# SERADAM

## Software Project-II

## Submitted By

| | |
|---|---|
| 16-32070-2 | Islam , Md  Nazmul |
| 16-32170-2 | Sakib , Ahmed Shahriar |
| 16-32020-2 | Saha , Jishnu |
| 16-32079-2 | Mazumder, Md. Abdullah Al Noman |

## Department of Computer Science

## Faculty of Science & IT

## American International University Bangladesh

## August 2019

# Declaration

We declare that this thesis is our original work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

_____
**Islam , Md Nazmul**
16-32070-2
SE

_____
**Sakib , Ahmed Shahriar**
16-32170-2
SE

_____
**Saha , Jishnu**
16-32020-2
SE

_____
**Mazumder, Md. Abdullah Al Noman**
16-32079-2
CSSE

# Approval

The thesis titled "**SeraDam**" has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science on (21st August 2019) and has been accepted as satisfactory.

_____
**VICTOR STANY ROZARIO**
Assistant Professor & Supervisor
Department of Computer Science
American International University-Bangladesh

_____
**MD AL-AMIN**
Lecturer & External
Department of Computer Science
American International University-Bangladesh

_____
**Dr. M.M Mahbubul Syeed**
Associate Professor & Head (Undergraduate)
Department of Computer Science
American International University-Bangladesh

_____
**Professor Dr. Tafazzal Hossain**
Dean
Faculty of Science & Information Technology
American International University-Bangladesh

_____
**Dr. Carmen Z. Lamagna**
Vice Chancellor
American International University-Bangladesh

# Acknowledgement

At first, we want to thank our almighty. We also want to thank our respectable parents for their kind and supports.

We want to thank our honourable supervisor Md. Shamsur Rahim sir. His outstanding supervision and support makes this project possible. We also thanks to our Head of Computer Science department Dr. Dip Nandi, Director of the Computer Science department Mashiour Rahman, Dean of FSIT Professor Dr. Tafazzal Hossain and our honourable Vice Chancellor, Dr. Carmen Z. Lamagna for their genuine support.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

On the other hand price comparison engines, better known as metasearch engines or simply metasearch, have been around since 1995.

In travel, price comparison sites have been in existence since the early 2000s. Early flight metasearch entrants were Skyscanner and Kayak, followed by Momondo and Google Flights. Hotel metasearch was also introduced in the early 2000s with early entrants such as Trip.com, SideStep.com (acquired by Kayak) and, over the past 13 years, Trivago, Momondo, TripAdvisor and Google Hotel Ads.

There are many e-commerce product based price comparison sites are out there. Such as – MySmartPrice, PricePanda etc.

Price comparison website or comparison shopping website acts like a search engine where shoppers can sort through products (based on category, price, features etc.) and compare it simultaneously. Once a customer decides to buy a product, he/she is redirected to the owner's website once they click on 'Buy'.

## 1.2 Objective

The objective of this project is to develop an e-commerce system where people can see a comparison list of prices of branded products from various retailers online. It is specialized in meta-searching and UX along with common features of an e-commerce system.

However, for implementation purposes, this paper will deal with a website where mainly electronics products specifically prices of laptop will be displayed and user can search and filter products and for extra feature they have to deal with basic registration-login system.

# Chapter 2: Statement of Work

## 2.1 Documentation History and Distribution

### Table 2-A: Revision History

| Revision | Revision Date | Description of change | Author(s) |
|---|---|---|---|
| 1.1.0 | 08.10.2018 | N/A | 1. Islam , Md Nazmul<br><br>2.Sakib,Ahmed Shahriar<br><br>3. Saha , Jishnu |
| 2.1.0 | 08.22.2018 | Added test plan | 1. Islam , Md Nazmul<br><br>2.Sakib,Ahmed Shahriar<br><br>3. Saha , Jishnu |

### Table 2-B: Distribution

| Recipient Name | Recipient Organization | Distribution method |
|---|---|---|
| VICTOR STANY ROZARIO | AIUB | Soft copy |

## 2.2 Purpose/Objective
- Analysis existing e-commerce metasearch application
- Improve UX via existing metasearch application
- Figure out the requirements and add features
- Improvise some features and find proper solution

## 2.3 Anticipated benefits
- Online system
- User friendly design
- More customization  and features in the user-end
- Faster comparison result

## 2.4 Software Technology proposed

| | |
|---|---|
| Programming Language | Python, javascript |
| Front end Design | Reactjs |
| OS | Windows 7/8/10 |
| Database | Mongodb |
| Project Management | Trello, Slack |
| SDLC model | Rapid Application Development(RAD) |
| Back end framework | Django |

## 2.5 Customer Impacted
- General people

## 2.6 Requirements
- Device- Desktop/laptop/tablet/phone
- Internet connection

## 2.7 Deliverable include in scope

- Full software

- Technical documentation

- User manual

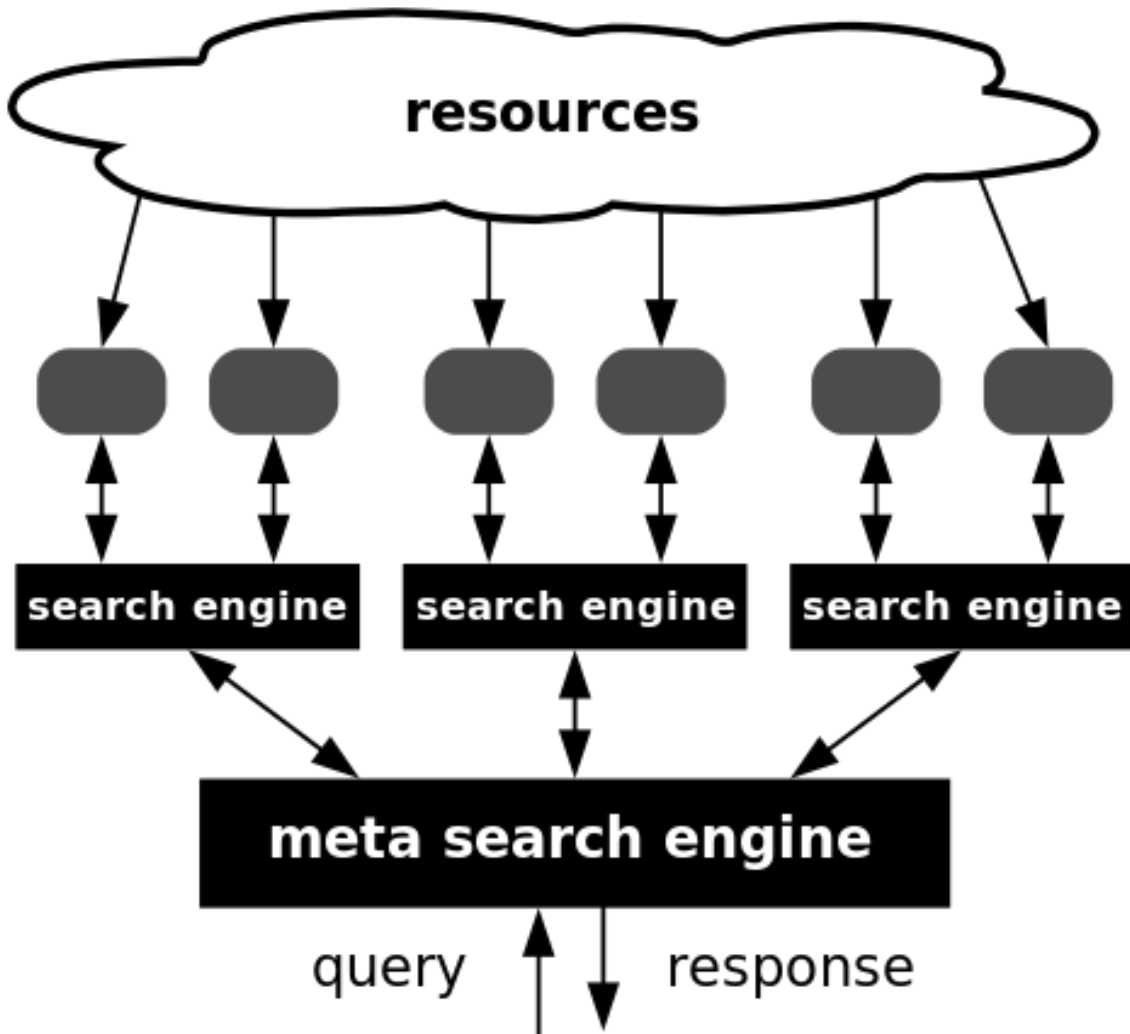# Chapter 3: Related Work Study

## 3.1 Meta-search Engine:

**Fig 3-1: Meta-search Engine**

A metasearch engine (or aggregator) is a search tool that uses another search engine's data to produce its own results from the Internet. [1][2] Metasearch engines take input from a user and simultaneously send out queries to third party search engines for results. Sufficient data is gathered, formatted by their ranks and presented to the users.

## 3.2 Existing E-commerce websites:

## 3.2.1 MYSMARTPRICE



**Fig 3-2: MYSMARTPRICE**

This website provides electronics products' comparison price list from various e-shop such as flipcart, paytmmall,tatacliq [3]

## 3.2.2 TRIVAGO



**Fig 3-3: Trivago**

Trivago's search tool scans hotel booking sites for prices, availability, images and reviews within seconds. When users choose a hotel, they are redirected to a partner website to complete the booking [4]

## 3.2.3 TRIPADVISOR



**Fig 3-4: TripAdvisor**

TripAdvisor is an American travel and restaurant website company that shows hotel and restaurant reviews, accommodation bookings and other travel-related content. It also includes interactive travel forums. [5]

# Chapter 4 : Study on SDLC

## 4.1 What is SDLC?

Software development life cycle SDLC is the systematic approach to complete the software development process within the time and maintain quality of the software. System development life cycle provides the set of activities to be carried out during the system development and it is often called that software development life cycle. Software development is divided into set of activities that allow any software development company to control the software product easily. The software development life cycle models use the step by step approach to complete the software development process [6]
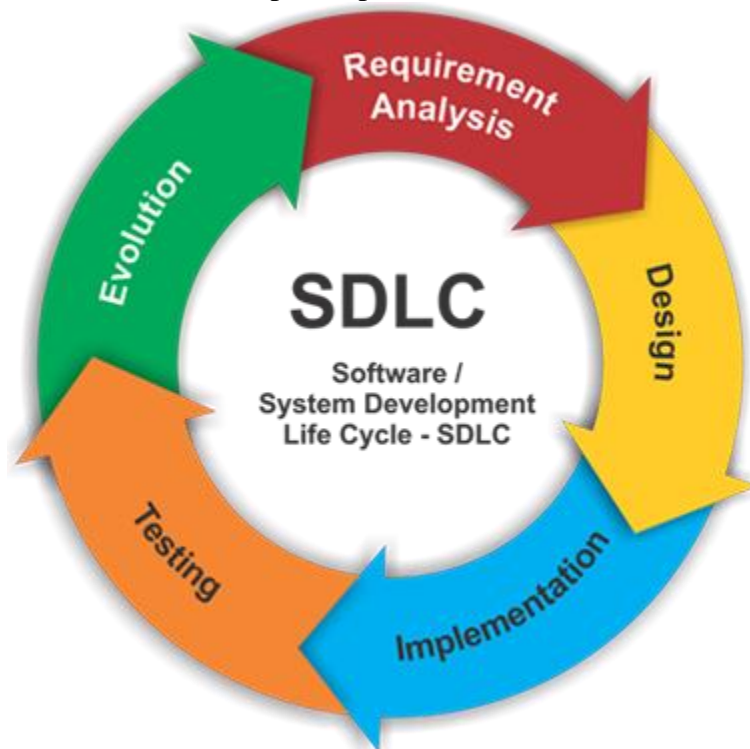


**Fig 4-1: SDLC Model**

# 4.2 THE SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

## 4.2.1 The Waterfall Model

This model was proposed by Royce in 1970. This is the classical model of the software engineering. The model is one of the oldest models used in the government projects and in many of the major companies. The model is also called as the linear sequential model or classic life cycle. This is the linear sequential SDLC model followed by the following phases Requirement Analysis, Design, Implementation, Testing and Maintenance . The phases are fixed in such a manner the phase once cannot be repeated again. As the waterfall model is the classical model so it serves as the baseline of all other models. The waterfall model consists of several non overlapping stages as shown in the figure bellow. It is one way street with no turning back. You cannot move backward once phase "X" is over, you can move to phase "Y". There is no provision of moving backward.

## 4.2.2 The V- Model

This model can be considered as extension of waterfall model. In waterfall model we move in a linear way while in V shaped model process steps are bent upwards the coding phases to form typical V-Shape. The relationship between each phase of the development process and its associated phase of testing. As testing is one of the important part of Software development the V shaped model emphasis is more on testing. The V shaped Model is almost same as waterfall model both the models are sequential. Requirements have to be very clear before the project starts because it is very expensive to go back to requirements and make changes for the existing projects.

## 4.2.3 The Spiral Model

The spiral model was developed on the end of the 1980s; it was outlined by Barry Boehm, and introduces something that other models did not take into account, which is risk analysis. In essence, the spiral model attempts to bring together key aspects of some other prominent models (namely the waterfall, incremental, and evolutionary prototyping), in an attempt to gather the most appropriate traits from each one, because specific projects might be more or less adaptable to specific models.

## 4.2.4 The Prototyping Model

Prototype model is the evolutionary process model. Evolutionary Process models are the iterative type models using this model the developer cam develop increasingly more complete version of software. In prototype model here, we have the quick design phase through which the developer has to quickly design the software after requirement gathering as shown in figure.

## 4.2.4 The Prototyping  Model

Prototype model is the evolutionary process model. Evolutionary Process models are the iterative type models using this model the developer cam develop increasingly more complete version of software. In prototype model here, we have the quick design phase through which the developer has to quickly design the software after requirement gathering as shown in figure.

## 4.2.5 The Iterative Waterfall   Model

The problems with the classical waterfall model created a demand of new model. The iterative model came into existence to cope with the problems of the original waterfall model the iterative waterfall model is enhanced version of classical waterfall model which could provide faster results require less time and heaving good flexibility. In iterative model project is divided into small parts and this allows developer team to go easily and quickly for their goal and obtain their valuable feedback from users. The project that is divided into small parts here each part is actually a mini waterfall process.

## Agile Model :

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frame) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

The most popular Agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995),XP etc.

## 4.3 Our Model: RAD (Rapid Application Development)

RAD or Rapid Application Development process is an adoption of the waterfall model; it targets at developing software in a short span of time. RAD follow the iterative [7]

RAD model has following phases

- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation
- Testing and Turnover
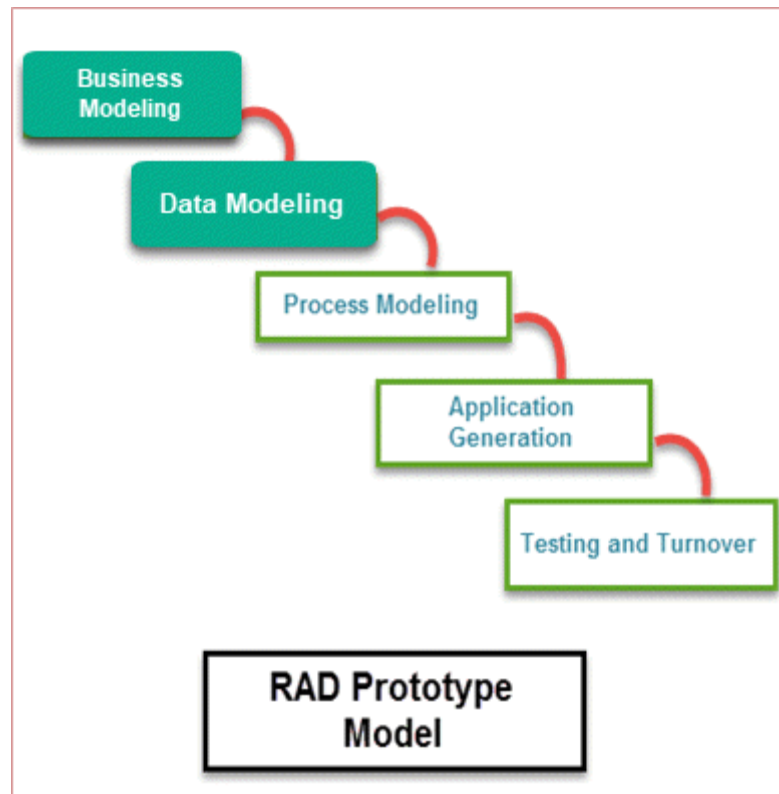
**Fig 4-2: RAD SDLC Model**

## 4.4 Why RAD Model :

It focuses on input-output source and destination of the information. It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects. The main features of RAD model are that it focuses on the reuse of templates, tools, processes, and code.
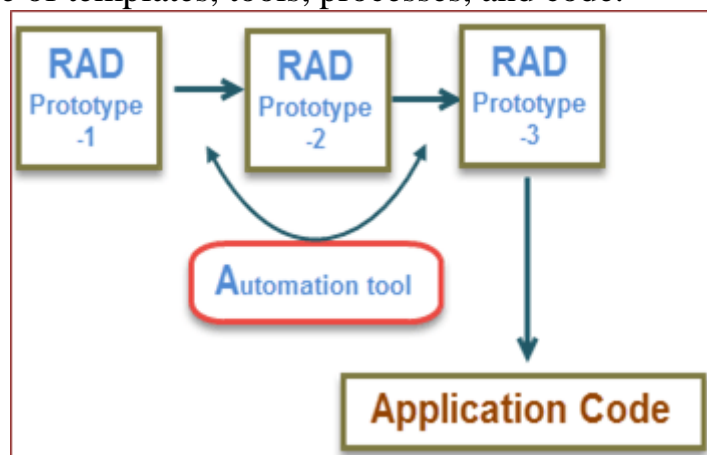


**Fig 4-3: RAD Structure**

**Advantages :**

- Flexible and adaptable to changes
- It is useful when you have to reduce the overall project risk
- It is adaptable and flexible to changes
- It is easier to transfer deliverables as scripts, high-level abstractions and intermediate codes are used
- Due to code generators and code reuse, there is a reduction of manual coding
- Due to prototyping in nature, there is a possibility of lesser defects
- Each phase in RAD delivers highest priority functionality to client
- With less people, productivity can be increased in short time

**When to use RAD Model:**
- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When there is a necessity to create a system that can be modularized in 2-3 months of time
- When a budget is high enough to afford designers for modelling along with the cost of automated tools for code generation.

# Chapter 5 : Software Requirements Specification

## 5.1 Project summary:
### 5.1.1 Field Study:

We choose to develop a price comparison website. Right now, there are very few websites in the world, which provide this type of service. In our country, there is no services like this. Price comparison websites such as trivago, shopzilla, tripadvisor, Kayak etc. are providing services in specific sector. However, in our case, we are building a system where various sectors such as electronics, home appliances etc. sectors will be covered.

### 5.1.2 Problematic field:
- There are no reliable price comparison website in Bangladesh.
- We have to check various website for the same products, which requires time.
- We need to keep browsing and be updated for a specific product for different websites.
- We have to bookmark manually for specific products for different websites.

### 5.1.3 Background to the problem:
Most of the existing websites belong from abroad. On the top of that, there are few reliable websites out there, which deal with such type of services. With the rise of E-commerce in our country, consumers have urge to check prices for a specific product in different websites, as there are many e-commerce sites out there. This takes a user lot of time to pick the product that he/she desired to buy in proper budget. Moreover, there is no service out there like tracking a specific product price.

## 5.1.4 User story:

### Table 5-A: User Story

| Use Case | Actor | Corresponding user story |
|---|---|---|
| View search results | User | As a user, I want to view search results. |
| Display all search results | System | As the system , I want to display all search results |
| Search specific branded products | User | As a user, I want to search specific branded products |
| Search specific site products | User | As a user, I want to search specific site products |
| Search specific category products | User | As a user, I want to search specific category products |
| Filter specific category | User | As a user , I want filter specific category |
| Filter specific brand | User | As a user , I want filter specific brand |
| Filter specific site | User | As a user , I want filter specific site |
| Filter specific price range | User | As a user , I want filter specific price range |
| Sort searched products | User | As a user, I want to sort searched result. |
| View details of products | User | As a user, I want to view the details of the products |

| View all the deals of products | User | As a user, I want to view various deals of the products |
|---|---|---|
| View photos of products | User | As a user, I want to view the photos of the products |
| Get notification on price change | User | As a user , I want to get notification if the price changes |
| Add product to wish list | User | As a user , I want to add products to my wish list |
| Redirect to website when click on the deal | User | As a user, I want to redirect to the parent website of the product when I click the "Buy" button |

### 5.1.5 User story summary:

Our main concern is to provide a platform where a user can view a price comparison list for various products based on the prices from various e-retailers. User can search and filter specific products and check basic info of the products and all the deals included. He/she can also get notification on price change of the desired product.

## 5.2 Project Scope:

**In scope:**
- User registration
- Log in
- Update profile
- Display all products
- View products
- Add to wish list
- View product details
- Get notification
- Edit profile
- Change password
- Password recovery

**Out of scope:**
- Statistics for admin panel
- Automation service for parsing product info

## 5.3 Exclusive features:

- Receive notification on product status change
- Live chat for customer service

### 5.3.1 Operating Environment:

The project will run in any operating system

### 5.3.2 Dependencies:

- Desktop, tab, laptop, mobile etc. devices.
- Internet connection

## 5.4 System description

In design specification plan there will be some system diagrams, some software UI screenshots, architecture plan, test plan and system overview.

## 5.5 System Overview

## 5.5.1 Use Case
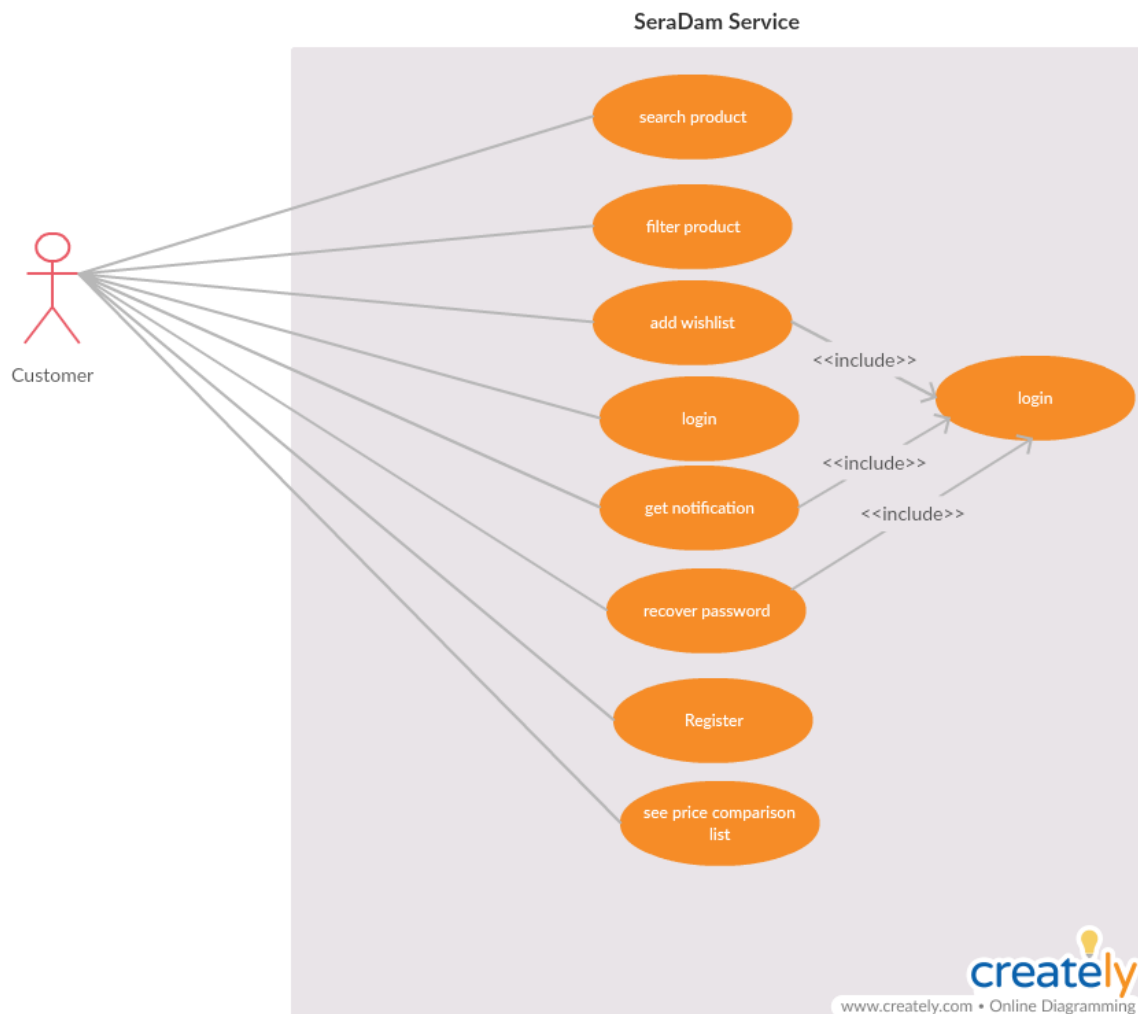


**Fig 5-1: Use case diagram for Seradam service**
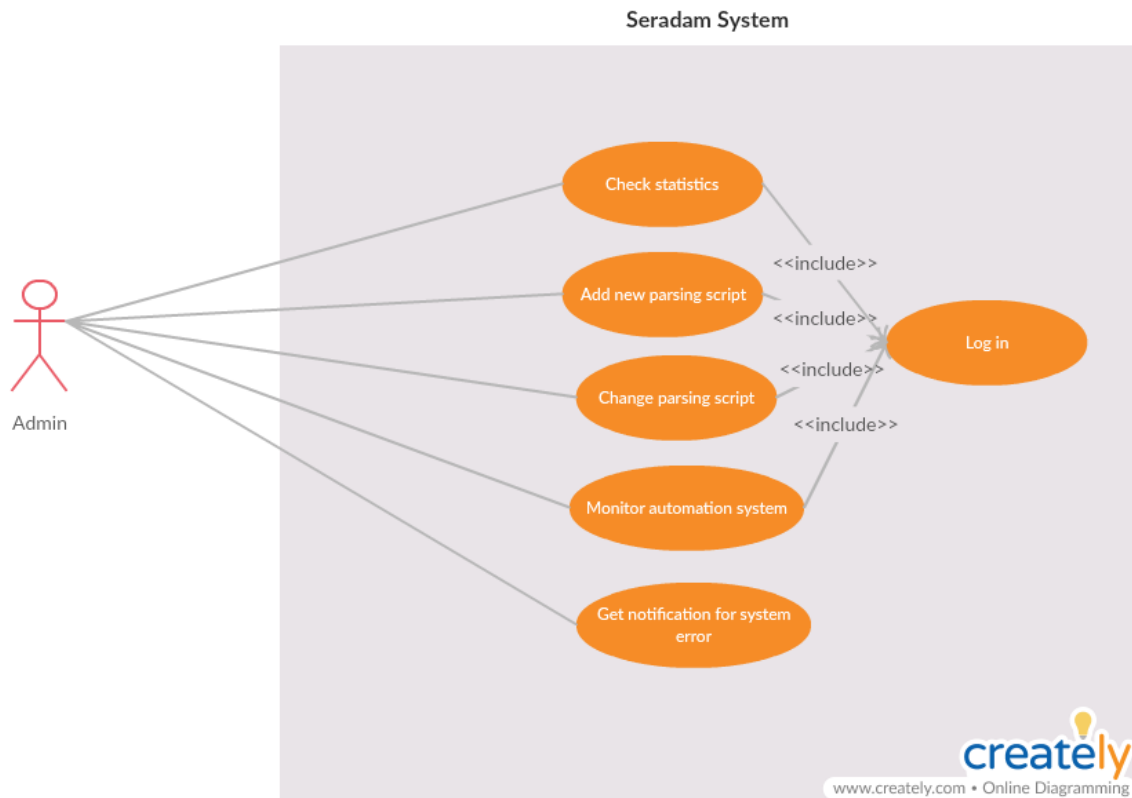
**Fig 5-2: Use case diagram for Seradam system**

**5.5.2 Activity diagrams:**
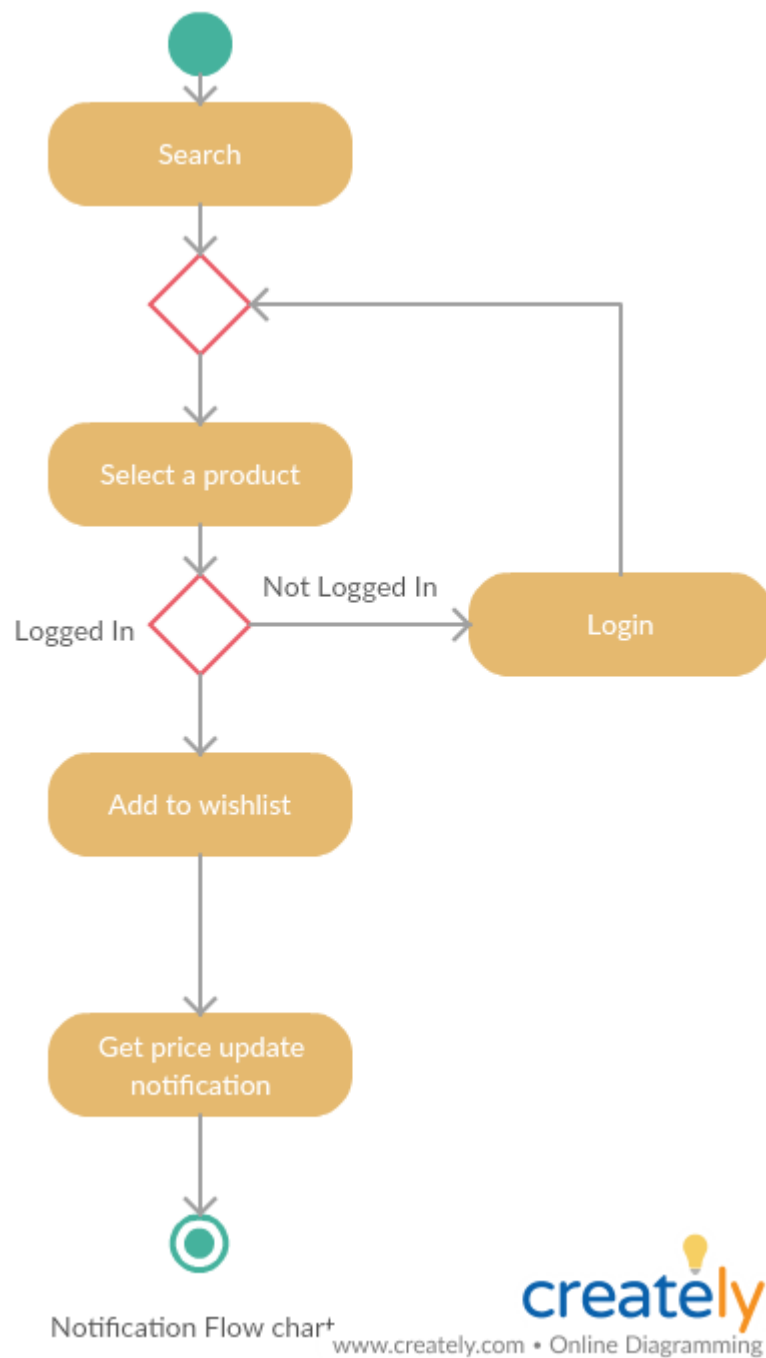


**Fig 5-3: Activity diagram for Seradam notification system**

## 5.5.3 Entity Relationship Diagram:
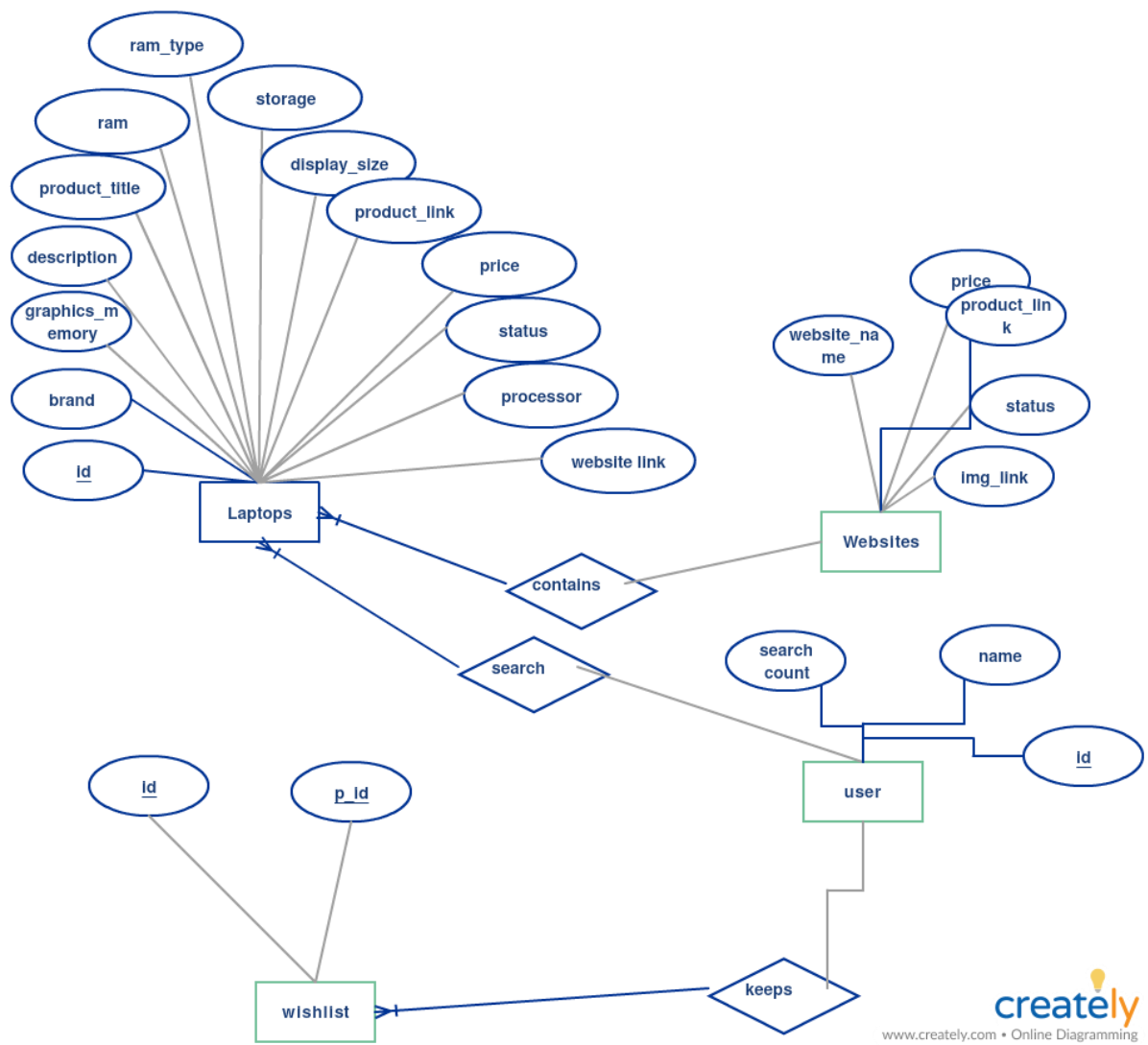


**Fig 5-3: ER diagram for Seradam system**

## 5.6 Human Interface Design [UI Design]:

## User-End:

### 5.6.1 Landing page:
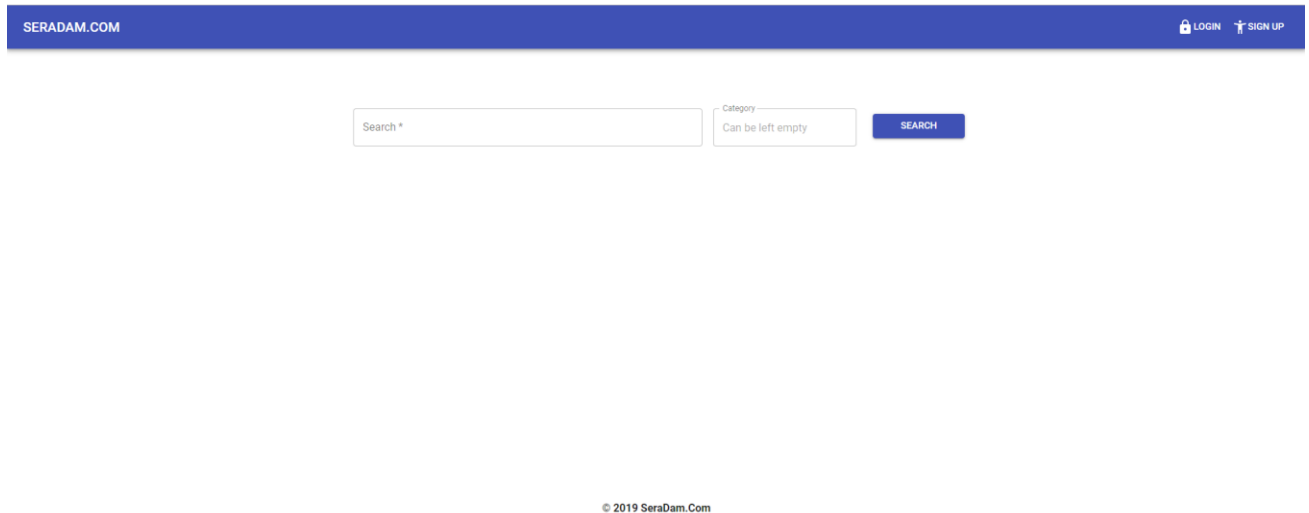


**Fig 5-4: Seradam landing page**

### 5.6.2 Search Result page:
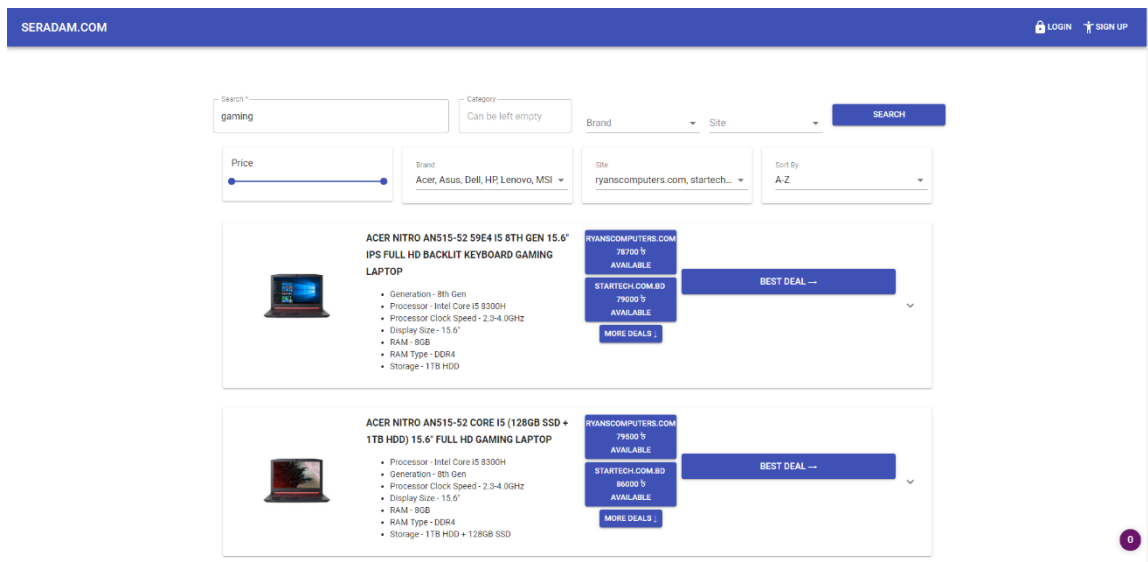


**Fig 5-5: Seradam search result page**

### 5.6.3 Price filter:
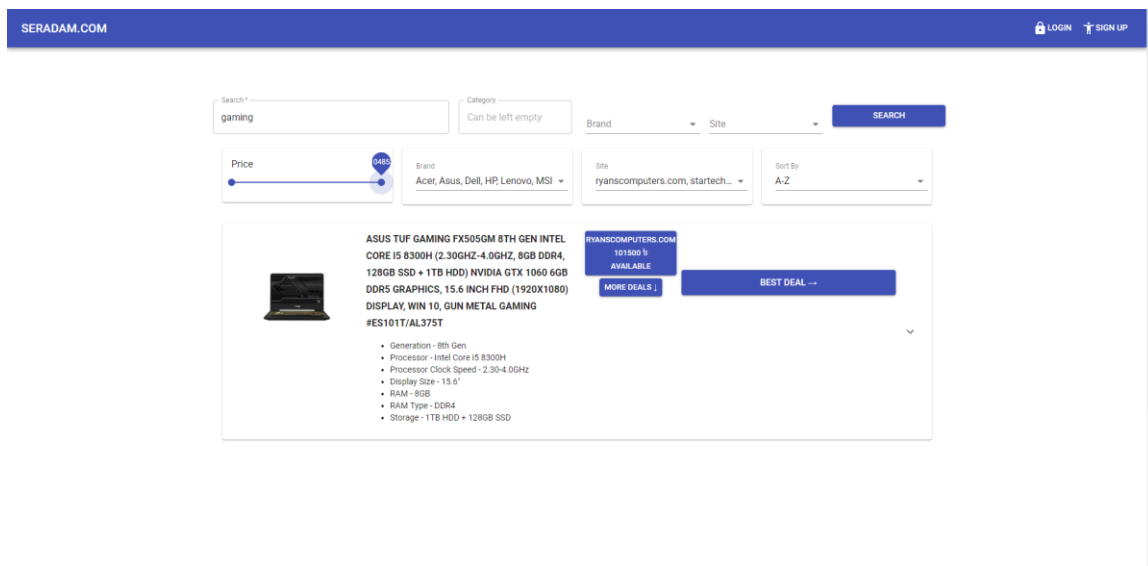


**Fig 5-6: Seradam price filter**

### 5.6.4 Brand filter:
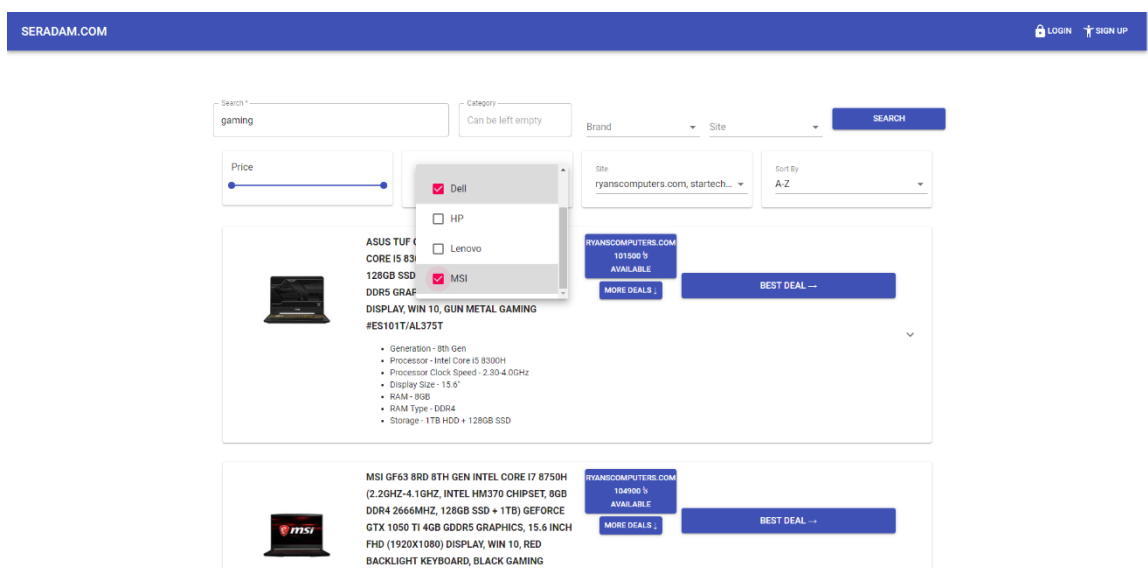


**Fig 5-7: Seradam brand filter**

### 5.6.5 Website filter:
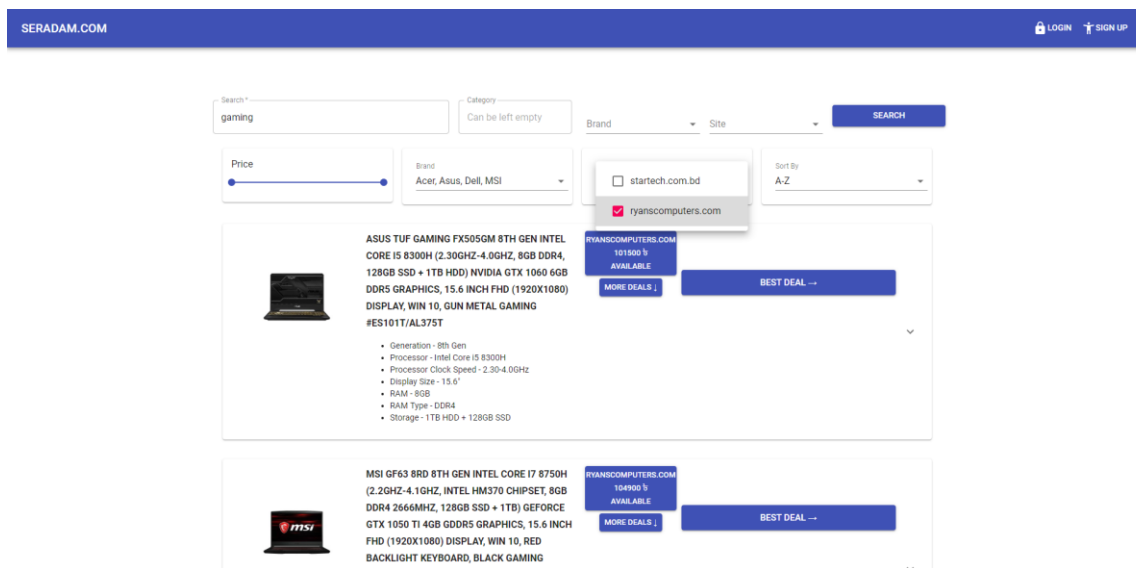


**Fig 5-8: Seradam site filter**

### 5.6.6 Sign up/registration pop-up:
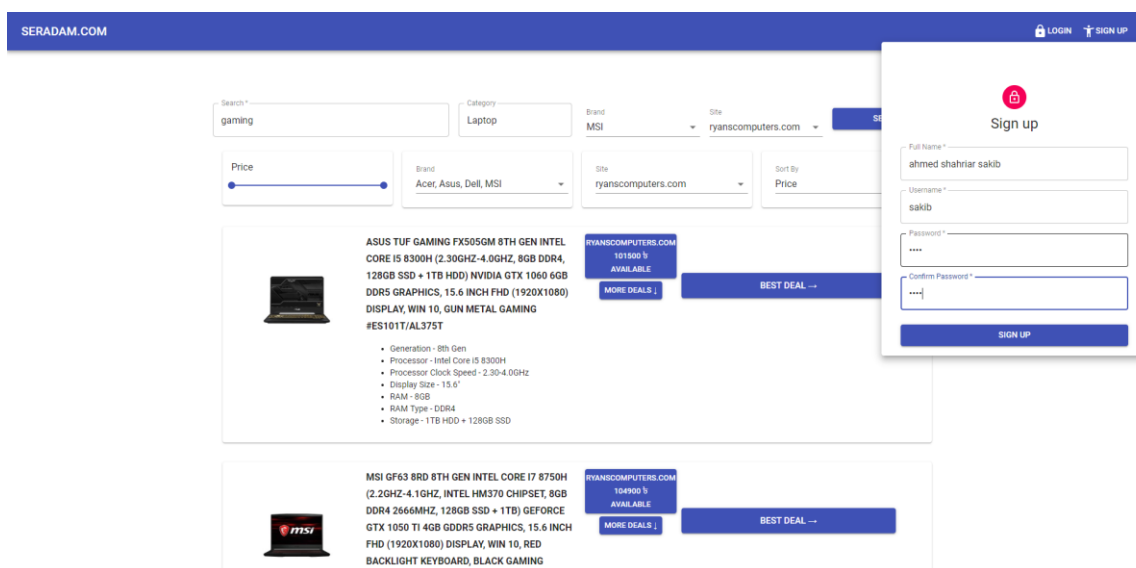


**Fig 5-9: Seradam sign up**

## 5.6.7 Login Pop-up:
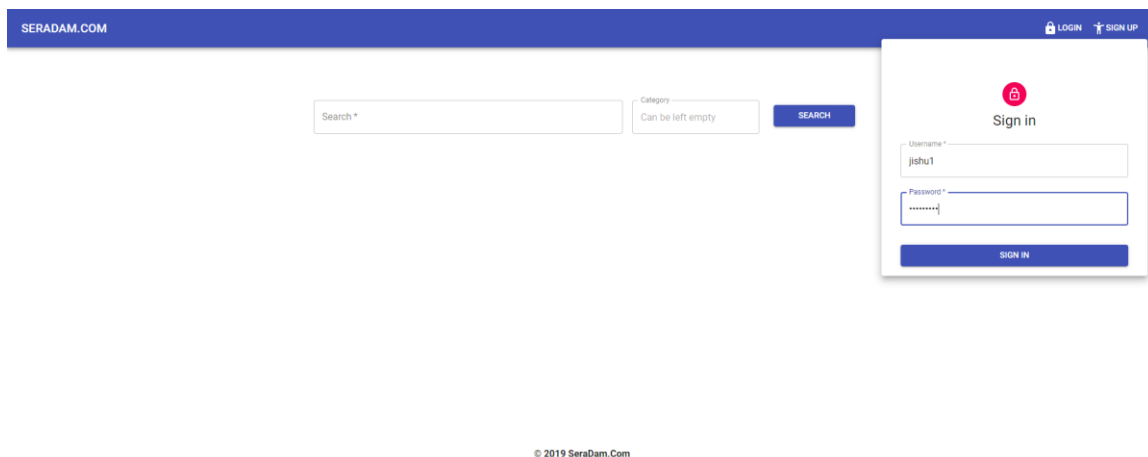


**Fig 5-10: Seradam login**

## 5.6.8 Expand/collapse Product details (Section-Info):
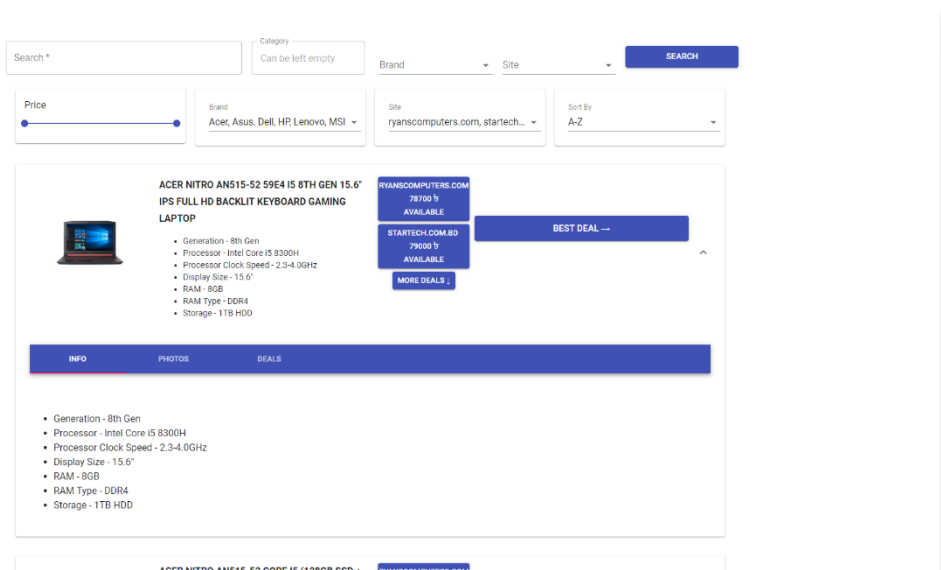


**Fig 5-11: Seradam product details**

### 5.6.9 Expand/collapse product details (Section-photos):
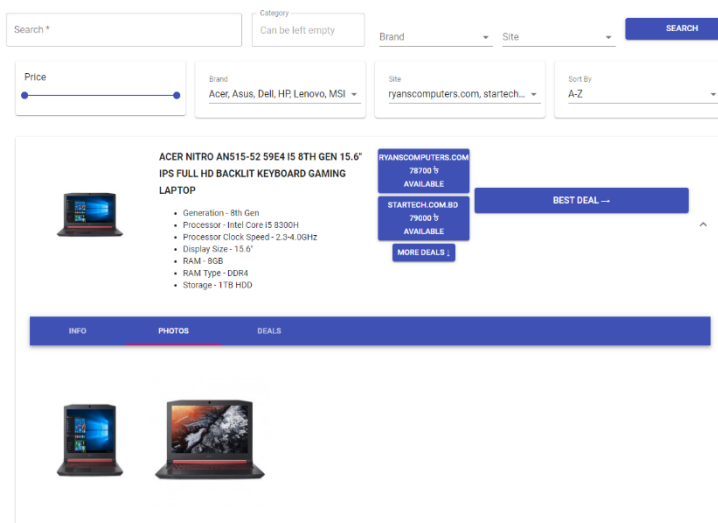


**Fig 5-12: Seradam more info (photos)**

### 5.6.10 Expand/collapse product details (Section-More deals):
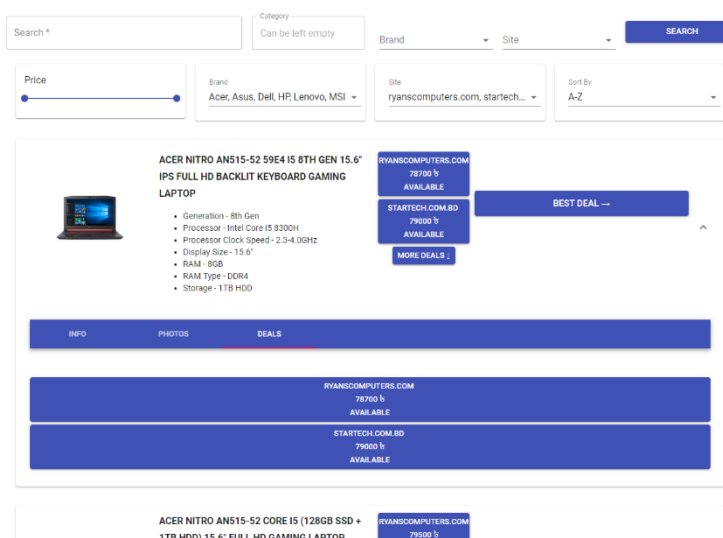


**Fig 5-13: Seradam more info (deals)**

## 5.6.11 Add to wish-list section (Logged in feature):
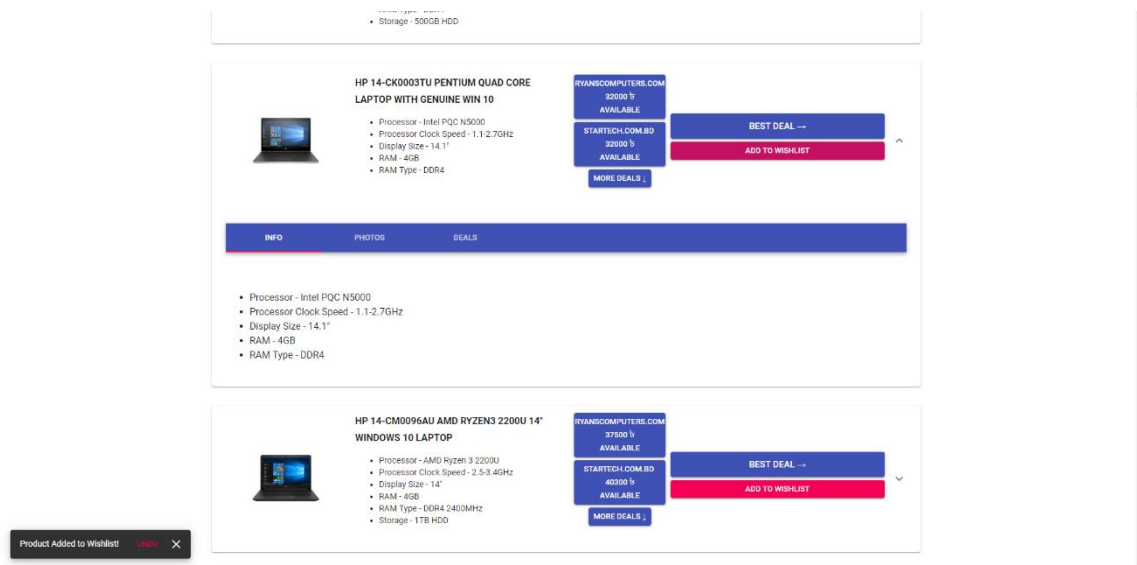


**Fig 5-14: Seradam wish list feature**

## 5.6.11 Wish list:
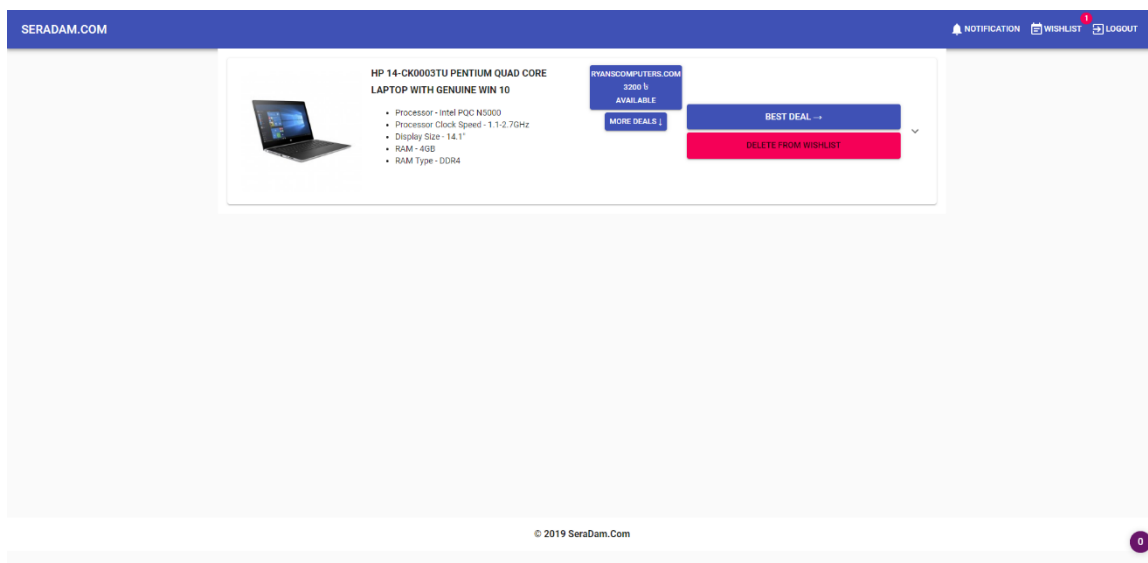


**Fig 5-15: Seradam wish list**

## 5.6.12 Notification:
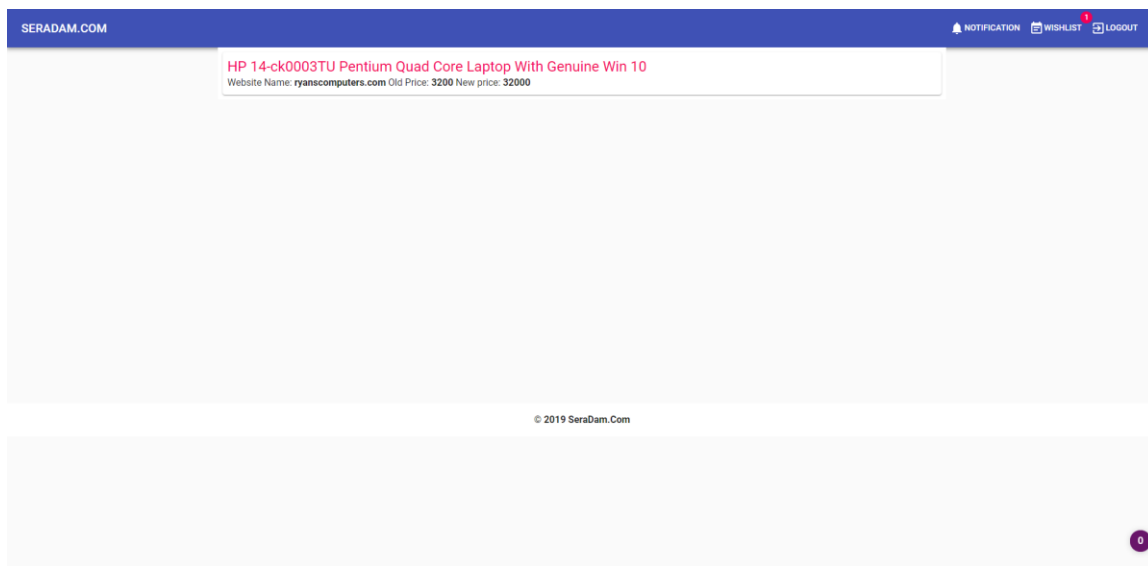


**Fig 5-16: Seradam notification**

# Admin-End:

## 5.6.13 Basic info:
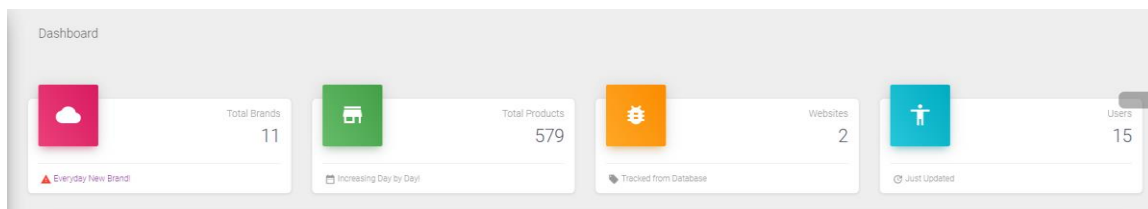


**Fig 5-17: Seradam admin panel basic info section**

## 5.6.14 User statistics:



**Fig 5-18: Seradam admin panel user statistics section**

## 5..15 Search hits statistics:
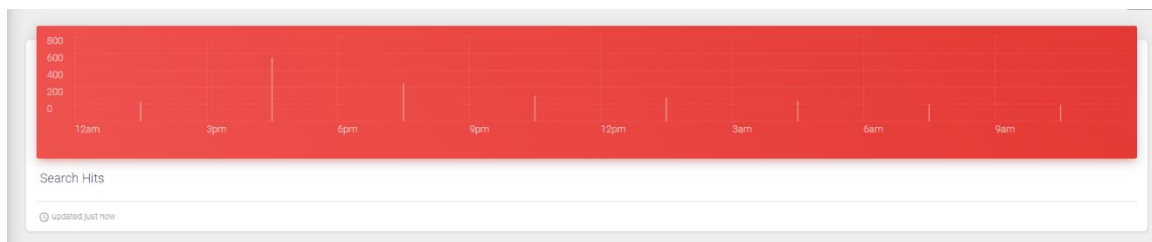


**Fig 5-19: Seradam admin panel search hits graph**

## 5.56.16 Active user statistics:



**Fig 5-20: Seradam admin panel active users graph**

## 5.6.17 User registration statistics:



**Fig 5-21: Seradam admin panel registration graph**

## 5.6.18 Admin Landing Page:
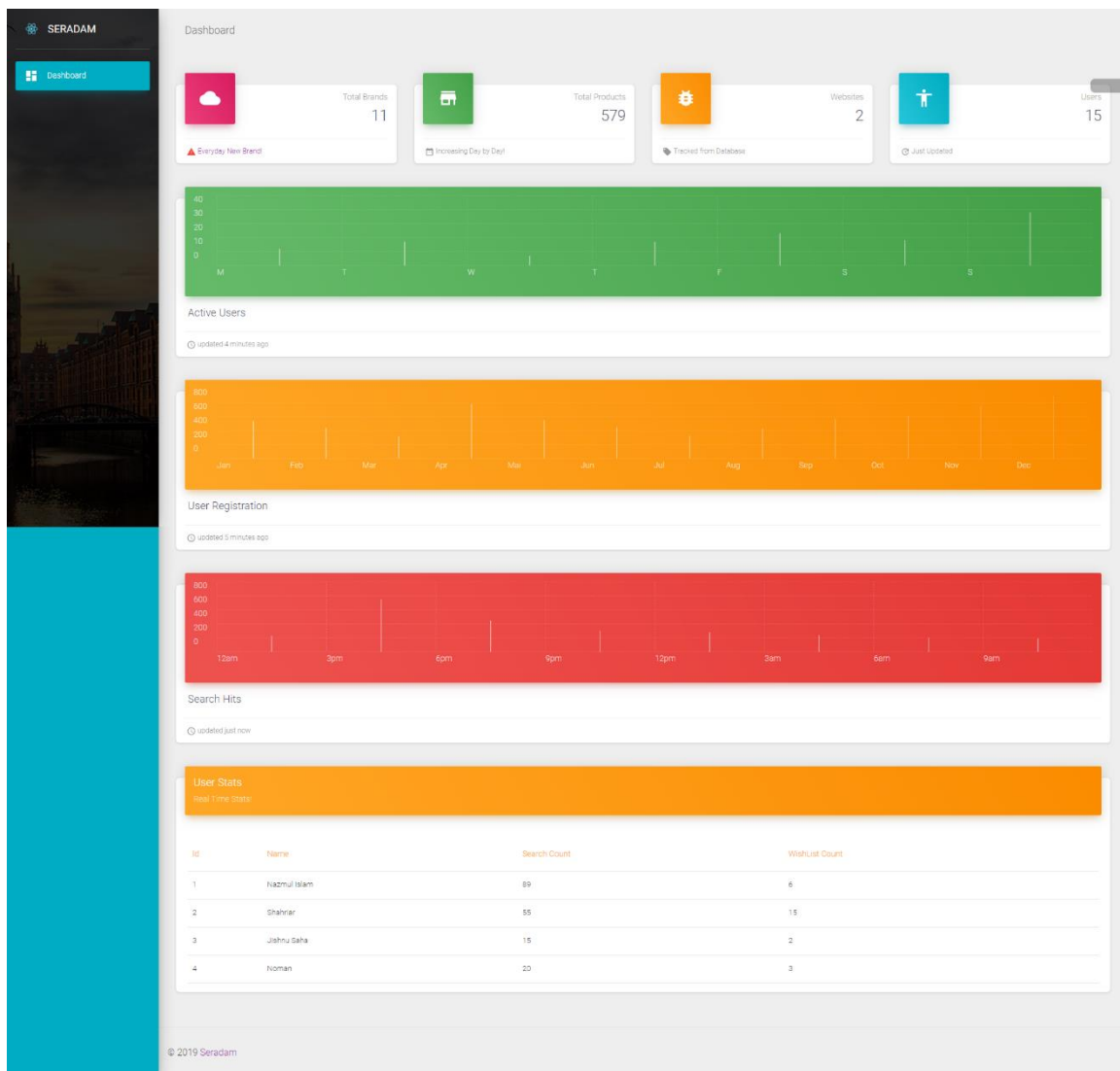


**Fig 5-17: Seradam admin panel landing page**

# Chapter 6 : Project Management

## 6.1 Work Breakdown Structure (WBS)

The WBS for this project is aligned with the System Development Life Cycle rather than focusing on the project management process groups. This WBS will help the team to understand the overall picture of what is going to happen next, without having to be familiar with the project management processes. [8]
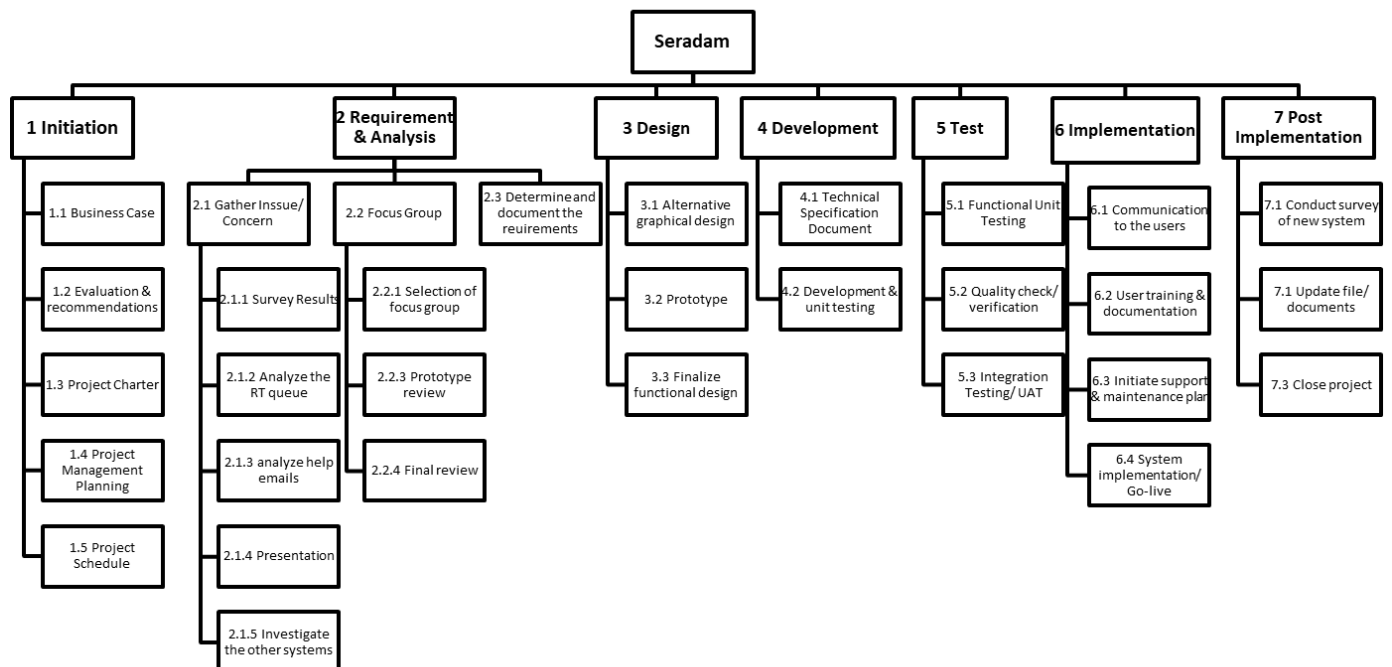


**Fig 6-1: Seradam WBS**

## 6.2 Quality Gate:

**Table 6-1: Quality gate**

| Work Product | QA Method |
|---|---|
| 1.S/w development project plan | Words spellcheck & inspection |
| 2.S/w requirements specification | Formal technical review |
| 3.Design document | Formal technical review |

| | |
|---|---|
| 4.Schedule | Formal technical review |
| 5.Testing plan document | Formal technical review & words spellcheck |
| 6.Coding | Code inspection |
| 7.System testing | Test coverage measurement |
| 8.Alpha testing | One month testing in local server |
| 9.Beta testing | One month testing after live |

## 6.3 Project Scheduling

**Table 6-2: Project schedule**

| Task | Time (Week) |
|---|---|
| Field study | 0-2 |
| Study on previous works | 2-4 |
| Study on SDLC | 1-3 |
| Prepare user story | 3-5 |
| Identify user requirements | 5-6 |
| Develop Use case diagram | 6-7 |
| Develop Activity diagram | 6-7 |
| Develop E-R diagram | 6-8 |
| Prototyping | 8-10 |
| Create Database | 11-12 |
| Create user interface | 13-14 |
| Software Development | 15-21 |
| Software Testing and Debugging | 21-24 |
| Documentation | 19-24 |

**Total Project Time: 24 Weeks**

**Start Date: 5 March 2019**

**Release Date: 20 August 2019**

## 6.4 Milestones

**Table 6-3: Milestones**

| Serial No | Major Milestone | Date |
|:---:|:---:|:---|
| 1 | Planning phase | 05th March |
| 2 | Requirement analysis | 21th April |
| 3 | Preliminary design | 11th May |
| 4 | Final design | 20th May |
| 5 | Design prototype | 10th June |
| 6 | Coding | 25th July |
| 7 | System testing | 05th August |
| 8 | Release | 20th August |

## 6.5 Staffing Plan

**Table 6-4: Staffing Plan**

| Name | Role | Backup |
|:---:|:---:|:---:|
| Jishnu | Project co-ordinator & developer | Sakib |
| Noman | Developer | Nazmul |
| Nazmul | Designer & developer | Noman |
| Sakib | Project Manager & developer | Jishnu |

## 6.6 Monitoring and Controlling Mechanism

Status meeting conducted bi-weekly (task: compare actual start date with planned start date)

To estimate our progress and take corrective action we will use Earn Value Analysis (EVA) method.

## 6.7 Risk Management

| ID: risk01 | Date : 23/07/19 | Probability : 69% | Impact : Medium |
|---|---|---|---|

Description: The website DOM might change while parsing which will give error product info

Mitigation plan: Create an alert system whenever parsing gives an error and save it in the log file.

Status: Initial identification

| ID: risk02 | Date : 30/07/19 | Probability : 80% | Impact : Medium |
|---|---|---|---|

Description: Live website might not work properly while parsing
Mitigation plan: Create a separate database while parsing is active and shift the database
Status: Initial identification

| ID: risk03 | Date : 25/07/19 | Probability : 80% | Impact : Medium |
|---|---|---|---|

Description: Website might not work properly when new parsing script is updated
Mitigation plan: Shift the database with the old one while running new script for parsing
Status: Initial identification

| ID: risk04 | Date : 23/07/19 | Probability : 90% | Impact : High |
|---|---|---|---|

Description: When an e-retailer website is down the parser cannot parse through that website and give backdated result.
Mitigation plan: Remove that e-retailer website from the parsing script
Status: Initial identification

| ID: risk05 | Date : 26/07/19 | Probability : 40% | Impact : Low |
|---|---|---|---|

Description: System does not provide real time parsing hence, cannot provide real time prices of the products.
Mitigation plan: Create an automation method for parsing script, which runs it once every six (6) hours.
Status: Initial identification

## 6.8 List of Deliverables

**Table 6-5: Deliverables**

| Deliverables | Date |
|---|---|
| SRS | 20th March |
| Analysis document | 25th April |
| System design | 11th May |
| Final design | 20th May |
| Test plan | 25th July |
| Test manual | 01th August |
| Test cases | 05th August |
| User manual | 20th August |

## 6.9 Schedule Tracking Process

We will update the Gantt chart considering changes and other circumstances.

## 6.10 Metrics

This process plan is to collect metrics-

**Table 6-6: Metrics**

| Metrics | Quantity |
|---|---|
| Actual task completion time | 180 days |
| Total no of defects | 70 approx |
| No of defects/KLOC | 4 |
| No of pages of documentation | 30 |
| No of total inputs | 15 |
| No of total outputs | 10 |
| No of total files | 80 |

## 6.11 Post-mortem

**Review Meeting**

There has been a meeting after every major phase, a post-mortem review and after the project there has been another one.

**Purpose**

To collect good and bad practices for further improvement

## 6.12 Project Management Tool (Trello)
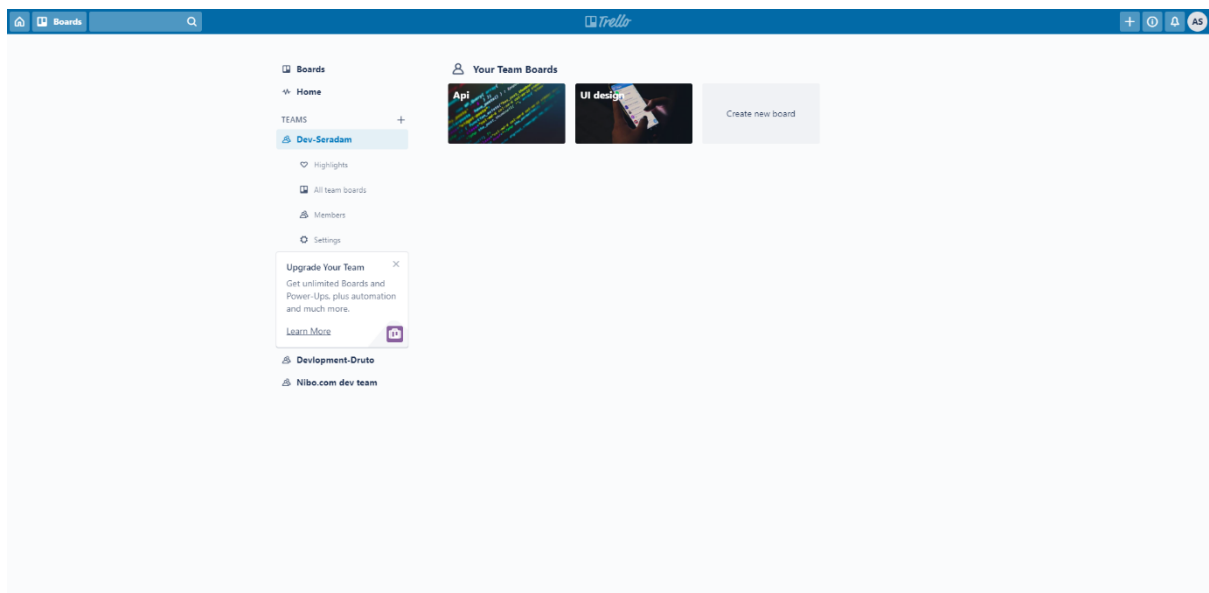


**Fig 6-2: Trello(project view)**
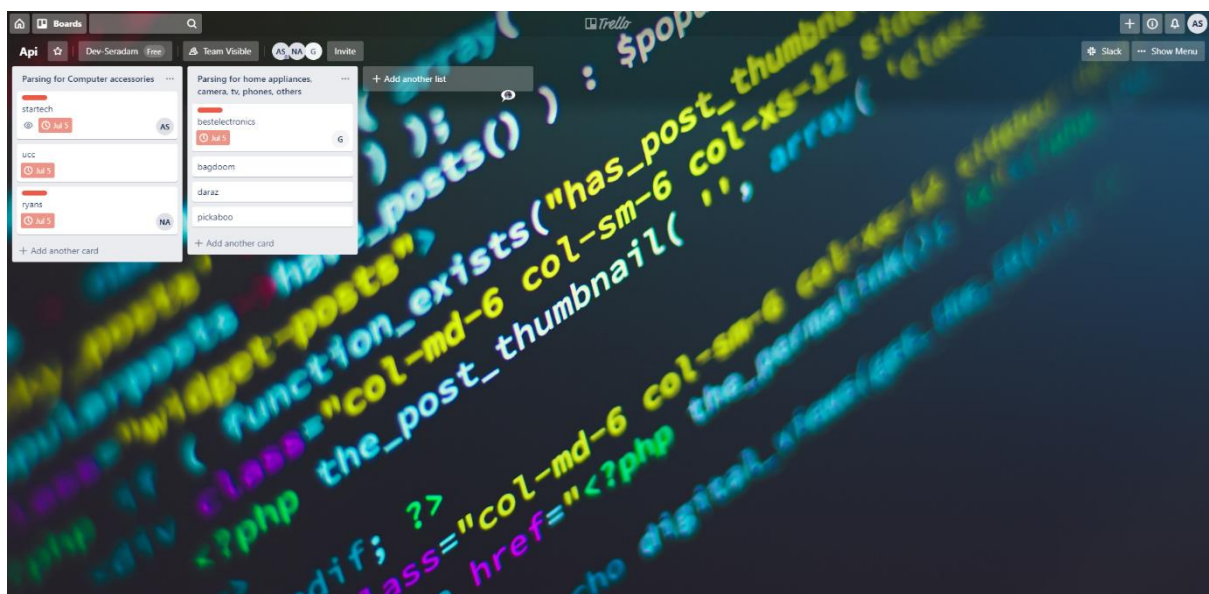


**Fig 6-3 : Trello(API )**
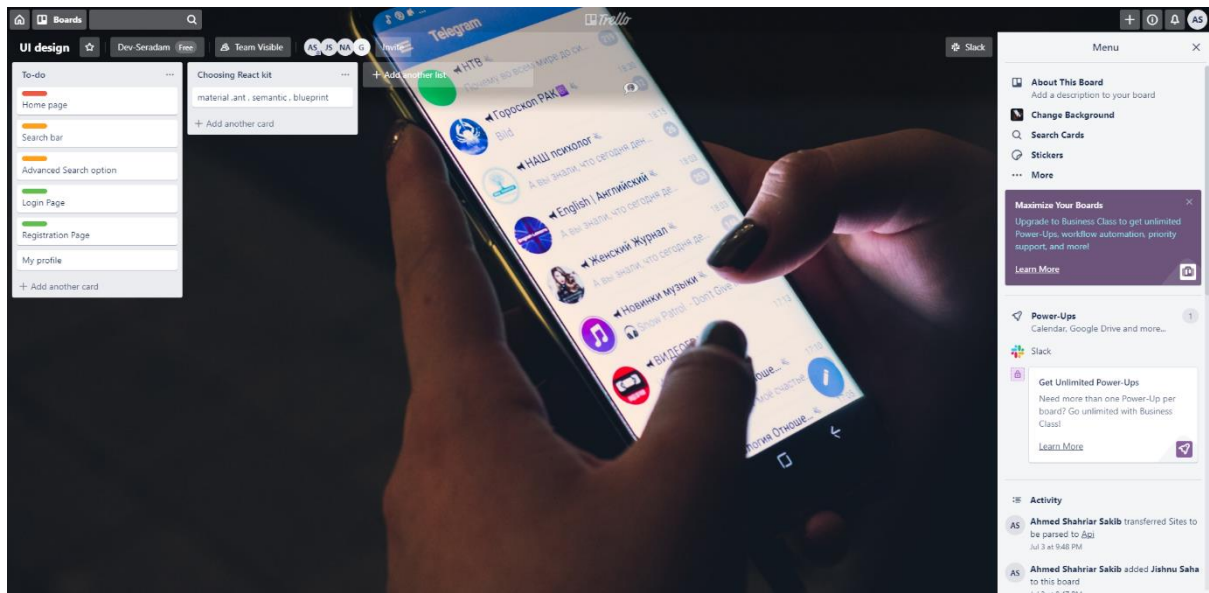
**Fig 6-4 : Trello(Front end)**

## 6.13 Project Management Communication Platform(Slack)
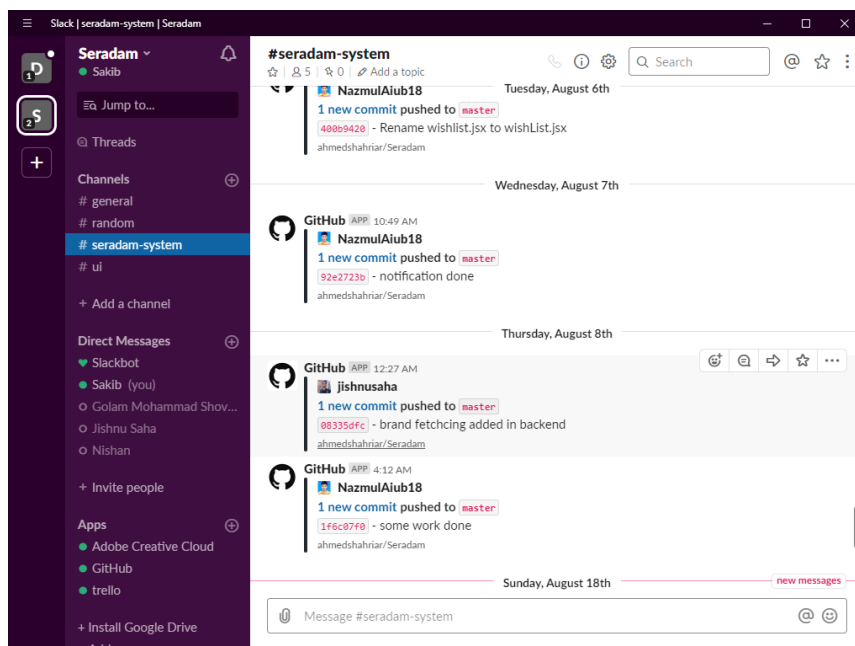


**Fig 6-5: Slack**

## 6.14 Project Management (Version control – git and github)

We used Github as our online repository. Using Git, we used the repository from everyone's computer.

Some sample screenshots of relate works are given:
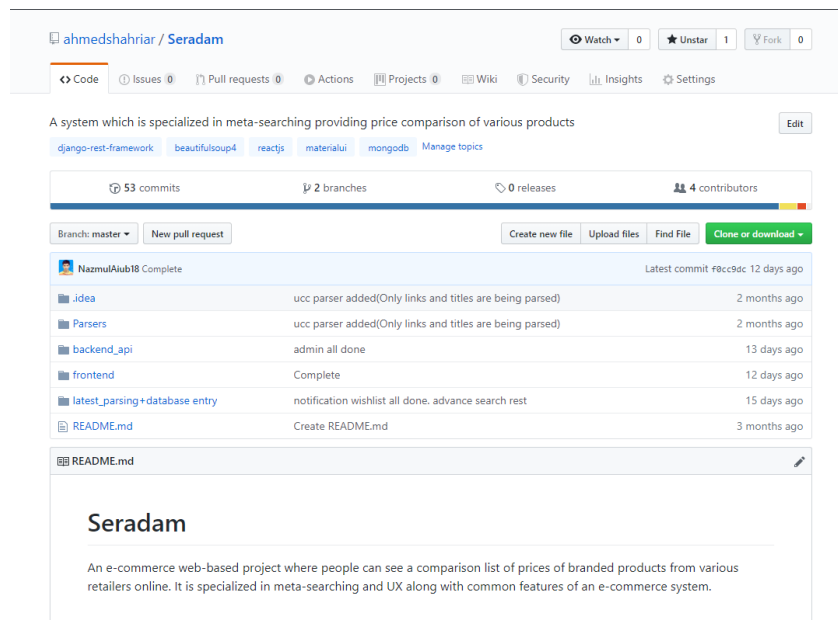


**Fig 6-6: Github repository folders**
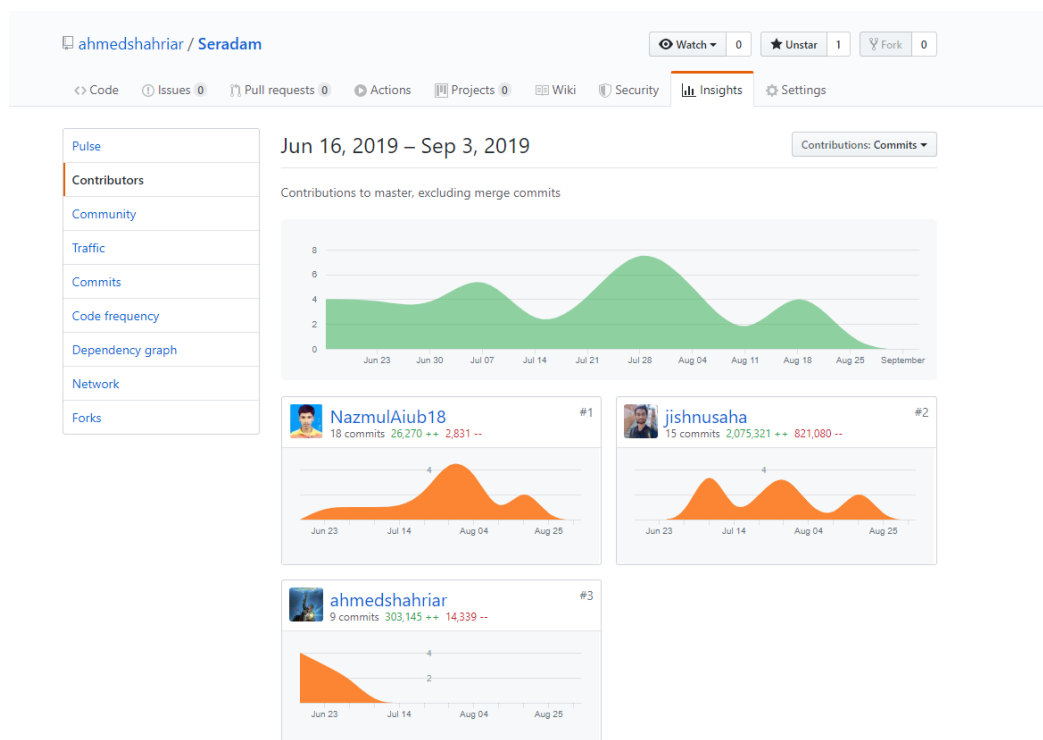


**Fig 6-7: Github repository contributors' timeline**

# Chapter 7 : Testing

## 7.1 Test Plan:

### 7.1.1 Identifier:

Seradam -V1.001.

Author: Jishnu Saha [16-32020-2], Nazmul Islam [16-32070-2], Ahmed Shahriar Sakib [16-32170-2] Mazumder, Md. Abdullah Al Noman [16-31079-1].

### 7.1.2 References:

- Requirements Specifications
- Design Details
- Methodology Guidelines

### 7.1.3 Overview:

For Seradam system, this test plan is based on the demands and functionalities and design a scheme that can meet all the demands. The project will have three testing stages, Unit, System / Integration and Acceptance.

### 7.1.4 Purpose:

The objective of this test plan is

### 7.1.5 Scope:

**The scope of the project is:**

Sign up module:  This will help the users to register from anywhere through internet.

Login module:  Users who have the valid id and password only they can log in to the system.

User module:  This section covers search, sort, filter, notification, and wish list

Functionalities for the users.

Admin module: This section covers admin panel functionalities.

### 7.1.6 Test Items

- Login Module, Version 1.001
- User Module, Version 1.001
- Registration Module, Version 1.001
- Admin Module, Version 1.001

### 7.1.7 Software Risk Issues:

1. **Critical Areas:**

   - Interruption of a third party Software.
   - Errors while searching
   - Handle many users.
   - Database switching during automated parsing

2. **Complexities:**

   - Reliability.
   - Server load
   - Updated DOM of e-retail shop.

### 7.1.8 TEST DELIVERABLES

- Test cases
- Test plans for different kind of testing
- Problem reports and corrective actions
- Error logs and execution logs.
- Detailed test plan document
- Test design specifications.
- Simulators
- Tools and their outputs
- Problem reports and corrective actions
- Statistical documents

## 7.1.9 Test Plan

## 7.1.9.1 Register Account

### Table 7-1: Test Plan for Register Account

| No | Testing module | Case | Coverage |
|---|---|---|---|
| 1 | Register Account | Check all the fields | Check the boxes with valid/invalid of input |
|   |   |   | Store data in the database with successful registration |
|   |   |   | Registration without typing required data |
| 2 |   | Check the button | Sign Up button navigation |
|   |   |   | Register Button navigation |
|   |   |   | Check button's properties |
|   |   |   | Check the pop up is working upon clicking sign up button |
| 3 |   | Check register functionalities | System process time and functionality after completing successful registration |

## 7.1.9.2 Login Account

### Table 7-2: Test Plan for log in to account

| No | Testing module | Case | Coverage |
|---|---|---|---|
| 4 | Login Account | Check all the fields | Check the boxes with valid/invalid of input |
|   |   |   | Store data in the database with successful registration |
|   |   |   | Login without typing required data |
| 5 |   | Check the button | Login button navigation |
|   |   |   | Submit  Button navigation |
|   |   |   | Check button's properties |
|   |   |   | Check the pop up is working upon clicking sign in button |
| 6 |   | Check login functionalities | System process time and functionality after completing successful registration |

## 7.1.9.3 User

**Table 7-3: Test Plan for user**

| No | Testing module | Case | Coverage |
|----|----------------|------|----------|
| 7 | Search product | Check all the fields ,boxes, options and functionalities | Check the boxes with valid/invalid of input |
| | | | Search without typing any data |
| | | | |
| 8 | Filter product | Check all the fields ,boxes, options and functionalities | Check the boxes with valid/invalid of input |
| | | | Filter without typing any data |
| | | | Check different dynamic filter options |
| 9 | Add to wish list | Check button and functionalities | Check the button navigation |
| | | | Check button after pressed state |
| 10 | Get notification | Check button and functionalities | Notification functionality after change in the database |
| 11 | | | |

## 7.1.9.4 Admin

**Table 7-4: Test Plan for Admin**

| No | Testing module | Case | Coverage |
|----|----------------|------|----------|
| 12 | Check statistics | Check the graphs, data table and functionalities | Check the real-time change of graph value |
| | | | Check the data table functionalities |

## 7.2 Test Case

### Table 7-5: Test Plan for Admin

| Test Case # | Action | Requirement # | Expected Results | Actual Results | Pass/Fail | Comments |
|---|---|---|---|---|---|---|
| TC_1 | Browse to landing page | 110.0 | Landing page should contain a text box for searching , a select box and a search button | Landing page should contain a text box for searching , a select box and a search button | Passed | The test case passed. |
| TC_2 | Click the Seradam title in the navigation bar | 111.1 | It will redirect to the Seradam landing page | Nothing happens while clicking the title. | Failed | Missing redirect to Landing page when click on the title |
| TC_3 | Clicking on 'Search' When search box is empty | 111.2.1 | Clicking on the button 'Search' brings to search result page containing a message "No result" | Clicking on the link 'Home' brings to search result page but does not display "No result" | Failed | Does not display "No result" after an empty search |
| TC_4 | Clicking on 'Search' when field isn't empty | 111.2.2 | Clicking on the button 'Search' brings to search result page contains result | Clicking on the button 'Search' brings to search result page contains result | Passed | The test case passed. |
| TC_5 | Clicking on 'Category' | 111.2.3 | Clicking on the category component displays multiple | Clicking on the category component displays multiple | Passed | The test case passed. |

| | | | | category option | category option | | |
|---|---|---|---|---|---|---|---|
| TC_6 | Clicking on 'Brand' beside category | 111.2.4 | Clicking on the category component displays multiple brand option | Clicking on the category component displays multiple brand option | Passed | The test case passed. |
| TC_7 | Clicking on 'Site beside category | 111.2.5 | Clicking on the category component displays multiple site option | Clicking on the category component displays multiple site option | Passed | The test case passed. |
| TC_8 | Clicking on 'Search' selecting all previous options | 111.2.5 | Clicking "Search" refreshes the search result page with new result | Clicking "Search" refreshes the search result page with new result | Passed | The test case passed. |
| TC_9 | Clicking on 'Search' selecting all previous options | 111.2.5 | Clicking "Search" refreshes the search result page with new result | Clicking "Search" refreshes the search result page with new result | Passed | The test case passed. |

All test cases has been conducted using selenium and we used POSTMAN to test the API.

# Chapter 8 : Conclusion & Future Work

The focus of the project was to deliver a working product in this case a product comparison website within the schedule. The project is maintained with SDLC and other practices. New technology was used such as Django for parsing and other back-end functionality for robustness, mongodb for scalability and reliability and reactjs for front to make it more customizable.

Future work of the project concludes –

- To improve the API
- Parse more e-retailers' website
- More categorical product.
- Improve filter and search options
- Add more filter options
- Develop a mobile app for the system.

# References

[1] Sandy Berger's Great Age Guide to the Internet by Sandy Berger. Que Publishing, 2005, ISBN 0-7897-3442-7.

[2] "Architecture of a Metasearch Engine that Supports User Information Needs". Proceedings of the Eighth International Conference on Information Knowledge Management. 1999.

[3] MySmartPrice - https://www.mysmartprice.com/

[4] Trivago - https://www.trivago.com/

[5] TripAdvisor - https://www.tripadvisor.com/

[6] What is the Software Development Life Cycle (SDLC)? Available: https://airbrake.io/blog/sdlc/what-is-the-software-development-life-cycle

[7] What is RAD Model? Advantages & Disadvantages? Available: https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html

[8] WBS: https://uwaterloo.ca/ist-project-management-office/methodologies/project-management/planning/work-breakdown-structure

[9] POSTMAN - https://www.getpostman.com/

[10] GitHub. Available: https://github.com/

[11] Trello: https://www.trello.com/

[12] Slack: https://slack.com/

[13]Hotel metasearch - https://www.phocuswire.com/End-of-hotel-metasearch

[14] Shopping comparison sites - https://indianonlineseller.com/2014/09/10-best-shopping-comparison-engines-increase-ecommerce-sales/

[15] Shopping search engines - https://www.bigcommerce.com/blog/shopping-search-engines/