“Wise Shopper”

An Independent Price Comparison Service Website

Prepared By:

Shafayeat Kabir Sumit

B.sc in Software Engineering

Daffodil International University

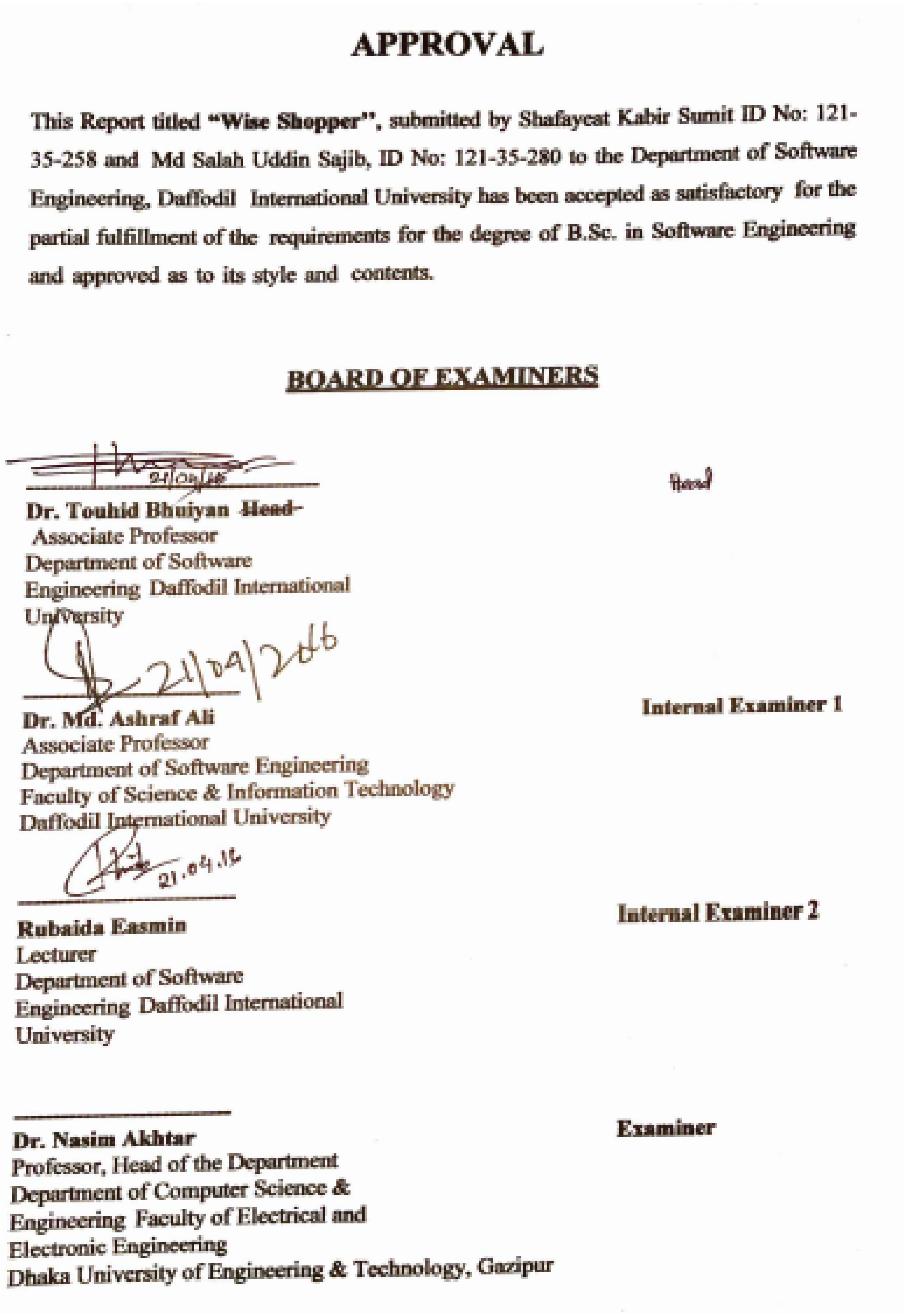
ID: 121-35-258

Md Salah Uddin Sajib

B.sc in Software Engineering

Daffodil International University

ID: 121-35-280



APPROVAL

This Report titled **"Wise Shopper"**, submitted by Shafayeat Kabir Sumit ID No: 121-35-258 and Md Salah Uddin Sajib, ID No: 121-35-280 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

21/04/16

Dr. Touhid Bhuiyan ~~Head~~ Head
Associate Professor
Department of Software
Engineering Daffodil International
University

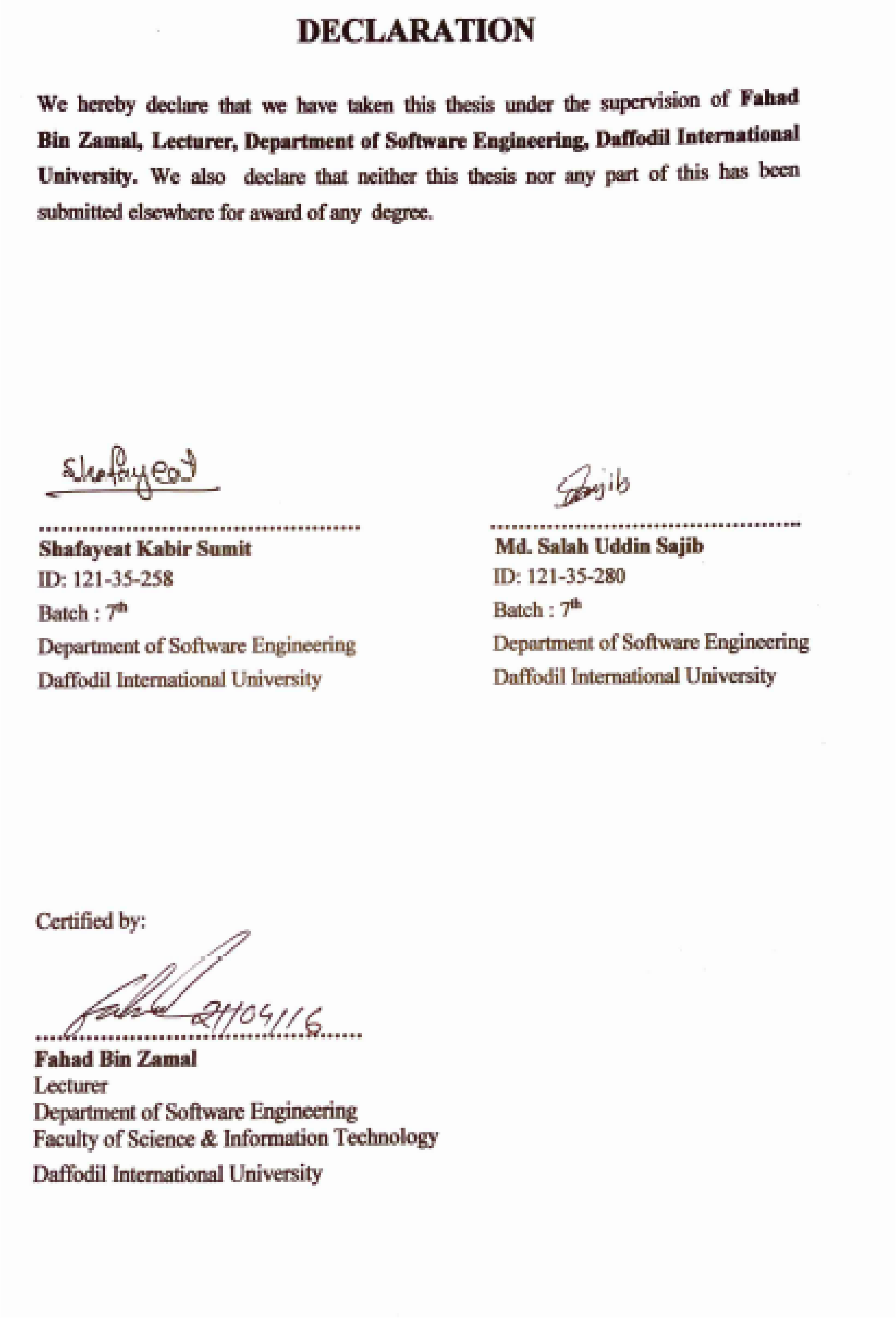
21/04/2016

Dr. Md. Ashraf Ali **Internal Examiner 1**
Associate Professor
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

21.04.16

Rubaida Easmin **Internal Examiner 2**
Lecturer
Department of Software
Engineering Daffodil International
University

Dr. Nasim Akhtar **Examiner**
Professor, Head of the Department
Department of Computer Science &
Engineering Faculty of Electrical and
Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur



DECLARATION

We hereby declare that we have taken this thesis under the supervision of **Fahad Bin Zamal, Lecturer, Department of Software Engineering, Daffodil International University.** We also declare that neither this thesis nor any part of this has been submitted elsewhere for award of any degree.

Shafayeat

....
Shafayeat Kabir Sumit
ID: 121-35-258
Batch : 7th
Department of Software Engineering
Daffodil International University

Sajib

....
Md. Salah Uddin Sajib
ID: 121-35-280
Batch : 7th
Department of Software Engineering
Daffodil International University

Certified by:

21/04/16
....
Fahad Bin Zamal
Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

ACKNOWLEDGEMENT

First offal, wear grateful to The Almighty Allah for making us eligible to complete this project.ThenwewouldliketothankoursupervisorFahadBinZamal,Lecturer, Department of SoftwareEngineering.Weareextremelygratefulandindebtedtohisforhersexpert,sincereandvaluableguidanceandencouragementextendedtous.Besideoursupervisor,we would like to express our profound gratitude to **Abdur Razzak Javed**, Doctoral Researcher, Process Quality Research Group Lero the Irish Software Research Centre ,University of Limerick,Irelandforhispatience,motivation,enthusiasmandimmenseknowledge.Wewishto expressoursincerethanksto **Dr.Touhid Bhuiyan**, Associate Professor, Head of Software Engineering department for his constant encouragement. Lastly, we would like to thank to our parents, for their unconditional support, love and without this we would not have come this far.

ABSTRACT

This project entitled “wiseshopper” isawebbased price-comparison shopping site and distributed content commerce service. It also come swath an pith at helpsdevelopersanddataanalysttoplaywiththedatagrabbedfromdifferentecommerce/B2bwebsites.Acomparisonshoppingwebsite, sometimes called a price comparison websites a vertical search engine that shoppers use to filter and compare products based on price, features, and other criteria. The main purpose of this project can be divided into two sections. First, the buyers who shops frame commerce websites no wave applet form wheretheycancomparethepricesofthesameproductsoldbydifferentvendorsorecommercesitesthemselves.Secondly,developersanddataanalysthavetheaccesstotheapiofthisprojectwheretheycanmakeuseofthese data for their own purposes. Most of thee- commerce websites doesn’t’ have api for the developers so, this site will help them by ensuring greasy access of data without screen scraping. Technologies issued for the websites are Django(pythonwebframework),Scrapy(frameworkforscrapping),DjangoRestFramework(forcreatingRestAPI).Also some client-side framework has used to make it more user friendly.

Table of Contents

Contents

LISTOFFIGURES

LISTOFTABLES

CHAPTER 1
INTRODUCTION

1.1 Purpose:

The document specifies all required deliverables of the wise shopper price comparison engine. The document provides Software Requirement Specifications, Software Design with diagrams, Test Case and Test Report of proposed system, Physical Design/Codes.

1.2 Document Conventions

- Main Chapter
- Font: Times New Roman
- Face: Bold
- Size :20
- Main Section Titles
- Font: Times New Roman
- Face : Bold
- Size:16
- Sub Section Titles
- Font : Times New Roman
- Face: Normal
- Size: 11
- Other Text Explanations
- Font: Times New Roman
- Face: Normal
- Size: 11

1.3 Intended Audience and Reading Suggestions

This document description project scope for software developers, users and testers to understand the basic system architecture. The reader types can be: customers: It is no surprise that the E- Commerce sector has developed rapidly in Bangladesh and there are many websites catering to the shopping needs of local consumers. B2B websites one of the the biggest constituents of the e- commerce sector in Bangladesh. This system will provide customers to buy the best second hand product from those B2B sites. It will help consumers; the opportunity to compare prices from different retailers at a glance helps them save money by finding good deals.

Developers: Whenever a developer or tool requests information from our powerful web API, they can call or, more technically speaking, create a request to the API. Our rest api provides them with easy access to the data.

Vendors: typical small business owner learn from a price comparison website. Some of the uses are listed below.

- Price comparison websites allow you to compare many of the suppliers of a product in one place, giving you a consumer eye-view of the market. It's easy to search for the products you sell to see exactly who you are up against and find out what makes each of them special

- do they have good terms of service, for example, or an easy-to-use website? It's worth finding out what else your competitors are selling, too, because you might be missing out on popular product lines and potentially lucrative business

.

- In a difficult market and with consumers looking for bargains, price is more sensitive than ever before. As a result, retailers are constantly adjusting prices to try and stay ahead of the game, both on the high street and online. As a small business, you can't afford to be out of touch on price. You may not be able to compete purely on price with the biggest firms but you must make sure that your price is competitive.

- Reputation is critical and many price comparison websites facilitate word-of-mouth recommendation through customer ratings and reviews. This is basically free market research that tells you which products are rated most highly. What's more, most reviews include feedback about service as well as the products themselves so you can compare how well your competitors are meeting their customers' expectations. Could you do better?

1.4 Scope:

According to BASIS (Bangladesh Association of Software and Information Services), in just three years, more than 2000 online trading platforms have made its way to Bangladesh. Businesspeople and trading houses state that the rapid growth in internet networking and mobile phone coverage induces them to meet customers online and deliver a smooth and transparent shopping experience. Including e-ticket sales and sales on Facebook the market has yearly transactions around Tk 10 billion . As the virtual shopping becoming more and more popular here in this country there comes the need of a price comparison engine where consumers have the opportunity compare products based on price, features, and other criteria. Although the ecommerce sites is growing rapidly most of them don’t have the web api so it is making things harder for developers to get access to their data so the integrated api will provide them with the ability to access , manipulate and freely work with those data.

The overall SDLC steps that will be followed in this project is given below:

- ✓ **Pre-analysis**
- ✓ Scenario/Issue Initiation
- ✓ Requirement collection

- ✓ **Analysis**
- ✓ Customer Requirement Specifications(crs)
- ✓ Software Requirement Specifications(SRS)
- ✓ Software Test Plan(STP)
- ✓ Design
- ✓ High Level Design
- ✓ Detail Level Design
- ✓ Test Case

- ✓ **Implementation**
- ✓ Codebase
- ✓ Unit Test

- ✓ **Completion**
- ✓ Software Test Plan(System test Plan)
- ✓ SDC(Software Development Completion)

- ✓ **System Test**
- ✓ System Test Report(STR)
- ✓ Software Development Completion Report(SDC)

CHAPTER: 2

PRE ANALYSIS

2.1 Scenario

Scenario is a process used during requirement analysis to describe the use of proposed system. Scenario is a story about proposed system which describes an overview of the whole system.

The scenario of proposed system is given below:

The system aggregates all the b2b websites of Bangladesh and turn them into a single platform where consumers can compare price of single product sold by different vendors. It also comes with a webApi which helps developer to get access to the products and their related data in Jason format.

The system users can be divided into two types:

I. Consumers

II. Developers

The following table shows the brief description about different users with their permission based on different operations.

Users	**Description**
consumers	Consumers who want to buy product will be able to search for the product a list of the same matched product available from the same vendor will be shown.

Developers	Developers are registered user of the system. A developer can access to the provided rest webApi. According to the product Id he can get access to a specific product.

Table 2.1: user permission chart

For building a price comparison engine, we need to collect, compare and display the data. There are many ways to collect data for comparison. Price comparison sites can collect data directly from merchants. Retailers who want to list their products on the website then supply their own lists of products and prices through merchant feeds, and these are matched against the original database. There are sites which provide data through API's so just invoking proper endpoints will get us proper data. Web scraping is another option to collect data which focus more on the transformation of unstructured data on the web in a structured form. As our intended sites don't have api for this reason we will scrape data from those sites.

Our comparison shopping website will work like vertical search engine that shoppers use to filter and compare products based on price, features, and other criteria. They can also choose specific criteria of products to view them.

There should be an web api in this system which allows the developers (registered users) easy access to those scrapped data. According to specific product id they can access to a single products information. The api should be browsable so, they can get access according to the category. There should be an user friendly good looking interface for browsing the api .

2.2 Peer Review:

Peer review is a process used for checking the work performed by one's equals (peers) to ensure it meets specific criteria. Peer review is used in working groups for many professional occupations because it is thought that peers can identify each other's errors quickly and easily, speeding up the time that it takes for mistakes to be identified and corrected. In software development, peer review is sometimes used in code development where a team of coders will have a meeting and go through code line by line (even read it aloud possibly) to look for errors. Generally, the goal of all peer review processes is to verify whether the work satisfies the specifications for review, identify any deviations from the standards, and provide suggestions for improvements.

The following Products are reviewed by developer:

Review 1:**Shopping.com**

Shopping.com is a price comparison service owned by eBay and operates websites in USA, United Kingdom, France, Germany and Australia. Shopping.com started out with the name DealTime.com which still operates as a related, but otherwise separate website.

Description: the original business model was to create a downloadable client that would monitor changes in prices of products the user seeks to buy over time, notifying the user when the product price reached a predetermined level (hence the site's original name, DealTime). Originally targeting Americanconsumers, the basic concept was quickly modified from a downloadable client to a purely web-based service, a concept the company has kept ever since.

Review 2: **NextTag**

Nextag is an independent price comparison service website for products, travel, and education.Categories include: Baby, Books, CDs and DVDs, Clothing, Accessories, Collectibles, Art, Computers & Software, Electronics and many other things. It started originally as a website where buyers and sellers could negotiate prices for computers and electronics products. The current business model with a focus on comparison shopping has existed since 2000.

Date: 20/10/2015

Description: NexTag is an example of what Google does best: scraping data from around the

Web and presenting it to you in a way that's clean, straightforward, and usable. And best of all, the price comparison feature is part of the search engine itself.To use it, all you have to do is search for a product under the Shopping section. After selecting one of the results, you'll see a few photos, a product description, any available reviews, and most importantly, pricing information from dozens of retailers.

The product result page automatically shows you the lowest available price, but if it comes from a retailer that you can't or don't want to use, then you can expand to see the complete table comparison. It also has a convenient toggle for when you want to buy used or refurbished.

CHAPTER 3
SYSTEM ANALYSIS

After gathering requirements from different sources, the next phase is to analysis on requirements. In this stage, software engineers work with customers and system end-users to find out about what services the system should provide, the required performance of the system, hardware constraints and so on.

3.1 Requirement analysis

There requirement analysis is divided into several parts all of them listed (in details) below.

3.1.1 Requirement Elicitation:

Requirements engineering, requirements elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders. The practice is also sometimes referred to as "requirement gathering”. The term elicitation is used in books and research to raise the fact that good requirements cannot just be collected from the customer, as would be indicated by the name requirements gathering. Requirements elicitation is non-trivial because you can never be sure you get all requirements from the user and customer by just asking them what the system should do OR NOT do (for Safety and Reliability). Requirements elicitation practices include interviews, questionnaires, user observation, workshops, brainstorming, use cases, role playing and prototyping. Some common techniques are used to gather proper requirements. Techniques that are used to collect requirements are

Interviewing: As this project totally depends on the consumers who prefers virtual shopping. So we tried to speak with them took interview and told them about the idea of our price comparison engine to see their response and as well as to gather requirement for the project. We also studied several survey those based on virtual shoppers interview and analyzed their shopping habit.

Studying similar project: Most of the requirements are collected by studying similar projects. We studied several project very deeply and gathered requirement from there. Some of price comparison engines that we have studied are Pricegrabber.com, Pricezilla.com, Nexttag.com and shopping.com.

Brainstorming: Some of the requirement is elicited by brainstorming. We discussed with different developers and people from different domain. That helps us to generate new, useful ideas and promoting creative thinking for finding the solution to a specific issue.

Functional Requirements:

1. System should work like a vertical search engine so there will be a search bar (top centered) in the starter page
2. Below the search there should be some popular category of products clicking on the category will take us to the list of products
3. All the search results (products) must contain these information's - title, image of the product, part of the description, postdate , vendor's phone number and the logo of the website where the product is scraped from
4. Products should be listed as horizontal list
5. There would be a view more detail option for the product which will redirect to the page where the data is fetched from
6. Starter page need to have some featured product listed below of the starter page
7. System's Webmail should have an authentication process .so only the registered user can get accessto the api's data.
8. Systems Api should be brow sable and a user friendly interface should be built.

9. Developer would have the ability to access all data
10. Developers would have the ability to access a single product's data using product ID.
11. System should have the ability to remove similar product of it is scraped from two different site.

Non Functional Requirements:

1. System scheduler need to run every hour
2. As the system deals with a huge amount of data it must have a proper backup
3. The most important feature of the System is the search option so this should work as efficiently as possible .
4. The browsableapi should be as user friendly as it can be made.
5. The interface should not be complicated and it should be easy to be followed by the user
6. The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

3.1.2 Software Requirement Specification

Requirement ID	Requirement Title	Priority

FR_01	System should work like a vertical search engine so there will be a search bar (top centered) in the starter page	High
FR_02	Below the search there should be some popular category of products clicking on the category will take us to the list of products	High
FR_03	All the search results (products) must contain these information's - title, image of the product, part of the description, postdate , vendor's phone number and [illegible]	High
FR_04	Products should be listed as horizontal list	Medium
FR_05	There would be a view more detail option for the product which will redirect to the page where the data is fetched from	High
FR_06	Starter page need to have some featured product listed below of the starter page	Low
FR_07	System's WebAPI should have an authentication process .so only the registered user can get access to the api's data	High
FR_08	Systems Api should be browsable and a user friendly interface should be built .	High
FR_09	Developer would have the ability to access all data	High
FR_10	Developers would have the ability to access a single product's data using product ID.	High

FR_11	System should have the ability to remove similar product of it is scraped from two different site.	High
NFR_01	System scheduler need to run every hour	High
NFR_02	As the system deals with a huge amount of data it must have a proper backup	High
NFR_03	The most important feature of the System is the search option so this should work as efficiently as	High
NFR_04	The browsableapi should be as user friendly as it can be made.	High
NFR_05	The interface should not be complicated and it should be easy to be followed by the user	High
NFR_06	The effort required to move the software to a different target platform. The measurement is most commonly person months or % of modules that need	Medium

Table 3.1: requirements specification table

3.1.3 Systems legality issues: Why we are discussing this?

This system completely depends on web scraping and sometimes crawling. There are some legality issues when someone scrape other websites. Now we will discuss about whether it is legal or not and what is the terms and conditions of those sites which we are scraping.

What is the legality of web scraping?

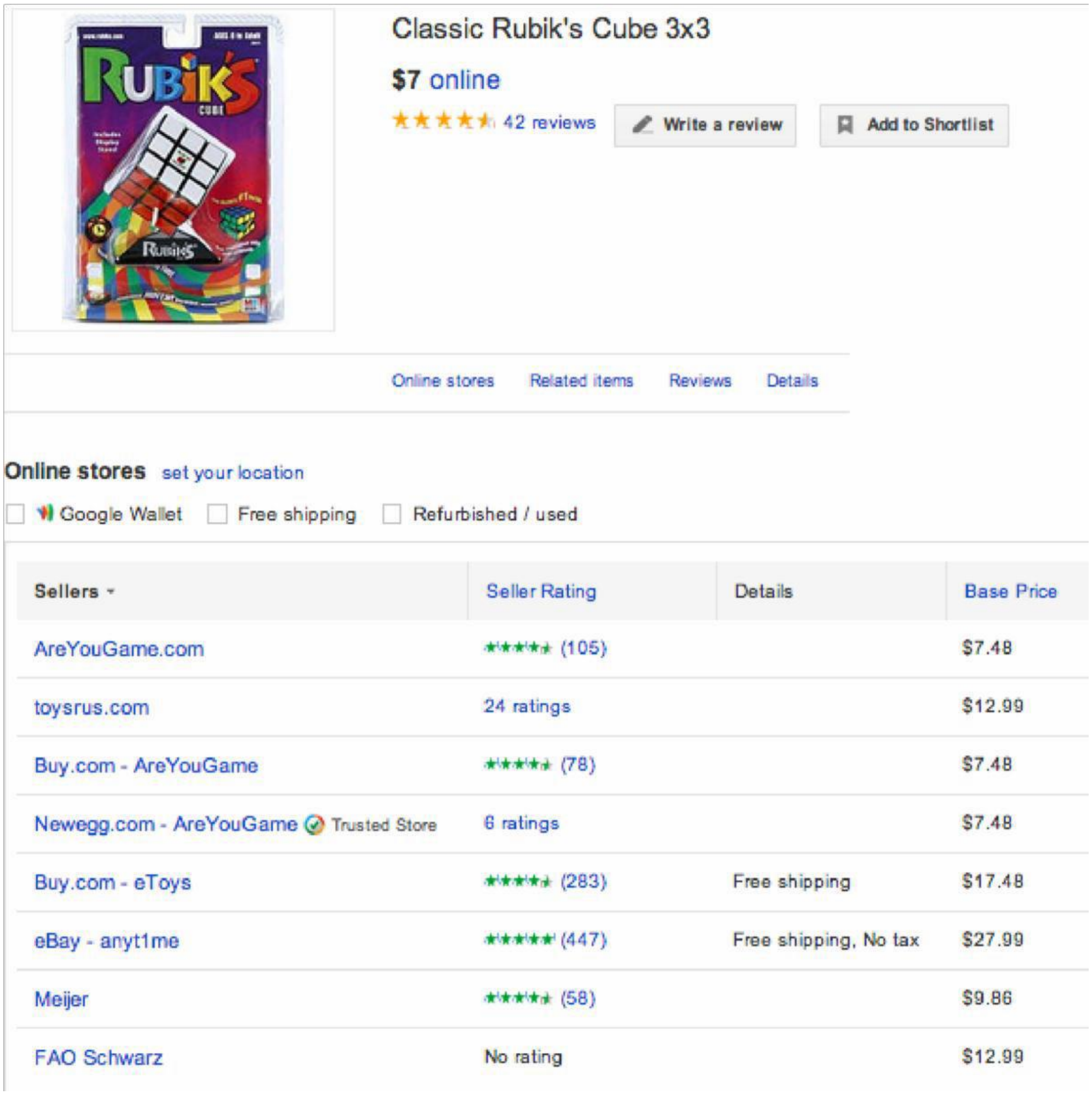
If the owner of the web site objects to the way we are using the data, they have a reasonably good way of forcing us to stop. As long as we don’t scrape any copyrighted content we don’t have any problem with the legality according to the cyber law of our country.

3.2 System Development Plan & Process:

3.2.1 Project Overview:

The project named as “wiseshopper” is the first ever price comparison engine in Bangladesh. We will discuss about how price comparison engines work in this overview. Comparison shopping engines collect product information, including pricing, from participating retailers and then display that collective information on a single results page in response to a shopper's search query. In this way, shoppers can compare each retailer's price, shipping options, and service on a single page and choose the merchant that offers the best overall value.

Example: Let’s say I want to buy a Rubik's Cube and want to see what my options are. I go toGoogle Shopping (one of the most popular CSE's around) and do a search. Here's what comes up:



Classic Rubik's Cube 3x3

$7 online

★★★★½ 42 reviews

Online stores set your location

□ Google Wallet □ Free shipping □ Refurbished / used

Sellers	Seller Rating	Details	Base Price
AreYouGame.com	★★★★★ (105)		$7.48
toysrus.com	24 ratings		$12.99
Buy.com - AreYouGame	★★★★★ (78)		$7.48
Newegg.com - AreYouGame ✓ Trusted Store	6 ratings		$7.48
Buy.com - eToys	★★★★★ (283)	Free shipping	$17.48
eBay - anyt1me	★★★★★ (447)	Free shipping, No tax	$27.99
Meijer	★★★★★ (58)		$9.86
FAO Schwarz	No rating		$12.99

Fig 3.1: shopping comparison site

3.2 Proposed system and its benefits:

Find out who's selling what. Price comparison websites allow you to compare many of the suppliers of a product in one place, giving you a consumers' eye-view of the market. It's easy to search for the products you sell to see exactly who you are up against and find out what makes each of them special

- do they have good terms of service, for example, or an easy-to-use website? It's worth finding out what else your competitors are selling, too, because you might be missing out on popular product lines and potentially lucrative business.

Keep up to date on price. In a difficult market and with consumers looking for bargains, price is more sensitive than ever before. As a result, retailers are constantly adjusting prices to try and stay ahead of the game, both on the high street and online. As a small business, you can't afford to be out of touch on price. You may not be able to compete purely on price with the biggest firms but you must make sure that your price is competitive.

Compare levels of service. Customer service is an important selling point for small businesses and can help them stand out from bigger operators. Service terms should cover delivery speeds and costs, refund policies and payment security. Crucially, they should also offer an easy way to ask questions and make complaints. Use price comparison sites to compare the service being offered by the different sellers in your market.

Check out the detail. Shoppers don't always choose the cheapest product, but will take into account all the extras, including shipping costs, the service promise and the reputation of the retailer, before committing to buy. So it's worth checking out the details and not just comparing price tags - that's what your customers will be doing after all.

Pay attention to user ratings and reviews. Reputation is critical and many price comparison websites facilitate word-of-mouth recommendation through customer ratings and reviews. This is basically free market research that tells you which products are rated most highly. What's more, most reviews include feedback about service as well as the products themselves so you can compare how well your competitors are meeting their customers' expectations. Could you do better?

Watch out for hot products and spot trends. Many price comparison sites pick out best-selling items or feature popular products on their home page. By spotting trends, you can make sure you stock the right goods, promote them prominently and price them competitively to take advantage of demand.

Look for weaknesses. When you analyses your competitors on price comparison sites, don’t just look at what they are doing - check for the gaps, too. Do you have something unique to offer that can make you stand out from the crowd? What are your strengths and do your customers know about them? Compare unique selling points and make sure yours are distinct. Comparison websites can give you a good snapshot of a sector - is there a gap in the market for a new product or service?

Click through to websites. Some price comparison websites allow you to buy then and there. But most offer lists and links through to individual suppliers. This is a crucial stage in the buying processwhen shoppers often rethink their choices. To investigate your competitors fully, you need to click through to their websites and see what they are getting right and wrong. Shoppers will be put off by badly-designed landing pages, poor navigation, and inadequate security or by simply having to jump through too many hoops to be able to buy something. Make sure your website is clear and simple to use.

Project Development Process Overview:

The entire project depends on web scraping. First of all we will extract data from the popular b2b websites like Bikroy.com and Ekhanei.com then we will run filter on the grabbed data. Using those data we will do two things one is we will create an web api from the data and another is we aggregate this in a place from which consumers can search their product.

Data scraping is normally associated with the programmatic collection of visual data from a source, instead of parsing data as in Web scraping. Originally, screen scraping referred to the practice of reading text data from a computer display terminal's screen. This was generally done by reading the terminal's memory through its auxiliary port, or by connecting the terminal output port of one computer system to an input port on another. The term screen scraping is also commonly used to refer to the bidirectional exchange of data. This could be the simple cases where the controlling program navigates through the user interface, or more complex scenarios where the controlling program is entering data into an interface meant to be used by a human. More modern screen scraping techniques include capturing the bitmap data from the screen and running it through an OCR engine, or for some specialized automated testing systems, matching the screen's bitmap data against expected results. This can be combined in the case of GUI applications, with querying the graphical controls by programmatically obtaining references to their underlying programming objects.

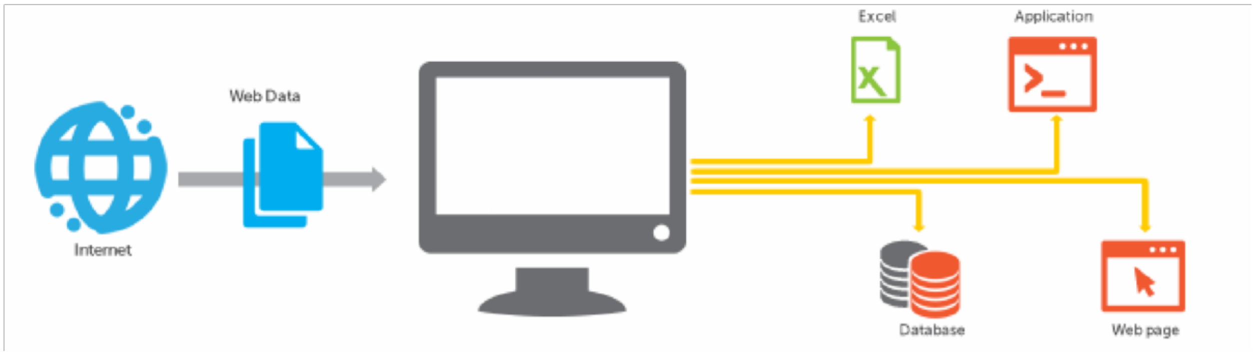


Fig 3.2: How data scraped from the web

Typically a fully automated process, web scraping is also possible through manual means. In the case of hand scraping, human operators copy and paste large volumes of information from specific pages into a text file or database. Very labor-intensive and slow, this method succeeds in collecting more precise information than other technical techniques.

Automated data scrapers make the process faster. From HTTP socket programming to UNIX grep commands, they are well equipped to use varied techniques with ease. For instance, a scraper tool can view dynamic web pages in full and then parse them into DOM trees--as and when required.

These programs boast of full-fledged web browsers that are embedded into the software for better automation.

3.2.4 Technologies Used:

In this part we will be discussing about the technologies used for building that system and why we choose those technologies, Framework and Libraries. List of technologies used are following

Programming Language: Python Web Framework: Django (Backend)

For creating API: Django Rest Framework

For scraping and Crawling: Scrapy(web scraping framework)

3.3 Project Schedule (Detail breakdown in a .doc table):

A. Project Scope

Primary Phase				
SL	Job Description	Start date	End Date	Total Date
1	Idea Searching	02/10/2015	7/10/2015	5
2	Domain Knowledge Gathering	8/10/2015	28/10/2015	20
3	Resource Availability Checking	28/10/2015	3/11/2015	5
4	Brainstorming	10/11/2015	15/11/2015	5
5	Feasibility Study	15/11/2015	20/11/2015	5
			Total Days	40

Table No 3.2: Schedule Project Scope table

B. Idea Proposal

Advanced Phase				
SL	Job Description	Start date	End Date	Total Date
1	Idea Searching with Supervisor	20/12/2015	23/12/2015	3
2	Advanced Domain Knowledge Gathering	23/12/2015	25//12/2015	2
3	Advanced Resource Gathering from Supervisor and from online open resources	25/12/2015	30/12/2015	5
4	Advanced Brainstorming with Supervisor	1/1/2016	5/1/2016	5
5	Advanced Feasibility Checking with Supervisor	5/1/2016	10/1/2016	5
6	SDLC Model Selection for System Development	10/1/2016	15/1/2016	5
			Total Days	25

Table No: 2 Idea Proposal

C. Requirement Collection

SL	Job Description	Start date	End Date	Total Date
1	Develop System Scenario	15/1/2016	17/1/2016	2
2	Requirement Identify from System Scenario	17/1/2016	25/1/2016	8
3	Row Requirement Collection from Identified Requirement	25/1/2016	27/1/2016	2
4	Functional Requirement Collection	27/1/2016	30/1/2016	3
5	Software requirement Specification(SRS)	30/1/2016	31/1/2016	1
6	Manage All Requirement and Arrange all information	1/2/2016	2/2/2016	1
7	Non Functional Requirement Specification	2/2/2016	3/2/2016	1
			Total Days	18

Table No 3.3: Requirement Collection

System Design

Physical System Design				
SL	Job Description	Start date	End Date	Total Date
1	Graphical User Interface(GUI) UX Modeling	3/2/2016	5/2/2016	2
2	System Prototype Designing	5/2/2016	6/2/2016	1
			Total Days	3

Fig 3.3: System Design

Logical System Design

1	System Environment Design	6/2/2016	8/2/2016	2
2	Use Case Diagram Design	8/2/2016	9/2/2016	1
3	Detail Flow Diagram DFD-Level:0 Designing	9/2/2016	10/2/2016	1
4	Detail Flow Diagram DFD-Level:1 Designing	10/2/2016	11/2/2016	1
			Total Days	5

Table No 3.4: System Design

Technology Used				
SL	Job Description	Start date	End Date	Total Date
1	Building first scraper	11/2/2016	21/2/2016	10
2	Building second scraper	21/2/2016	25/2/2016	4
3	Refactoring Codebase	25/2/2016	27/2/2016	2
4	Backend Integration	27/2/2016	28/2/2016	1
5	Database PostgreSQL Integration	28/2/2016	29/2/2016	1
6	Refactoring Code & Front-End design	1/3/206	11/3/206	10
			Total Days	28

Table No 3.5: System Implementation

Black Box Testing				
SL	Job Description	Start date	End Date	Total Date
1	Boundary Value Testing			n/a
2	System Functionality Testing			n/a
			Total Days	
			Total Days	119

Table No 3.6: System Testing

3.4 Software Test Plan:

A test plan is a document detailing the objectives, target market, internal beta team, and processes for a specific beta test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow.

3.4.1 Introduction:

Test Plan document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

3.4.2 Test Items:

The test items are features to be tested that are deliverable to the clients. This is the list of what to be tested Test items for proposed system. The list of things need to be tested are listed below.

- Registration Process for api module
- Login process for api module
- Testing multiple spiders (the part of the scraper where xpath/css path is defined and data grabbed according to those path).
- Testing the filter (Item Pipeline)
- Testing the scheduler
- Testing the search option
- Individual product category selection process

3.4.3 Features to be tested:

Feature ID	**Title**	**Description**	**Involved User**
001	Registration process	Registration for Api users who want to gain access to the provided api.	Api user
002	Login Process	User Login for the api users.	Api user
003	Spider Testing	Spiders need to be tested for to check whether they are properly selection data using xpath/css selector	Developer

004	Data filtering process	Data filtration need to be checked to ensure avoid delicacy.	Developer
005	Testing the scheduler	As the spider needs to run after a certain period of time so it is important to check the scheduler	Developer
006	Searching option	The search option needs to be tested properly to ensure the proper data is coming from the backend	Consumer
007	Category selection	While a specific category selected we should make sure that other category's product is not coming from backend.	consumer

Table 3.6: Features testable

CHAPTER 4

SYSTEM DESIGN

4.1 Flowchart diagram of a scraper

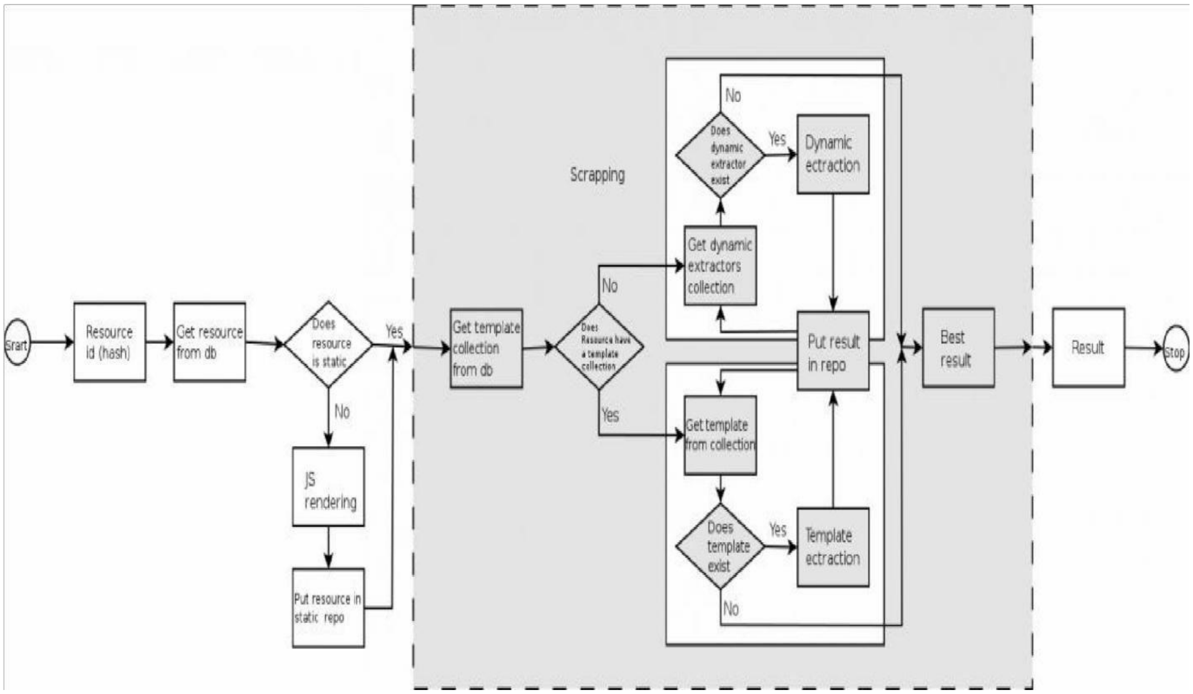


Figure 4.1: flow chart diagram of the single scraper

4.2 Use Case Diagrams of the systems:

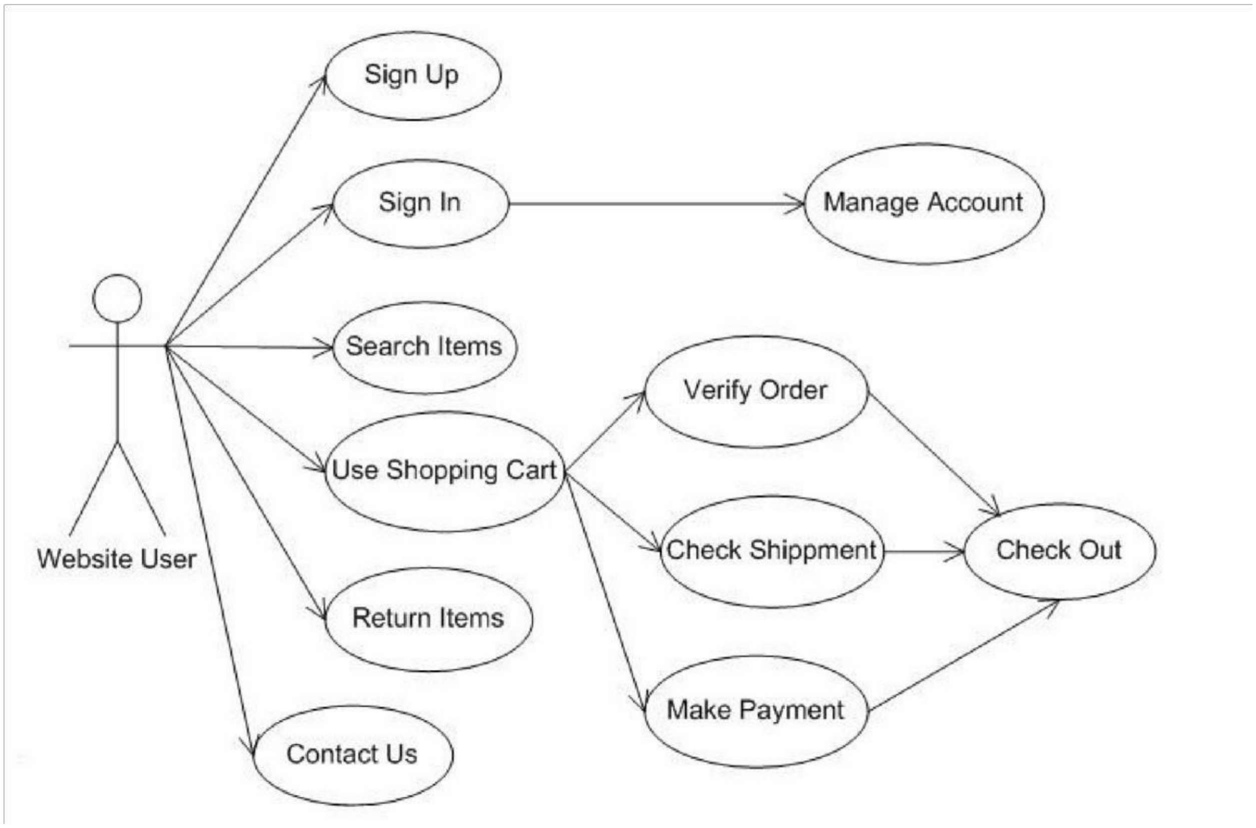


Figure 4.2: Use case Diagram

4.3 Entity Relationship Diagram:

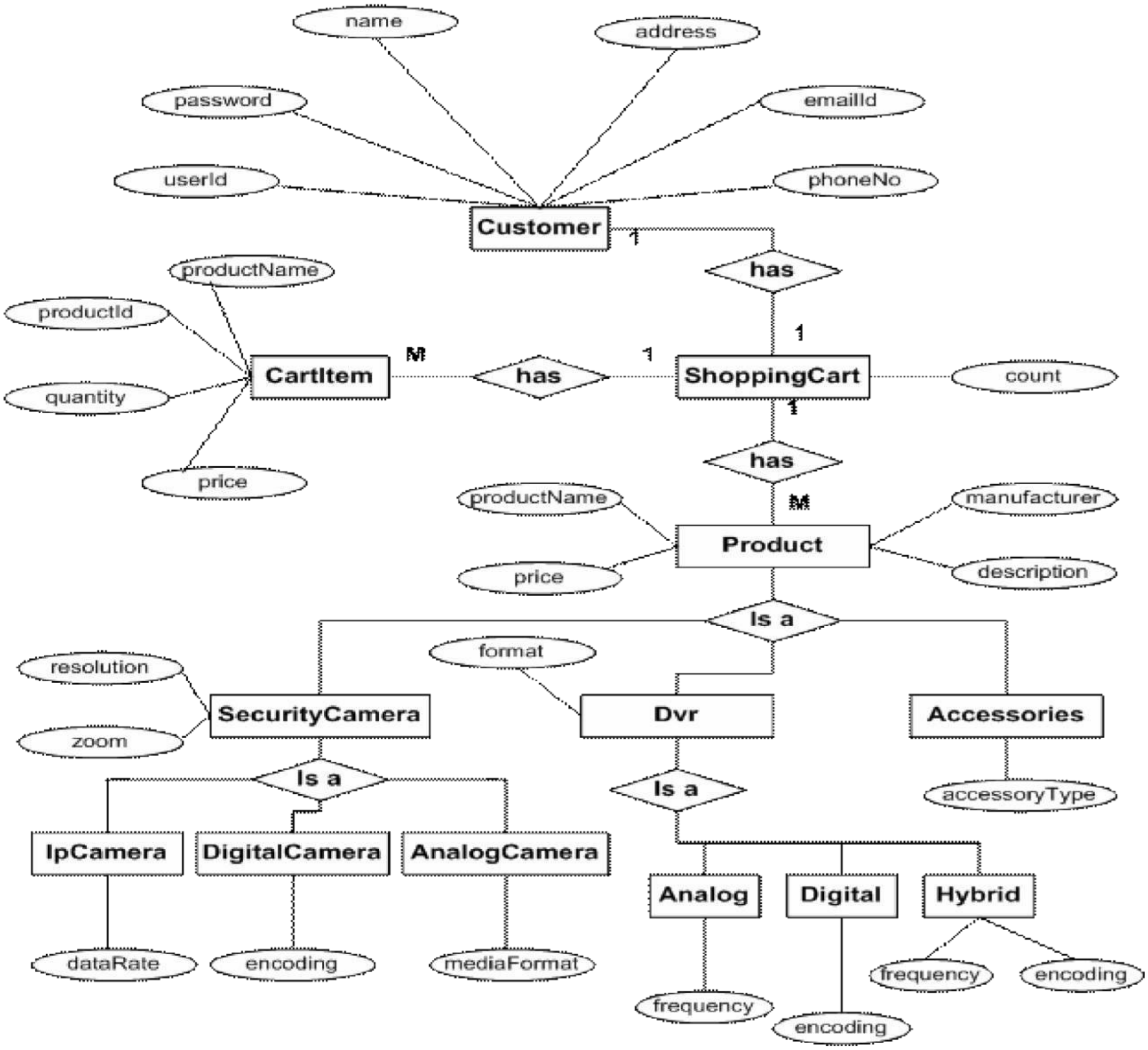


Fig 4.3: ER diagram of wises hopper

4.4 Data Flow Diagramof the system:

Thefollowingdiagramshowsanoverviewofthescraperarchitecturewithitscomponentsandan outlineofthedataflowthattakesplaceinsidethesystem(shownbythegreenarrows).Abrief descriptionofthecomponentsisincludedbelowwithlinksformoredetailedinformationaboutthem. Thedataflow is also described below.

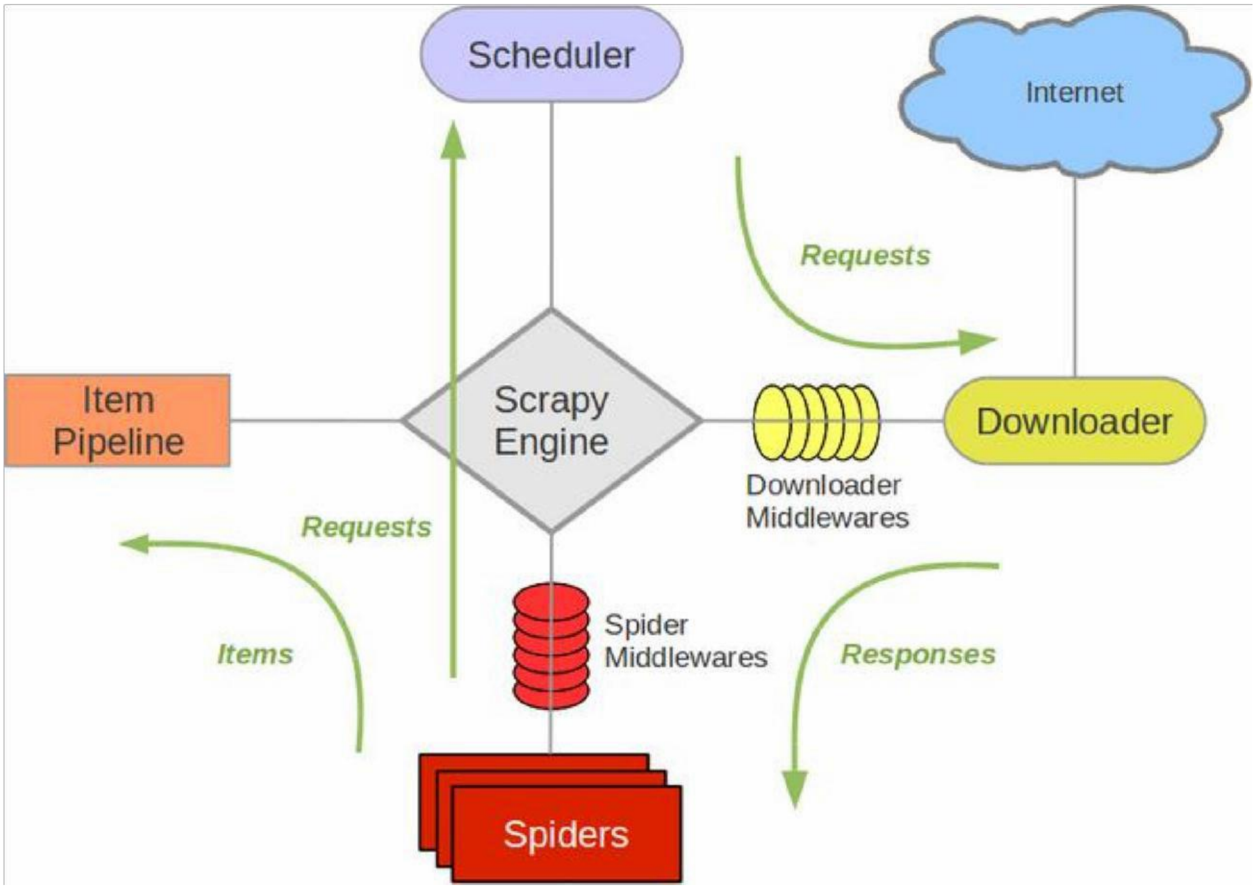


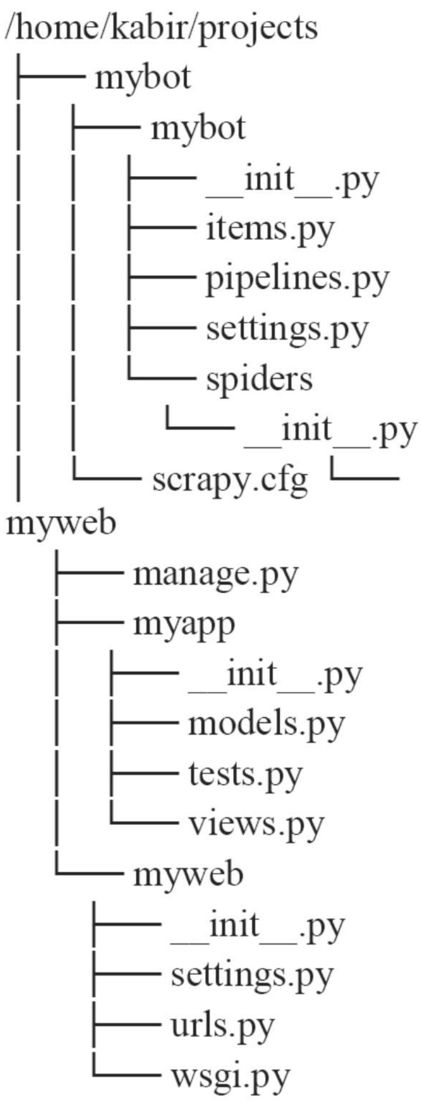
Fig 4.4: Data flow diagram of a sample scrapper

CHAPTER 5

SYSTEM IMPLEMENTATION AND MAINTENANCE

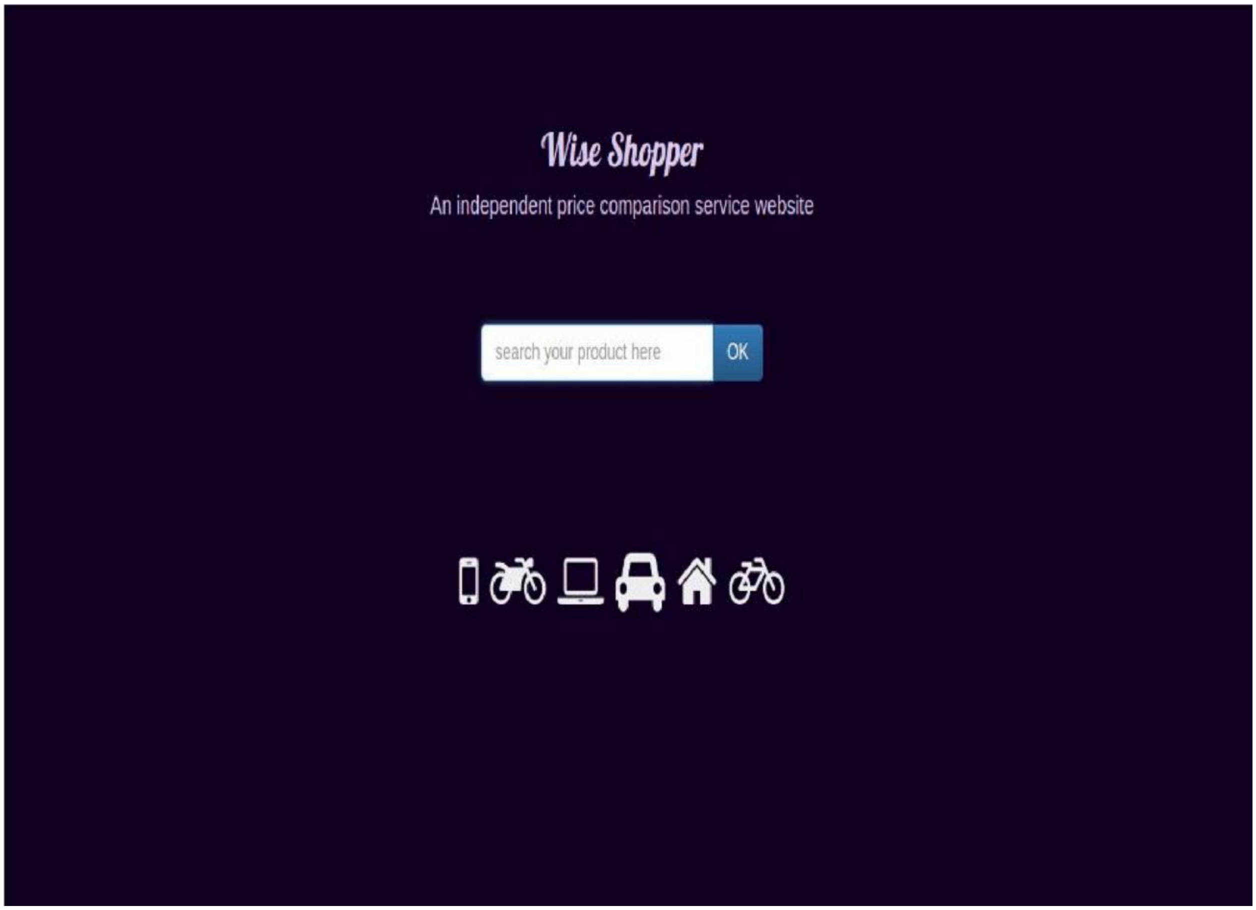
5.1 Physical Design/Codes:

All the data are scraped using Scrappy (a framework for scraping data from the web) and stored in PostgreSQL dB and the backend is done using Django(a python web framework) . While using both of the frame work at a same project right folder structure is crucial. Here is the basic folder structure of the system. Data are scraped using Scrappy (a framework for scraping data from the web) and stored in PostgreSQL dB and the backend is done using Django(a python web framework) . While using both of the frame work at a same project right folder structure is crucial. Here is the basic folder structure of the system.



```
/home/kabir/projects
├── mybot
│   ├── mybot
│   │   ├── __init__.py
│   │   ├── items.py
│   │   ├── pipelines.py
│   │   ├── settings.py
│   │   └── spiders
│   │       └── __init__.py
│   └── scrapy.cfg └──
myweb
    ├── manage.py
    ├── myapp
    │   ├── __init__.py
    │   ├── models.py
    │   ├── tests.py
    │   └── views.py
    └── myweb
        ├── __init__.py
        ├── settings.py
        ├── urls.py
        └── wsgi.py
```

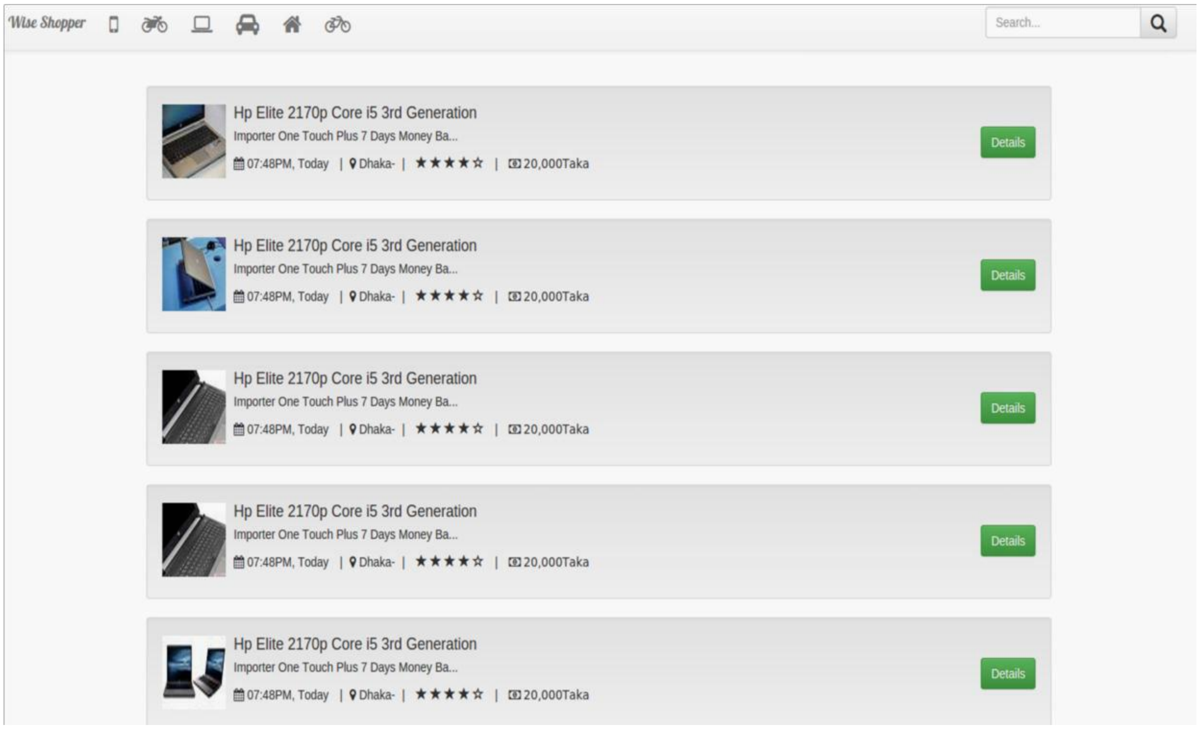
Project Screen Shots:



Wise Shopper

An independent price comparison service website

Fig 5.1: System Screen shot 1



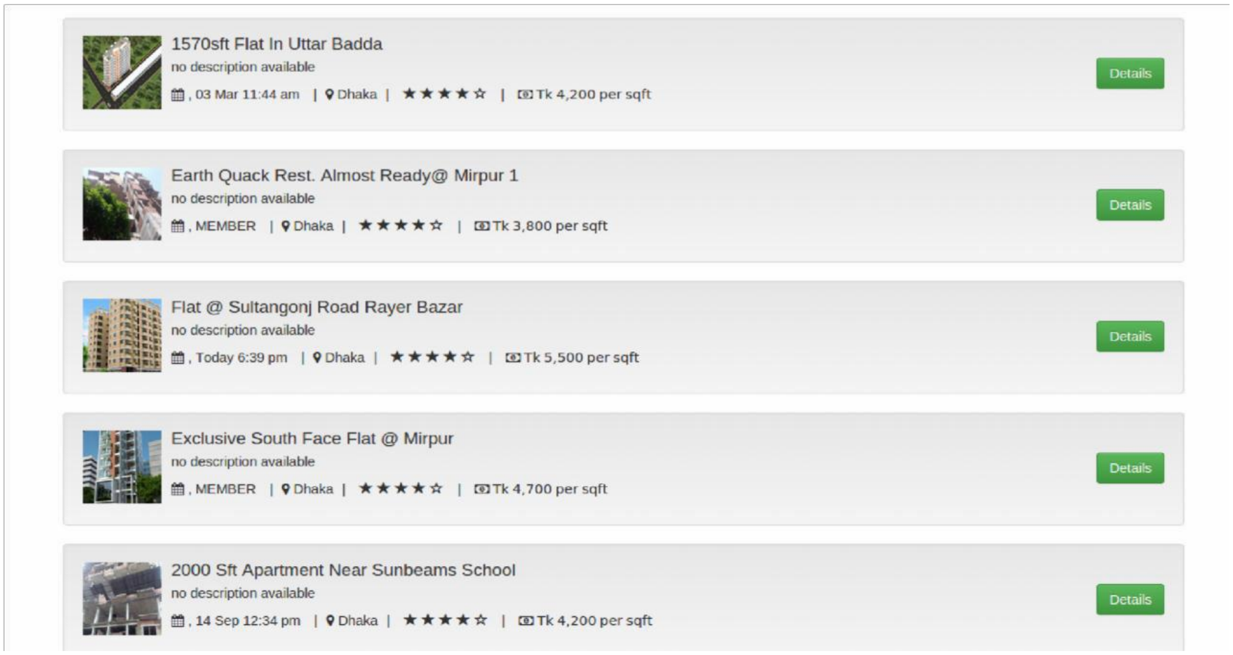
Hp Elite 2170p Core i5 3rd Generation
Importer One Touch Plus 7 Days Money Ba...
07:48PM, Today | Dhaka- | ★★★★☆ | 20,000Taka

Hp Elite 2170p Core i5 3rd Generation
Importer One Touch Plus 7 Days Money Ba...
07:48PM, Today | Dhaka- | ★★★★☆ | 20,000Taka

Hp Elite 2170p Core i5 3rd Generation
Importer One Touch Plus 7 Days Money Ba...
07:48PM, Today | Dhaka- | ★★★★☆ | 20,000Taka

Hp Elite 2170p Core i5 3rd Generation
Importer One Touch Plus 7 Days Money Ba...
07:48PM, Today | Dhaka- | ★★★★☆ | 20,000Taka

Hp Elite 2170p Core i5 3rd Generation
Importer One Touch Plus 7 Days Money Ba...
07:48PM, Today | Dhaka- | ★★★★☆ | 20,000Taka



1570sft Flat In Uttar Badda
no description available
, 03 Mar 11:44 am | Dhaka | ★★★★☆ | Tk 4,200 per sqft

Earth Quack Rest. Almost Ready@ Mirpur 1
no description available
, MEMBER | Dhaka | ★★★★☆ | Tk 3,800 per sqft

Flat @ Sultangonj Road Rayer Bazar
no description available
, Today 6:39 pm | Dhaka | ★★★★☆ | Tk 5,500 per sqft

Exclusive South Face Flat @ Mirpur
no description available
, MEMBER | Dhaka | ★★★★☆ | Tk 4,700 per sqft

2000 Sft Apartment Near Sunbeams School
no description available
, 14 Sep 12:34 pm | Dhaka | ★★★★☆ | Tk 4,200 per sqft

Fig 5.2: system screen shot 2

5.2 Unit Test Case

A unit is the smallest testable part of an application like function, classes, procedures, and interface. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use [10]. Unit tests are created and executed by software developer during the development process.

Some major unit test on each module is shown below:

Module 1: Registration and Login

UTC ID	**UTC_1**
UTC Name	User registration page value check
Code Module	User registraion.py design page
UTC Description	1. Enter user Id 2. Enter Email
Input Values	Input 1: Id: summit input 1: Email:summit @gmail.com Input 3: Id: summit@ input 4: Email:summit.gmail.com
Expected Output	Input 1:true, input 1: true, Input 3:false 4:false
Actual Output First Iteration	Input 1:true, input 1: true,
Actual Output Second Iteration	Input 1:true, input 1: true, Input 3:false Input 4:false

UTC ID	**UTC_2**

UTC Name	Test Input values of Login Page
UTC Name	Test Input values of Login Page
Code Module	User login.py design page
UTC Description	1. The Password textbox must hide input values as **** 2. Enter invalid Email, then system show error message 3. Enter invalid password, then system show error message 4.
Input Values	Input 1: Id: summit input 1: Email:summit @gmail.com Input 3: Id: summit@ input 4: Email:summit.gmail.com
Expected Output	Input 1:true, input 1: true, Input 3:false Input 4:false
Actual Output First Iteration	Input 1:true, input 1: true, Input 3:true Input 4:false
Actual Output Second Iteration	Input 1:true, input 1: true, Input 3:false 4:false
pass/Fail	Pass: Complete $ of $. Total Percent:100%

Table 5.1: unit test case table module-1

Module 2: Project Request and Acceptance

UTC ID	**UTC_3**
UTC Name	Test Input values of Login Page
Code Module	User login.py design page
UTC Description	1. The Password textbox must hide input values as **** 2. Enter invalid Email, then system show error message 3. Enter invalid password, then system show error message 4. 123456(******) [Valid password]
Input Values	Input 1: Id: summit input 1: Email:summit @gmail.com Input 3: Id: summit@ input 4: Email:summit.gmail.com
Expected Output	Input 1:true, input 1: true, Input 3:false 4:false
Actual Output First Iteration	Input 1:true, input 1: true, Input 3:true 4:false
Actual Output Second Iteration	Input 1:true, input 1: true, Input 3:false 4:false
pass/Fail	Pass: Complete $ of $. Total Percent:100%

UTC ID	**UTC_4**

UTC Name	Test Input values of Login Page
Code Module	User login.py design page
UTC Description	1. The Password textbox must hide input values as **** 2. Enter invalid Email, then system show error message 3. Enter invalid password, then system show error message 4.
Input Values	Input 1: Id: summit input 1: Email:summit @gmail.com Input 3: Id: summit@ input 4: Email:summit.gmail.com
Expected Output	Input 1:true, input 1: true, Input 3:false Input 4:false
Actual Output First Iteration	Input 1:true, input 1: true, 3:true 4:false
Actual Output Second Iteration	Input 1:true, input 1: true, Input 3:false t 4:false
pass/Fail	Pass: Complete $ of $. Total Percent:100%

Table 5.2: unit test case table module-2

Module 3: Product Search

UTC ID	**UTC_5**
UTC Name	Find product search
Code Module	Product.py Design page, product search()
UTC Description	1. Click search box, system show search 2. By entering invalid product system show message “No product found”

Input Values	Input 1: Blank, input 2: iPhone, Input 3: 000,
Expected Output	Input 1:false, input 1: true, Input 3: false, Input 4: true,
Actual Output First Iteration	Input 1:false, input 1: true, Input 3:false, Input 4:false,
Actual Output Second Iteration	Input 1:true, input 1: true, Input 3:false Input 4:false
pass/Fail	First Iteration: 2 Failed in 4, Passed Percent:50% Second Iteration: 2 Passed in 2, Passed Percent:100%

Table 5.3: unit test case table module-3

UTC ID	**UTC_5**
UTC Name	Find product search
Code Module	Product.py Design page, product search()
UTC Description	1. Click search box, system show search 2. By entering invalid product system show message “No product found”

Input Values	Input 1: Blank, input 2: iPhone, Input 3: 000,
Expected Output	Input 1:false, input 1: true, Input 3: false, Input 4: true,
Actual Output First Iteration	Input 1:false, input 1: true, Input 3:false, Input 4:false,
Actual Output Second Iteration	Input 1:true, input 1: true, Input 3:false Input 4:false
pass/Fail	First Iteration: 2 Failed in 4, Passed Percent:50% Second Iteration: 2 Passed in 2, Passed Percent:100%

Table 5.3: unit test case table module-3

Iteration	**Number of unit test case**	**100% in Success first iteration**	**Less than 100%**	**Total succession %**
Cycle-1	Total:15	6	9	(50+75+66.66+66.66+75+75+75+75)=618.32 (618.32+600)/15=81%
Cycle-1	Total:15	15	0	100%

Fig 5.4: unit test case cycle evaluation

CHAPTER 6

COMPLETION

6.1 System Test Report

Scope of system test: The Test Plan defines the unit, integration, system and Client Acceptance testing approach. The test scope include the following

- System Test Objective and Goal: Software testing is the process that ensures the quality of software by identifying defects and failures. The major purpose of system testing is to reduce defect and deliver a error free good quality software.
- Software defects are solved in each stage of development. The correction percentage of software defects are 90% that shown in a table.

Severity	Description	correction

Critical	The defect that result in the termination of the complete system or one or more component of the system and causes extensive corruption of the data [12]. The types of defects make the system unusable. example : Installation failed,	100%
Moderate	The defect that result in the termination of the complete system or one or more component of the system and causes extensive corruption of the data [12]. The types of defects	90%
Minor	The defect that result in the termination of the complete system or one or more component of the system and causes extensive corruption of the data [12]. The types of defects	80%
Total correction defects		90%

Table 6.1: System Defects Testing

System Test Method: As the software is developed in SDLC waterfall model, it maintain testing consistency with the development process. It follows "V" model has been need for system testing to ensure project deliverables accurately.

V-Model: V-model means verification and validation model. Just like the waterfall model, the V- Shaped life is a sequential path the extension of process. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development.

Test-Result: The test result of system is shown in a following table. Total number of defects are

Defects	Number

Critical Defects	3
Major Defects	10
Minor Defects	12
Total Defects	26

Table 6.2: system test result 1

Total resolved defects are following:

Defects	Number
Critical Defects	3(100%)
Major Defects	9(90%)
Minor Defects	10(836%)
Total Defects	22(91%)

Table 6.3: system test result 2

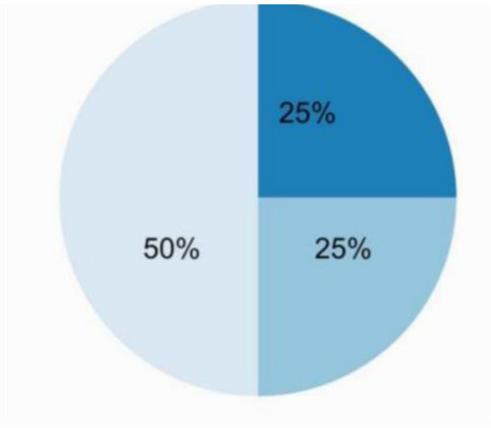


Fig 6.1: Defect Distribution pie chart

❑ Critical

❑ Moderate

❑ Minor

6.2 software development completion report (SDC)

Project Information			
Project Acronym	WS		
Project Title	Wise Shopper		
Start Date	12/11/2015	End Date	12/12/2015
Lead Institution	Daffodil International University		
Project Director	Fahad Bin Zamal		
Partner Institutions	Daffodil International University		
Project Web URL	wiseshopper.pythonanywhere.com		

Fig 6.2: SDC Table

6.3 Project Outcomes

It is difficult to describe the outcomes and impacts without going into details.

Annual project reports are required for all standard and continuing grants and cooperative agreements. Final reports are required for all standard and continuing grants, cooperative agreements and fellowships. Interim project reports are not required and are used to update the progress of a project any time during or before the award period expires

All submitted annual and final reports must be approved by an NSF Program Officer to meet the submission requirements.

Key features of Project Reporting System in Research.gov:

- ✓ A consolidated project reporting dashboard that includes annual, final, interim, and project outcomes report

- ✓ Ability to deposit published journal articles and juried conference papers in NSF Public Access Repository to be compliant with the Public Access requirement. PIs and co-PIs can also submit publications in the NSF's Public Access repository through their project reports and comply with the Public Access requirement.

- ✓ Upload multiple Products via BibTex upload feature

6.4 Project Achievement

- ✓ Ability to deposit published journal articles.

  - ○ Project Objective One: To support and educate individuals, families/carers and the service system, to identify, build and sustain natural supports around the goals and aspirations or the people they work with.
  - ○ Five individuals with disabilities (including psychiatric disabilities) and their families and carers, wanting to engage natural supports within their local, mainstream community will be identified and engaged in the project development.
  - ○ Consult with individuals, their families/carers about their fears and vulnerabilities in engaging natural supports; as well as the resources and interventions required to overcome these and achieve their vision and goals.
  - ○ Support these families with the knowledge, skills, and strategies and establish natural supports in their local communities.

  - ○ Achievements:

  - ○ 10 people with disabilities and four carers were interviewed about their experiences in engaging natural supports within their local mainstream communities.

  - An organisational perspective was sought from a mainstream group who actively engage with people with disabilities.
  - The interviews with project participants identified fears and vulnerabilities in engaging natural supports; as well as the resources and interventions required to overcome these barriers.
  - The Consumer Advisory Group supported and guided the project
  - The project report and tip sheets will assist people with disabilities and families and carers to build relationships with natural supports if they wish too.

- ✓ To strengthen local, natural supports to embrace the participation of individuals with disabilities through education and training.

  - Map the LoddonMallee Region to identify natural supports interested in supporting the inclusion of people with disabilities e.g. clubs, groups.
  - Identify the strengths and needs of these natural supports, including the knowledge and resources that would benefit their practice.
  - Create, develop and deliver training to natural supports, identifying a range of inclusive practices and strategies.

  - The project focused on an individual approach to accessing natural supports. In the light of this, it was decided that the development of a resource directory was not the best use of the project resources. This was for two reasons; the project focused on the individual and their
  - Experience: and the project was an opportunity to develop the materials that would assist people with disabilities and family careers to develop individualized resource directories. This reflects the philosophy of Disability Care Australia.

- Tip sheets and posters on accessing and using natural supports, tailored to the needs of people with disabilities, family careers, workers and organizations have been developed. These resources are available to be used on staff training.

6.5 Project Maintenance

Projects at any level and at any magnitude require careful planning to ensure success. A component of planning is preparing for risk and managing this risk if it occurs. Up to 80 percent of project risk comes from human resources; therefore, it's important for this risk to be adequately factored when planning projects.

Project risk management is a process by which risk is planned to increase the probability and impact of positive events, and decrease the probability and impact of events harmful to the project. All projects follow a similar path:

□□Initiation

□□Planning

□□Execution and control

□□Close-out

CHAPTER: 7

REFERENCES

1. The Mythical Man-Month: Essays on Software Engineering (Anniversary 2nd Edition) by Frederick Brooks, Jr.
2. The Art of Computer Programming by Donald Knuth
3. Introduction to Algorithms (3rd Edition) by Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein
4. Code: The Hidden Language of Computer Hardware and Software by Charles Petzold
5. Working Effectively with Legacy Code by Michael Feathers Peopleware: Productive Projects and Teams (3rd Edition) by Tom DeMarco and Timothy Lister
6. The Pragmatic Programmer: From Journeyman to Master by Andrew Hunt and David Thomas
7. Hello World! Computer Programming for Kids and OtherBeginners(http://helloworldbookblog.com), 2nd ed., Warren Sande and Carter Sande, Manning
8. Invent your Own Computer Games with Python (http://inventwithpython.com), 2nd edition, Al Sweigart
9. Python for Software Design: How to Think Like a Computer Scientist, Allen B. Downey, Jeff Elkner and Chris Meyers, Green Tea Press
10. Python Programming for the Absolute Beginner (http://www.cengage.com/search/pr...), 3rd ed., Michael Dawson, Course Technology
11. www.scrapy.org
12. www.djangoproject.com
13. www.Docs.djangoproject.com
14. Django-haystack.readthedocs.org
15. speakerdeck.com/pycon2014
16. tutorial.djangogirls.org/
17. showmedo.com/videotutorials/django

18. www.youtube.com/user/HackedExistence

19. django-marcador.keimlink.de/en/

20. support.google.com/merchants/answer/188494

21. www.shopzilla.com/

22. www.finder.com.au/

23. www.comparisonfocus.com/shopping/shopping-price-comparison-scripts/

24. www.ebaycommercenetwork.com/

25. www.become.com/

26. www.nextag.com/