched. Threads that are created in the thread-pool must be limited so that they do not eat up the entire memory. And each thread that's been started has to be terminated. The Coordinating thread distributes the crawl job to the processing threads. These processing threads fetches the URLs and returns to the Coordinating threads. Thus the fetched URLs that we have in the set visited nodes are given to the scraper for scraping purpose.

Web crawler is one of the main component of the Project. Since the product is price comparison engine, the first thing that is required is to collect large amount of data in terms of products from different E-commerce websites. Manually ,the collection of such large amount of data was not possible. So the best way to get these data is to create a web crawler also known as spider. For crawler to be more effective, it is necessary that the crawler is efficient ,concurrent and multi-threaded. For crawler to be multi-threaded, it is important that the synchronization among the threads are maintained. So use of blocking queue came into picture. The main purpose of the crawler is to crawl different E-commerce websites and to fetch the URLs of the products from these websites.

Every E-commerce website can be considered as a graph consisting of several nodes(Links or URLs)as shown in figure 1. The crawler must traverse to all these nodes and fetch these nodes .Once it has fetched the node, that node must be kept in a set of visited nodes so that no two same URLs are fetched. Threads that are created in the thread-pool must be limited so that they do not eat up the entire memory. And each thread that's been started has to be terminated. The Coordinating thread distributes the crawl job to the processing threads. These processing threads fetches the URLs and returns to the Coordinating threads.Thus the fetched URLs that we have in the set visited nodes are given to the scraper for scraping purpose.

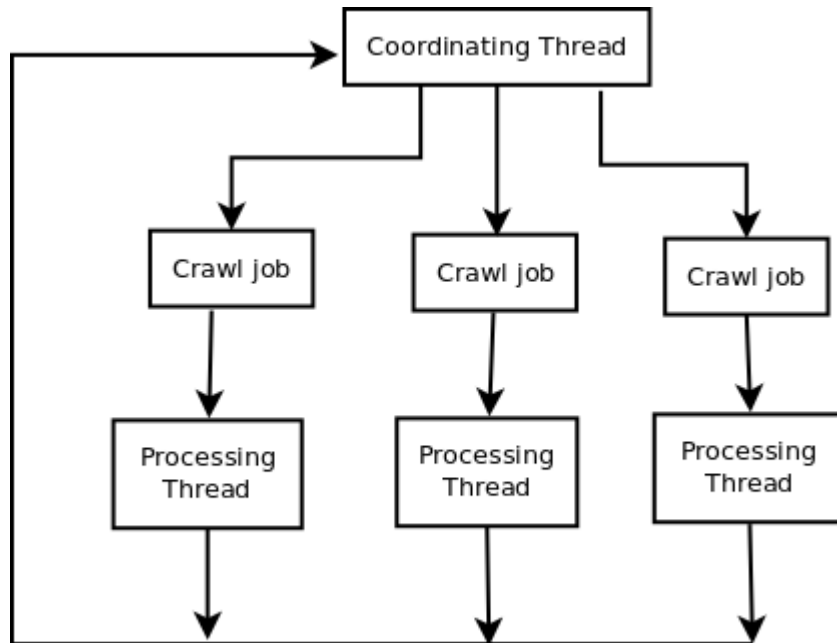


Figure 1.1: Webcrawler

b) Scraper:

Web scraping can be defined as a process of extracting HTML data from the URLs and then using this data for personal purposes. Once we have the fetched URLs with us then the job is to get the information that is abstracted within the URL. For example the mentioned URL[9] contains information such as the name of the product and the price and other related information on the link. These information are to be extracted for the purpose of comparison. The scrapper scrapes the information on this page on the basis of the tags in which the element. In this way the information can be extracted that are abstracted within the URLs. The extracted information is then stored in the database in the unstructured format.

1.1.1 Motivation

A frameworks which is completely based on theoretical leads to the development of shopping bots or smart tool agents can found in market of e-commerce websites, where Stigler [1] argued that consumers who value time will stop searching marginal benefits of search no longer outweigh the marginal cost. Hence the use of shopping bots or Real Time Product Analysis is not limited to simply typing just few keywords and waiting for the result. This technology helps to consumers in entire purchase decision-making process.

In the past, technologies or application were often referred to as agents, intelligent agents, software agents or intelligent assistants. In this technology, they are treated as regular software-based applications. It should be noted whenever the customer want to buy any product through online from any e-commerce website he/she does visit the many e-commerce websites for getting the desire product. Like this customer surf lots of time in visiting of e-commerce websites for getting the desire product. The customer not only surfs lots of time in visiting of e-commerce sites, and quality of products, but also he/she suffers from limited option to choose the product[2].

The proposed architecture, by making use of WebCrawler, offers a solution to those problems. That is, for customers there is no need to visits many e-commerce websites for buying the desire the product. These technology acts as an agent on the behalf of many e-commerce websites, a user can buy desire product without visiting many e-commerce website as well as without wasting of time in visiting of e-commerce website. Real Time Product Analysis using Data Mining provides the a stage to customer where the user can get various of variety of a particular product in reasonable price or where the customer can get particular product which fulfill the customer requirements. The user only needs to use this technology, the agent or shopbot or smart tool agent will provide the desire results for particular search. Hence this application minimizing the use of resources[3].

The Real Time Product Analysis is applicable in many application fields from which some areas are specially recognized as e-commerce, electronic auctions and comparison shopping websites. These areas need high transaction of money. Now the user can buy the product and put his money when he is confident that his money is secure and the agents can be trusted which are dealing with his money or transmitting some secure information and which providing the desire results. This is one reason which instigates much research effort in Real Time Product Analysis using Data Mining or shopping bots which has its own benefits when used in these areas.

1.1.2 Advantages Over Current System

A large number of people nowadays gives the priority to Online Shopping over the Traditional Shopping for that they are using mobile devices such as personal digital assistants (PDA) or mobile phones, Laptop Desktop to access E-Commerce websites through the Internet. In addition, they want to get their product in minimum price. In order to get desire product in minimum price they survey or searches the number of E-commerce sites. Current e-commerce websites, such as Flipkart, Snapdeal , Amazon, Jabong etc are typically providing the large range of different products over their websites.

On these E-commerce sites users can purchase their desire product from their personal computers, public terminals and their mobile but not in their desire or minimum price. For purchasing their desired product in minimum price they need to visit lots of sites and which is always a time-consuming process. To address these challenges, several weird agents-based e-commerce systems and add-on have been proposed. For example, a Jungle proposed by Amazon, Makkhi-choose add-on where user can get their desired product in minimum price on behalf of some e-commerce sites. Nevertheless, these sites and technology do not satisfy the users demand due

to restriction, limitations all these technology limited to its range of domain. The Real time product analysis or shopbot or smart Shopping agent overcomes these issues. This technology satisfied the user demand. Here by using this technology user can get their desire product in minimum price apart from this features this technology recommends best to buy product to user over the numbers of e-commerce websites[4].

1.2 Proposed System Architecture

E-commerce applications have a certain model and standards that is followed in the industry. In Real Time product analysis using Data Mining, Intelligent agent is used to crawl through to different websites to fetch URLs of different products. The intelligent agent is a web crawler or a ShopBot that is an automated program that fetches the URLs from different Ecommerce applications. Most E-commerce applications only provide the products that are available with them at a particular rate. In most cases, users before purchasing the products online, they need to go different E-commerce applications to find the particular products at the cheapest price.

Real Time product Analysis using Data Mining solves this problem of user by providing the user Products from different E-commerce applications at one place with different prices and schemes and offers that are offered by different E-commerce firms. This will provide the user privilege to choose products from different E-commerce applications which they consider is the best for them. For this, we make use of machine Learning algorithms that keep Track of the users behavior and searching patterns. Figure 1.2 describes the system architecture and its working procedure. An Intelligent agent is a web crawler running on the back-end of the website whereas fronted technology provides a graphical user interface (GUI) for the users to communicate with the system.

The explanation of the above architecture is as follows: The WebCrawler visits different E-commerce websites and fetch URLs from different Ecommerce websites. The Filter performs filtration so as to remove unuseful URLs. Then the filtered URLs are stored in the local database. The database used is MongoDB. Web crawler periodically fetches the data from different E-commerce websites and if updates are available, then web crawler carries the updates and updates the local database. Whenever client searches for the products in the search bar of the comparative website, the local database is queried so as to retrieve the required results.

The user can then compare products based on prices from different E-commerce websites. When user selects the best deal according to him and click on the buy button of the product then on clicking on the buy button, it triggers the user to original website to purchase the product.

1.2.1 Formulation of Problem With using Technology

It is really a tedious and time-consuming process that clients/buyers go through when shopping over the Internet. The process might start with a search for a particular product from which

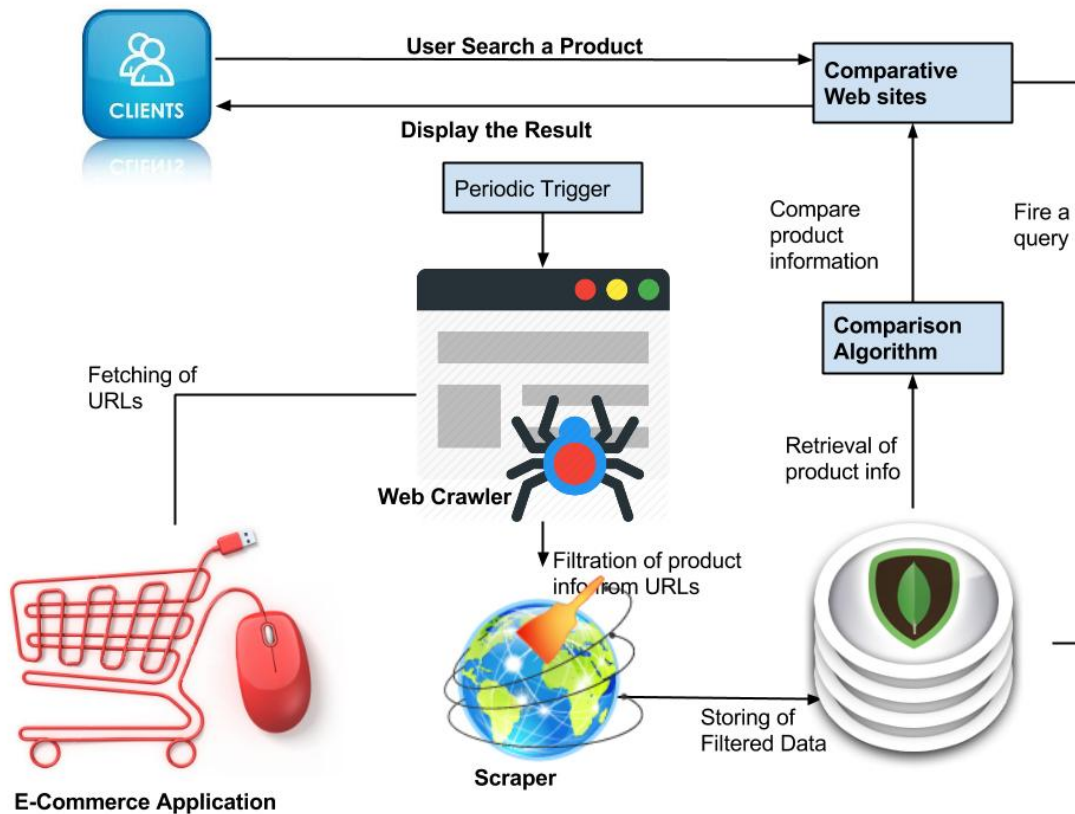


Figure 1.2: System Architecture

several links to stores are returned. The buyer typically must visit each website to check pricing and additional terms. This could involve considering alternate products from an online catalog, product availability, delivery options, return policy, or payment methods. After all relevant information is gathered, the buyer will then purchase the product using a credit card as payment method. It should be noted whenever the customer want to buy any product through online from any e-commerce website he/she does visit the many e-commerce websites for getting the desire product. Like this customer surf lots of time in visiting of e-commerce websites for getting the desire product. The customer not only surfs lots of time in visiting of e-commerce sites, and quality of products, but also he/she suffers from limited option to choose the product.

The Real Time Product Analysis is applicable in many application fields from which some areas are specially recognized as e-commerce, electronic auctions and comparison shopping websites. These areas need high transaction of money. Now the user can buy the product and put his money when he is confident that his money is secure and the agents can be trusted which are dealing with his money or transmitting some secure information and which providing the desire results. This is one reason which instigates much research effort in Real Time Product Analysis using Data Mining or shopping bots which has its own benefits when used in these areas.

The proposed architecture, by making use of WebCrawler, offers a solution to those problems. That is, for customers there is no need to visits many e-commerce websites for buying the desire the product. These technology acts as an agent on the behalf of many e-commerce websites, a user can buy desire product without visiting many e-commerce website as well as without wasting of time in visiting of e-commerce website. Real Time Product Analysis using Data

Mining provides the a stage to customer where the user can get various of variety of a particular product in reasonable price or where the customer can get particular product which fulfill the customer requirements. The user only needs to use this technology, the agent or shopbot or smart tool agent will provide the desire results for particular search. Hence this application minimizing the use of resources.

This project is also to recognize all countermeasures taken to avoid the attacks caused by a malicious host. These attacks may include the stealing of private data like credit-card number or e-money which it uses to buy the items, or manipulation in the results of an agent that it generates on the remote host and finally to protect itself from manipulation or alteration of its code itself. This work shows the use of cryptographic methods to solve the basic attacks and to protect the agent from the bigger attacks it uses the combination of encryption to solve the problem.

1.3 Organization of the Project

The remaining part of the project is organized as follows.

Chapter 2 presents a review of related work.

Chapter 3 introduces the Software and Hardware Requirement of the project.

Chapter 4 proposes the Project Design of the Project . It represent the architectural design, front end design and database design of the project.

Chapter 5 introduces the system model and some basic assumptions and Dependencies of our work.

Chapter 6 presents the Results and Test cases related work.

Chapter 7 Described the time management and time utilization during the Project implementation.

Chapter 8 Described the Workload distribution.

Chapter 9 provides some concluding remarks and directions of our future work.

Chapter 2

Review Of Literature

2.1 A Survey of the Comparison Shopping Agents Based Decision Support Systems.

2.1.1 Discription

In the paper by Bhavik Pathak, the Focus was to analyze shopping Comparison agents in the Context of Decision Support Systems (DSS). In this Paper, Four Component : Data, Models, interfaces & User Specific Customization were used. This four Component were used to Synthesizer the research work in the context of decision support system & to explore contemporary comparison shopping agents. In This Paper Provides suggestions for improving the decision support aspect of the shopping comparison agents. In this paper, the author provides purchasing patterns, preferences & price Index required to provide state of the art decision.

CSAs provide decision support tools to shoppers for comparing their purchase alternatives based on both the price and non-price (e.g. product, merchant reputation) based factors. They are increasingly becoming popular among shoppers. The CSA-based decision support systems are also proactively integrated by some merchants like Buy.com. CSAs do not have an equivalent counterpart in the offline world and they have generated a significant amount of interest among researchers in economics, marketing, and information systems fields[5].

2.1.2 Pros

- CSAs gives the detail specification of product and also the review and rating of a particular product. You can often rank search results by price, review rating, bestsellers etc.

2.1.3 Cons

- It may not include all retailers, meaning that search results may not be comprehensive Some rely on feeds from retailers.
- It may not be updated regularly enough therefore there may be chances of getting wrong result.

2.1.4 How we overcome Those problem in Project

- In our Project, by comparing the product from various E-commerce websites where there are larger number of retailers selling same product we can compare and can get best deal among large number of retailers and like this we are overcome the drawback of CSAs.
- We are using a periodic trigger in our project which will regularly update the product records stored in database.

2.2 A Multi-Agent System for Electronic Eommerce

2.2.1 Discription

In this paper by henri lopes,system follows a multi-party and multi-user negotiation approach.The system was tested by comparing the performance of agents that use multiple tactics with ones that include learning capabilities based on a specific kind of reinforcement learning technique.This paper uses the architecture of the SMACE(Sistema multi-agent pera comercio Electronica) system which is a multiagent system where user can create buyer and seller agents hat negotiate autonomously inorder to make deals on services they are requesting or offering[6].

2.2.2 A System for Electronic Commerce

In this research the author described basic negotiation approach and the architecture of the SMACE (Sistema Multi Agente para Comercio Electronico) system. It is a multiagent system for electronic commerce, where users can create buyer and seller agents that negotiate autonomously, in order to make deals on services they are requesting or offering. SMACE has been used as a test-bed for different negotiation paradigms, both user controlled and self-adaptive.

2.2.3 SMACE

SMACE allows users to create buyer and seller agents that negotiate under the model and protocol described above. The system was implemented with the JDK1.1.4 API, and uses the JATLite package to easily build agents that exchange KQML messages. The agents communicate with each other in the MarketPlace, which is an enhanced JATLite router, facilitating the message routing between the agents and working as an information center for the agents to announce themselves and search for contacts. According to the author the SMACE API consists of three layers built on top of the JATLite packages. These layers also consist of packages, and using the SMACE system can take place at any of them:

- **Infrastructure** this layer consists of two fundamental parts:
 1. **MarketAgent**: a template for the creation of market agents. It already has implemented the model of negotiation and its associated protocol. The only task left to the user starting in this layer is providing his own negotiation tactics.
 2. **MarketPlace**: the application that represents the marketplace, as a space where the agents meet and trade. It includes message routing and agent brokering facilities.
- **PlugTrade** this layer includes predefined market agents that can also be seen as examples of how an agent can be built using the MarketAgent template:
 1. **MultipleTacticAgent (MTA)**: a market agent that is able to use a weighted combination of three tactics, one from each of the tactic families described in the next section, to generate its negotiation proposals.
 2. **AdaptiveBehaviourAgent (ABA)**: a market agent that is able to weight the several tactic that it is using in an adaptive way, using Reinforcement Learning techniques.
- **UserInterface** this layer consists of an application that provides both an HTML user interface for the creation and monitoring of PlugTrade market agents operation and their persistence.

2.2.4 Pros

- In this a multiagent system is used where user can create buyer and seller agents hat negotiate autonomously in order to make a deals on service they are requesting.

2.2.5 Cons

- Since It is based on the multiagent system so it may create the conflict among the services of seller and the buyer.

2.2.6 How we overcome Those problem in Project

- We are providing a sophisticated methodology that preserves the confidentiality and integrity of the data by scraping and parsing that prevents XSS attacks and avoid the conflict among the seller and buyer services.

2.3 Agent Based Market Place System for more Fair Efficient Transaction

2.3.1 Description

In this paper, the author has suggested a broken-base synchronous transaction algorithm that could generate a more fair efficient transaction deal for both seller buyer. The algorithm was implemented in visual C++. The result showed better performance for comparison with the current two model types. The number of transaction increases by 21% the prices were adjusted upto 280% more efficiently in some transaction cases[7]. The agent carries out transactions in accordance to the transaction algorithm of a certain market system. The two major transaction algorithms are the direct transaction algorithm and the algorithm of transactions done by broker agents. These transaction algorithms show weakness in providing a fair and profitable deal to both the seller and buyer. The present paper suggests the broker-based synchronous transaction for overcoming such weaknesses.

2.3.2 Simple Market Place (SMP)

SMP is the transaction system applying the direct transaction algorithm. It is as shown below in Fig. When a user applies to sell or buy a product on the e-marketplace, the interface agent requests for the information on the product that must be registered into the product database. And based on the information, the interface agent generates the user agent. The generated user agent carries out the transaction in the e-marketplace for the user. And when a deal is made, the user agent notifies the results to the user. The user agent is classified into the buyer agent and the seller agent.

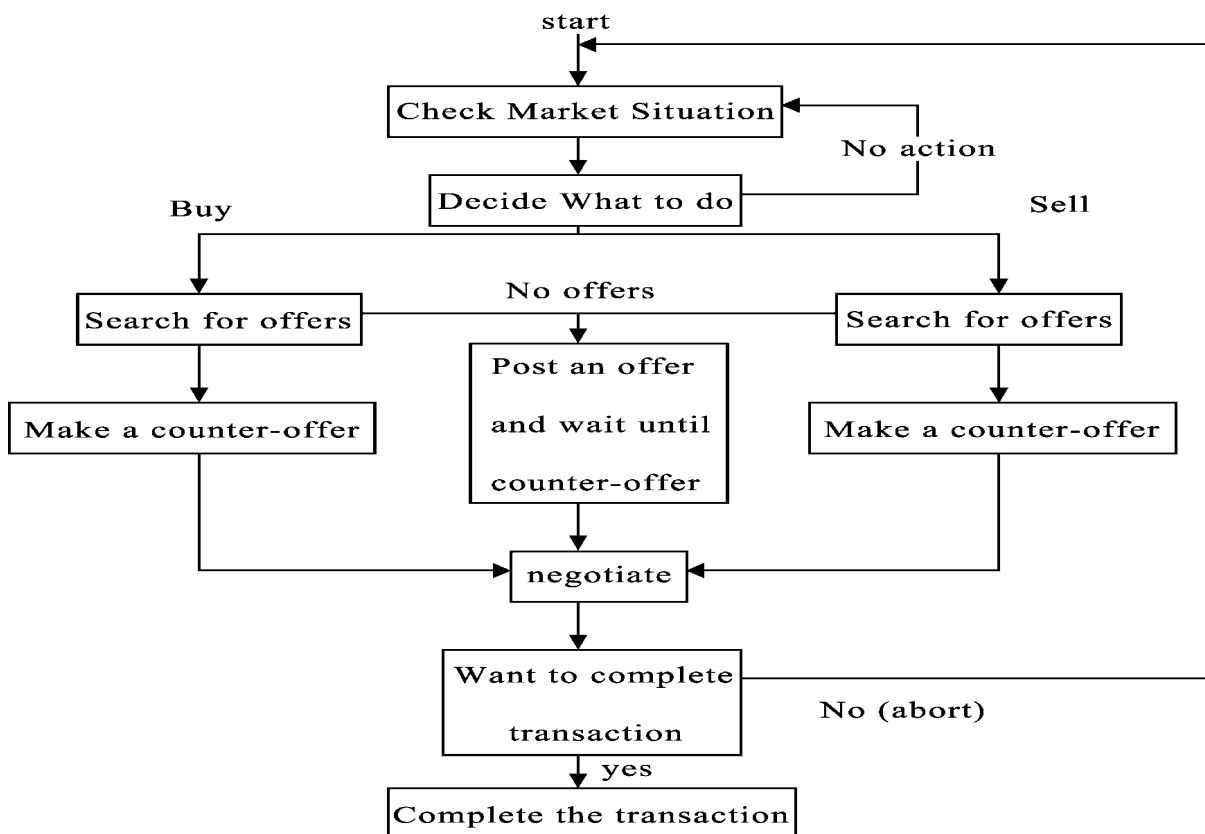


Figure 2.1: Algorithm for direct transaction

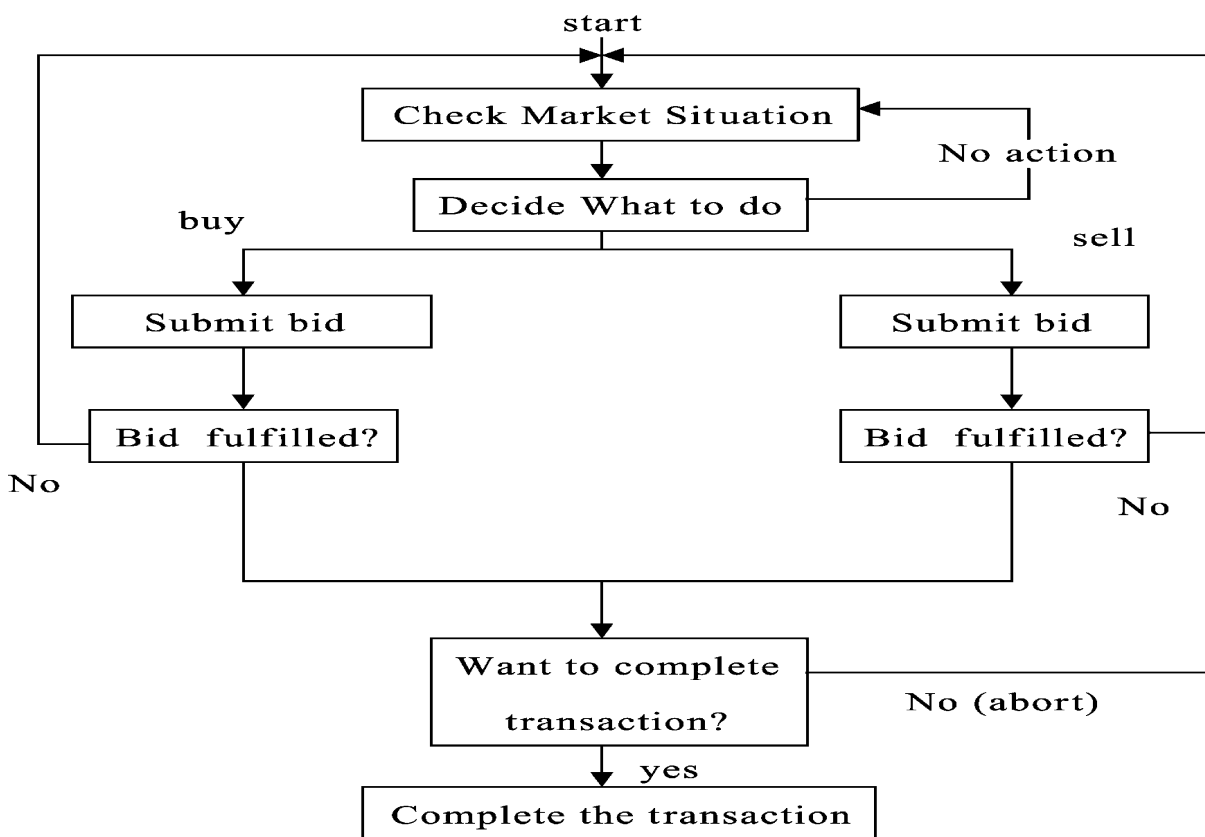


Figure 2.2: Algorithm for brokerbased transaction

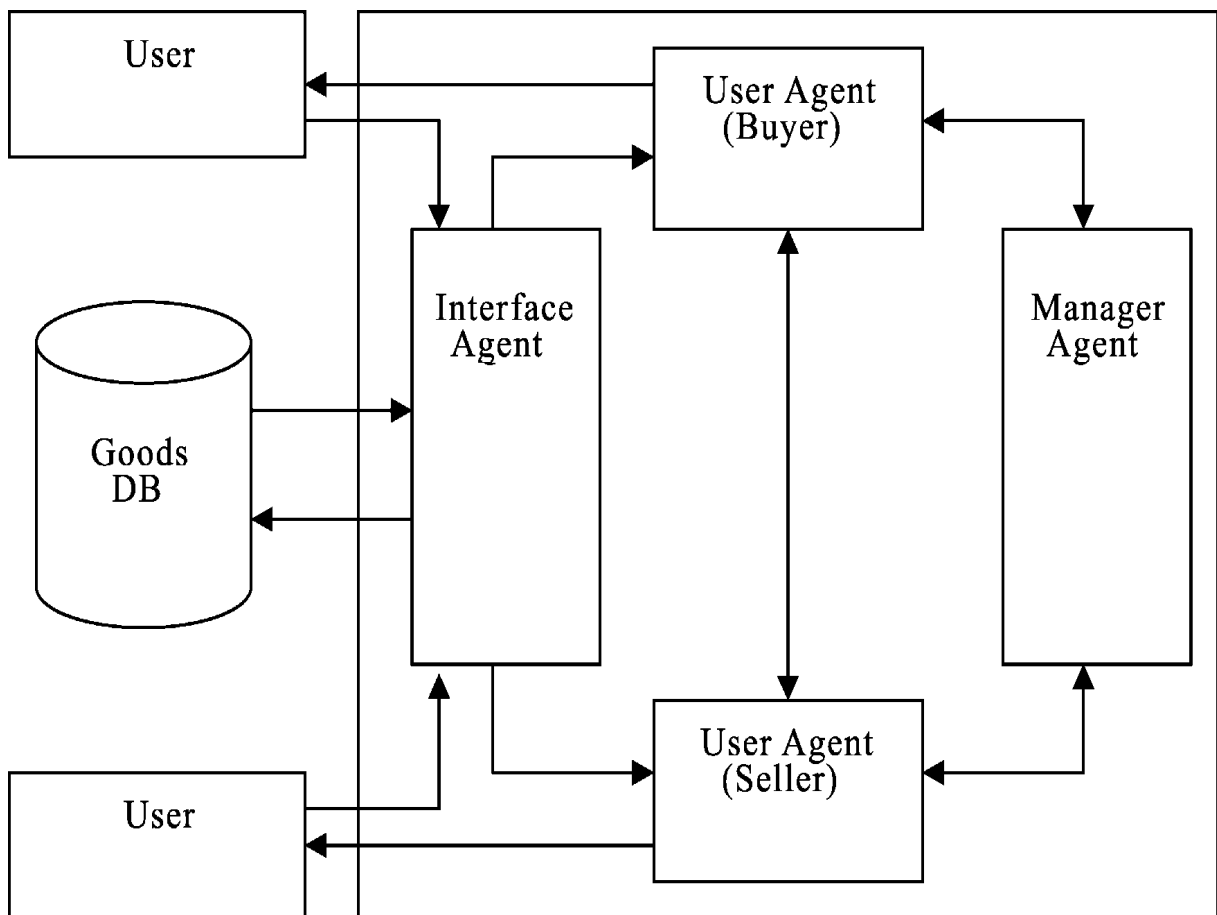


Figure 2.3: Simple Market Place

2.3.3 Broker-Based Market Place (BBMP)

BBMP is the transaction system applying transaction through broker agents as the transaction algorithm. It is organized as shown below in Fig. The interface agent and the product database operate just the same as those in the SMP described previously. The BBMP user agentTM's basic functions are similar to that of the SMP user agent, except for the fact that the BBMP user agent cannot search for the transaction partner itself. Instead, the user agent requests to the broker agent for the right transaction partner. The broker agent searches for the partner that offers the best price to the user agent.

2.3.4 Pros

- It uses the synchronous transaction algorithm that provide the more fair and efficient transaction deal for both seller and buyer.
- Broker-based synchronous transaction showed greater performance in terms of number of transactions achieved and rate of best deals made.

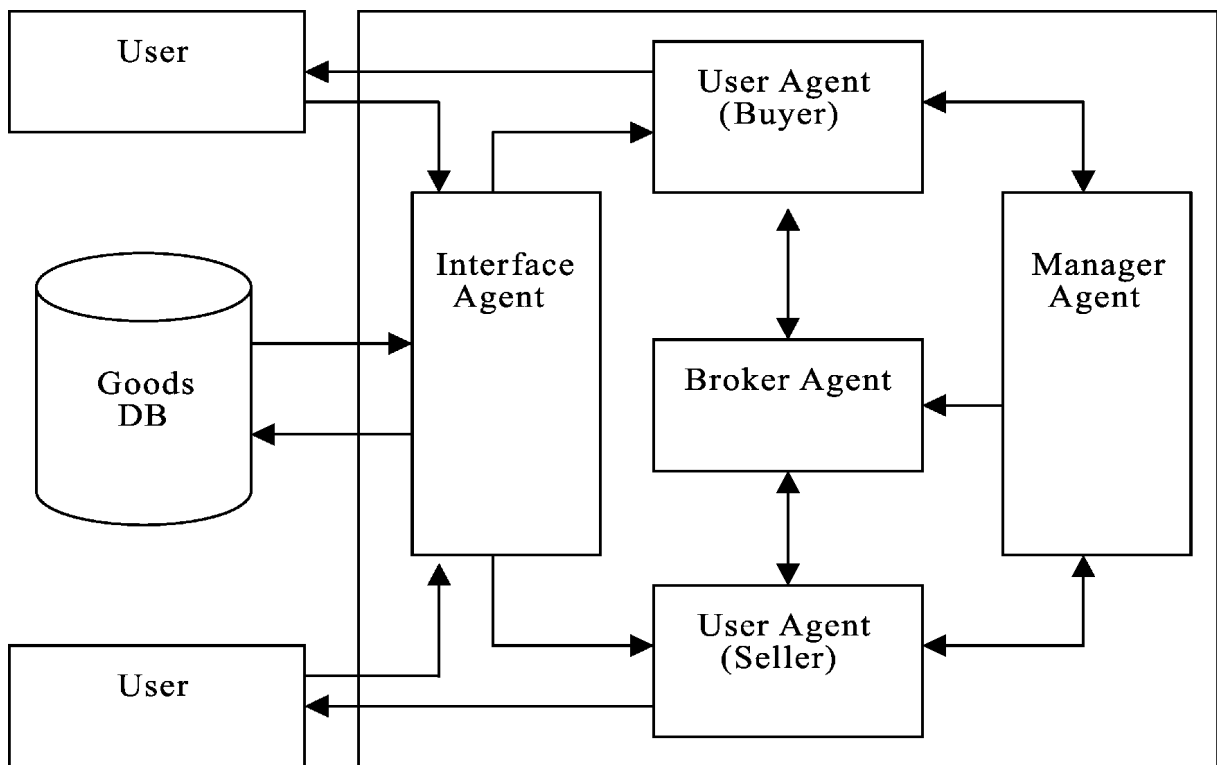


Figure 2.4: Broker Market place

2.3.5 Cons

- The agents took more time in making a deal in the broker-based synchronous transaction than in the others.
- The transaction processing time is weak in broker-based synchronous transaction system.

2.3.6 How we overcome Those problem in Project

- In our project we are making the efficient agent (scraper) and parser which will be capable to process the transaction quickly. We are using the Elastic-search tool that will help to make searching and transaction very efficient.

2.4 Building Comparison Shopping Brokers on the Web

2.4.1 Description

This paper presents an architecture for comparison shopping brokers that provides peculiar information in unstructured sources like the web heterogenous business data on online data.

The objective of this paper is to propose an architecture for information brokers based on agent technology that retrieves data from e-commerce websites , providing information about products vendors to consumer with focus an comparison[8]. In this paper author discussed about the online shopping task and ShopBot: A comparison agent.

2.4.2 The Online-Shopping Task

In this the author long-term goal is to design, implement, and analyze shopping agents that can help users with all aspects of online shopping. The capabilities of a sophisticated shopping assistant would include: 1) helping the user decide what product to buy, e.g., by listing what products of a certain type are available, 2) finding specifications and reviews of them, 3) making recommendations, 4) comparison shopping to find the best price for the desired product, 5) monitoring What's new lists and other sources to discover new relevant online information sources, 6) and watching for special offers and discounts.

2.4.3 ShopBot: A Comparison-Shopping Agent

Author initial research focus has been the design, construction, and evaluation of a scalable comparison shopping agent called ShopBot. ShopBot operates in two phases: in the learning phase, an online learner creates a vendor description for each merchant; in the comparison shopping phase, a real-time shopper uses these descriptions to help a person decide which store offers the best price for a given product.

2.4.4 Pros

- This method presents a unique idea of brokering system that is revolutionary.
- This method proposes an architecture for information brokers based on agent technology that retrieves data from e-commerce websites , providing information about products vendors to consumer with focus an comparison.

2.4.5 Cons

- The agents in the system provides a security threat to the application leaving many loopholes that can act as an onion ring.
- This can hamper the confidentiality and integrity of the data where different agents can misinterpret the data according to their own profit.

2.4.6 How we overcome Those problem in Project

- We are providing a sophisticated methodology that preserves the confidentiality and integrity of the data by scraping and parsing that prevents XSS attacks .Therefore scrapers and parsers are written to overcome the above cons.

2.5 A More Scalable Comparison Shopping Agent

2.5.1 Description

This paper propose a more scalable comparison shopping agent. The system presents a robust automatic shopping mall learning algorithm a ontology generation method.The main idea of the proposed algorithm is to determine the position of a product description used from the html source of a search result page by recognizing a repeated pattern of logical line information. Real time Analysis using DataMining has incorporated the model & methods that the above paper have used & has attempted to make shopping comparison engine more accurate by implementing recognition system that makes use of choice set by using agent that continuously runs behind the back 24/7 enabling Real time data analysis[9].

2.5.2 Pros

- This agent provides a very sophisticated methodology of scaling shopping comparison engine. This agent proposes an algorithm that determines the position of a product description used from the html source of a search result page by recognizing a repeated pattern of logical line information.

2.5.3 Cons

- This agent entirely focuses on scaling the shopping agent while it neglects the efficiency that is required for searching the products precisely.

2.5.4 How we overcome Those problem in Project

- To overcome the searching problems ,the use of elastic search is being made that is a search server based on lucene. Better indexing is being performed that leads to better searching.

2.6 Technological Review

2.6.1 Java

In implementing the project, we are making use of java language as the main language. The crawler, scraper and parser have been developed in java. Since java is platform independent and has rich functionality such as sophisticated multi-threading facility therefore java has been roped in to develop the crawler ,scraper and parser.

2.6.2 Jsoup

Jsoup is parser library provided by MIT .The use of jsoup is to parse the html pages. The scraper , crawler and parser have been developed by making use of jsoup library. jsoup provides functionality such as selector elements and parser elements to parse the html page and selects the html tags.

2.6.3 MongoDB

Mongodb is used as the databases. Since mongo provides better search functionality and collectivity therefore mongodb is opted as the central database. MongoDB is one of many cross-platform document-oriented databases. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas, making the integration of data in certain types of applications easier and faster.

2.6.4 Elasticsearch

Elasticsearch is a search server based on Lucene. It provides a distributed, multitenant-capable full-text search engine with a RESTful web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. Elasticsearch is the second most popular enterprise search engine.

Chapter 3

Requirement Analysis

3.1 Platform Requirement :

3.1.1 Supportive Operating Systems :

The supported Operating Systems for client include:

- Windows xp onwards
- Linux any flavour.

Windows and Linux are two of the operating systems that will support comparative website. Since Linux is an open source operating system, This system which is will use in this project is developed on the Linux platform but is made compatible with windows too.The comparative website will be tested on both Linux and windows.

The supported Operating Systems for server include: The supported Operating Systems For server include Linux. Linux is used as server operating system. For web server we are using apache 2.0

3.2 Software Requirement :

The Software Requirements in this project include:

- Java
- Python

- Jsoup
- MongoDB
- Flask
- Internet Explorer, Mozilla Firefox, Google Chrome etc
- Elastic-search

In this project, the use of Jsp and servlet are used for creating the backbone structure of the comparative are python web technologies that are used for developing website modules such as inventory system, cart etc. Jsoup is a specialized tool developed by MIT written in java programming language for extraction of URLs in Real Time. Jsoup libraries are imported in the project for developing web crawler. Through Jsoup libraries, scraper can perform efficient crawling and thus increasing the performance of the comparative website. Elastic search is used in searching method. it is a search server based on lucene. Elastic search provides Exceptional functionality such as indexing and searching that makes it extremely important in the entire application.

Java language is the fundamental language being used in the development of the project. MongoDB is a Nosql which is used as a database for storing URLs. MongoDB reduces the complexity of maintaining relationships since in Nosql unlike RDBMS we don't need to maintain relationships. With the help of MongoDB, Map reduce techniques can be performed on the data Thus facilitating operations of data mining on the database.

3.3 Hardware Requirement :

3.3.1 Hardware Required For Project Development:

- 1 GB Ram.
- 40 GB Hard Disk Minimum.
- Intel Core i3-3xxx

Chapter 4

Project Design

4.1 Design Approach

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.2 Software Architectural Designs

Our system is follow the three tier architecture . First tier consist of GUI, Comparison tier and the Database.

1. GUI: The GUI(Graphical User Interface) in our project deals with the interface for the user where the user enters the name of the product he/she wants to search. The GUI provides a platform for the user to communicate with the database.

2. Comparison block: The comparison block is the block where the actual processing of our project is done. This block connects the GUI to the database i.e. It acts as a connector as well as communicator which connects the database and helps in transfer of data between the GUI and the database. It's main function is to index the products in our database and do the

comparison on the basis of the product price and other aspects.

3. Database: Database tier is the tier used for the storage of data. This tier contains all the data that is need for the processing of the whole project. The data in this tier is related to the product details such as the name, price and other details that are present on the website regarding the product.

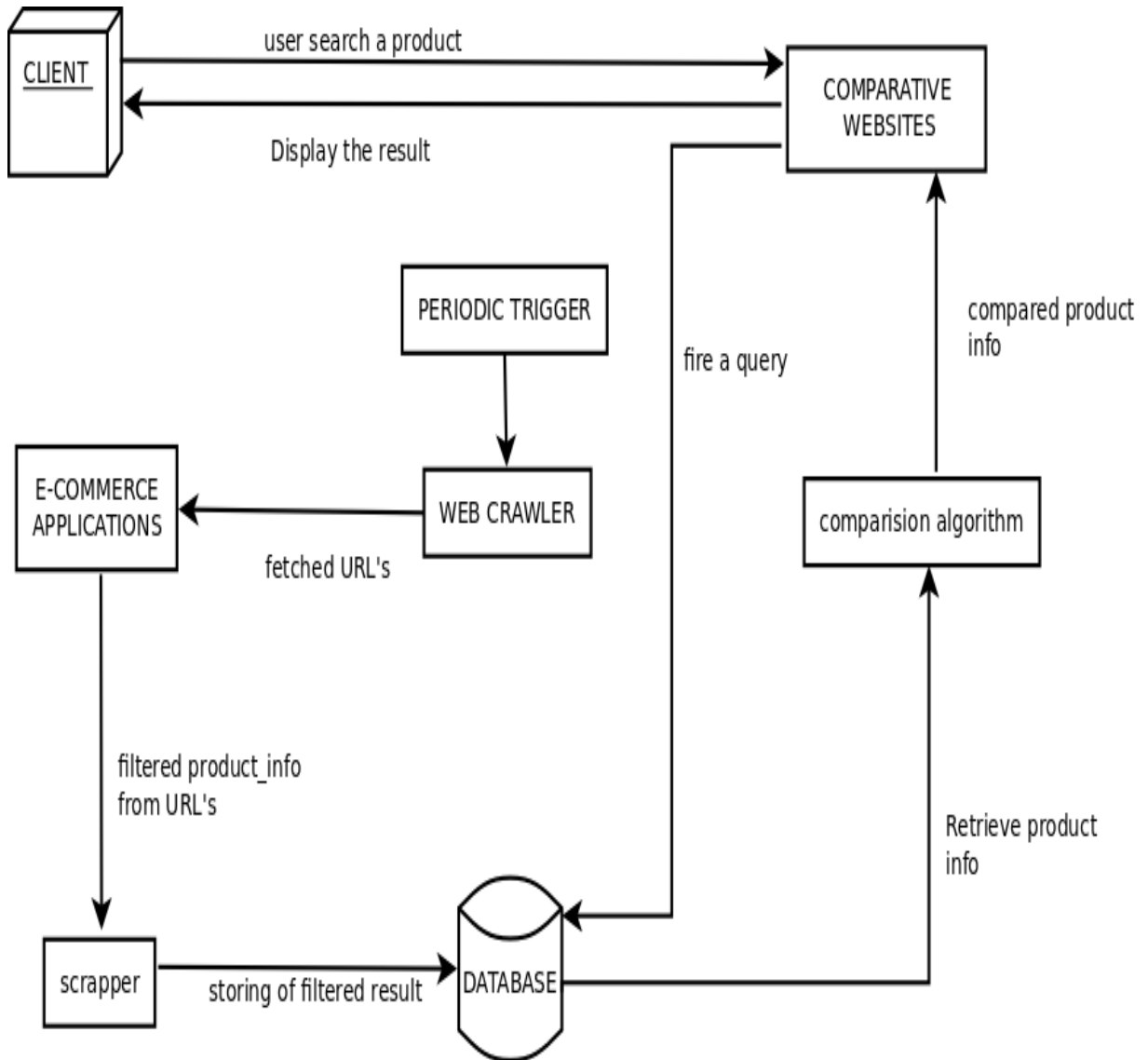


Figure 4.1: Software architecture Design

4.2.1 Front End Designs

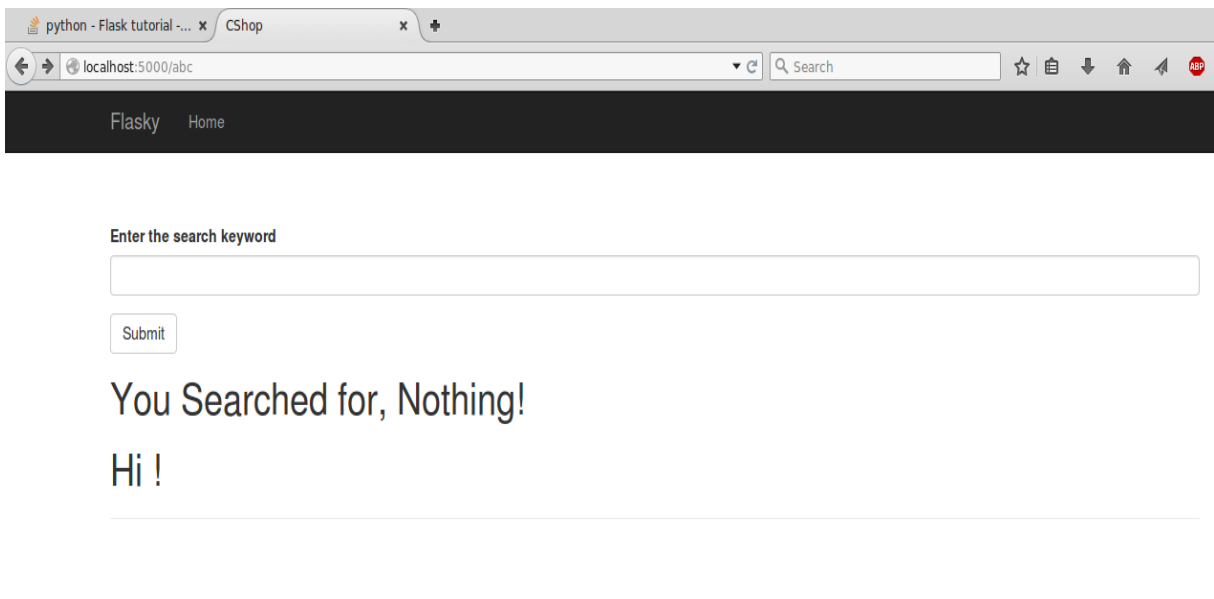


Figure 4.2: Front End Design

4.2.2 Component Diagram

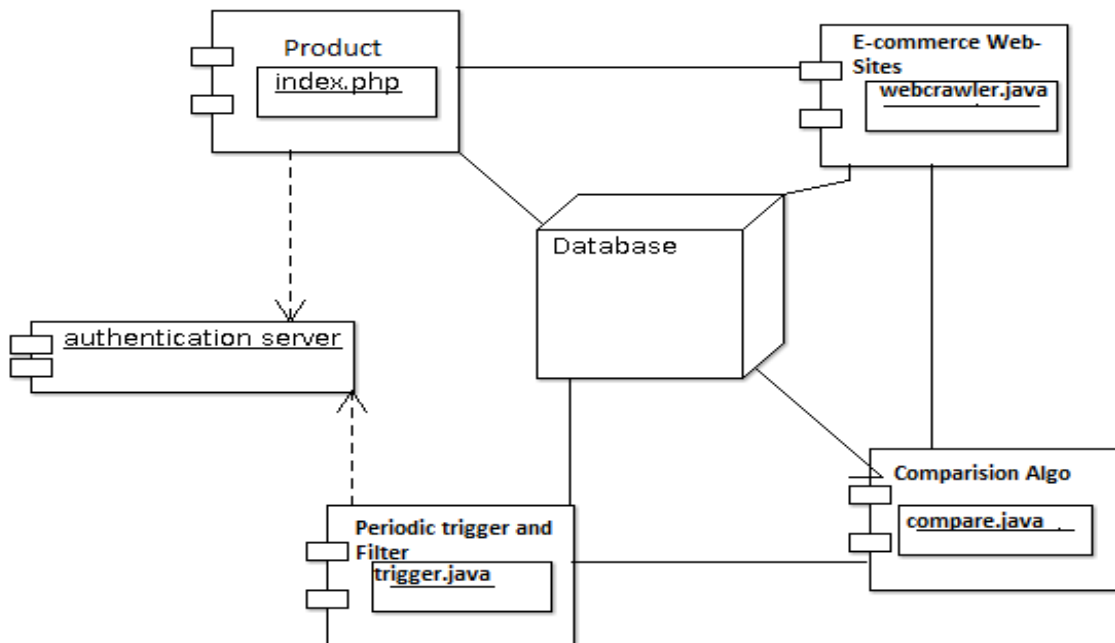


Figure 4.3: Component Diagram of Real Time Product Analysis

4.2.3 Deployment Diagram

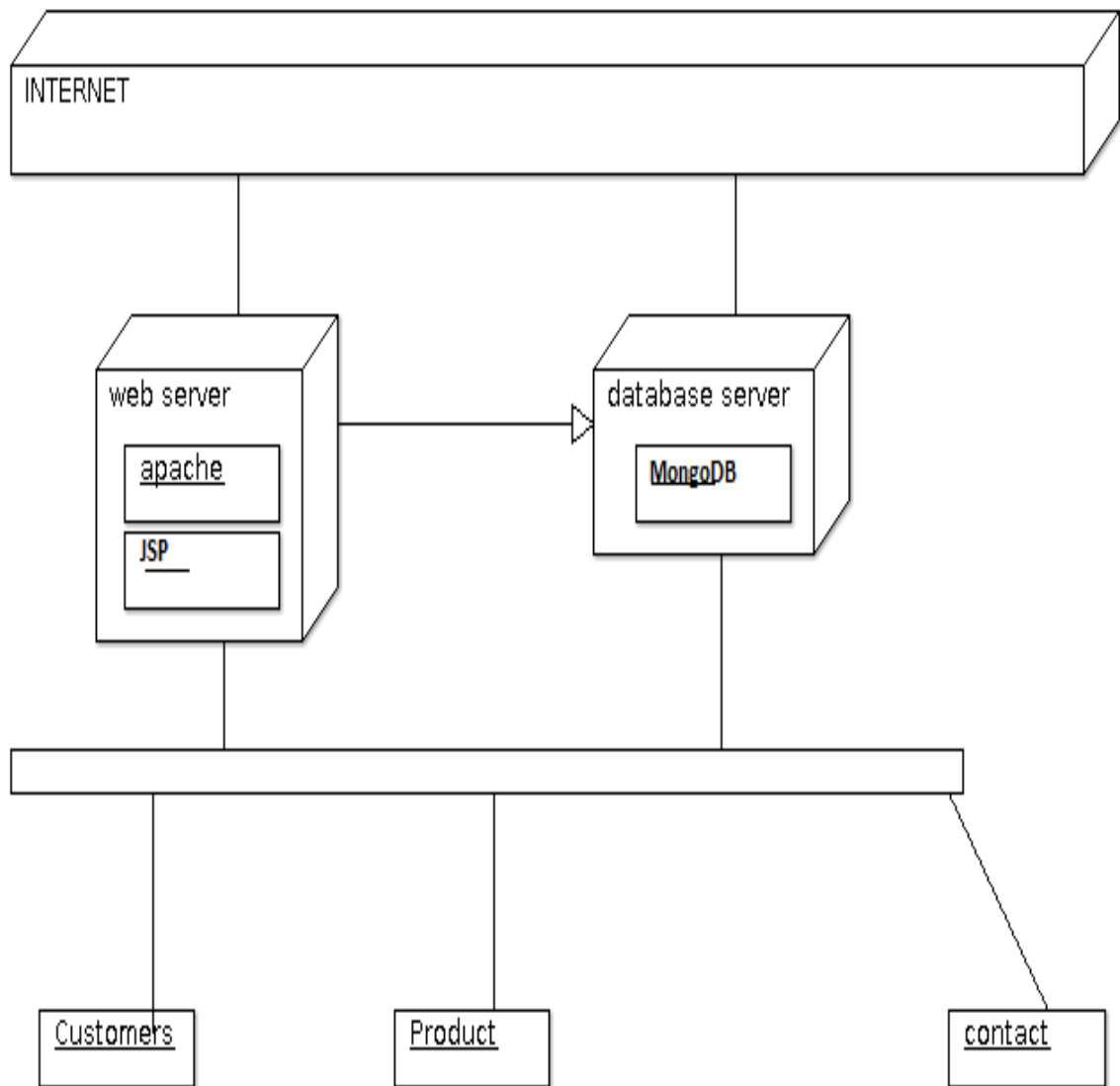


Figure 4.4: Deployment Diagram of Real Time Product Analysis system

4.3 Database Design

4.3.1 E-R Diagram

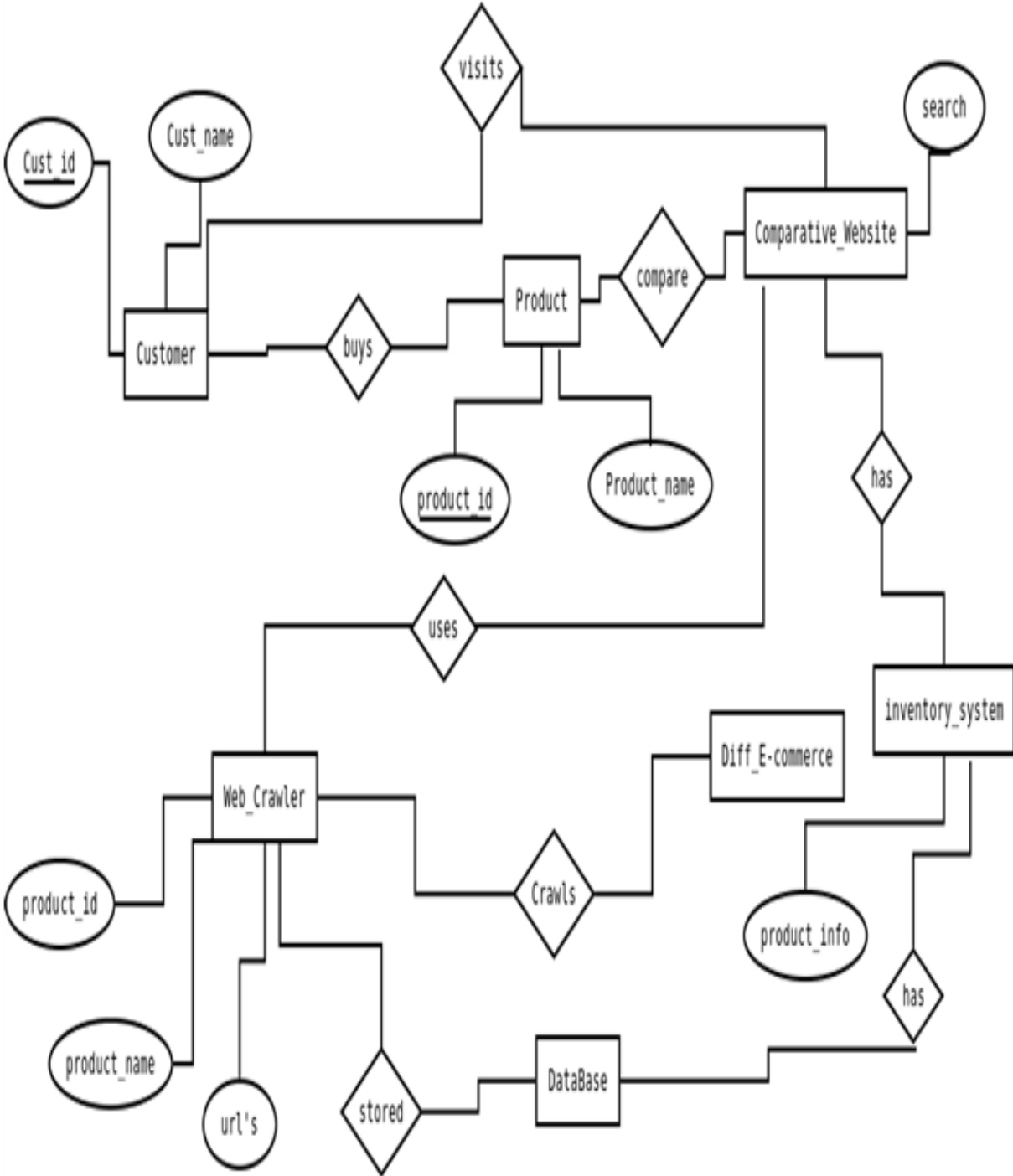


Figure 4.5: E-R Diagram of Real Time Product Analysis system

4.4 Work-flow Design

4.4.1 Flow Diagram

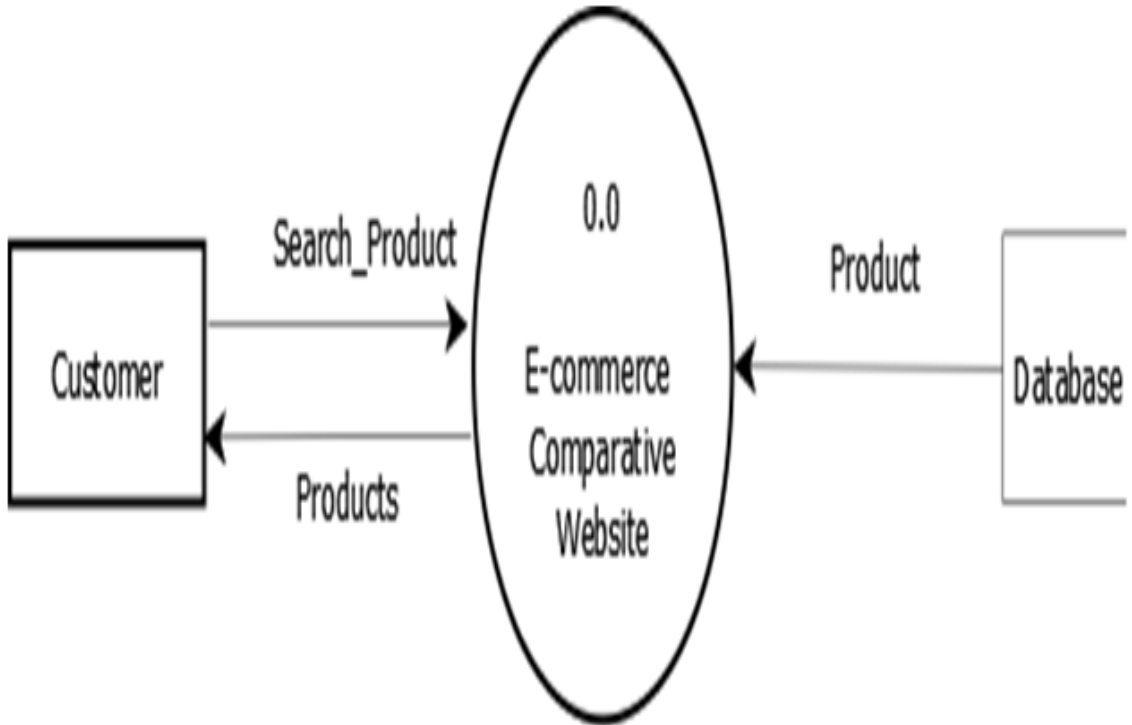


Figure 4.6: Level 0 DFD of Real Time Product Analysis system

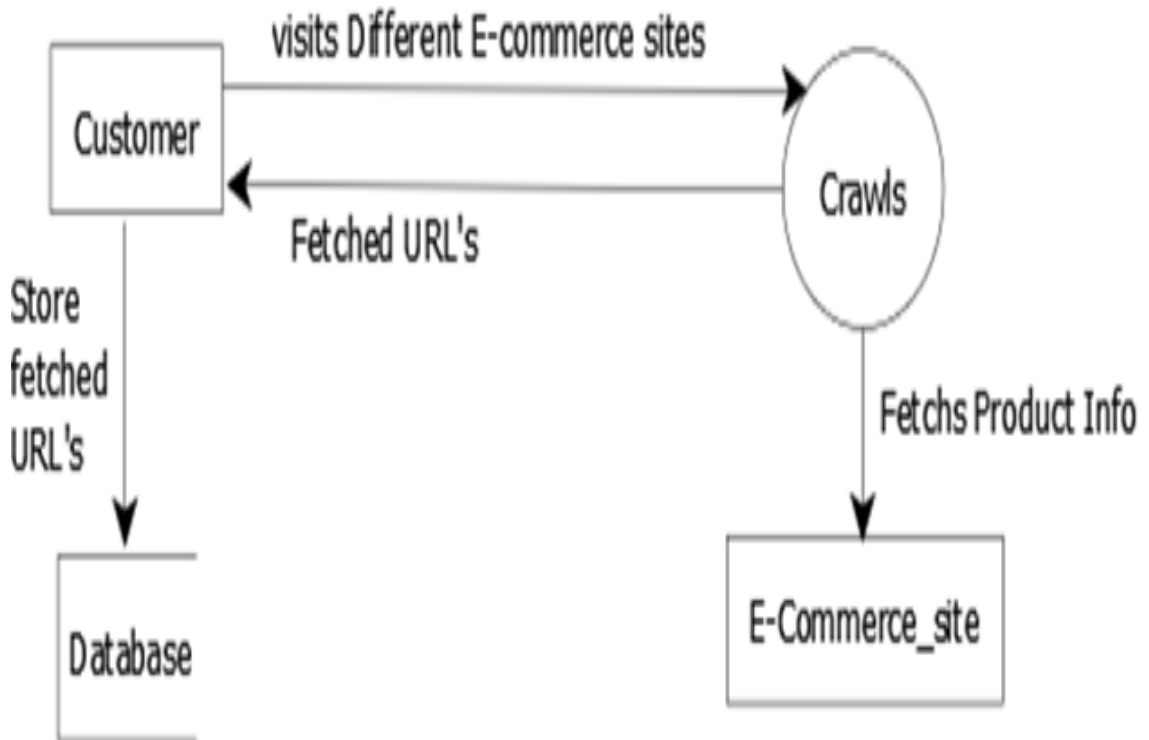


Figure 4.7: Level 1 DFD of Real Time Product Analysis system

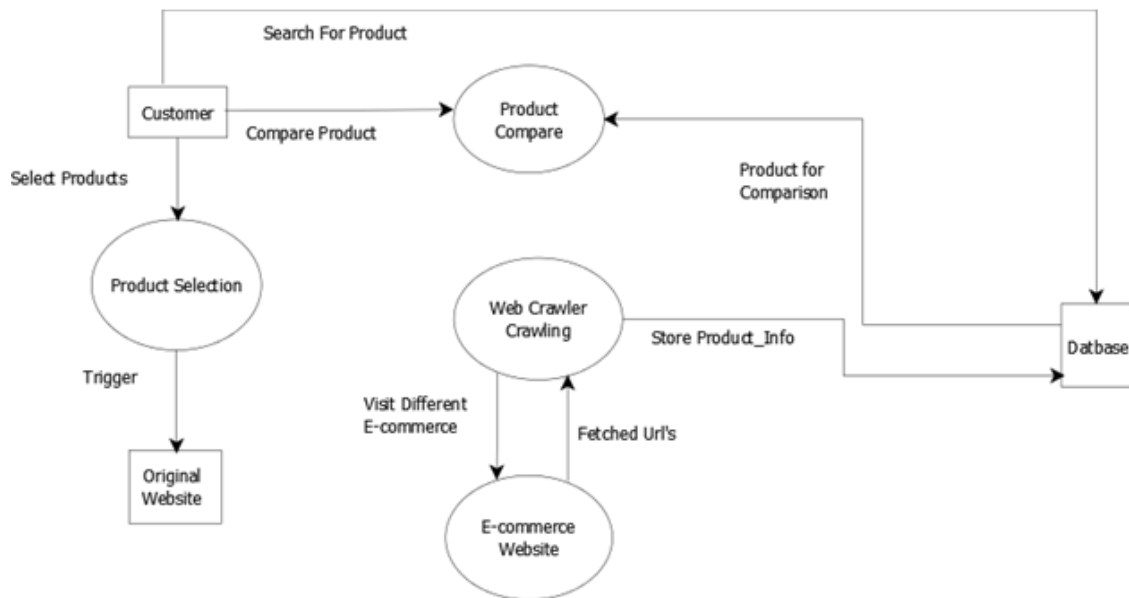


Figure 4.8: Level 2 DFD of Real Time Product Analysis system

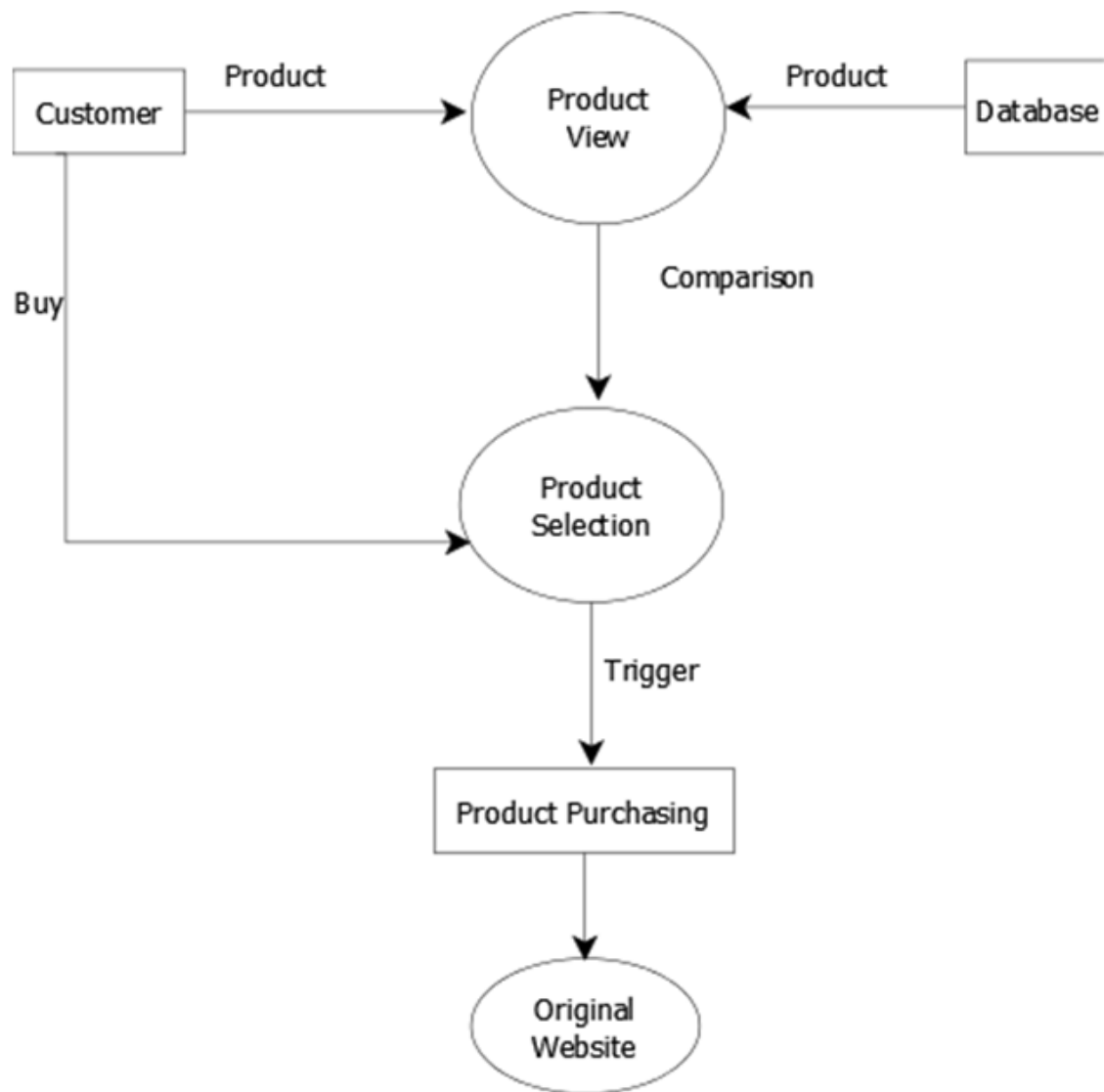


Figure 4.9: Level 3 DFD of Real Time Product Analysis system

Chapter 5

Implementation Details

5.1 Assumptions And Dependencies

5.1.1 Assumptions

The following Assumption was taken into consideration:

- The web crawler fetches the data from various E-commerce websites and stores in the local database. The crawler periodically fetches the data from different E-commerce websites and updates the local database if any updates are made by the E-commerce websites. Therefore it is assume that the crawler has to be very effective and fast in fetching the database. The local database is assumed to be scalable and robust so that it can store huge amount of data with time and maintain consistency of the data.
- The search algorithm has to very optimize in performing the search. As soon as the user queries the database the search algorithm is assumed to bring the required and accurate results. The user interface should be simple and clean that allows soothing effect to the user. The comparative algorithm is assumed to be very effective that allows comparison amongst different products.

5.1.2 Dependencies

The dependencies are as follows:

- For backend processing, Jsp and Servlet is being used. The backbone structure of the system is developed by making use of Jsp and Servlet. In creation of web crawler, Jsoup, java and MongoDB is used. Jsoup is a specialized tool developed by Mit which is written in java language For URL extraction. The extracted URLs are stored in MongoDB

database.

- Tomcat is the server which is put into use for handling server side request and response. Real Time product analysis using data mining is only a web agent that serves as a mediator. It depends on data extracted from different E-commerce websites.

5.2 Implementation Methodologies

Different Modules created are WebCrawler, inventory, storescripts, style, and cart. In WebCrawler module, the WebCrawler is created using Jsoup libraries, java and MongoDB. The WebCrawler module contains code for Fetching URLs and filtering the URLs. The filter URLs are then stored in the database. The WebCrawler module is then queried to store the URLs in the inventory.

5.2.1 Modular Description of Project

5.3 Detailed Analysis and Description of Project

Inventory Module: In this module, the admin can add, delete and update products. Store script module contains all the queries for the creation of Database in the MongoDB and the required queries to perform operations on the database.

Style Module: In style all the CSS coding is done and the front end is developed. In cart module, the customer can select the database and put it into cart. When the customer clicks on the buy button, then it is triggered to the original E-commerce website.

Web Crawler: The web crawler fetches the data from various E-commerce In this project, a web crawler/scrapper is used. The job of the web crawler is to periodically fetch data from different E-commerce websites. Since E-commerce websites hold huge amount of data, the WebCrawler has to be very efficient in extracting URLs. For this purpose, a specialized tool called Jsoup is being used. The Jsoup is a specialized tool developed by Mit which holds specialization in URLs extraction. Then we perform filtration of the URLs in order to get the right and useful URLs and removes the unuseful URLs.

URL Extraction: The URLs extracted from different E-commerce websites by the web crawler has to be store somewhere? For this purpose the use of Nosql MongoDB is being used MongoDB is a Nosql database which provides key features like map reduction, indexing, listing etc. MongoDB removes the complexity of maintaining relationships since no relationships have to be maintained in MongoDB. Now the extracted URLs are stored in MongoDB database.

User Search: Whenever user searches for the product in the search bar, the database is queried

in order to retrieve the accurate results. Now the user is presented with a particular product with different E-commerce websites. This allows the user to compare products based on prices. In this project, the web crawler is required to be very effective and efficient. For this purpose various factors must be taken into consideration such as time, address of the website etc. For creating the crawler, Jsoup libraries are imported which holds specialized functions in URL extraction. Then a function is written in java that filters the unuseful URLs. For deciding which of the URLs are useful or not, web crawler visits all the URLs and the URLs it finds as spam, it filters it. Since different websites have different naming standards therefore it will become difficult for the crawler to fetch and compare different URLs. For this purpose machine learning algorithm is used for URLs comparison. Also to track user behavior and search patterns, algorithms are used that keeps track on users and their behaviors and their searching patterns in order to provide user with better search experience.

Map Reduction Module: Map reduction techniques are performed on the data present in the database. Search optimization algorithms are used to optimize the searching in order to provide the user with better search results. The database is queried in order to get the results which the user searches. Then a comparative algorithm is used to provide the user with the best deal. The comparative algorithm compares a particular product from different websites on basis of prices and then recommends the user that which website is offering the product with the cheapest price. Thus, the comparative website is acting as agent providing the user the ability to compare products based on prices offered by different websites.

5.3.1 Usecase Report

Title:	Real Time Product Analysis Using Data Mining
Description:	Real Time Product Analysis Using Data Mining(Shopping Bot), provide the way to customer that allows the customer to view and compare prices of a particular product from different websites and purchase the product which he/she finds suitable for him/her. This tends to reduce Time and effort put by customer providing customer ease and satisfactory results
Primary Actor:	Customer
Preconditions:	Customer visit the Website
Post conditions:	Customer Search the Product
Main Success Scenario:	<ol style="list-style-type: none"> 1. Customer searches the desire Product through the comparative website and WebCrawler Fetches the products information from different E-Commerce website Periodically & stores that info in the database 2. Comparative Website retrieve requested information from the database and will display to the customer 3. Customer will Buy their desire product
Frequency of Use:	User can use any number of time
System Requirement:	Normal. No Specific Requirement
Trigger:	A WebCrawler, Which Fetches the products information from different E-Commerce website Periodically & stores that info in the database

Table 5.1: Usecase Report

5.4 Class Diagram

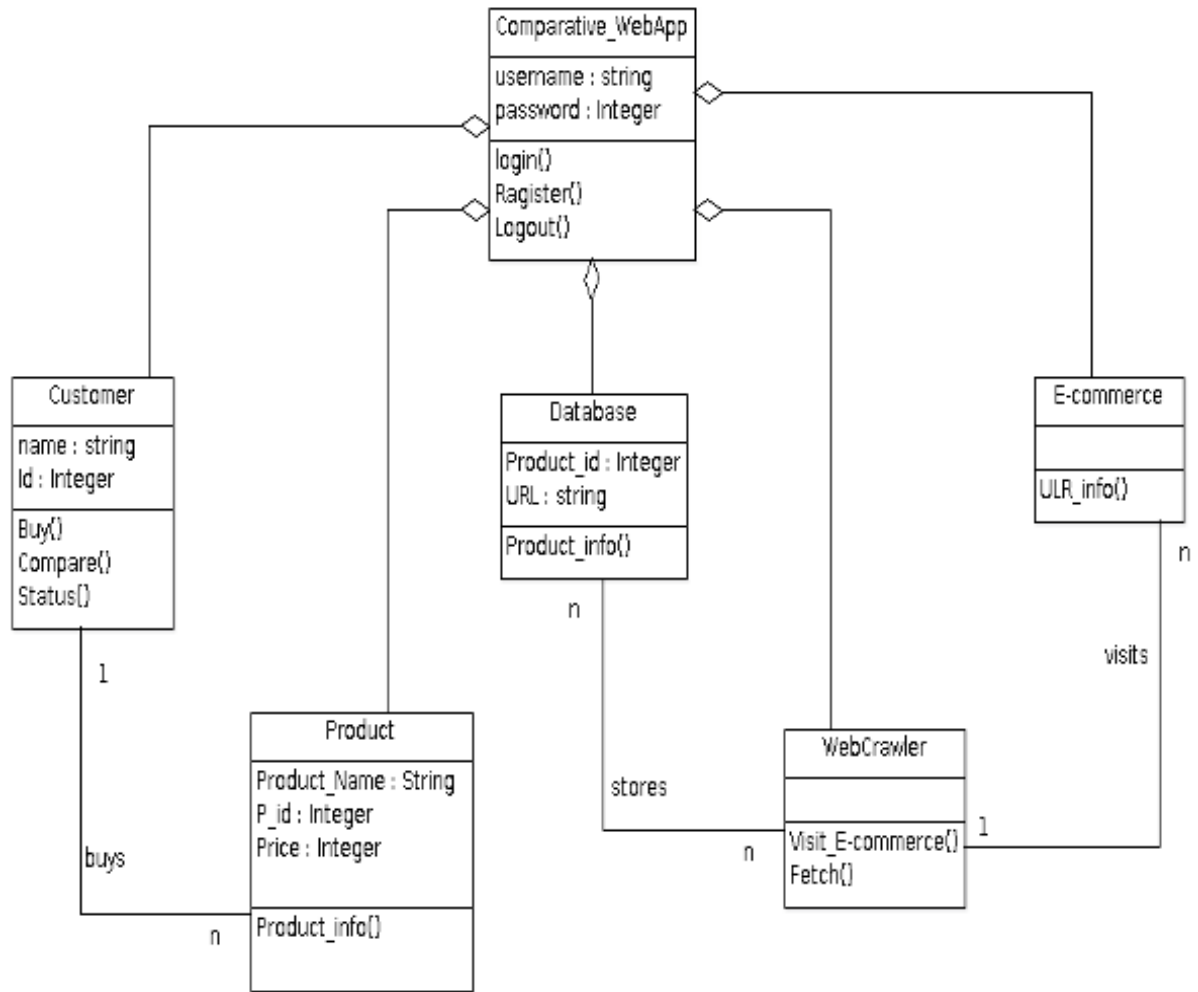


Figure 5.1: Class Diagram

5.4.1 Class Diagram Report

Title:	Real Time Product Analysis Using Data Mining
Description:	Real Time Product Analysis Using Data Mining(Shopping Bot), provide the way to customer that allows the customer to view and compare prices of a particular product from different websites and purchase the product which he/she finds suitable for him/her. This tends to reduce Time and effort put by customer providing customer ease and satisfactory results
Primary Actor:	Customer
Preconditions:	Customer visit the Website
Post conditions:	Customer Search the Product
Comparative Website :	<ol style="list-style-type: none"> 1. This Entity Website will Perform the Comparison among the customer desire product which is store in the database and fetched from the different E-commerce websites by WebCrawler. 2. After comparison it will display the result to customer. 3. This Entity also perform the customer registration task etc.
E-commerce websites:	It Includes the different E-commerce websites from where the comparative website fetches the product details for further operation.
WebCrawler:	This entity perform the fetching task i.e. it fetches product details from the product URL of different E-commerce sites.
Database:	Database is used for storing the product details fetched from the different E-commerce websites by the WebCrawler.

Table 5.2: Class Diagram Report

Chapter 6

Results and Discussion

6.1 Test cases and Result

We have tested our web application by considering following test cases:

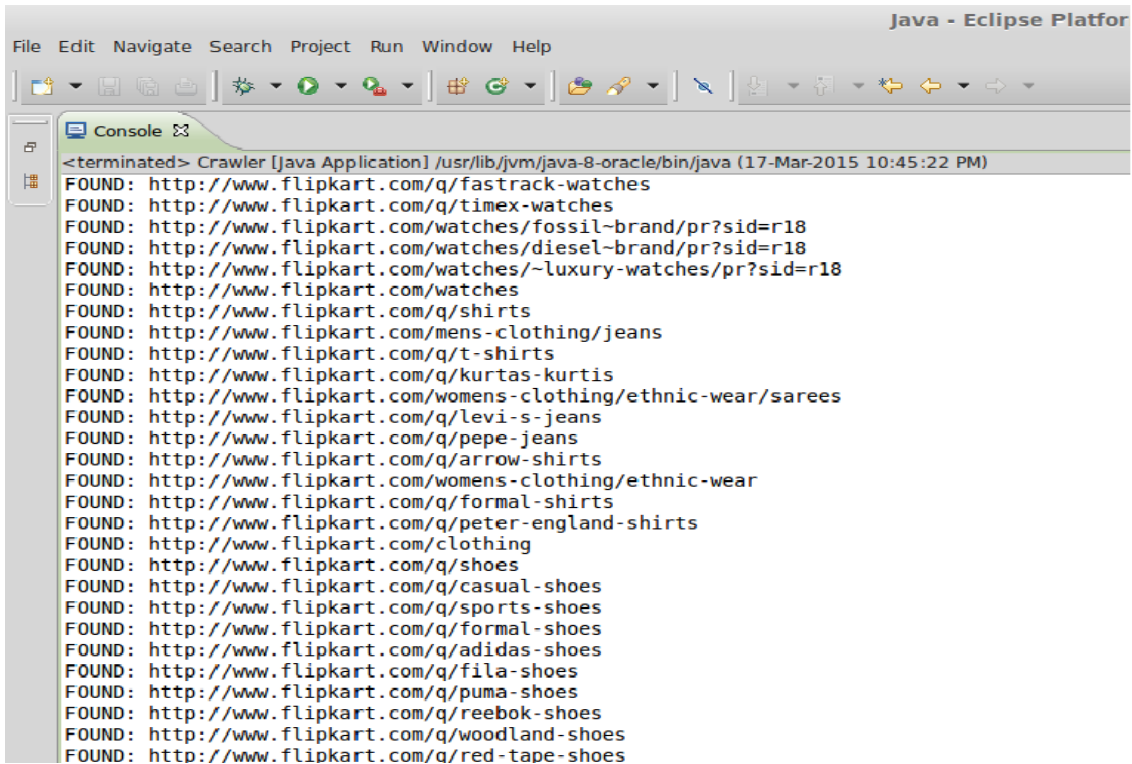
6.1.1 Unit Testing

We are comparing products of two E-commerce website. The webcrawler fetched Urls from these E-commerce websites. Example , Below picture shows the urls that we are fetching from different E-commerce websites.

From these urls , we scrape the details of the product.

Then these scrape details we are adding into the database i.e. MongoDB.

By making use of combination program which is implemented in python, we are performing indexing and also we are making the combination of the same products which is available on different E-commerce websites.

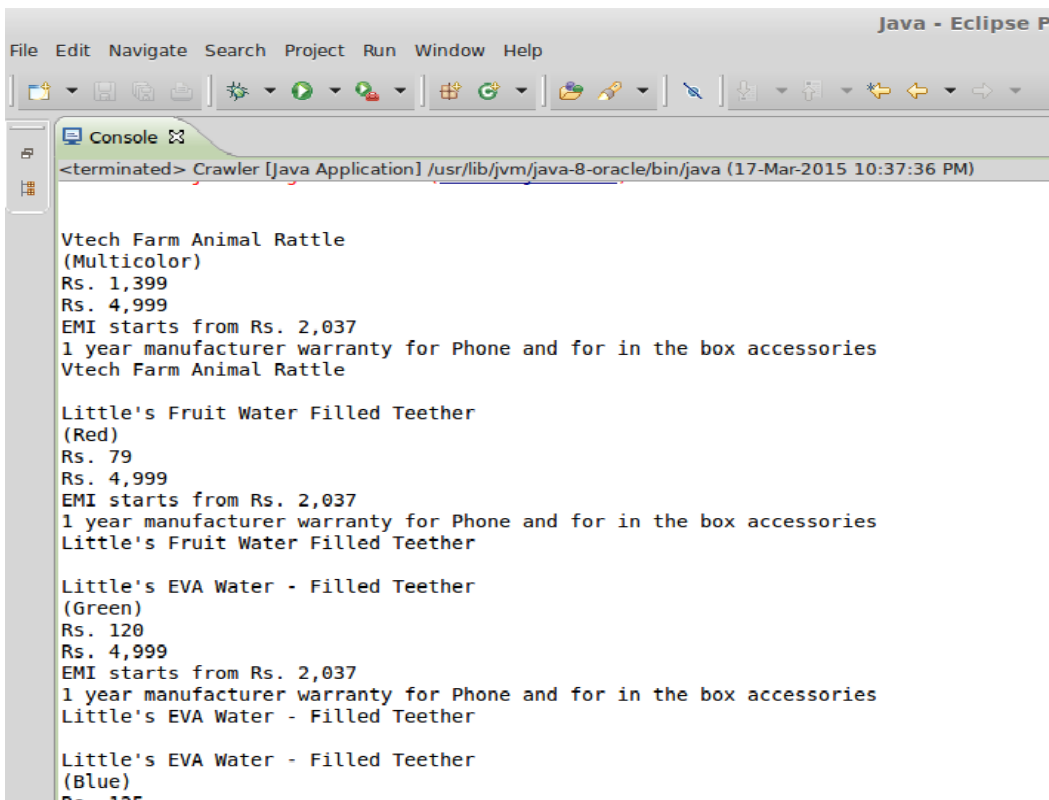


```

Java - Eclipse Platform
File Edit Navigate Search Project Run Window Help
<terminated> Crawler [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (17-Mar-2015 10:45:22 PM)
FOUND: http://www.flipkart.com/q/fastrack-watches
FOUND: http://www.flipkart.com/q/timex-watches
FOUND: http://www.flipkart.com/watches/fossil-brand/pr?sid=r18
FOUND: http://www.flipkart.com/watches/diesel-brand/pr?sid=r18
FOUND: http://www.flipkart.com/watches/~luxury-watches/pr?sid=r18
FOUND: http://www.flipkart.com/watches
FOUND: http://www.flipkart.com/q/shirts
FOUND: http://www.flipkart.com/mens-clothing/jeans
FOUND: http://www.flipkart.com/q/t-shirts
FOUND: http://www.flipkart.com/q/kurtas-kurtis
FOUND: http://www.flipkart.com/womens-clothing/ethnic-wear/sarees
FOUND: http://www.flipkart.com/q/levi-s-jeans
FOUND: http://www.flipkart.com/q/pepe-jeans
FOUND: http://www.flipkart.com/q/arrow-shirts
FOUND: http://www.flipkart.com/womens-clothing/ethnic-wear
FOUND: http://www.flipkart.com/q/formal-shirts
FOUND: http://www.flipkart.com/q/peter-england-shirts
FOUND: http://www.flipkart.com/clothing
FOUND: http://www.flipkart.com/q/shoes
FOUND: http://www.flipkart.com/q/casual-shoes
FOUND: http://www.flipkart.com/q/sports-shoes
FOUND: http://www.flipkart.com/q/formal-shoes
FOUND: http://www.flipkart.com/q/adidas-shoes
FOUND: http://www.flipkart.com/q/fila-shoes
FOUND: http://www.flipkart.com/q/puma-shoes
FOUND: http://www.flipkart.com/q/reebok-shoes
FOUND: http://www.flipkart.com/q/woodland-shoes
FOUND: http://www.flipkart.com/q/red-tape-shoes

```

Figure 6.1: Url Extraction



```

Java - Eclipse P
File Edit Navigate Search Project Run Window Help
<terminated> Crawler [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (17-Mar-2015 10:37:36 PM)

Vtech Farm Animal Rattle
(Multicolor)
Rs. 1,399
Rs. 4,999
EMI starts from Rs. 2,037
1 year manufacturer warranty for Phone and for in the box accessories
Vtech Farm Animal Rattle

Little's Fruit Water Filled Teether
(Red)
Rs. 79
Rs. 4,999
EMI starts from Rs. 2,037
1 year manufacturer warranty for Phone and for in the box accessories
Little's Fruit Water Filled Teether

Little's EVA Water - Filled Teether
(Green)
Rs. 120
Rs. 4,999
EMI starts from Rs. 2,037
1 year manufacturer warranty for Phone and for in the box accessories
Little's EVA Water - Filled Teether

Little's EVA Water - Filled Teether
(Blue)
Rs. 125

```

Figure 6.2: Product Details

One Simple Elegant query is used to compare different products from the database:

```

Terminal
type "it" for more
db.flipkart.find().pretty();
  "_id" : ObjectId("54fa886f44ae2090b6bc9c68"),
  "name" : "Soyer SI102 Champion Series Steam Iron",
  "subtitle" : "(Blue)",
  "sell price" : "Rs. 500",
  "exchange sell price" : null,
  "emi" : null,
  "warranty" : "2year+200 Days",
  "keywords" : [
    "Soyer",
    "SI102",
    "Champion",
    "Series",
    "Steam",
    "Iron"
  ]

  "_id" : ObjectId("54fa88e244ae677fd501d9e3"),
  "name" : "PremiumAV Anti-virus 1 PC 1 Year (Activation Card)",
  "subtitle" : null,
  "sell price" : "Rs. 69",
  "exchange sell price" : null,
  "emi" : null,
  "warranty" : null,
  "keywords" : [
    "PremiumAV",
    "Anti-virus",
    "1",
    "PC",
    "1",
    "Year",
    "(Activation",
    "Card)"
  ]

```

Figure 6.3: Data Storage

6.1.2 Functional Testing

We have tested our web application on the server by integrating all the units of our project. In this testing we focus on the output is as per requirement or not. which is as follow:

In our comparative websites there is a Home page of our websites.


```

{
    "productName"=Samsung galaxy s6
    "vandar"=flipkart
    "sellingPrice"=Rs 14,500
    "EMI"=Rs 704
    "Devilivery"=+ Rs100
}
{
    "productName"=Samsung galaxy s6
    "sellerName"=ebay
    "sellingPrice"=Rs 14,140
    "EMI"=Rs 700
    "Devilivery"=+ Rs100
}

```

Figure 6.4: Indexing Result

for checking the outputs and functionality first we typed the product name in the search bar and press the enter button or click on submit button.

These search result will display the products details which is available on different E-commerce websites in a manner of best deal. Product detail includes the product name , price, EMI details , Exchange price, Vendor name, warranty etc. Now here user has the freedom to choose the product from any vendor i.e any E-commerce websites as per his or her wish and as per their Budget to buy their desire product.

Output:
Samsung galaxy s6 from flipkart
Rs. 14,500
Selling Price
EMI starts from Rs. 704 .
+ Rs 100 Delivery

Samung galaxy s6 from ebay
Rs. 14,140
Selling Price
EMI starts from Rs. 700.
+ Rs 150 Delivery

Figure 6.5: product output

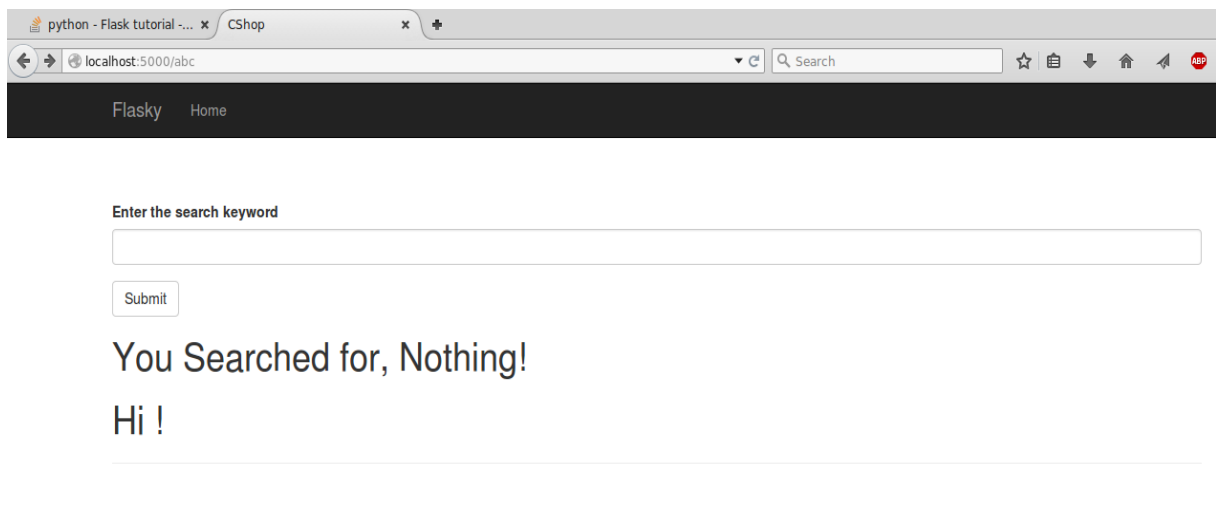


Figure 6.6: Home Page

When user click on the product name or image our comparative websites will redirect the user to vendor website or E-commerce websites.

Enter the search keyword

You Searched for, samsung galaxy s6!

Hi !



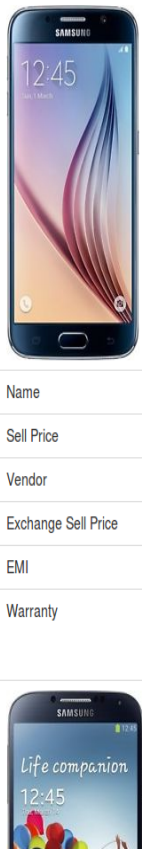
Figure 6.7: Searched Product Result



Name	Samsung Galaxy S6 (Black Sapphire, with 32 GB)
Sell Price	Rs. 49,900
Vendor	Flipkart
Exchange Sell Price	none
EMI	EMI starts from Rs. 2,420
Warranty	1 year manufacturer warranty for Phone and 6 months warranty for in the box accessories

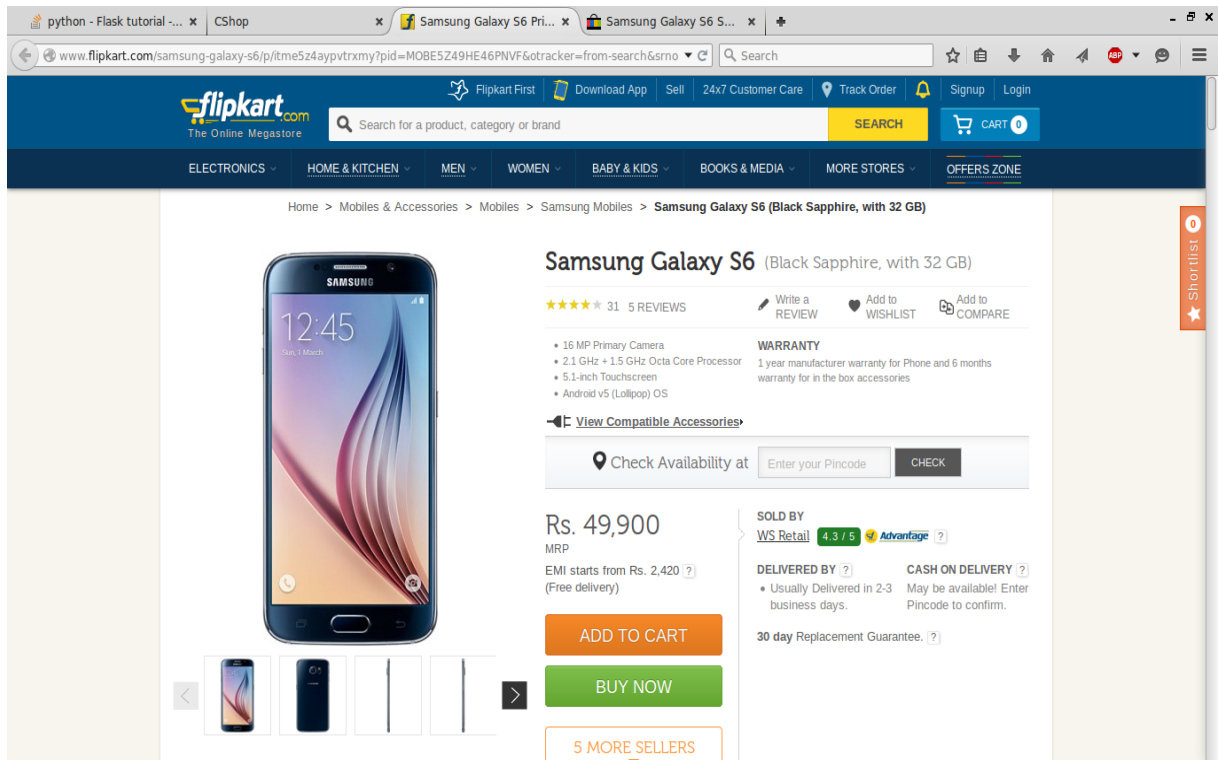


Figure 6.8: Searched Product Result



Name	Samsung Galaxy S6 SM-G920I Dispatch in 48hrs from Launch Date i.e 10th April'15
Sell Price	Rs. 49,900.00
Vendor	Ebay
Exchange Sell Price	none
EMI	none
Warranty	warranty 1 year Manufacturer Warranty

Figure 6.9: Searched Product Result



Home > Mobiles & Accessories > Mobiles > Samsung Mobiles > Samsung Galaxy S6 (Black Sapphire, with 32 GB)

Samsung Galaxy S6 (Black Sapphire, with 32 GB)

★★★★★ 31 5 REVIEWS

- 16 MP Primary Camera
- 2.1 GHz + 1.5 GHz Octa Core Processor
- 5.1-inch Touchscreen
- Android v5 (Lollipop) OS

WARRANTY
1 year manufacturer warranty for Phone and 6 months warranty for in the box accessories

Rs. 49,900
MRP
EMI starts from Rs. 2,420 (Free delivery)

ADD TO CART

BUY NOW

5 MORE SELLERS

Figure 6.10: vendor sites

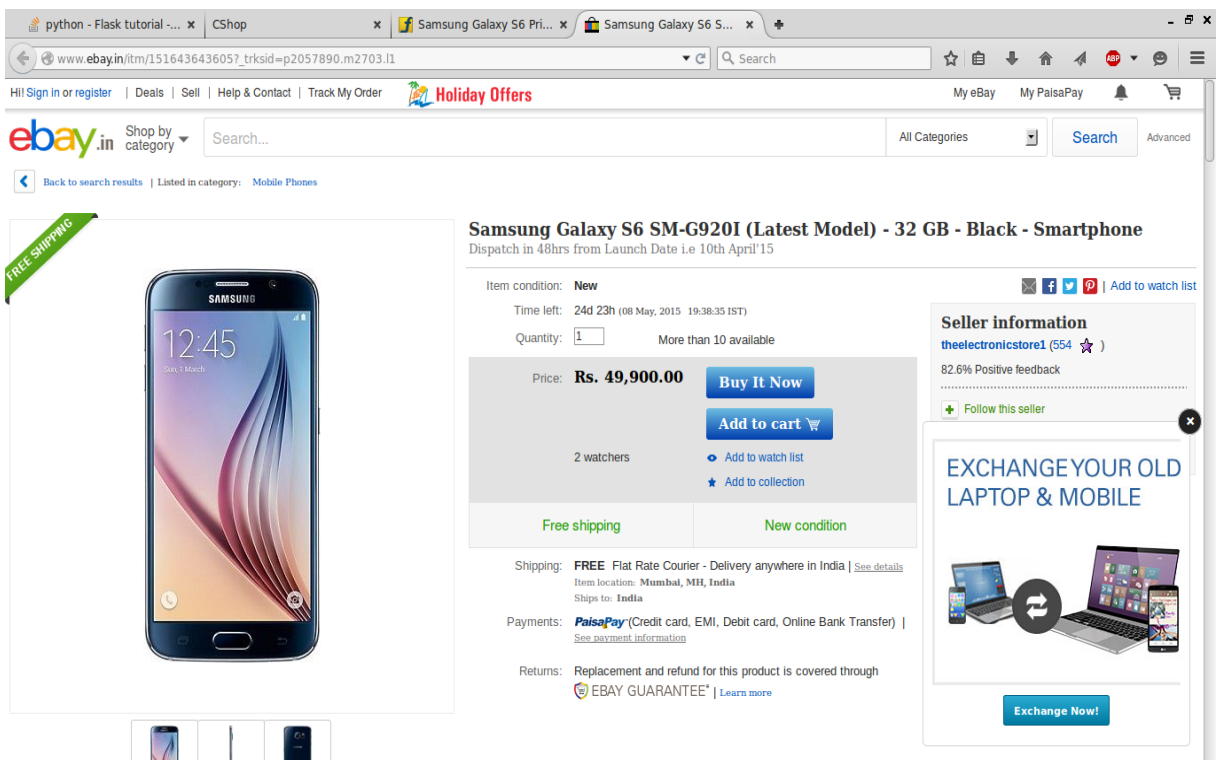


Figure 6.11: vandor sites

Chapter 7

Project Time Line

7.1 Project Time Line Matrix

	⊙	Name	Duration	Start	Finish	Predecessors
2		1(a) Requirements Gathering	8 days	1/1/15 8:00 AM	12/1/15 5:00 PM	
3	☐	1(b) Confirm Requirements	1 day	13/1/15 8:00 ...	13/1/15 5:00 PM	
5	☐	2(a) Frontend-User Interface	7 days	14/1/15 8:00 ...	22/1/15 5:00 PM	
6	☐	2(b) Backend Database Designing	10 days	23/1/15 8:00 ...	5/2/15 5:00 PM	
8	☐	3(a) Frontend Coding	12 days	6/2/15 8:00 AM	23/2/15 5:00 PM	
9	☐	3(b) Database Creation	10 days	24/2/15 8:00 ...	9/3/15 5:00 PM	
10	☐	3(c) Coding for screen,tables	8 days	10/3/15 8:00 ...	19/3/15 5:00 PM	
11	☐	3(d) Creation of test case	9 days	20/3/15 8:00 ...	1/4/15 5:00 PM	
13	☐	4(a) Unit Test	6 days	2/4/15 8:00 AM	9/4/15 5:00 PM	
14	☐	4(b) System Test	7 days	10/4/15 8:00 ...	20/4/15 5:00 PM	
15	☐	4(c) Alpha and Beta Test	6 days	21/4/15 8:00 ...	28/4/15 5:00 PM	
17	☐	5(a) Deployment	2 days	29/4/15 8:00 ...	30/4/15 5:00 PM	

Figure 7.1: Time Line Matrix

7.2 Project Time Line Chart

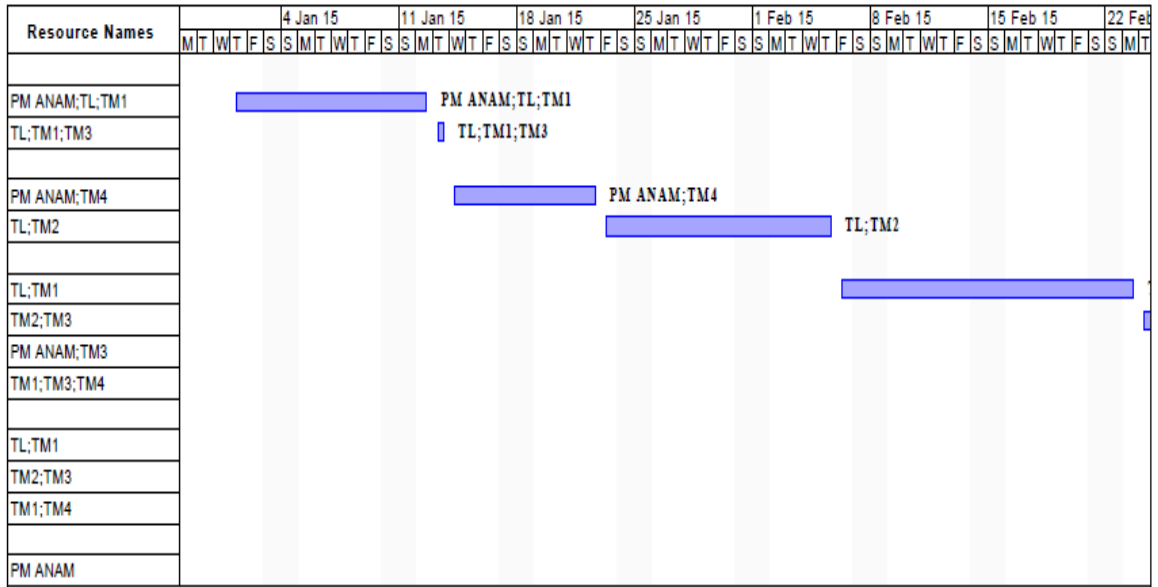


Figure 7.2: Time Line Chart

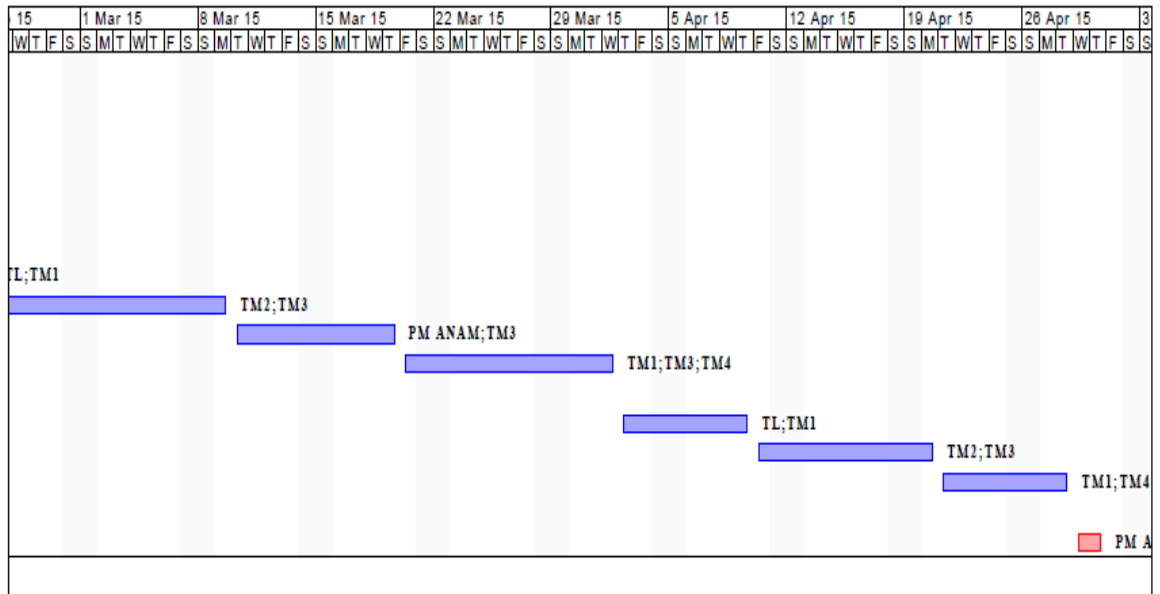


Figure 7.3: Time Line Chart

Chapter 8

Task Distribution

8.1 Distribution of Workload

8.1.1 Scheduled Working Activities

Activity	Time Period	Comment
Requirement Gathering	08 Days	Requirement gathering has took placed through searching on internet and taking the ideas, sharing the views among group members.
Planning	04 Days	Planing has done by Reviewing of literature of IEEE papers and by taking the walkthrough.
Design	04 Days	Designing has done by creating UML diagram, By creating Charts,
Implementation	90 Days	Implementation has done First creating the backend and then front end module by module.
Testing	10 Days	Testing has done by perfoming unit testing, alpha & Beta Testing, integrated testing and system testing.
Deployment	05 Days	Deployment has done by installing project on the server.

Table 8.1: Scheduled Working Activities

8.1.2 Members activities or task

Member	Activity	Time Period	Start Date	End Date	Comment
M1, M2, M3, M4	Requirement Gathering	4 Days	01/01/15	04/01/15	M1 and M2 has performed the searching for project requirement on the internet by reviewing the related literature and by analysing the related project which is already available in the market. Regularly inform to the other member of team.
M1, M2, M3, M4	Analysing of the requirement	3 Days	05/01/15	07/01/15	M1, M2, M3, M4 done the requirement analysing of project by sharing the ideas, and by discussing on related information which is gather by the M1, And M2. M3 and M4 has created the list of requirement after every meeting
M1, M2, M3, M4	Finalysing the requirement	1 Day	08/01/15	08/01/15	Whole team finalize the requirement. M1 and M4 has created a list of finalise requirement.
M1, M2, M3, M4	Planning	4 Days	09/01/15	12/01/15	Planning has done by walkthrough and by analysing the available product. M2 and M3 creates a list of funtion which will be implement in the project. Each and every module were discuss in every group meeting and M1 and M2 creates a blue print for project .

M1, M2	Front End design	4 Days	13/01/15	16/01/15	M1 and M2 creates the UML diagram for front end of the system and data flow diagrams and informed to the whole team regularly.
M3, M4	Back End design	4 Days	13/01/15	16/01/15	M3 and M4 creates the UML diagram for back end of the system and data flow diagrams and informed to the whole team regularly.
M3, M4	Installation of tools and technology for front end	1 Days	17/01/15	17/01/15	M3 and M4 installed the all the require tools which is use for front end design.
M1, M2	Installation of tools and technology for back end	2 Days	17/01/15	19/01/15	M3 and M4 installed all the require tools which is use for back end design.
M3, M4	Implementation of GUI	4 Days	20/01/15	24/01/15	M3 and M4 creates the GUI of the project and informed to other member.
M1	Implementation of Webcrawler	10 Days	20/01/15	10/02/15	M1 implemented the webcrawler and discuss on it with other team member
M2	Implementation of Scraper	10 Days	20/01/15	02/02/15	M2 implemented the scraper and discuss on it with other team member
M1, M2	webcrawler and scraper connectivity with GUI	4 Days	02/02/15	06/02/15	M1 and M2 makes the connection among scraper and webcrawler. M1 and M2 Explain the codes to the other member of team.

M3, M4	webcrawler and scraper connectivity with GUI	8 Days	07/02/15	14/02/15	M3 and M4 makes the conection with GUI and discuss on it with other team member regularly.
M1, M2	Database connectivity	4 Days	06/02/15	10/02/15	M1 and M2 created the database connectivity with the scraper and webcrawler .
M3, M4	GUI Connectivity	4 Days	11/02/15	14/02/15	M3 and M4 created the connectivity GUI with database
M1, M2	Implementation of Comparison algorithm	10 Days	15/02/15	25/02/15	M1 and M2 implemented the comparison program for comparison of product which will be fetched from the different E-commerce websites and regularly updates to the other member of team.
M3, M4	Data gathering into database	2 Days	15/02/15	17/02/15	M3 and M4 gather the data of products from different E-commerce websites by crawling & scraping with the webcrawler and scraper and made the collection into the database.
M1, M2	Connectivity of Comparison program	8 Days	26/02/15	04/03/15	M1 and M2 created the connectivity of comparison program with scraper, crawler, and Database. They explain the code to other member of team.
M3	Indexing	8 Days	26/02/15	04/03/15	M3 implements indexing programs in the python. And Connect it with the GUI. He explain the code to other member of team.

M4	Connectivity of indexing with database	2 Days	05/03/15	07/03/15	M4 makes the indexing program connectivity with database. And informed to the other member of team.
M3,M4	Implementation of searching algorithm	5 Days	07/03/15	12/03/15	M3 and M4 implemented searching algorithm and regularly discussed this implementation with the other member of team.
M1,M2	Searching program connectivity	2 Days	12/03/15	14/03/15	M1 and M2 makes the connectivity of searching algorithm program with GUI.
M1, M2, M3, M4	Integration of all modules	10 Days	16/03/15	25/03/15	M1, M2,M3 and M4 integrated all the module. Implemented whole system properly.
M1,M2	Unit testing	2 Days	26/03/15	28/03/15	M1 and M2 performed the unit testing and noted down results and discuss with other member of team.
M3, M4	Functional testing	5 Days	26/03/15	30/03/15	M3 and M4 performed the functional testing and noted down results and discuss the result of testing with other member.
M1, M2, M3, M4	Deployment		30/03/2015		

Table 8.2: Member Activities and Task

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

The Real Time product Analysis using Data mining is a price comparison engine that aims to facilitate the buyers to compare products from different E-commerce websites and purchase the product at the cheapest rate with best price. This way, the buyer has more power in his/her hands and can take better decision on different products at different price. Thus this project saves buyers efforts, time and money and also avoids user to physically visit each and everywhere E-commerce website. The beauty of this project is that it can be customize for a specific business segment and also can be used. It also helps the different E-commerce applications to boost their business by providing them a platform to compete and do business in a more reasonable manner. By involvement of data intelligence we can also comment and predict about the services and quality provided by the different E-Commerce. So that user can be help to choose a better deal.

9.2 Future Scope

In the future, the product aims to include several features like suggesting the user about the expected future changes in prices on products on the basis of business analytic that can predict the situation and demand of the products in the market. In the future we can make the add-on of this project and include it in the browser, when the customer visits any E-commerce site for any product the add-on will suggest the best deal.

References

- [1] Stigler, G. *The Economics of Information*, *Journal of Political Economy*, Vol. 69, pp.213-225, Iss. Jan-Feb, (1961).
- [2] Rehman S.U *Smart agent for automated E-commerce UK IEEE pp.124-128, 7-10 7-10 Nov.2011*, School of information System, Computer. Math, Brunel Univ, Uxbridge.
- [3] Serenko and Hayes J *Investigating the functionality and performance of online shopping bots for E-commerce, Electronic Business 8(1): pp.1-15, (2009)*.
- [4] Seddin K, Serenko and Hayes J , *Online shopping bots for E-commerce, Int. J. Electronic Business,pp. 556-589, (2007)*.
- [5] Bhavik Pathak. 'A Survey of the Comparison Shopping Agent-Based Decision Support System' *Journal of Electronic Commerce Research, VOL 11, NO 3, 2010*.
- [6] Henrique Lopes Cardoso, Max Schaefer, Eugenio Oliveira. 'A Multi-Agent System for Electronic Commerce including Adaptive Strategic Behaviors' 7-10 Nov.2011.
- [7] Namoo Kang*, Sangyong Han 'Agent-based e-marketplace system for more fair and efficient transaction', *N. Kang, S. Han / Decision Support Systems pp 157-165, (2002)*.
- [8] Robert Bo Doorenbos, Oren Etzioni, and Daniel So Weld, *A Scalable Comparison-Shopping Agent for the World-Wide Web* Department of Computer Science and Engineering University of Washington Seattle, WA 98195.
- [9] Clark, D *Shopbots become agents for business change, IEEE Computer, Vol. 33, Issue 2', (2000)*.
- [10] <http://www.postonline.co.uk/post/analysis/2319717/the-rise-of-price-comparison-sites-in-south-east-asia>
- [11] <http://e27.co/former-lazada-thailand-ceo-now-working-on-financial-products-comparison-site/>
- [12] <http://www.consumerfutures.org.uk/>

[13] <http://www.snapdeal.com/product/htc-desire-526-g/681509924248>

Chapter 10

Appendix I

10.1 What is Elasticsearch?

Elasticsearch is a tool for querying written words. It can perform some other nifty tasks, but at its core it made for wading through text, returning text similar to a given query and/or statistical analysis of a corpus of text.

More specifically, elasticsearch is a standalone database server, written in Java, that takes data in and stores it in a sophisticated format optimized for language based searches. Working with it is convenient as its main protocol is implemented with HTTP/JSON. Elasticsearch is also easily scalable, supporting clustering and leader election out of the box.

10.1.1 What problems does Elasticsearch solve well?

There are myriad cases in which elasticsearch is useful. Some use cases more clearly call for it than others. Listed below are some tasks which for which elasticsearch is particularly well suited.

- Searching a large number of product descriptions for the best match for a specific phrase(say *“chef’s knife”*) and returning the best results
- Auto-completing a search box based on partially typed words based on previously issued searches while accounting for mis-spellings
- Storing a large quantity of semi-structured (JSON) data in a distributed fashion, with a specified level of redundancy across a cluster of machines

10.2 Flask

Flask is a micro web application framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed. Examples of applications that make use of the Flask framework are Pinterest, LinkedIn, as well as the community web page for Flask itself.

Flask is called a microframework because it does not presume or force a developer to use a particular tool or library. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions, that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

10.2.1 Features of Flask

- Contains development server and debugger
- Integrated support for unit testing
- RESTful request dispatching
- Uses Jinja2 templating
- Support for secure cookies (client side sessions)
- 100% WSGI 1.0 compliant
- Unicode-based
- Extensive documentation
- Google App Engine Compatibility
- Extensions available to enhance features desired.

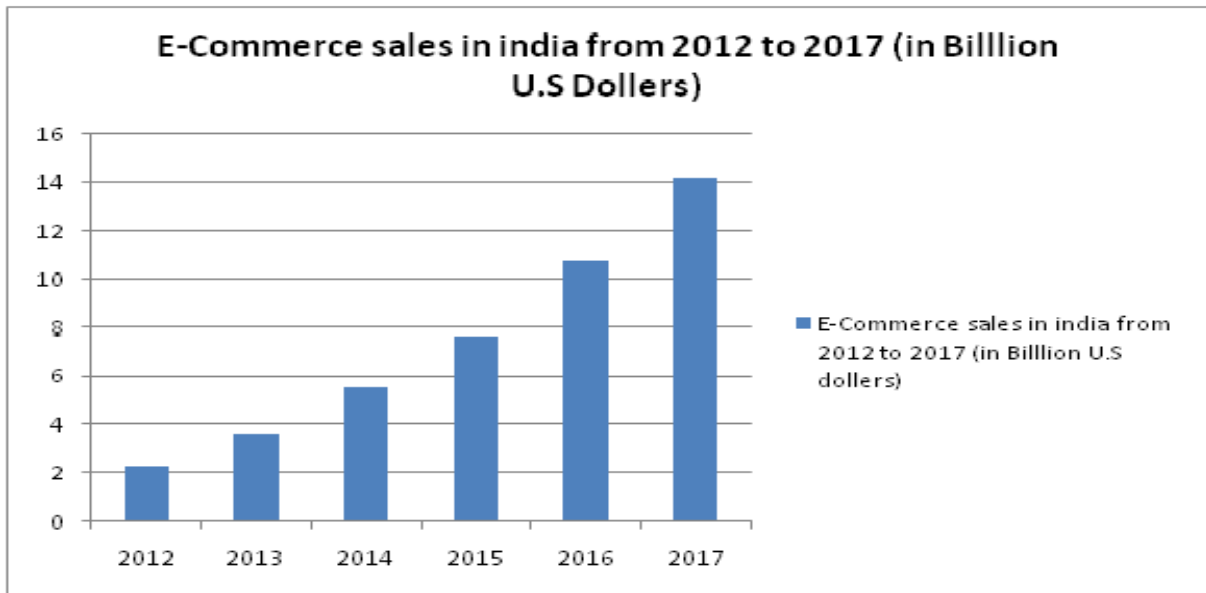


Figure 10.1: E-commerce sales in India

10.3 Retail E-Commerce sales in India from 2012 to 2017 (in billion U.S. dollars)

This statistic gives information on retail e-commerce sales in India in 2012 and 2015 and provides a forecast until 2017. In 2014, retail e-commerce sales amounted to 6 billion US dollars and are projected to grow to 14.18 billion US dollars in 2017.

ACKNOWLEDGMENT

We would like to take the opportunity to express our sincere thanks to our guide **Prof. Tabrez Khan**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout our project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped us track and schedule the process effectively. His time, ideas and encouragement that he gave his help us to complete our project efficiently.

We would also like to thank **Dr. Abdul Razak Honnutagi**, AIKTC, Panvel, for his encouragement and for providing an outstanding academic environment, also for providing the adequate facilities.

We are thankful to **Prof. Tabrez Khan**, HOD, Department of Computer Engineering, AIKTC, School of Engineering, Panvel and all my B.E. teachers for providing advice and valuable guidance.

We also extend our sincere thanks to all the faculty members and the non-teaching staff and friends for their cooperation.

Last but not the least, We are thankful to all our family members whose constant support and encouragement in every aspect helped us to complete our project.

Ansari Mohd. Azam Mohd. Salim Reshma Bano (11CO53)

Jawahire Nakash Karim Zohra (11CO56)

Shaikh Anas Munawwar Aisha (11CO38)

Siddiqi Muzammil Ahmad Mansoor Ahmad Razia Begum (11CO55)

(Department of Computer Engineering)

University of Mumbai.