BSCS-F18-015

03-134152-028 MANSOOR AHMAD

03-134151-052 BILAL BIN MAJID

Domain Specific Search Engine For Mobile Phone Review Aggregation

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Munaza Sher

Department of Computer Sciences

Bahria University, Lahore Campus

January 2020

Certificate

We accept the work contained in the report titled

"DOMAIN SPECIFIC SEARCH ENGINE FOR MOBILE PHONE REVIEW

AGGREGATION"

written by

MANSOOR AHMAD

BILAL BIN MAJID

as a confirmation to the required standard for the partial fulfilment of the degree of

Bachelor of Science in Computer Science.

Approved by:

Supervisor: Munaza Sher

_____   (Signature)

January 27, 2020

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134152-028	MANSOOR AHMAD	
03-134151-052	BILAL BIN MAJID	

Date   :   January 27, 2020

Specially dedicated to

My beloved parents, friend and my teachers

(Mansoor Ahmad)

My beloved parents, friends and my teachers

(Bilal-Bin-Majid)

ACKNOWLEDGEMENTS

DOMAIN SPECIFIC SEARCH ENGINE FOR MOBILE PHONE REVIEW AGGREGATION

ABSTRACT

Domain specific search engine for mobile phone reviews aggregation is a web-based project which gathers valid reviews of different mobile phones from websites. The reviews include both positive and negative reviews. Once all the valid reviews are gathered, they are aggregated and shown on the web application. Statistics and ratings of the mobile phone will be shown based on the reviews that have been gathered.

The basic focus is to target what the people have to say about the mobile phone to form the statistics rather than what the manufacturer has to say about the device. The search engine reduces the time require to get the complete statistics and valid reviews of the device in one single web application. We will be using the concept of data scraping to gather all the valid reviews and the search engine would be python based. After scraping the data our focus would be to remove all the irrelevant reviews using NLP API.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

DSSEMPRA   Domain specific search engine for mobile phone review aggregation

API   Application Programmable Interface

ORM   Object Relational Mapping

JSX   Java Script XML

XML   Extensible Mark-up Language

UI   User Interface

REST   Representational State Transfer

NPM   Node Package Manager

JSON   Java Script Object Notation

HTML   Hyper Text Mark-up Language

URL   Uniform Resource Locator

WWW   World Wide Web

HTTP   Hyper Text Transfer Protocol

NLP   Natural Language Processing

JS   Java Script

UC   Use Case

CHAPTER 1

INTRODUCTION

1.1   Background

The project is a web-based search engine which also consists of techniques of data scraping to gather valid reviews to form the statistics and ratings of the mobile phone. A search engine is a software system designed to search information. The information may be a mixture of web pages, images and other types of files. Some search engines also mine data available in databases or open directories. Our search engine provides the user to search for a mobile phone. After the search is complete the desired mobile phone with its specifications, ratings and reviews will be displayed on the user's screen. Data scraping is used for extracting data from websites. Data scraping software access the WWW directly using HTTP or through a web browser. Web scraping can either be done manually by a software user or automatically by using a bot or a web crawler. It is a form of copying in which specific data is gathered and saved to a JSON file. In scraping first, the page is fetched using a web crawler and then data is extracted from it using web scrapers. An automated scraping technique would be used to gather the required data.

The scraped reviews will further be filtered by using NLP. NLP is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, how to program computers to process and analyse large amounts of natural language data.

1.2   Problem Statements

Whenever a person wants to buy a new mobile phone it takes a lot of time to go to different website and see the reviews to make up his mind. This website allows the purchaser to see all the relevant information of different websites aggregated into one single platform which will reduce the time span to make up his/her mind thus saving a lot of time. Getting all the relevant data in one place also help the user. The search engine will allow the purchaser to find the phone he wants to buy.

1.3   Aims and Objectives

To develop a website that shows the rating, statistics and valid (positive and negative) reviews of mobile phones based on the reviews gathered from different sites using techniques of data scraping.

1.4   Scope of Project

The domain specific search engine is a web-based project that would collect and aggregate all valid reviews both positive and negative by removing all spam reviews, into one search engine-based website. All those aggregated reviews will then be displayed on the website along with the valid reviews. Data scraping using spiders and sentiment analysis of reviews will be done using Text Blob for Relevant comments. A search engine-based website will be created.

1.5 Tools & Technologies

1.5.1 Back End

Data Scraping:

- The actual extraction of data / information from a web page
- Uses HTML Tags to scrap data.
- Using web crawling to surf between pages.
- Scrapy a python-based framework is used.

Figure 1.1: Data Scraping

Django Framework:

- A python-based open source web framework.
- The framework emphasizes reusability and "pluggability" of component and the principle of don't repeat yourself.
- Provides with administrative features via Django Admin.
- Allows using ORM
    - Creating objects-based data models.
    - Each object contains lists of attributes.
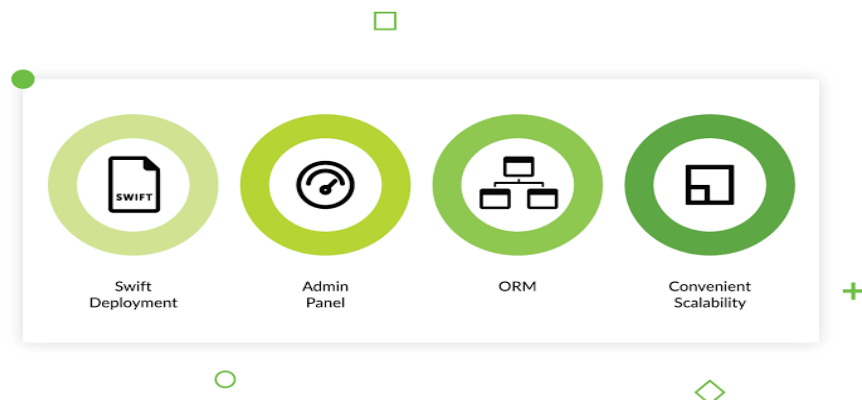
Figure 1.2: Django Framework

Text Blob:

- A Python (2 and 3) library for processing textual data.
- Provides a simple API for diving into common natural language processing tasks.
- Used for sentiment analysis of Reviews.
- Uses Polarity & Subjectivity of textual data to categorize it as positive or negative.
- Works through the Django framework.
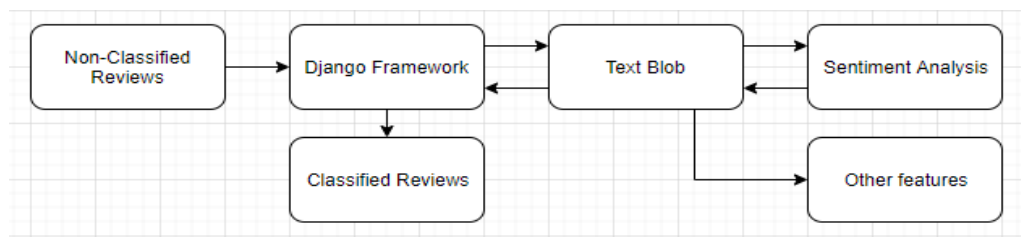
Figure 1.3: Text Blob

Node JS:

- Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications.
- Provides a rich library of various JavaScript modules which simplifies the development of web applications.
- Provides with environment for React.JS to work.
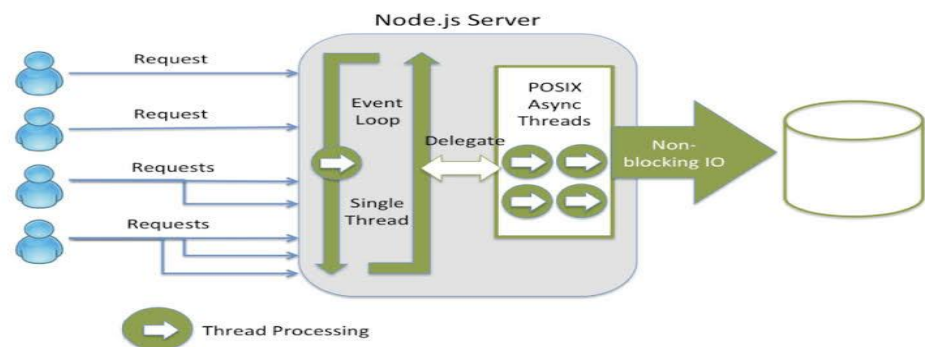
Figure 1.4: Node JS

1.5.2   Front End

REACT JS:

- React is a declarative, efficient, and flexible JavaScript library for building user interfaces.
- It lets you compose complex UI's from small and isolated pieces of code called "components".
- React Routers are used to run the components.
- React components are made using JSX.
- React works on top of the Node environment.
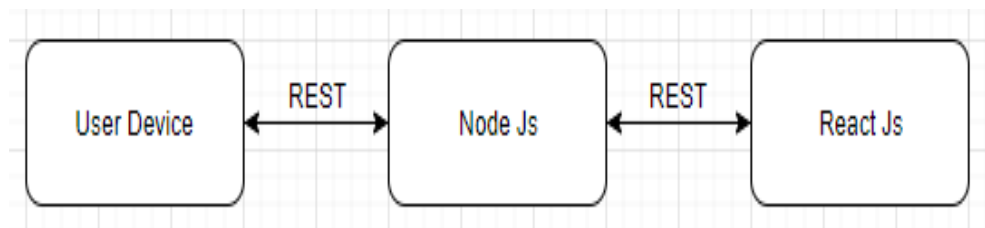
Figure 1.5: React JS

1.6 Workflow
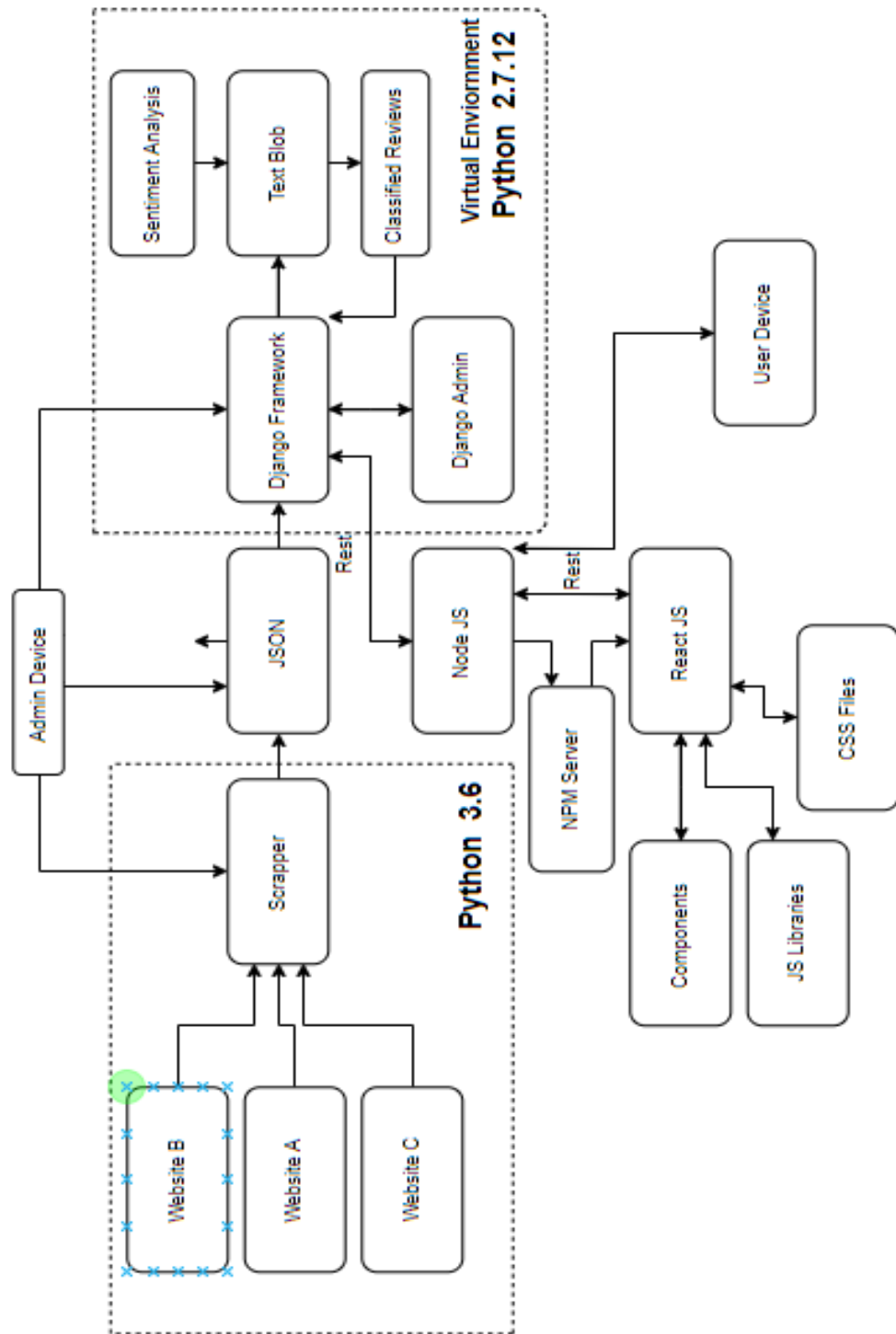
Figure 1.6: Workflow

CHAPTER 2

LITERATURE REVIEW

2.1 User Classes and Characteristics

2.1.1 Admin

- Add/Delete Sub-Admin.
- Add/Delete Announcements.
- Add/Delete Groups
    - Add/Remove Privileges.
- Add/Delete Phone.

2.1.2 User

- Search for reviews.
- Compare different mobile phones.
- View mobile phone specifications.
- View classified/Non-Classified reviews.

2.2 Operating Environment

- Operating System: Ubuntu, Windows 8 or above
- Web Browser: Google Chrome, Internet Explorer (ver. 8 or later), Mozilla Firefox, Safari (Mac).
- Sentiment Analysis: Text Blob.

- Data Scarping: Scrapy.
- Web Development: Django framework, Node JS, React JS.

2.3   Design and Implementation Constraints

It is a web-based system therefore front-end of application will be developed using JSX in React Js. JSON file will be created by scraping data from different sites. For data scraping Scrapy will be used. Text Blob will be used for sentiment analysis. It will be developed using agile methodology by making modular designs where features of every module are wrapped separately, and each feature will be divided into possible sub features.

The results are not 100 % accurate. The Phone specifications and reviews which are not scraped will not be available. The scraper requires at least 3 or more weeks of constant running on a very good internet connection to collect required data of one site.

2.4   Assumptions and Dependencies

Domain specific search engine for mobile reviews aggregation is a web-based application therefore it requires having compatible web browsers installed on the operating systems to run efficiently. It requires the following tools to run.

- Python 2.7.12
- Python 3.5.2
- Django 1.11.10
- Nodejs 11.0.0
- NPM 6.4.1
- Text blob 0.15.3
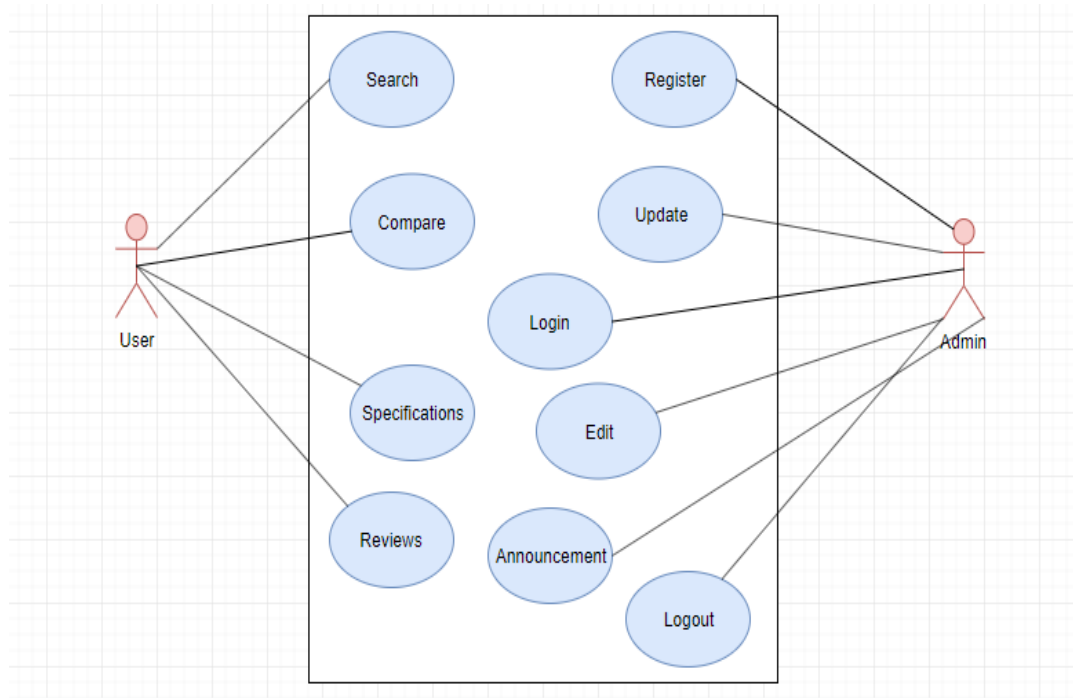- Scrapy 1.6.0

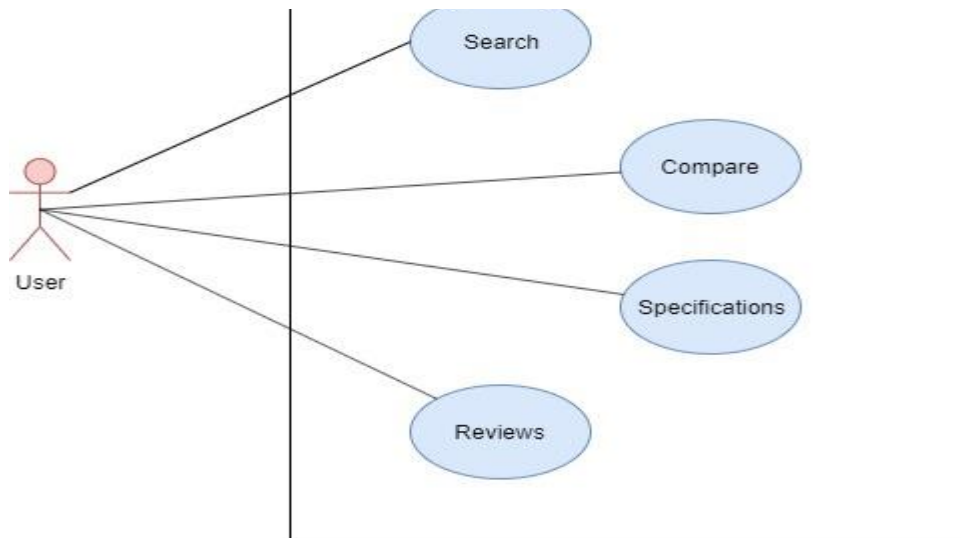## 2.5   System Use Cases

Figure 2.1: System UC Diagram

Figure 2.2: Users UC Diagram
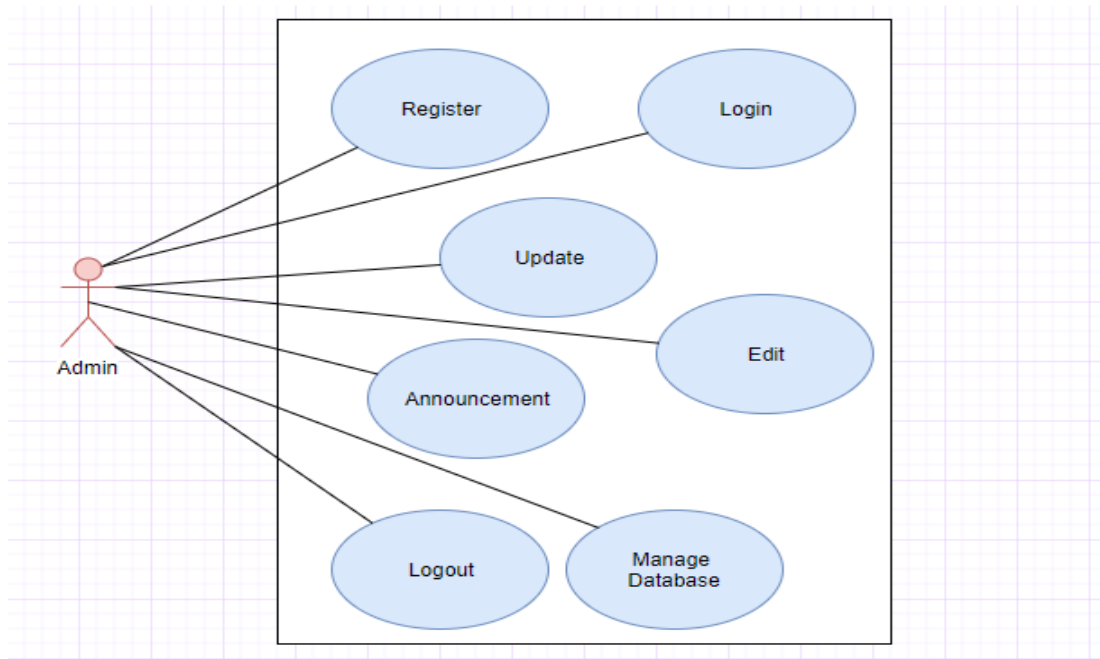
Figure 2.3: Admin UC Diagram

2.5.1   Register (U1)

Name	Register
Use-Case ID	U1
Priority	High
Primary Actor	Admin
Description	This use case describes the event of registration of admin, admin is already registered.
Pre-condition	Admin must open the website.
Trigger	This use case initiates when admin is going to register himself.
Typical flow of events	1. User open the website 2. Click on Admin button 3. Click on Registration button 4. Enter all required data 5. Press Register button to register himself.
Alternate flow of event	Alt-2. Click on Admin button Alt-3. Click on Login button Alt-4. Don't follow the rules for entering data
Post condition	Admin is registered.
Alternate post condition	Admin is not registered.

Table 1: Register (U1)

2.5.2   Login (U2)

Name	Login
Use-Case ID	U2
Priority	High
Primary Actor	Admin
Description	This use case describes the event of logging in of admin.
Pre-condition	Admin must register himself first
Trigger	This use case initiates when admin is going to login.
Typical flow of events	1. Admin open the website 2. Admin clicks on Login button. 3. Enter the username and password. 4. Admin is logged in after clicking the button login.
Alternate flow of event	Alt-2. Click on Admin button and login. Alt-3. Admin enter wrong username or password.
Post condition	Admin is logged in.
Alternate post condition	Admin is not logged in.

Table 2: Login (U2)

2.5.3 Announcements (U3)

Name	Announcements
Use-Case ID	U3
Priority	Medium
Primary Actor	Admin
Description	This use case describes the event of announcements by the Admin.
Pre-condition	Admin must log in.
Trigger	This use case initiates when admin is going to post the announcements
Typical flow of events	1. Admin log in into website 2. Click on Announcements button 3. Post Announcements
Alternate flow of event	Alt-1. Admin is not logged in due to wrong username or password Alt-2. Admin clicks on another button rather than Announcements
Post condition	Admin post Announcements
Alternate post condition	Admin was not post Announcements

Table 3: Announcements (U3)

2.5.4   Updates (U4)

Name	Update
Use-Case ID	U4
Priority	Medium
Primary Actor	Admin
Description	This use case describes the event of update by the Admin.
Pre-condition	Admin must log in.
Trigger	This use case initiates when admin is going to update.
Typical flow of events	1. Admin log in into website. 2. Click on update button.
Alternate flow of event	Alt-1. Admin is not logged in due to wrong username or password Alt-2. Admin clicks on another button rather than update.
Post condition	Admin update.
Alternate post condition	Admin not updated.

Table 4: Update (U4)

2.5.5   Edit (U5)

Name	Edit
Use-Case ID	U5
Priority	Medium
Primary Actor	Admin
Description	This use case describes the event of edit by the Admin.
Pre-condition	Admin must log in.
Trigger	This use case initiates when admin is going to edit.
Typical flow of events	1. Admin log in into website. 2. Click on edit button.
Alternate flow of event	Alt-1. Admin is not logged in due to wrong username or password Alt-2. Admin clicks on another button rather than edit.
Post condition	Admin edits.
Alternate post condition	Admin did not edit.

Table 5: Edit (U5)

2.5.6   Logout (U6)

Name	Logout
Use-Case ID	U6
Priority	High
Primary Actor	Admin
Description	This use case describes the event of Logout.
Pre-condition	Admin must log in.
Trigger	This use case initiates when admin is going to logout.
Typical flow of events	1. Admin log in into website. 2. Click on logout button.
Alternate flow of event	Alt-1. Admin is not logged in due to wrong username or password Alt-2. Admin clicks on another button rather than logout.
Post condition	Admin did not login.
Alternate post condition	Admin did not logout.

Table 6: Logout (U6)

2.5.7 Search (U7)

Name	Search
Use-Case ID	U7
Priority	Medium
Primary Actor	User
Description	This use case describes the event of search by the user.
Pre-condition	User must open web application.
Trigger	This use case initiates when user is going to search.
Typical flow of events	1. User must open the web application. 2. Click on search button.
Alternate flow of event	Alt-1. User must open web application. Alt-2. Admin clicks on another button rather than search.
Post condition	User search.
Alternate post condition	Admin was not searching.

Table 7: Search (U7)

2.5.8   Compare (U8)

Name	Compare
Use-Case ID	U8
Priority	Medium
Primary Actor	User
Description	This use case describes the event of compare by the user.
Pre-condition	User must open application.
Trigger	This use case initiates when user is going to compare.
Typical flow of events	1. User open website. 2. Click on compare button.
Alternate flow of event	Alt-1. User did not open web application. Alt-2. User clicks on another button rather than compare.
Post condition	User compares.
Alternate post condition	User did not compare.

Table 8: Compare (U8)

2.5.9   Specification (U9)

Name	Specification
Use-Case ID	U9
Priority	Medium
Primary Actor	User
Description	This use case describes the event of specs by the user.
Pre-condition	User must open web application.
Trigger	This use case initiates when user is wanting to see the specs.
Typical flow of events	**1.** User must open web application. **2.** Click on specs button.
Alternate flow of event	Alt-1. User did not open application. Alt-2. Admin clicks on another button rather than specs.
Post condition	User see specs.
Alternate post condition	User did not see specs.

Table 9: Specification (U9)

2.6   Other Non-functional Requirements

2.6.1   Performance Requirements

The system should perform all actions correct and frequent. The system should give a response within a reasonable time. Internet is very essential for this WEB APP. Because the website usually relies on internet, so faster internet will give better performance for the system. DSSEMPRA requires a Laptop which must have Windows installed in it. Laptop must have minimum 4GB of RAM, 80 GB of storage and i3 Processor. Graphics card is not a must in it because the WEB APP contains efficient graphics level. Screen size may vary, and the WEB APP runs in all screen sizes (Laptop, Desktop, Mobiles and Tablets and many more). Although for the backend a cloud-based database is required and a pc/laptop having 8GB of RAM and a minimum of i3 processor with at least 500GB of Storage.

2.6.2   Safety Requirements

Because the data which we would be dealing with would create a huge database of reviews we would require backing up the entire database either on a server or cloud service.

2.6.3   Security Requirements

The Admins will have full control of the website keeping it secure will be the duty of the admins. The database is to be secured and once different credentials are provided, they will be kept under surveillance by the super admin.

2.6.4   Software Quality Attributes

DSSEMPRA provides their users with both simple and advanced features. Due to its well designed and easy to use interface and frequent modules transitions it can be used by both experts and typical users. However, users must already have a basic knowledge of using a WEB APP before using it.

2.7   Other Requirements

2.7.1   Availability:

The web app would be available 24/7 once up and running on a domain. It would not be available when web app is being updated on server end.

2.7.2 Maintainability:

The system would be maintained and configured easily by the admin.

2.8 System Requirements Chart

ID	Priority	Type	Source	Used in Use case	Description
1	High	Functional	University	N/A	Open the web application.
2	High	Functional	University	U1	Registration is done by the admin to interact.
3	Medium	Functional	University	U1	Reset Password
5	Medium	Functional	University	U6	Logout
6	Medium	Functional	University	U4	Update by admin
7	Medium	Functional	University	U4	Update set by admin
8	Medium	Functional	University	U3	Announcements are made by admin
9	Medium	Functional	University	U5	Edit by admin

10	Low	Functional	University	U7	Search by users
11	Medium	Functional	University	U8	Compare by users
12	Medium	Functional	University	U9	Specs for users
14	High	Functional	University	U6	Logout
15	High	Non-Functional	University	N/A	Display all content after loading web page.
16	High	Non-Functional	University	N/A	Server Response
17	High	Non-Functional	University	N/A	Power loss and system failure safety
18	High	Non-Functional	University	N/A	All the passwords should be in encrypted form
19	High	Non-Functional	University	N/A	Components of the project code will be tested alongside the implementation phase to ensure that they are functional.
20	High	Non-Functional	University	N/A	Final, integrated project code will test to ensure Working

Table 10: System Requirement Chart

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Methodology

We are going to be using agile Method. The Agile Method is an approach to project management that is utilized in software development. This method assists teams in responding to the unpredictability of constructing software. It uses incremental, iterative work sequences that are commonly known as sprints.
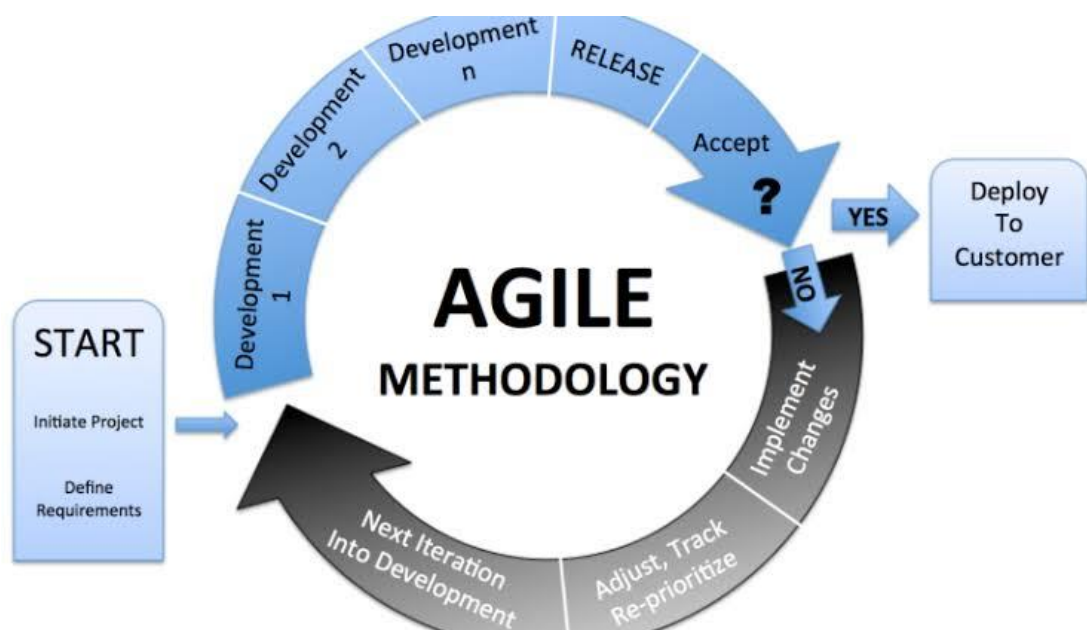
Figure 3.1: Agile Methodology

3.2 Domain Model

Domain models represent the set of requirements that are common to systems within a product line. There may be many domains, or areas of expertise, represented in a single product line and a single domain may span multiple product lines. The requirements represented in a domain model include:

- Definition of scope for the domain
- Information or objects
- Features or use-cases, including factors that lead to variation
- Operational/behavioral characteristics

A product line definition will describe the domains necessary to build systems in the product line.
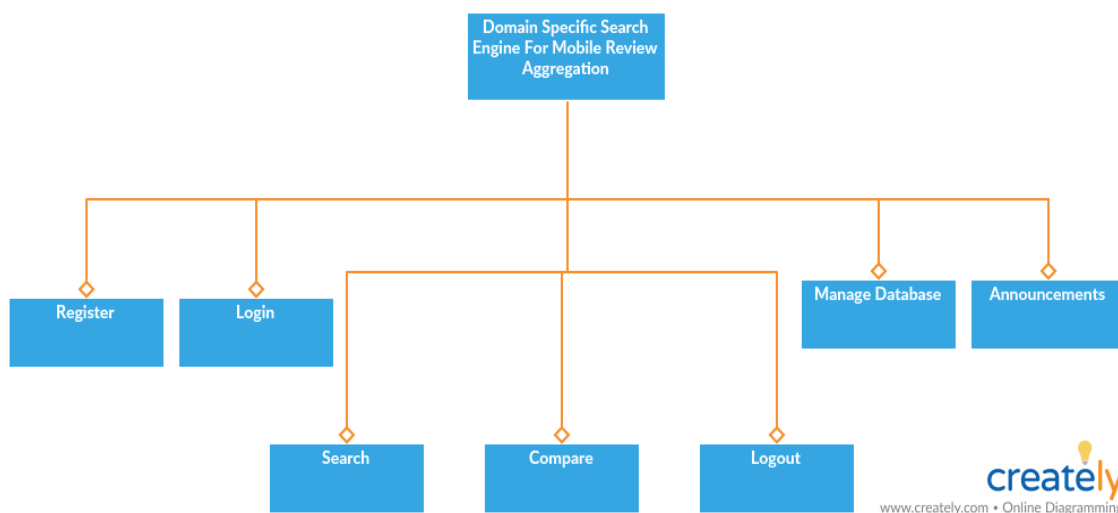
Figure 3.2: Domain Model

3.3 Collaboration Diagram

A collaboration diagram describes a pattern of interaction among objects; it shows the objects participating in the interaction by their links to each other and the messages that they send to each other.
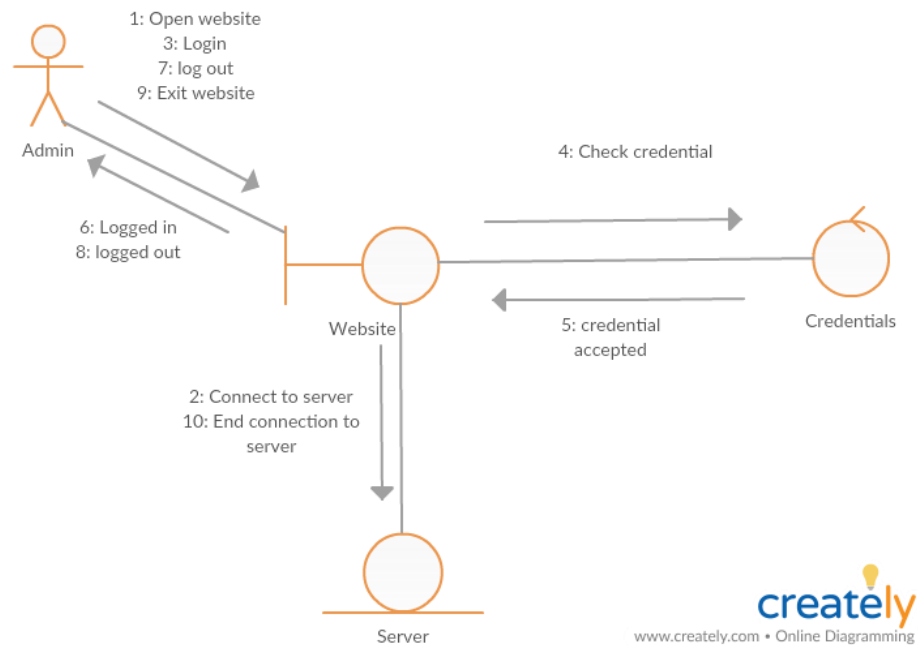
Figure 3.3: Collaboration diagram (Login/Log out)
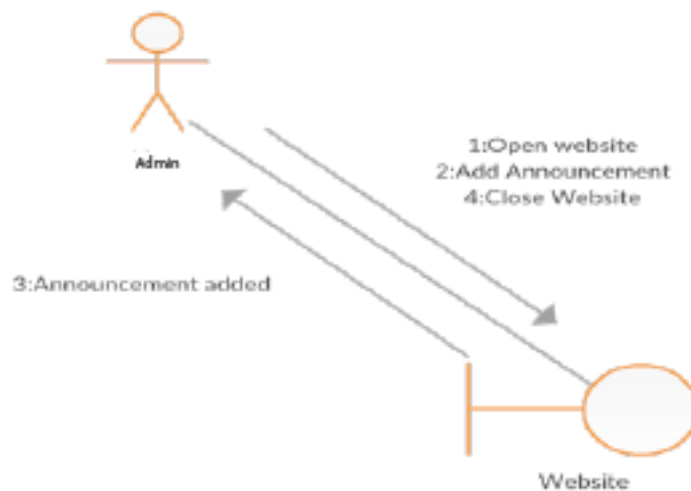
Figure 3.4: Collaboration Diagram (Announcement)
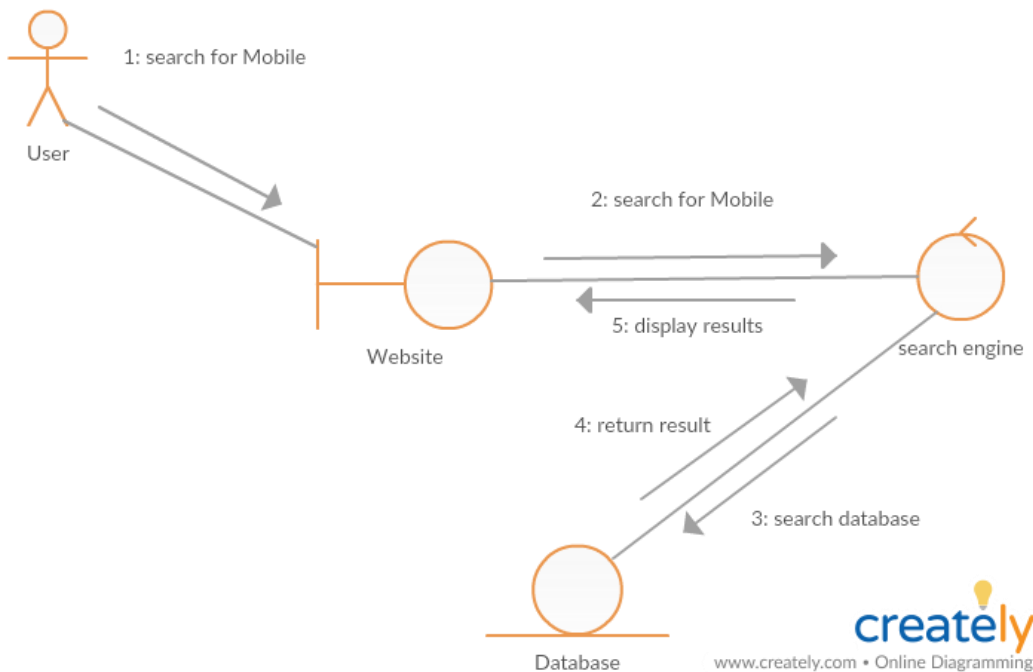
Figure 3.5: Collaboration Diagram (Search)
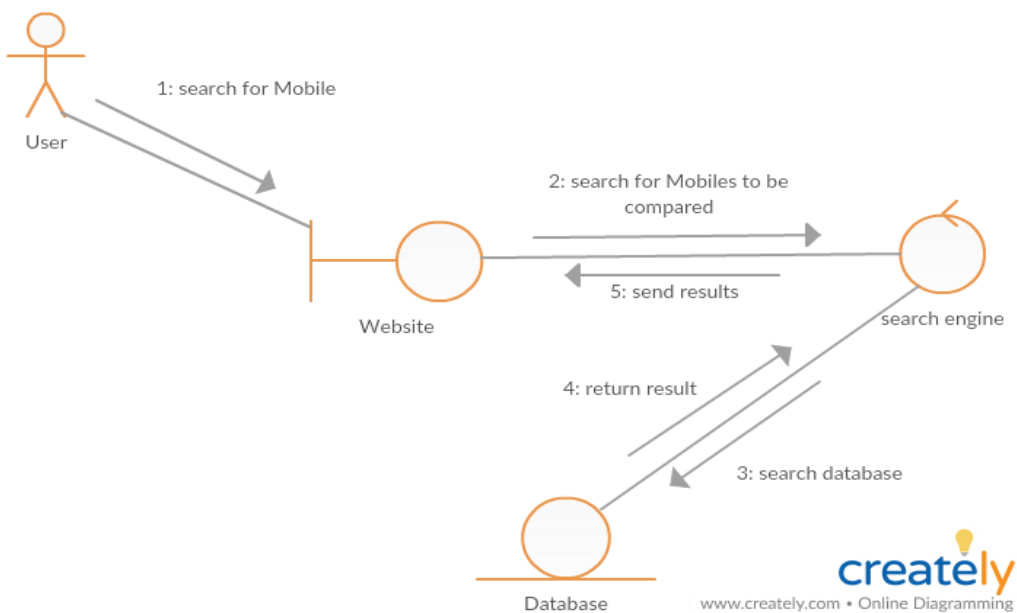
Figure 3.6: Collaboration Diagram (Compare)

3.4   **Operation Contracts**

A UML Operation contract identifies system state changes when an operation happens. Effectively, it will define what each system operation does. An operation is taken from a system sequence diagram. It is a single event from that diagram. A domain model can be used to help generate an operation contract.

3.4.1   **Operation Contract Registration**

>   **Name:** Register
>
>   **Responsibilities:** To register admin.
>
>   **Cross References:** U1
>
>   **Exceptions:** Admin entered wrong credentials.
>
>   **Preconditions:** Must open the web application.
>
>   **Post conditions:** Admin registered.

3.4.2   **Operation Contract Login**

>   **Name:** Login
>
>   **Responsibilities:** To allow login to web application.
>
>   **Cross References:** U2
>
>   **Exceptions:** Entered wrong credentials.
>
>   **Preconditions:** Must open the web application.
>
>   **Post conditions:** Admin logged in.

3.4.3   Operation Contract Announcements

Name: Announcements

Responsibilities: The event of announcements by the Admin.

Cross References: U3

Exceptions: Admin didn't select correct option.

Preconditions: Admin must log in.

Post conditions: Admin post Announcements.

3.4.4   Operation Contract Manage Database

Name: Manage Database

Responsibilities: The event of updating and editing Database by the Admin.

Cross References: U4

Exceptions: Admin didn't select correct option.

Preconditions: Admin must log in.

Post conditions: Admin post updated or edited database.

3.4.5   Operation Contract Logout

Name: Logout

Responsibilities: To allow logging out from web application.

Cross References: U5

Exceptions: Admin didn't select correct option.

Preconditions: Admin must log in.

Post conditions: Admin logged out.

3.4.6   Operation Contract Search

Name: Search

Responsibilities: The event of searching by the user.

Cross References: U6

Exceptions: User didn't select correct option.

Preconditions: User must open web application.

Post conditions: User searched.

3.4.7   Operation Contract Compare

Name: compare

Responsibilities: The event of comparing mobiles by the user.

Cross References: U7

Exceptions: User didn't select correct option.

Preconditions: User must open web application.

Post conditions: User compared mobile phones.

3.4.8   Operation Contract Specification

Name: Specification

Responsibilities: The event of checking specifications by the user.

Cross References: U8

Exceptions: User didn't select correct option.

Preconditions: User must open web application.

Post conditions: User checked specification.

CHAPTER 4

USER MANUAL

4.1   Home page
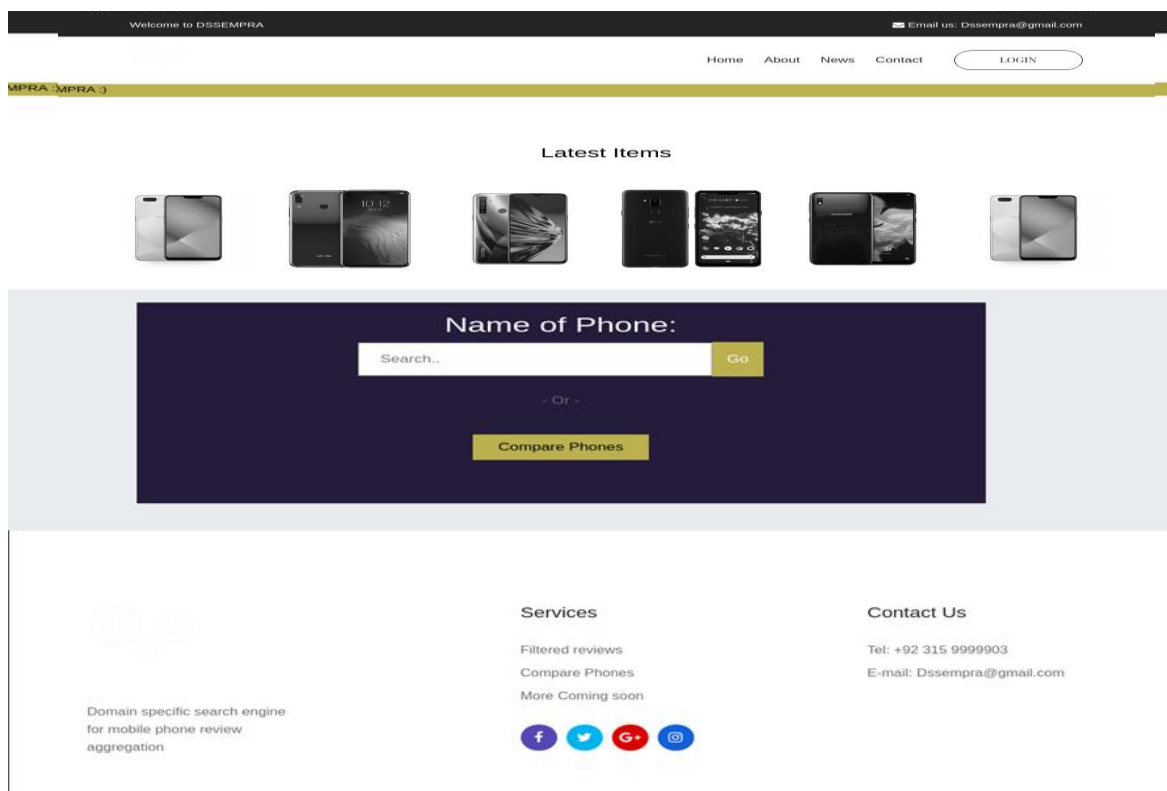
Figure 4.1: Homepage

4.2   Search
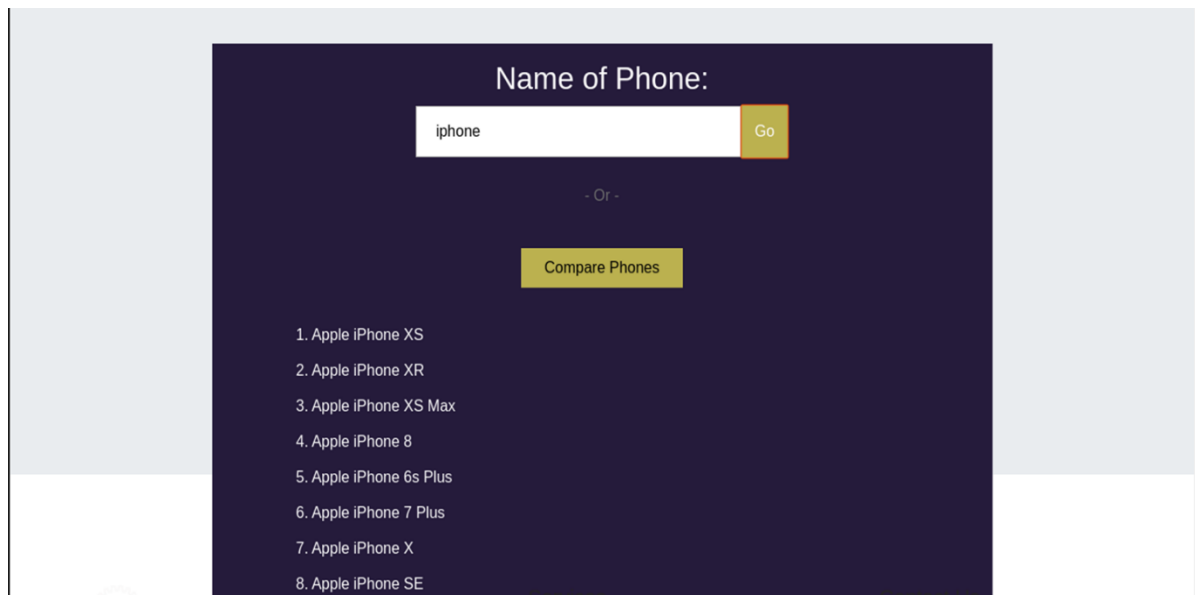
Figure 4.2: Search Phone

4.3   Non-Classified Reviews

Figure 4.3: Non-Classified Reviews

4.4   Classified Reviews

Figure 4.4:Classified Reviews

4.5   Compare phones

Figure 4.5: Compare Phones