|  |  |
| --- | --- |
|  | **Qatar University**  **College of Engineering**  **Department of Computer Science and Engineering** |

Senior Project Report

**Multi-facets online Shopping System Structuring**

**Project Group Members:**

Afnan Esameldin - 201201777  
Haya Nasser Al-Nuaimi - 201402885  
Inas Ibrahim Eissa - 201300268

**Supervisor**: Dr. Ali Jaoua

**2017**

This project report is submitted to the Department of Computer Science and Engineering of Qatar University in partial fulfillment of the requirements of the Senior Project course.

# Declaration

This report has not been submitted for any other degree at this or any other University. It is solely the work of us except where cited in the text or the Acknowledgements page. It describes work carried out by us for the capstone design project. We are aware of the university’s policy on plagiarism and the associated penalties and we declare that this report is the product of our own work.

Student: Afnan Esameldin Date:

Signature:

Student: Haya Nasser Al-Nuaimi Date:

Signature:

Student: Inas Ibrahim Eissa Date:

Signature:

# Abstract

Nowadays people tend to use online shopping system, since it provides guidance and useful feedbacks. The objective of this project is to design, implement, test and validate a multi-facet online shopping system structuring approach. The target system should help users to have better shopping experience. It should also minimize the number of clicks for the users, help them to get more specific query and find their target product in a rational way.

Starting from existing online system as Amazon, the purpose is to offer to the user different browsing trees depending on his/her preference. Advanced browsing should help the user to make best decisions during the purchase process. Customers should be able to view the structured results at run time starting from one or several attributes (i.e. Price, category, description, mark, rank, etc...). The system should also display association rules or implications within some support and confidence. As we have limitation on the number of retrieved data, we need approximate the calculation of the support and confidence to capture the hidden data factor. This project should help sellers to improve the purchases, and improve user experience by offering different attributes that can be helpful in their purchase. Obtained system should be tested and validated.

# Acknowledgment

We would like to thank all of those who supported us and made it achievable for us to end the project requirements easily and effectively. Exceptional on account of our supervisor Dr. Ali Jaoua, who never hesitated to help us during the entire semester. We are also very thankful for Dr. Ali research team, Mr. Aboubakr Aqle, Ms. Eman Rezk, and Mr. Fahad Islam for their infinite help and support.

This contribution was made possible by NPRP grant #07- 794-1-145 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

# Table of Contents

[Declaration ii](#_Toc501386021)

[Abstract iii](#_Toc501386022)

[Acknowledgment iv](#_Toc501386023)

[Table of Contents v](#_Toc501386024)

[List of Figures vi](#_Toc501386025)

[List of tables vii](#_Toc501386026)

[1. Introduction and Motivation 8](#_Toc501386027)

[1.1. Problem statement 8](#_Toc501386028)

[1.2. Project significance 9](#_Toc501386029)

[1.3. Project objectives 9](#_Toc501386030)

[2. Background and related work 10](#_Toc501386031)

[2.1. Background 10](#_Toc501386032)

[1. Extracting Data from Online Systems with Hidden Database: 10](#_Toc501386033)

[2. Formal concept analysis (FCA) / Apriori Algorithm: 11](#_Toc501386034)

[3. Stemming and Lemmatization 13](#_Toc501386035)

[4. Conceptual Browsing tree algorithm 14](#_Toc501386036)

[2.2. Related work 15](#_Toc501386037)

[3. Requirements analysis 21](#_Toc501386038)

[3.1. Functional requirements 21](#_Toc501386039)

[3.2. Non-functional requirements 22](#_Toc501386040)

[3.3. Professional Code of Ethics 23](#_Toc501386041)

[3.4. Assumptions 25](#_Toc501386042)

[4. Proposed solution 25](#_Toc501386043)

[4.1. Possible Solutions and tradeoffs 25](#_Toc501386044)

[4.2. Selected Solution overview 28](#_Toc501386045)

[4.3. High level architecture 31](#_Toc501386046)

[4.4. Hardware/software to be used 33](#_Toc501386047)

[5. Project plan 35](#_Toc501386048)

[5.1. Project milestones 35](#_Toc501386049)

[5.2. Project timeline 39](#_Toc501386050)

[5.3. Anticipated risks 42](#_Toc501386051)

[References 44](#_Toc501386052)

[Appendix A – Use cases specification 47](#_Toc501386055)

[Appendix B – Survey and Survey Analysis 56](#_Toc501386056)

# List of Figures

Figure 1. Amazon AWS API retrieved attributes 11

Figure 2. Apriori Algorithm flowchart 13

Figure 3. Compute Hyper Concept for each attribute 15

Figure 4. Hyper Concept Tree construction 16

Figure 5. Retrieved data from Amazon 17

Figure 6. Use cases diagram 20

Figure 7. Flow chart of Porter Algorithm. 28

Figure 8. Illustrates Spiral life cycle model [6] 29

Figure 9. High level Architecture 30

Figure 10. Project Flow chart 31

Figure 11. Interface design 31

Figure 12. Survey questions #1. 55

Figure 13. Survey questions #2. 56

Figure 14. Analysis of Question # 1. 57

Figure 15. Analysis of Question # 2. 57

Figure 16. Analysis of Question # 3. 58

Figure 17. Analysis of Question # 4. 58

Figure 18. Analysis of Question # 5. 59

Figure 19. Analysis of Question # 6. 59

Figure 20. Analysis of Question # 7. 60

Figure 21. Analysis of Question # 8. 60

Figure 22. Analysis of Question # 9. 61

# List of tables

Table 1. Binary Context 11

Table 2. Binary Relationship of retrieved data from Amazon 17

Table 3. Comparison between research papers 19

Table 4. Use cases summary 21

Table 5: non-functional requirements 22

Table 6: Code of ethics 23

Table 7: Many-valued context of university majors 25

Table 8: Scaled Many-value context of university majors 26

Table 9: Platform comparison 27

Table 10: Hardware 33

Table 11: Software 33

Table 12. Milestone Fall 2017 35

Table 13. Milestone Spring 2018 38

Table 14. Timeline Fall 2017 39

Table 15. Timeline Spring 2018 41

Table 16. Anticipated risks 42

# Introduction and Motivation

## Problem statement

In our project we will provide to the user a straightforward way to get his goal using conceptual structure-browsing algorithm. This new multi-facet system will help us to give a suitable interface and accurate results to the user in a minimum run time.

We solved/observe some technical issues which are:

* Retrieving data from hidden database using API.
* Writing an algorithm that will scale the data in acceptable run time.
* Finding support and confident concepts to extract the rules from generated formal context using Apriori algorithm.
* Creating a tree interface for the user.
* Improving conceptual browsing algorithm to fit with our system (multi-facet).
* researching about different algorithms like conceptual browsing algorithm, and Apriori algorithm.
* Understanding the mechanisms of API’s for different online shopping system.
* Figure out the most important attributes to the user. (select some attributes)

We are excepting to face some challenges while developing this system like:

* Lack of theoretical knowledge.
* Lack of programming knowledge (language that we will use).
* Lack of management tools for SQLite database.
* Subscription to a premium account to get better response time as the Amazon AWS gives us low responsiveness with free account.

Our target system is described in the (Figure 11- Interface design). The web application goes through important stages which are:

1. Reading the query from user.
2. Retrieving results from online shopping system.
3. Saving data in our database.
4. Generating tree and rules.
5. Displaying result in acceptable interface.

## Project significance

The proposed solution will focus on popular online marketplaces items. Those marketplaces provide websites with search capabilities, but the results are shown in a way that can’t help the user to reach his/her target in an organized way. Our project has many benefits to the stakeholders. For us, it gives us an experience to work with hidden databases’ APIs. It also has a benefit to the users by providing helpful interface that divide the page into parts, one part contains the facets that the user chose according to his desired filter, another part includes the tree and other has the rules that extracted from formal context. This means that user can see the results in more organized way using a tree where the user can navigate through different paths. If user clicks on the leaf node, it will be a link that route him to specific item based on his choice and display only the item that are relevant. Also, the user has the option to go and choose directly an item from the extracted rules. The project idea is unique and new in the region. The benefit of the project in Qatar and gulf region is that it is a new way of searching that significantly help to improve the user experience and it will put Qatar in high position in technology researching.

## Project objectives

The project objective is to create a web application, which is a structured view for the user to reach his desired item(s) with minimum amount of time. The goal now is to apply it on Amazon and it might be extended to eBay and other online shopping system. The user submits a query to our system search box, the system will get the user input and search for it on different online shopping system. Then stores the retrieved data inside a database. After that multi-facets box, tree and extracted rules will appear to the user. The user chooses one or more attributes for browsing in the tree in a multi-facet way. He is also able to go through different paths in the tree until finding the desired items. The user may also select an item from the set of rules to make better and shorter item selections.

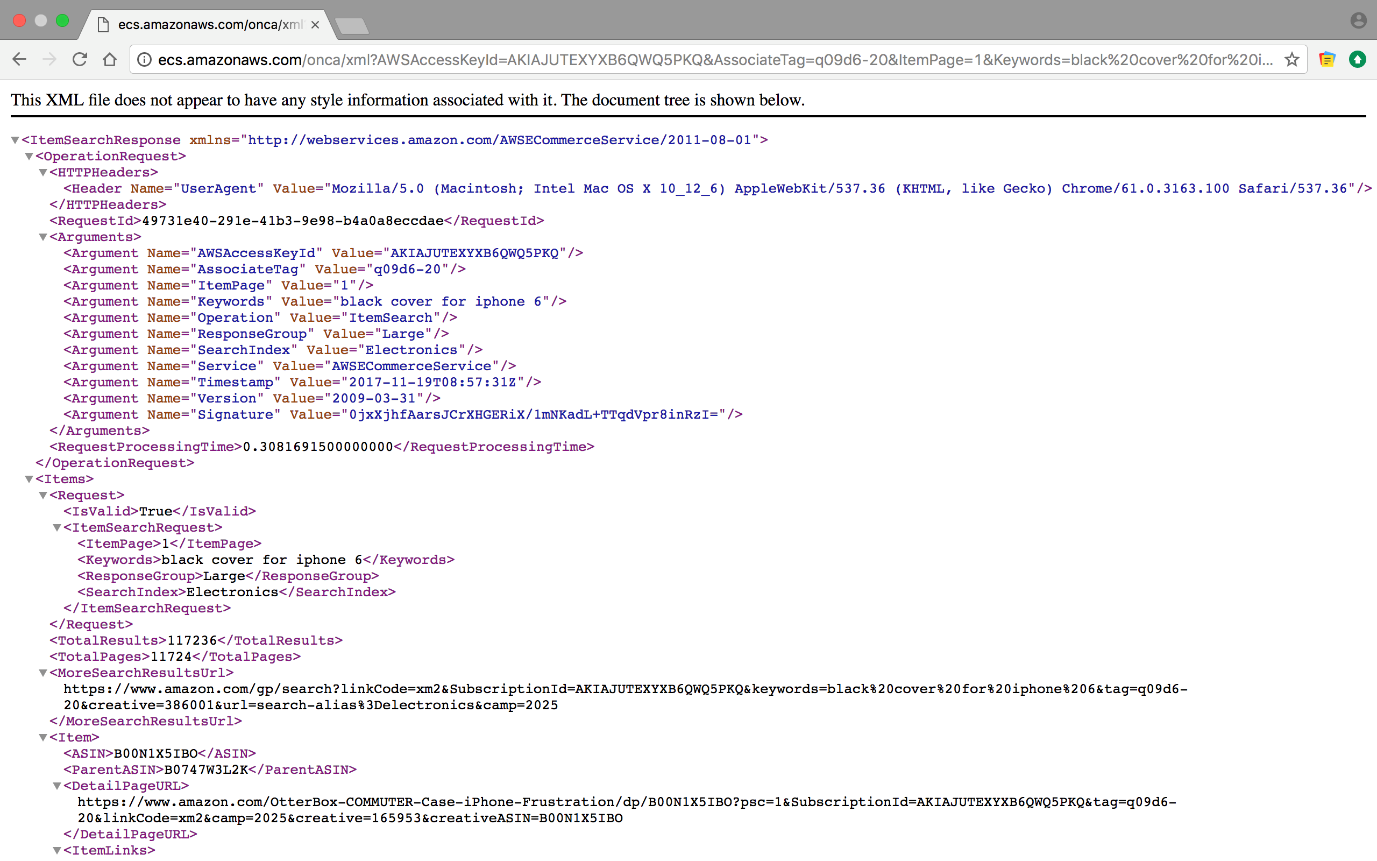
# Background and related work

## Background

### Extracting Data from Online Systems with Hidden Database:

One important originality of our project is the context of hidden database; since APIs offered by online systems give limited access to the database and only top k results. Although most of online shopping system hidden database’s can be access through a specific provided APIs at query time [8]. APIs are a query interfaces that act as a channel to access the source databases [1]. For example, Amazon offers some services that enable us to read from their source database when you register for Product Advertising APIs. But there are some Security Credentials that AWS (Amazon Web Services) declines the attempt in case of temporary overloading and response with 503 error code. We overcome this problem by giving time between each request. Using the Product Advertising APIs feature, we can send a request through calling an HTTP (Hypertext Transfer Protocol) GET method using name-value pairs, which are:

1. Endpoint = http://ecs.Amazonaws.com/onca/xml?
2. AWSAccessKeyId = xxxxxxxxxxxxxxxxxxxxx (specific key to each registered user)
3. AssociateTag = xxxxxxxxxxxxxxxxxxxxxx (tag to each registered user)
4. Version = 2009-03-31
5. Operation = ItemSearch (type of operations provided by Amazon)
6. ResponseGroup = Large (to enlarge the number of attributes)
7. SearchIndex = Electronics (targeted category for our project)

The received response is obtained in a XML (Extensible Markup Language) format as shown in (Figure 1- Amazon AWS API retrieved attributes). The XML doc contains many attributes regarding the stored items as well as the header of the request. The retrieved data is reflected as an instance of Amazon’s product database since it is only a fraction of the whole database, this is considered as an issue since we need to estimate the fraction. 

##### Figure 1. Amazon AWS API retrieved attributes

### Formal concept analysis (FCA) / Apriori Algorithm:

Formal Concept Analysis is a method of data analysis with growing popularity across various domains, concept hierarchy from a collection of objects and their properties [3]. It has been developed as an area of applied mathematics grounded on building a mathematical model of concepts and concepts hierarchy. The approach takes an input binary matrix identifying a set of attributes for those objects (columns) and objects (rows), finds the natural concepts defined in the data, and then arranges the concepts in a fractional order structure [7]. Each concept in the final diagram is a pair of sets of objects and attributes that are maximally associated with each other. A table with logical attributes can be represented by a triplet (O, A, I) where I is a binary relation between O and A. Elements of O are called objects and correspond to table rows, elements of A are called attributes and correspond to table columns. For example, the equivalent table (O, A, I) is given by O = {O1, O2, O3}, A = {A1, A2, A3} and we have (O1, A1) ∈ I, (O2, A3) ∉ I as shown in (Table 1-Binary Context).

###### Table 1. Binary Context

|  |  |  |  |
| --- | --- | --- | --- |
|  | A1 | A2 | A3 |
| O1 | 1 | 1 | 1 |
| O2 | 1 | 1 | 0 |
| O3 | 0 | 1 | 1 |

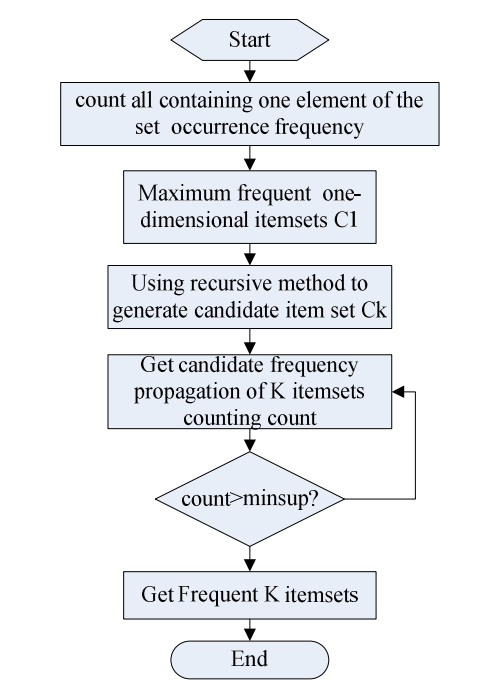
**Apriori algorithm**

The Apriori Algorithm is a persuasive algorithm for extracting frequent itemsets for Boolean association rules. Apriori utilizes a " bottom up" approach, where common subsets are stretched single item at any given moment in a step called Candidate generation, then candidates are examined alongside the data. Apriori is intended to work on database having transactions (for instance, accumulations of items purchased by clients). The algorithm mainly consists of two stages; Applying minimum support to locate all the frequent sets with k items in a database, then using the self-join rule to discover the frequent sets with k+1 items with the assistance of frequent k-item sets [21]. Reiterate the two stages from k=1 till we are incapable to apply the self-join rule (a rule to find 𝐿k, a set of candidate k-itemsets is generated by joining 𝐿k-1 with itself.) [21]. After generating the association rules, there are three definitions to measure them, support, confidence and calling frequent item set. Support is the relevancy of the association rule, while confidence is the reliability of the association rule, additionally calling frequent item set is when the item set support(X) is bigger than the threshold minsup. Let K: = (G, M, I) be a context, where G is a set of objects, M is a set of attributes (items), I G x M an association rule of the context K is an expression AB, where A, BM and(usually)AB=. Apriori algorithm flow chart is shown below [21].

Define cot

support (AB) =

confidence (AB) =



##### Figure 2. Apriori Algorithm flowchart

### Stemming and Lemmatization

Distinctive types of a word frequently convey basically a similar meaning [9]. For instance, there's most likely no distinction in intent between a query for hat and a look for hats. These syntactic contrasts between word frames are called inflections, and they make difficult task for query comprehension.

To avoid the variation of the inflections of words to affect the quality and the accuracy of the results we use the concept of Stemming and Lemmatization to get the most important keywords from the data. A similar root, for instance ‘covering’ is gotten from the stem or the root ‘cover’. So also, the word ‘offering’ gets from the verb ‘offer’. When dealing with a query, it is important to check every one of the expressions of similar origin and meaning to guarantee the pertinence and precision of the outcomes that will be recovered [17]. Stemma manages cutting the last couple of letters of a word to diminish the word back to its root, for example, ‘buying’ would progress toward becoming ‘buy’. Moreover, lemmatization is concerned about the base of the word, it forms of a set of related words like, the lemma of {paying, paid, pays} is pay. These strategies are utilized while building the term dictionary, by operating using porter’s algorithm to each term, while processing the queries to find a match in the dictionary.

**Porter Stemmer Algorithm**

The porter stemmer algorithm is a standout stemmer algorithm among stemmer algorithms like Lovins stemmer. It is an algorithm designed for removing the commoner morphological and inflexional endings from words in English, where they are removed by some standard removal rules [9]. In the first phase, a decrease rule is picked among a gathering based on the size of the suffix, these rules are

**Rule Example**

SSES → SS caresses → caress

IES → I ponies → poni

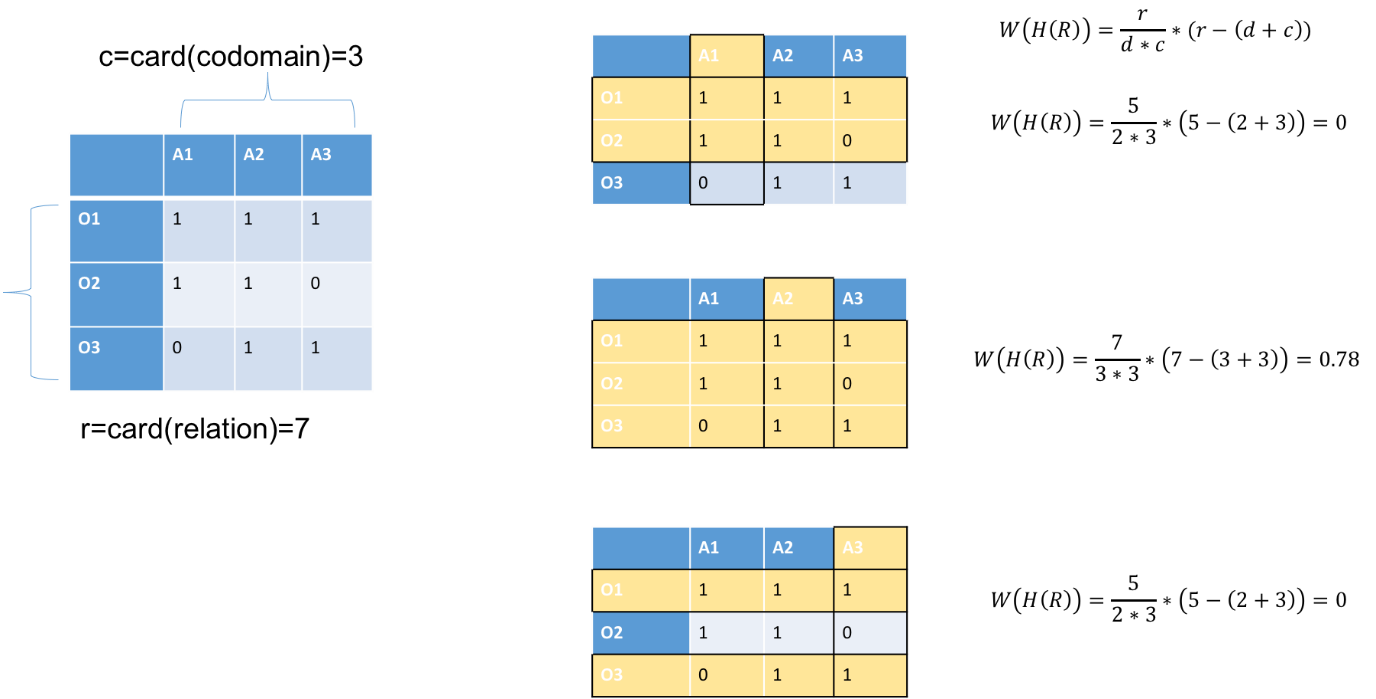
SS → SS caress → caress

S → cats → cat

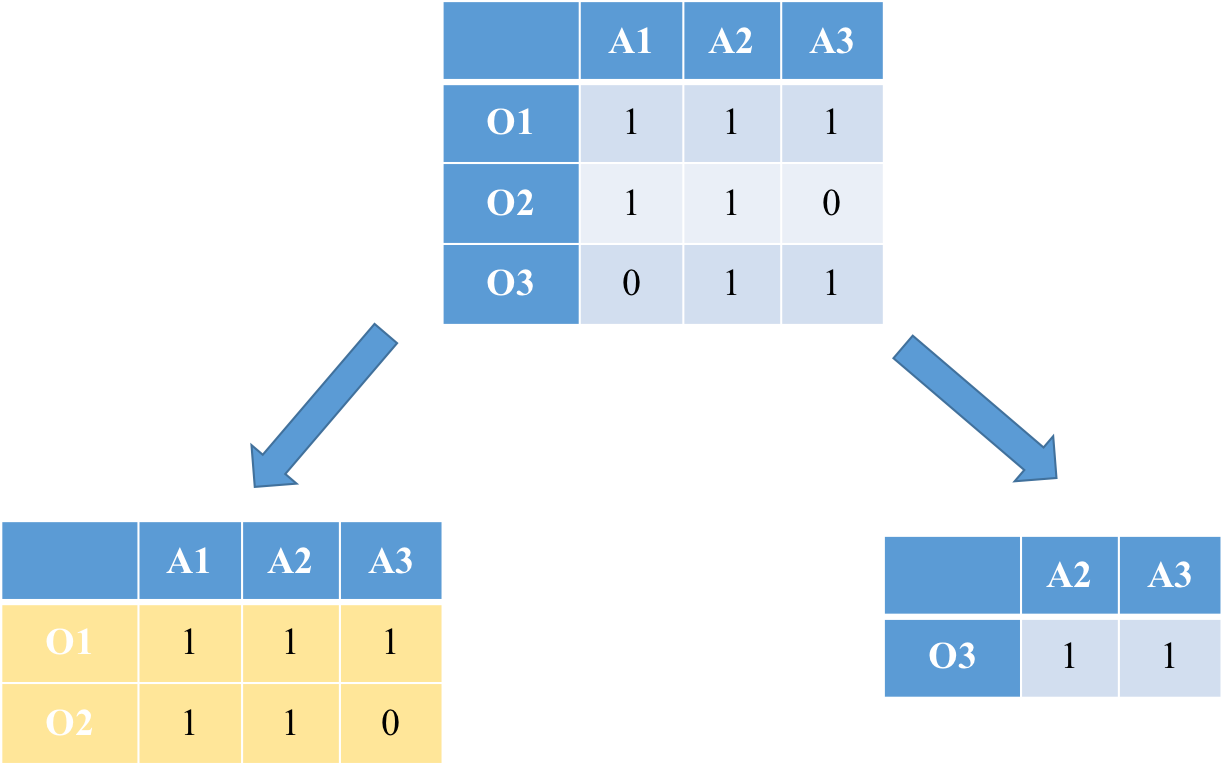
There is additionally another approach to check for suffix. For instant: the word “devotement”. There is a rule mentioned in Porter’s algorithm state that; if the term ends with “ement”, calculate the characters that prior “ement”, if the number of characters bigger than one; then we eliminate “ement”. As a result, the word “devotement” will be “devot”. If the number of characters is one or less, it will remain the same [9].

### Conceptual Browsing tree algorithm

The conceptual browsing algorithm used the hyper concepts method to extract concepts from the binary matrix to form a tree. The hyper concept method has a complexity of O (mn3), where n is the number of attributes and m is the number of objects. In the tree each concept act as a node in the tree and each item act as a leaf node. For example, referring to the (Table 1-Binary Context) using the Conceptual Browsing Algorithm we have demonstrated the steps needed to form the tree. First we compute the binary relation related to the objects. Then we calculate the maximum Hyper-concepts(HC) and the relations. We are trying to find the minimal coverage of the formal context by a set of Hyper-concepts(HC) covering the whole context. Then we can decompose each HC in the level to go a level deeper by avoiding the double selection of selected words [16].



##### Figure 3. Compute Hyper Concept for each attribute



##### Figure 4. Hyper Concept Tree construction

## Related work

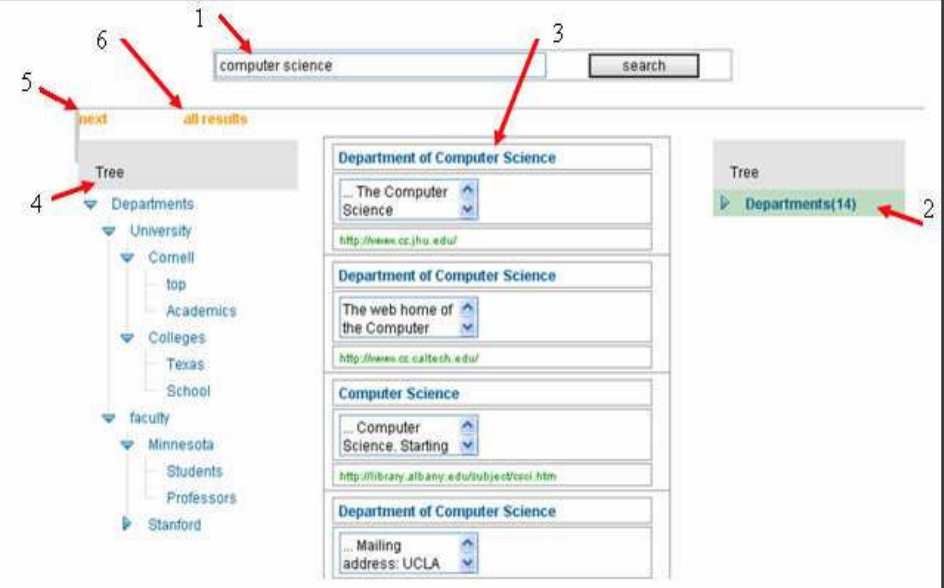
A lot of discussions and research has been done in the area of enhancing user experience in online shopping, and we address in our project closely related to these main topics: Multi-facets structuring system, Structuring result into trees, rule generation and Entity extraction.

[2] MOBILE APP FOR HIDDEN DATA ANALYTICS OF ONLINE MARKETPLACE SYSTEMS

The aim of this research is providing a good interface to user to help them to shop online by offering an Android application using the tree structure, the interface is minimized to only two navigation screens. A set of APIs were used to help extracting hidden data from several online marketplaces such as Amazon and eBay. Moreover, the implementation done using Java programming language and all responses are received in XML format. This paper is limited because the tree will be made based on only one attribute which is the item descriptor. In fact, the difference between this system and our system is that we have chosen the most important attributes for the user to achieve the shoppers' goal so by using multi-facets we will gain the user satisfaction. Also, we will generate rules based on item’s attribute to help user to get all the items that are related to the searched item

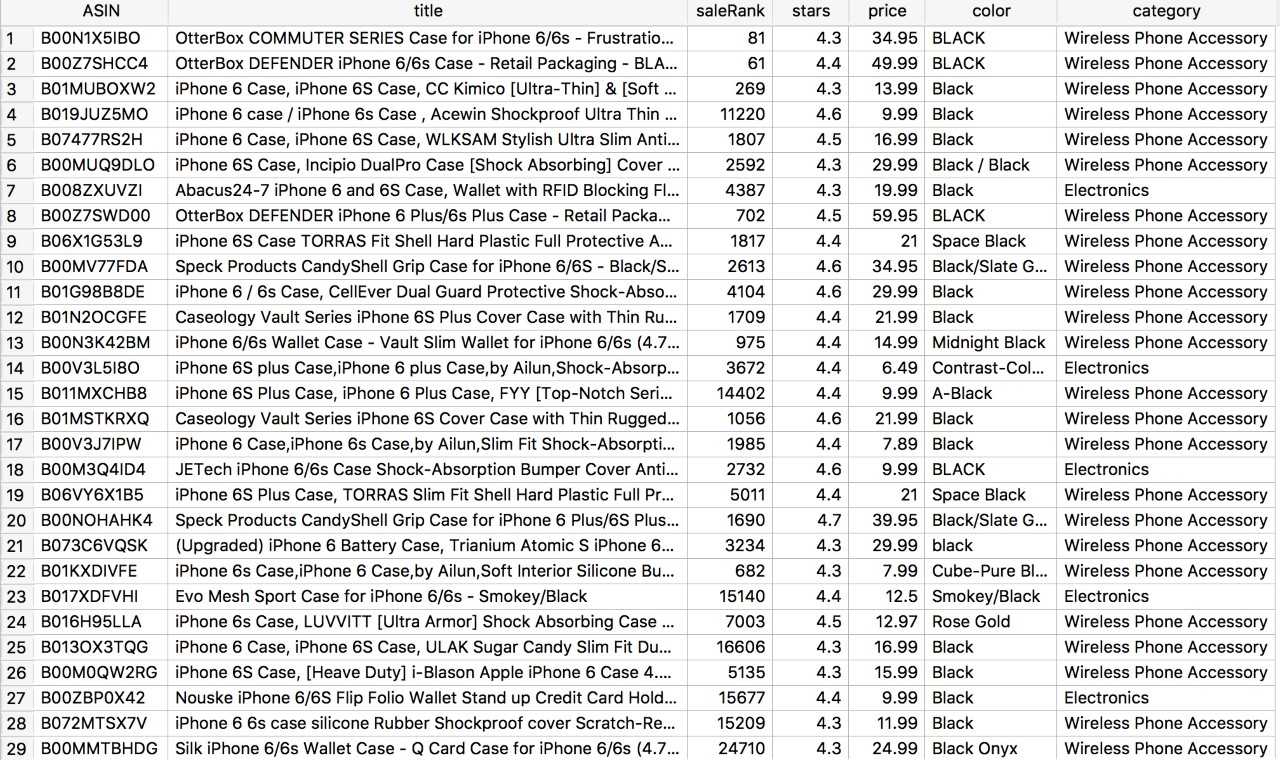
[12] TEXT SUMMARIZATION BASED ON CONCEPTUAL DATA CLASSIFICATION

In this paper, the authors discussed the Conceptual Browser Tree. The aim of this paper is to get a summary and correlate keywords in the document. First, creating a binary context containing objects (sentences) and properties (words). Then, after ignoring empty words, they get the optimal concept by finding the maximum weight from this binary context and generate summary keywords. An extension of tis work has been done in [22] the authors made use of the decomposition algorithm to build a browsing tree.



##### Figure 5. A view of Browsing tree [22]

In our system we used formal context which represent items as objects and their corresponding attributes as properties. This is done after we get results from online shopping system like Amazon using APIs as shown in (Figure 5-Retrieved data from Amazon) below:



##### Figure 6. Retrieved data from Amazon

###### Table 2. Binary Relationship of retrieved data from Amazon

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | S1 | S2 | S3 | S4 | S5 | Hprice | Mprice | Lprice | Hsale | Msale | Lsale |
| Item 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Item 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Item 3 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Item 4 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

We can easily discover the formal concepts from the above formal context

{item 1, item 2} {S4, Hprice, Msale}

{item 3, item 4} {s4, Mprice,Lsale}

[13] FACETEDPEDIA: DYNAMIC GENERATION OF QUERY-DEPENDENT FACETED INTERFACES FOR WIKIPEDIA

The research is concentrated on forming facet interface to help the user navigate through the resulted articles. This system will make the query generation easy for article researchers from Wikipedia. The user only enter keyword, then, Facetedpedia will provide categorized facets and each contain facets that will provides list of articles. The Facets are groups based on diverse views of the articles, in this paper they used articles attributes to form facets. First the system collects articles, then extract article attributes and group them by category, afterwards organize them by hierarchy facet category (HCF), lastly rank them. Although the system provides a dynamical generation of query-depended facets, the process is slightly time-consuming and unsteady. In this paper they gathered the data by downloading Wikipedia dump which is a complete copy of all Wikipedia wikis in XML format, they don’t have a client-server interaction with Wikipedia. The difference between this research and our system is that we have multiple providers for the data but they only use Wikipedia as the main source. On top of that, we have a direct connection with the providers since we are using API’s to get the last version of the data. They didn’t provide any kind of rules that will help the user to visualize the connection between facets.

[11] INCREMENTAL PSEUDO RECTANGULAR ORGANIZATION OF INFORMATION RELATIVE TO A DOMAIN

The authors introduce a new method that uses formal concept decomposition to come up with a system based on algorithms that handle incremental changes to information via addition operation. The aim of this paper is to manage incremental information organization and structuring in a dynamic environment. That means update the corresponding browsing tree may be optimized by using an incremental organization (i.e. update the browsing tree without recompiling the whole data). Using Incremental Pseudo Rectangular Organization can improve our solution since it supports scalability. Also, it gives our system the capability of delivering updates to the end users in a short time.

[15] STRUCTURED BROWSER OF THE DEEP WEB THROUGH SMALL SCREENS

The aim of this paper is to design and implement a structured conceptual browser for small screens. It deals with Arabic and English quires. It uses the tree to structure the retrieved results from Google. The goal of this research is to minimize the number of clicks for the user by reading the document and provide structured results, where the user will be able to navigate through the tree until he/she reaches his/her goal. In fact, the similarity between this system and ours is the features, but we have more improved system that will get better satisfaction of the user since our system is aiming to:

1. Minimize the user clicks
2. Minimize time to reach the goal
3. Represent results in a structured way.
4. Allow user to navigate through different paths.
5. Retrieved data from multiple providers
6. Querying using multi-facets

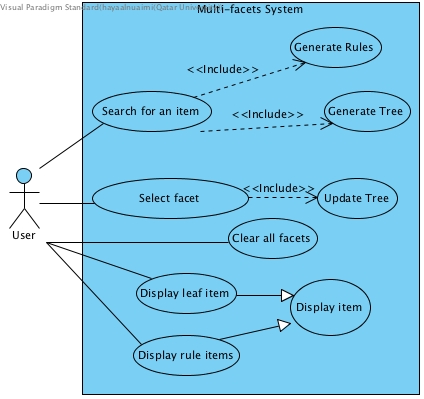
This (Table 3- Comparison between research papers) shows a comparison between the research papers and our system.

###### Table 3. Comparison between research papers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Mobile App for Hidden Data Analytics of Online Marketplaces Systems [2] | Facetedpedia [13] | Structured browser of the deep web through small screens [15] | Multi-facets online Shopping System Structuring |
| Multi-providers | √ |  |  | √ |
| Multi-facets |  | √ |  | √ |
| Extracting rules |  |  |  | √ |
| Use tree structure | √ |  | √ | √ |
| Extract data using APIs | √ |  |  | √ |
| Web application |  | √ |  | √ |
| Multiple  language | √ |  | √ |  |

# Requirements analysis

## Functional requirements



##### Figure 7. Use cases diagram

###### Table 4. Use cases summary

|  |  |
| --- | --- |
| Use case | Brief description |
| Search for an item | Searching for an item, retrieve it and store its attributes in the database |
| Select facet | Select the facet that the user wants to refine the result with |
| Generate tree | Generate tree based on user’s keyword/s typed in the search box |
| Generate rules | Generate rules and display them to the user |
| Display leaf item | Display the tree leaf item |
| Display rule item | Display item that satisfy the selected rule |
| Display item | Display an item in the provider page |
| Update Tree | Generate tree based on user facet selection |
| Clear facet | Clear all user’s previous selection |

## Non-functional requirements

###### Table 5: non-functional requirements

|  |  |
| --- | --- |
| 1. Scalability: | The system can handle a lot of users at the current time, as it depends on the online system like. For example, when dealing with data retrieved from Amazon, our system inherits the scalability from Amazon AWS. |
| 1. Modifiability: | More new attributes would be added to the system, some existing ones maybe be modified or removed to enhance further capabilities. The ability of the system to handle such modifications shows that the system is easily upgradable, modifiable and easily notified without changing or disrupting system interfaces. |
| 1. Portability: | As the program is written in java, and java is a platform independent language so, the system is could run on variety of operating system, meaning it is not specific or bounded to a certain platform, hence portable.  For example, Amazon AWS response in XML format and this format is portable to be parsed. |
| 1. Availability: | The project depends on the data from online providers, so the system will be available as long as they are. In general, the system is available to the users most of the time. |
| 1. Performance: | The project performance depends on Amazon AWS since we are running a free version and there might be some limitation in the response. |
| 1. Legal: | We use free versions of Amazon AWS API for online shopping system are used to implement the project.  Java language used in the project is free language. |
| 1. Usable: | The web application is easy to use, and the users can easily familiarize with the interface. |
| 1. Easy to use Interface: | Very easy interface, user can get familiar with. |

## Professional Code of Ethics

###### Table 6: Code of ethics

|  |  |  |
| --- | --- | --- |
| IEEE [10] | ACM (software Eng.) [19] | Project point of view |
|  | “3.13. Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.” | While developing the program, we might use Amazon AWS APIs, and possibly other marketplaces APIs for reading or downloading different types of documents, hence all resources and references used in this project are cited. |
|  | “5.09. Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed.” | A credit is given to all people who contributed in this project |
|  | “2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.” | - The work done to achieve the project goal is accomplished by students.  - The results and limitations of the system will be discussed honestly without any concealment. |
|  | “6.08. Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.” | An evaluation of the application and its performance will be carried out periodically to detect and then correct any errors that might arise later. |
| “7.To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.” |  | Accept criticism of the supervisor, and examiners, and use the comments to improve the design accordingly. |
| “8. to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression.” | “7.05. Give a fair hearing to the opinions, concerns, or complaints of a colleague.” | - The system is designed to be used and benefits any users who need it, regardless of any factor of race.  - The relationship among project team is based on respect and collaboration. |

## Assumptions

We assume that:

* The data of the providers are available.
* The categories were chosen based on electronics.
* The search word always takes the higher priority.
* The result will be based on user clicks order for the facets.

# Proposed solution

## Possible Solutions and tradeoffs

Our project has different stages and each stage can be defined as a subproject. Since every stage has a different methodology. The first Stage is about extracting data from providers, the only way is by using API’s.

The second stage, Scaling the data; converting the data to binary dataset. There are many types for scaling based on type of attributes: Nominal scaling, Dichotomic scaling and Ordinal scale. For example, this many-valued context of university majors can be scaled using several types of scaling.

###### Table 7: Many-valued context of university majors

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Gender | Age | Major |
| 1 | F | 19 | Computer Science |
| 2 | M | 20 | Computer Engineering |
| 3 | M | 19 | Computer Science |
| 4 | F | 25 | Business |

For long-string-descripted attributes we can use Porter or Lancaster algorithms to stem each word in the string. Theses stems will be provided as attributes to the scaling algorithm. Porter Algorithm uses Stemming and Lemmatization to make the scaling better and more accurate with minimum number of clusters/stems, for example the description of a bag item. On the other hand, Lancaster stemmer is the most aggressive stemming algorithm, since the word stems become too short that they are barely readable by a human anymore.

Nominal scaling can be used for “Major” attribute; this type of scaling is appropriate for binary representation of nominal attributes. We can take advantage of Porter or Lancaster algorithms as mentioned above to subdivide the string into its individual stems. A specific situation of nominal scaling is Dichotomic scaling [4], that is appropriate for attributes with dual similar values like “on” and “off”, this can be used for “Gender” attribute. Ordinal scale can be used for “Age” attribute; since it scales based on real number ranges. The resulting scaled context for our university majors example is shown below.

###### Table 8: Scaled Many-value context of university majors

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | F | M | ≤25 | ≤20 | ≤19 | Computer Science | Computer Engineering | Business |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

The third stage, Matrix Factorisation of the data; extracting concepts from binary matrix. There are some existing techniques, like, Non-Negative Matrix Factorisation (NMF), Boolean Matrix Factorisation (BMF), however these were not fully studied to be used in the framework of Information Retrieval and modern Data Analysis [4].

Boolean matrix factorisation (BMF) is a breakdown of the original matrix L ∈ {0,1} n×m, where Lij ∈ {0,1}, into a Boolean matrix product P ◦ Q of binary matrices P ∈ {0, 1} n×k and Q ∈ {0, 1} k×m for the smallest possible number of k [4]. After that building a conceptual browser tree; build a tree based on concepts using heap data-structure.

The fourth stage, generating Association Rules, mainly this is a market basket analysis problem and the most efficient proposed algorithm is Apriori, since it gives the minimal number of rules [4], compared to other algorithms like Galois connection. The later algorithm is hard to implement and we can use it for a specific case, but will not find the minimum number of rules. There is another type of Apriori algorithm called AprioriTID which is basically Apriori but it passes through the data once. AprioriTID require more space since it stores additional data structures [20]. Let <G, I, M> be a context, where n=|G| the number of objects and m=|M| is the number of attributes. Galois Connection is composed of two functions with known properties:

f:

g:

Assume that A belongs to is a subset of objects: Let B= F(A). If |A|=p, and a context (G, I, M), building B requires p\*m conjunctions where m is the number of attributes. Let q=|B|. From B we can build G(B) requiring n\*q conjunctions where q is the number of attributes of subset B and n the number of objects. The complexities of the two operators are respectively O(pm) and O(qn). So The complexity of the closures G(F(A)) is O(pm) + O(qn). The complexity of the Galois connection .

Suppose the number of unique elements is donated by R, the input transactions is N and the threshold is M. Hence the complexity of Apriori Algorithm is given as: . Another possible algorithm to use is Clustering; it is the process of making a group of abstract objects into classes of similar objects. Clustering is capable to be useful on any kind of data such as binary data, numerical data and categorical [5]. On which there are many clustering methods such as partitioning and hierarchical. Hierarchical method has two approaches Agglomerative (bottom-up) approach and Divisive (top-down) approach.

Lastly the Platform, there are 3 types of platforms Desktop application, website application or Mobile application. The differences between them can be better represented by this (Table 9-Platform comparison):

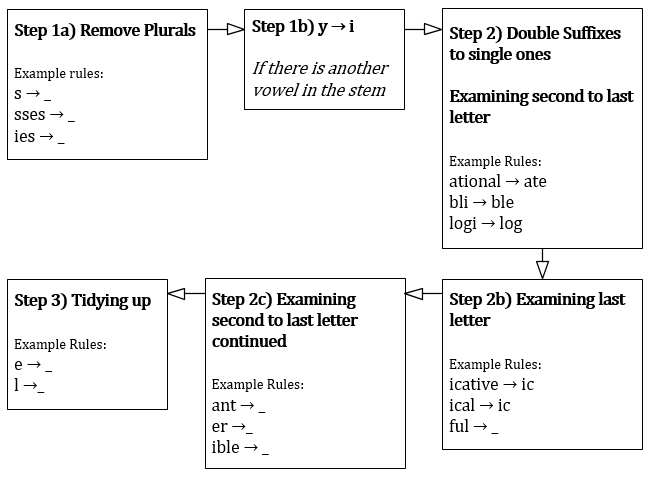
###### Table 9: Platform comparison

|  |  |  |
| --- | --- | --- |
|  | Advantages | Disadvantages |
| Desktop application | * More powerful. * **Responsive** in time matter. * Easily designed. | * **Localized**. * **Platform dependent**; application must be developed for and installed on a particular operating system. |
| Website application | * **Portable**; The user can access the application through web browser. * **Responsive** in design matter; on each different device there will be different design can be implemented using BootStrap [14]. * **Platform independent**; for clients. | * **Problem in Scalability**; needed to be taken under consideration while development. * **Performance**; need powerful hardware. * **Connectivity**; network problems can occur; while connecting to server. |
| Mobile application | * **Portable**. * **Responsive** in design matter; on each different device there will be different design can be implemented using BootStrap [14]. | * **Platform dependent**;(Android applications can’t run on IOS devices and vice versa). * **Performance**; need powerful hardware; since till now mobile hardware is not comparable to laptops. * **Connectivity**; network problems can occur; while connecting to server. |

## Selected Solution overview

Our project has different stages and we are aiming to choose the most suitable way to implement our project. In the first Stage we are going to extracting data from the provider database by using their API’s, detailed description of retrieved data mentioned above.

Secondly in Scaling we are going to use Nominal scaling for the item’s title by taking advantage of Porter Algorithm to stem each word in the title. We preferred using Porter Algorithm because it is the most commonly used stemmer. The Ordinal scale is important to us since most of our data are real numbers. Moreover, we customized the Ordinal scaling based on the attributes minimum and maximum. For example, in the price attribute we calculated the minimum, maximum and the range, then used the minimum to be the lower boundary of the first cluster, the minimum plus half of the range to be the lower boundary of the second cluster and the minimum plus the range to be the lower boundary of the third cluster. Hence we can have three equally divided clustered.



##### Figure 8. Flow chart of Porter Algorithm.

The third stage, extracting concepts from binary matrix using Boolean Matrix Factorisation (BMF) since it is the only factorization technique which is fully studied to be used in the context of Information Retrieval and modern Data Analysis. After that we are going to use Conceptual Browsing Tree Algorithm to build our tree of concepts. The fourth stage, generating Association Rules, the most efficient proposed algorithm is Apriori Algorithm, because it gives the minimum number of rules [18].

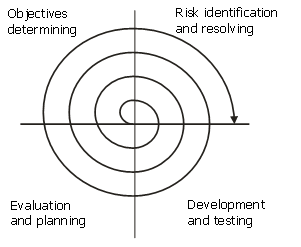
Lastly the in terms of Platform, we chose to build a website application hence it’s portable, responsive in design, platform independent and it reach our client’s need based on a survey we conducted. Showed that 55% of the people whom solved the survey prefer shopping using a Website rather than other platforms.

We can reuse open sources and adapt them to our multi-facet it would be a better usability since we can take advantage of other people work rather than building all algorithms from scratch then modify it to benefit our target.

**Life Cycle:**

Since our system should provide high maintainability, performance and accuracy. Therefore, spiral model has been used in our system to easy modify in case of failure or improve the system in the future.

The Spiral life cycle model is a type of iterative software development model. This model combines the features of both, prototype model and waterfall model. In Spiral model each iteration represents a stage of our system (and we can have any number of iterations according to the project). Each iteration has four sections: determine objectives, risk analysis, testing and evaluation. (Figure 8- Illustrates Spiral life cycle model) illustrates these sections.



##### Figure 9. Illustrates Spiral life cycle model [6]

1. Determine objectives: during this phase, we determine the objectives, alternatives and constraints [83]. We also collect the data and the requirements like system requirements specifications.

2. Risk analysis and evaluation of alternatives: here we identify risks and try to find possible solution that can be implemented in order to achieve the identified constraints.

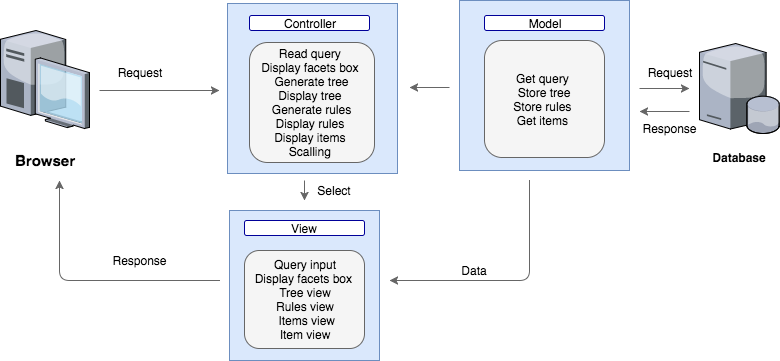
3. Development and testing: implementation using detail design, and testing to ensure that the application will not result into any failure. So, in this phase we validate and verify that the software meets the requirements, work as expected and can be implemented with the same characteristics.

4.Evaluation: here we review the progress and ask the supervisor to evaluate the output of the project before the project continues to the next spiral [83].

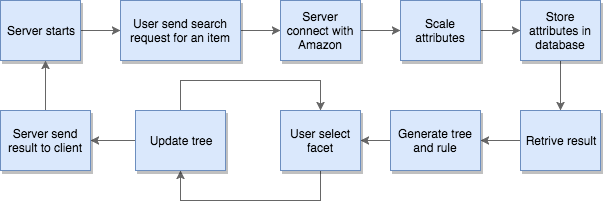
## High level architecture

The architecture used is MVC design pattern that consider a good approach in designing web application, since MVC design pattern supports different types of client at runtime. On top of that the complexity is reduced since each component has different logic.

MVC design pattern separates the view from the application functionality and behind this view there is an engine that is used to control the (request, response) procedure, this engine is called in programming manner (controller). Moreover, there are many benefits of using MVC design pattern, like designing, developing and modifying web application can be done by different people (separation of concerns - modifiability); because some people can design very attractive interface and others feel more comfortable with code, this distribution of work cannot be beneficial for our project. In our project, the view consists of five activities such that query input, display facets box, tree view and items view. The controller handles the callbacks of the actions the user provides, and it acts as an intermediate between the view and the model; it passes input to the model and provides data to the view. The model is responsible of communicating with Provider, creating the tree, and generating rules. (Figure 9- High Level Architecture) shows the architecture of our application.



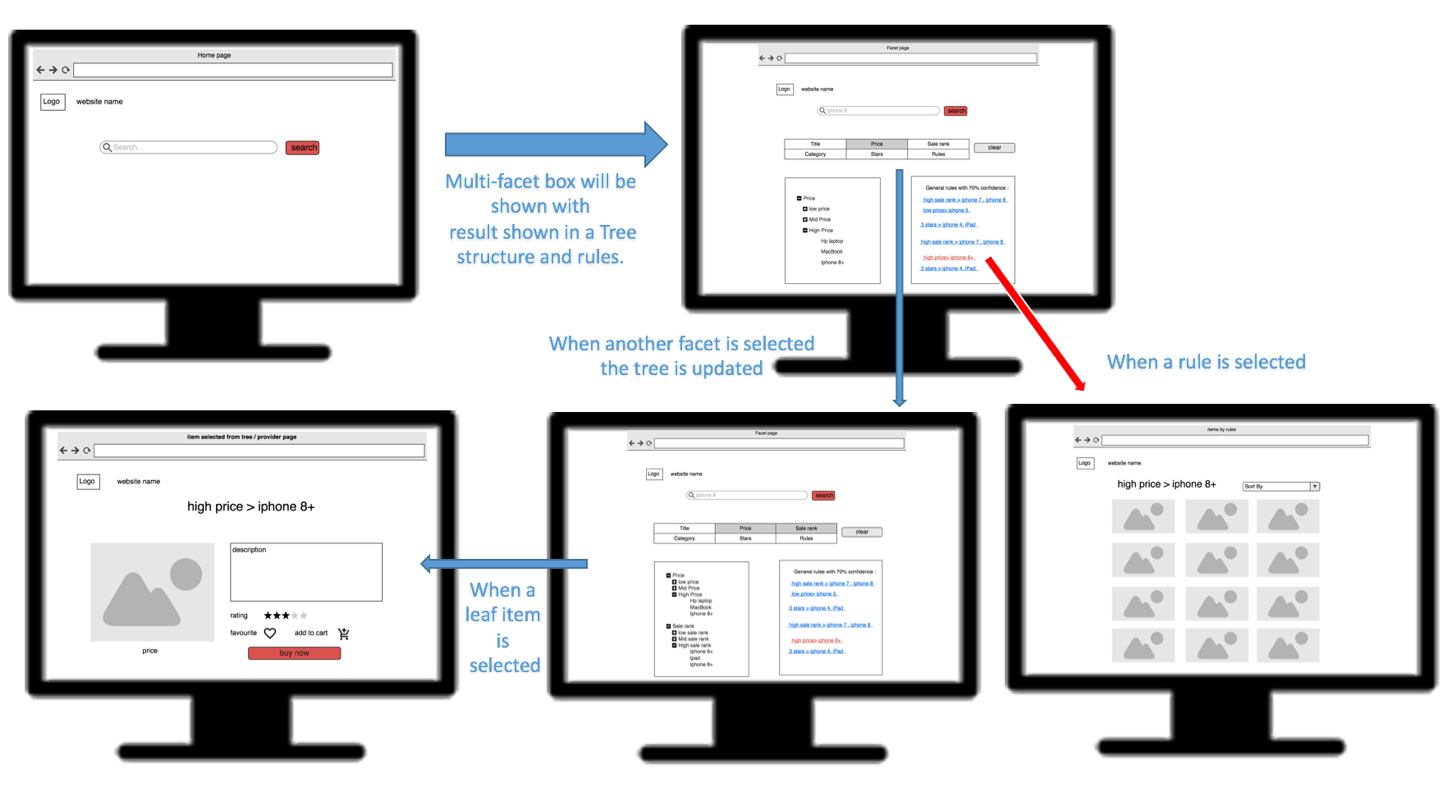
##### Figure 10. High level Architecture

****

##### Figure 11. Project Flow chart

**Interface Design**

To choose an architecture that will offer an application both its simplicity in utilization and to be operative. We proposed an interface for the diverse activities our application will have and determine how the client will explore through them.



##### Figure 12. Interface design

As shown in the interface design in (Figure 11- Interface Design), the detail web application we are going to develop will have a multi-facets filtering. The initial concern is to narrow the search query in a way that will structured in a tree consequently the user can easily browse the results.

## Hardware/software to be used

* **Hardware:**

###### Table 10: Hardware

|  |  |  |
| --- | --- | --- |
| **Device** | **Operating System** | **Justification** |
| **MacBook** | Mac OS Sierra (OS X 10.11) | To build the application |
| **Lenovo Yoga 2 pro** | Windows 10 |

* **Software:**

###### Table 11: Software

|  |  |  |
| --- | --- | --- |
| Category | Tools | Justification |
| **Graphics** | Adobe Photoshop - moqups  draw.io | To design the web application’s interface, pages, logo, mockup and layout. |
| **Development of source code** | Eclipse. | Eclipse: for implementing and testing some of the algorithms. |
| **Database development** | SQLite Studio | To create databases |
| **Programming languages** | Java.  Java, HTML, CSS and JavaScript | Programming language used to develop our application  To implement the web application, we will use Java, HTML, CSS and JavaScript. |
| **Modeling** | Visual Paradigm for (UML). | To model the project as class diagram, activity diagram etc. … |
| **Documentation** | Microsoft Office:(Microsoft PowerPoint – Microsoft Word) | To document every step in the project with references. Additionally, to create the project’s presentation, report. |

# Project plan

## Project milestones

###### Table 12. Milestone Fall 2017

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone | No. | Tasks | Description |
| M1:  Understand the project idea | T1 | Determine the problem | Identify the project problem. |
| T2 | Identify objectives | Identify the purpose of the project. |
| T3 | Project abstract | Write a short paragraph explaining the project. |
| T4 | Project background | Write some paragraphs about the project background. |
| M2:  Related Work | T5 | Read and search for related work | Read and search for work done before in the same field of our project. |
| M3:  Requirements Analysis  Merge (requirement analysis) | T6 | Use-case diagram | Determine our system use cases. |
| T7 | Use-case specifications | Provide the use-case specification for each use-case. |
| T8 | Write the non-functional requirements | Identify the non-functional requirements for the system. |
| T9 | Write Code of Ethics | Identify responsibilities relevant to our project. |
| T10 | Identify assumptions | List our project assumptions. |
| M4:  Proposed Solution | T11 | Solution overview | Provide all possible solutions for all stages of our system |
| T12 | Selected solution then (t3,t4) | Discuss appropriate solutions for our system |
| T13 | High level architecture | Choose an appropriate architecture and describe each component. |
| T14 | Hard & Software used to be used | Classify and list all the hardware and software that will be used in this system. |
| M5:  Extracting data | T15 | Learn about hidden data | Discover ways to find hidden data. |
| T16 | Explore ways to extract data | Search and read about ways to retrieve data from online shopping systems. |
| T17 | Extract data using APIs through a java code | Write a code using APIs to extract hidden data |
| T18 | Store data in a database | Store data using SQLite  [SQLite is an embedded SQL database engine] |
| T19 | Scaling data | Convert the extracted data into 0’s and 1’s matrix |
| M6:  Explore and study algorithms | T20 | Learn about different algorithms | Search and explore different algorithms that can be used and was used in the same field. |
| M7:  Project Plan | T21 | List project milestones and timeline | Identify project milestones and break them to tasks. Then, create the timeline. |
| T22 | Write anticipated risks | Write some assumptions about expected risks during implementation. |
| M8:  Implementation | T23 | Design mock interface | Design a mock interface and provide the application cycle. |
| T24 | Draw a flow chart | Draw a flow chart to illustrate all phases of the project |
| M9:  Report finalizing | T25 | Finalize the report | Proofread the report and check all parts. |

###### Table 13. Milestone Spring 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone | No. | Tasks | Description |
| **M1: System design** | T1 | Class diagram | Create structural model for our system |
| T2 | Sequence diagrams | Using sequence diagrams for each use-case we will determine the system’s behavioral model. |
| T3 | Activity diagram | Another diagram to determine the system behavior. |
| T4 | State diagram | Identify the system states. |
| T5 | Database design | Describe the entities and the relationship between them |
| T6 | Design pattern | Distinguish the design patterns applied for our design. |
| **M2:**  **Implementation** | T7 | Implement classes | Implement all classes in the class diagram. |
| T8 | Implement functions | Implement functions in the class diagram. |
| T9 | List tools | List all the tools used for implementation. |
| T10 | Identify challenges | List challenges met while implementing. |
| **M3:**  **Testing** | T11 | Testing | Test each milestone since we are using spiral model |
| **M4:**  **Finalizing** | T12 | Expected impact | Determine the social impact of our solution on individuals and society. |
| T13 | Conclusion | Write a short paragraph that describe the whole system |
| T14 | Future work | Discover parts in the project to enhance by attaching more features. |
| T15 | Students reflection | Each student writes about the gained experience of this project |

## 

## Project timeline

###### Table 14. Timeline Fall 2017

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M No. | T No. | Tasks | Duration | Beginning date | Completion date | Dependency | H | I | A |
| **M (1)** | T1 | Determine the problem | 1 week | **27/9/2017** | **3/10/2017** | - | √ | √ | √ |
| T2 | Identify objectives | 3 days | **3/10/2017** | **5/10/2017** | - |  |  | √ |
| T3 | Project abstract | 5 days | **5/10/-2017** | **9/10/2017** | - |  | √ |  |
| T4 | Project background | 1 week | **9/10/2017** | **15/10/2017** | - | √ | √ |  |
| **M (2)** | T5 | Read and search for related work | 3 weeks | **15/10/2017** | **17/10/2017** | M(1) | √ | √ | √ |
| **M (3)** | T6 | Draw the use case diagram | 3 days | **17/10/2017** | **19/10/2017** | - |  |  | √ |
| T7 | Write the use case specifications | 4 days | **19/10/2017** | **22/10/2017** | T6 |  |  | √ |
| T8 | Write the non-functional requirements | 2 days | **22/10/2017** | **23/10/2017** | - |  | √ |  |
| T9 | Write Code of Ethics | 2 days | **23/10/2017** | **24/10/2017** | - | √ |  |  |
| T10 | Identify assumptions | 1 days | **24/10/2017** | **24/10/2017** | - |  | √ |  |
| **M (4)** | T11 | Solution Overview | 3 days | **24/10/2017** | **26/10/2017** | - | √ |  |  |
| T12 | Selected solution | 2 days | **26/10/2017** | **27/10/2017** | T11 |  |  |  |
| T13 | High level-architecture | 2 days | **27/10/2017** | **28/10/2017** | T12 |  | √ |  |
| T14 | Hard & Software used | 2 days | **28/10/2017** | **29/10/2017** | - | √ |  |  |
| **M (5)** | T15 | Learn about hidden data | 3 days | **29/10/2017** | **31/10/2017** | - | √ | √ | √ |
| T16 | Explore ways to extract data | 2 days | **31/10/2017** | **1/11/2017** |  | √ | √ | √ |
| T17 | Extract data using APIs through a java code | 4 days | **1/11/2017** | **4/11/2017** | - | √ | √ | √ |
| T18 | Store data in a database | 2 days | **4/11/2017** | **5/11/2017** | T17 | √ | √ | √ |
| T19 | Scaling data | 5 days | **5/11/2017** | **9/11/2017** | T18 | √ | √ | √ |
| **M (6)** | T20 | Learn about different algorithms | 3 weeks | **9/11/2017** | **30/11/2017** | M(1) | √ | √ | √ |
| **M (7)** | T21 | List project milestones and timeline | 2 days | **30/11/2017** | **1/12/2017** | M(1)🡪M(6) | √ |  |  |
| T22 | Write anticipated risks | 2 days | **1/12/2017** | **2/12/2017** | - |  |  | √ |
| **M (8)** | T23 | Design mock interface | 2 days | **2/12/2017** | **3/12/2017** | - |  |  | √ |
| T24 | Draw flow chart | 3 days | **3/12/2017** | **5/12/2017** |  |  |  | √ |
| **M (9)** | T25 | Report finalization | 2 weeks | **5/12/2017** | **19/12/2017** | M(1)🡪M(9) | √ | √ | √ |

###### Table 15. Timeline Spring 2018

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M No. | T No. | Tasks | Duration | Beginning date | Completion date | Dependency | H | I | A |
| **M (1)** | T1 | Class diagram | 4 days | 11/2/2018 | 14/2/2018 | - | √ | √ | √ |
| T2 | Sequence diagrams | 10 days | 14/2/2018 | 23/2/2018 | T1 | √ | √ | √ |
| T3 | Activity diagram | 1 week | 23/2/2018 | 1/3/2018 | T2 | √ | √ | √ |
| T4 | State diagram | 1 week | 1/3/2018 | 7/3/2018 | T2 | √ | √ | √ |
| T5 | Database design | 2 weeks | 7/3/2018 | 21/3/2018 | T1 | √ | √ | √ |
| T6 | Design pattern | 3 days | 21/3/2018 | 23/3/2018 | - | √ | √ | √ |
| **M (2)** | T7 | Implement classes | 5 days | 23/3/2018 | 27/3/2018 | M1 | √ | √ | √ |
| T8 | Implement functions | 3 weeks | 27/3/2018 | 17/4/2018 | M1, T7 | √ | √ | √ |
| T9 | List tools | 2 days | 17/4/2018 | 18/4/2018 | - | √ | √ | √ |
| T10 | Identify challenges | 2 days | 18/4/2018 | 19/4/2018 | - | √ | √ | √ |
| **M (3)** | T11 | Testing | 2 weeks | 19/4/2018 | 1/5/2018 | T7, T8 | √ | √ | √ |
| **M (4)** | T12 | Expected impact | 3 days | 1/5/2018 | 3/5/2018 | - | √ | √ | √ |
| T13 | Conclusion | 2 days | 3/5/2018 | 4/5/2018 | - | √ | √ | √ |
| T14 | Future work | 4 days | 4/5/2018 | 7/5/2018 | - | √ | √ | √ |
| T15 | Students reflection | 2 days | 7/5/2018 | 8/5/2018 | - | √ | √ | √ |

## Anticipated risks

###### Table 16. Anticipated risks

|  |  |  |
| --- | --- | --- |
| Risk Type | Risk Conditions | Risk Action |
| Time | Failure to handle the time due to different schedules among the team members | We are trying to avoid this risk by arranging weekly meeting with the supervisor, and his research team. We have also regular meetings with our team members to make plan, discuss and distribute the work among us. Moreover, we are trying to follow the previously shown timeline table we planned in the beginning. |
| Technical | Machine Failure or damaged | We are using Version control such as GitHub to share files and keep track of all updates |
| Knowledge  Challenge | Lack of knowledge and misunderstanding of the idea of the project at the beginning.  We are not sure about which is the best algorithm we can use to make the generate tree in fastest and more efficient way | Professor Ali and his research team explain more about the idea and topic. we also did a lot of research on what algorithms are better than others, and compare the complexity and accuracy each of them will yield. |
| License expiration | UML license expired during work on the project | We asked Dr. Khaled Khan (software engineering experts) for student license for UML paradigm. It is one year, so it’s enough for us to continue working on the project. |

# 

# References

# [1] Ahuja, Babita. Juneja, Dimple. Anuradha ‘Hidden Data Extraction Using URL Templates Processing’. Big Data Analytics: *Proceedings of CSI 2015* .Eds. V.B. Aggarwal, Vasudha Bhatnagar and Durgesh Kumar Mishra. Springer Nature. 111-126.Print.

[2] Aqle, A. (2016, January). Mobile App for Hidden Data Analytics of Online Marketplaces Systems. Retrieved December 18, 2017, from http://qspace.qu.edu.qa/bitstream/handle/10576/5652/AboubakrAqle\_OGS\_Approved\_Manuscript.pdf?sequence=1&isAllowed=y

[3] Belohl ́avek, R.: Introduction to formal context analysis. Tech. rep., Dept of

Computer science. Palack y University, Olomouk, Czech Republic 2008.

[4] Dmitry I. Ignatov, Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields.

[5] E.-H. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering Based on Association Rule Hypergraphs. In *Proc. of the 1997 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Tucson, AZ, 1997.

[6] Figure 8 - Illustrates Spiral life cycle model retrieved from: https://www.google.com/search?biw=1536&bih=710&tbm=isch&sa=1&ei=mwU1Wt-GIo2P0gWV3qSQDQ&q=spiral+lifecycle+model&oq=spiral+life&gs\_l=psy-ab.3.0.0l4j0i8i30k1l6.1817.2734.0.5672.5.5.0.0.0.0.384.1069.0j4j0j1.5.0....0...1c.1.64.psy-ab..0.5.1068...0i67k1.0.Aysn9bc4Fdk#imgrc=jmqCGOJrufRW7M:

[7] Gemma C. Garriga (69). (1970, January 01). Claude Sammut. Retrieved November 13, 2017, from https://link.springer.com/referenceworkentry/10.1007/978-1-4899-7687-1\_316

[8] Gupta, S and Bhatia, K. K., *A Comparative Study of Hidden Web Crawlers*, International Journal of Computer Trends and Technology, Vol 12(3), 2014.

# [9] Hull, D. (n.d.). Stemming algorithms: A case study for detailed evaluation. Journal of the American Society for Information Science J. Am. Soc. Inf. Sci., 70-84.

[10] IEEE IEEE Code of Ethics. (n.d.). Retrieved November 26, 2017, from https://www.ieee.org/about/corporate/governance/p7-8.html#top

[11] Ismail, S., & Jaoua, A. (1970, January 01). Incremental Pseudo Rectangular Organization of Information Relative to a Domain. Retrieved December 18, 2017, from https://link.springer.com/chapter/10.1007%2F978-3-642-33314-9\_18

[12] Jihad M. Alja'am, Ali Jaoua, Ahmad Hasnah, F. Hassan, H. Mohamed, T. Mosaid, H. Saleh, F. Abdullah, H. Cherif: Text Summarization Based on Conceptual Data Classification. IJITWE 1(4): 22-36 (2006).

[13] Li, C., Yan, N., Roy, S. B., Lisham, L., & Das, G. (n.d.). *Facetedpedia: Dynamic Generation of Query-Dependent* ... Retrieved November 15, 2017, from http://www.bing.com/cr?IG=9B78635BE29749079C4D316BA8371165&CID=3438E4568A476C45334CEF0F8BE86DFD&rd=1&h=frWSxOno26h6PtESlYkFElRzc7y1lL9km9pyumLc7-8&v=1&r=http%3a%2f%2franger.uta.edu%2f%7ecli%2fpubs%2f2010%2ffacetedpedia-www10-lyrld-feb10.pdf&p=DevEx,5066.1

[14] Mark Otto, Jacob Thornton, and Bootstrap contributors. (n.d.). Bootstrap. Retrieved December 02, 2017, from https://getbootstrap.com/

[15] Mehawesh A, AlNisf M, AlMesaifri M. Structured Browser of the Deep Web Through Small Screens. Poster presented at, Eighth Annual Undergraduate Research Conference on Applied Computing. 2016 April 27-28; Dubai, UAE.

[16] Mecheter, S. (2014, December). Conceptual Data Reduction: Application for Text Summarization and Corpus Representation. Retrieved December 13, 2017, from http://qspace.qu.edu.qa/bitstream/handle/10576/3253/Mecheter,%20Souad%20Nourreddine.pdf?sequence=1

[17] Mogotsi, I. (2009). Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze: Introduction to information retrieval. Information Retrieval Inf Retrieval, 177-195.

*Proc. of the 20th Int'l Conference on Very Large Databases*, 1994.

[18] R. Agrawal, R. Srikant: "Fast Algorithms for Mining Association Rules",

Says, A., Says, J. K., Says, M. T., Says, A. S., Says, M. M., Says, C. T., . . . Says, P. P. (n.d.). Home. Retrieved December 16, 2017, from http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/

[19] Software Engineering Code. (2017, November 18). Retrieved November 26, 2017, from https://ethics.acm.org/code-of-ethics/software-engineering-code/

[20] Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Boston: Pearson Addison Wesley.

[21] Yang, Y., Ma, M., & Liu, B. (2013). *Information Computing and Applications 4th International Conference*, ICICA 2013, Singapore, August 16-18, 2013 Revised Selected Papers, Part II. Berlin, Heidelberg: Springer Berlin Heidelberg.

[22] Structures Conceptual Meta-Search Engine, A. Jaoua, M. Lazhar Saidi, Ahmed, Baina Salem, Noura Rashid Shereen Shareef, Suad Zaghlan. Fourth Intermational Conference on Concept Lattices and Their Application, Hammamet-Tunisia 2006,30/10 au 1/11/2006

# Appendix A – Use cases specification

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**01** | Search for an item | |
| **Brief Description** | Searching for an item, retrieve it and store its attributes in the database | |
| **Primary actors** | User | |
| **Preconditions:**  Connection to the network | | |
| **Post-conditions:**   1. All Items’ attributes stored in the database 2. Facets Box contains attributes displayed to the user | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User type a keyword/s inside the search box | | 2. Get the input from the search box |
|  | | 3. Search for the item inside the online provider’s database **(**See 3.a. For alternative flow) |
|  | | 4. Retrieve the matched items in XML |
|  | | 5. Extract specific attributes from each item retrieved |
|  | | 6. Store item’s attributes inside the database |
|  | | 7. Display the facets box contains those attributes for a user to select |
|  | | 8. <include: Generate rules use case>. |
|  | | 9. <include: Generate Tree use case>. |
| **Alternative flows:**  3.a. If the input is not existing, display massage “item not found, please reinter an item with English word” | | |
| **Special Requirements:**  Availability, scalability, performance, easy to use interface. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**02** | Select facet | |
| **Brief Description** | Select the facet that the user wants to refine the result with | |
| **Primary actors** | User | |
| **Preconditions:**  Item must exist | | |
| **Post-conditions:**  Facets selected | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User select a facet | | 2. Make the selected facet disabled (See 2.a. For alternative flow) |
|  | | 3. <include: Update Tree use case> |
| **Alternative flows:**  2.a if facet already selected, make facet enable. | | |
| **Special Requirements:**  Easy to use interface, performance. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**03** | Generate tree | |
| **Brief Description** | Generate tree based on user’s keyword/s typed in the search box | |
| **Primary actors** | User | |
| **Preconditions:**  User type a keyword/s in search box | | |
| **Post-conditions:**  Tree is generated | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1.user clicks on search button | | 2. Get user input from search box |
|  | | 3. Create a root node which is the keyword |
|  | | 4. Get data from contact database |
|  | | 5. Filter to matched items (See 5.a. For alternative flow) |
|  | | 6. Link root with children which is retrieved items |
| **Alternative flows:**  5.a. If there are no matched items, display a massage “not found” | | |
| **Special Requirements:**  Easy to use interface, performance | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**04** | Generate rules | |
| **Brief Description** | Generate rules and display them to the user | |
| **Primary actors** | User | |
| **Preconditions:**  User search for an item | | |
| **Post-conditions:**  Rules generated and displayed to the user | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User search for item by entering a keyword in the search box | | 2. Contact with database |
|  | | 3. Generate rules using Galois algorithm |
|  | | 4. Filter out rules that are less than 70% confidence |
|  | | 5. Display filtered rules |
| **Alternative flows:** | | |
| **Special Requirements:**  Easy to use interface, performance. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**05** | Display tree item | |
| **Brief Description** | Display the leaf item | |
| **Primary actors** | User | |
| **Preconditions:** Tree exist | | |
| **Post-conditions:** Item displayed in the provider page | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User clicks on one of the leaf nodes | | 2. Identify the selected leaf |
|  | | 3. Get ASIN for that node |
|  | | 4.Perform Display item use case. |
| **Alternative flows:**  2.a. If the user clicks on non-leaf node, extend tree | | |
| **Special Requirements:**  Easy to use interface, performance. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**06** | Display item | |
| **Brief Description** | Display the item in the provider page | |
| **Primary actors** | User | |
| **Preconditions:** Tree exist, or rules exist | | |
| **Post-conditions:** Item displayed in the provider page | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User clicks on an item | | 2. Contact the database |
|  | | 3. Get link for this item |
|  | | 4. Route user to the link |
|  | | 5. Display page |
| **Alternative flows:** | | |
| **Special Requirements:**  Easy to use interface, performance availability | | |

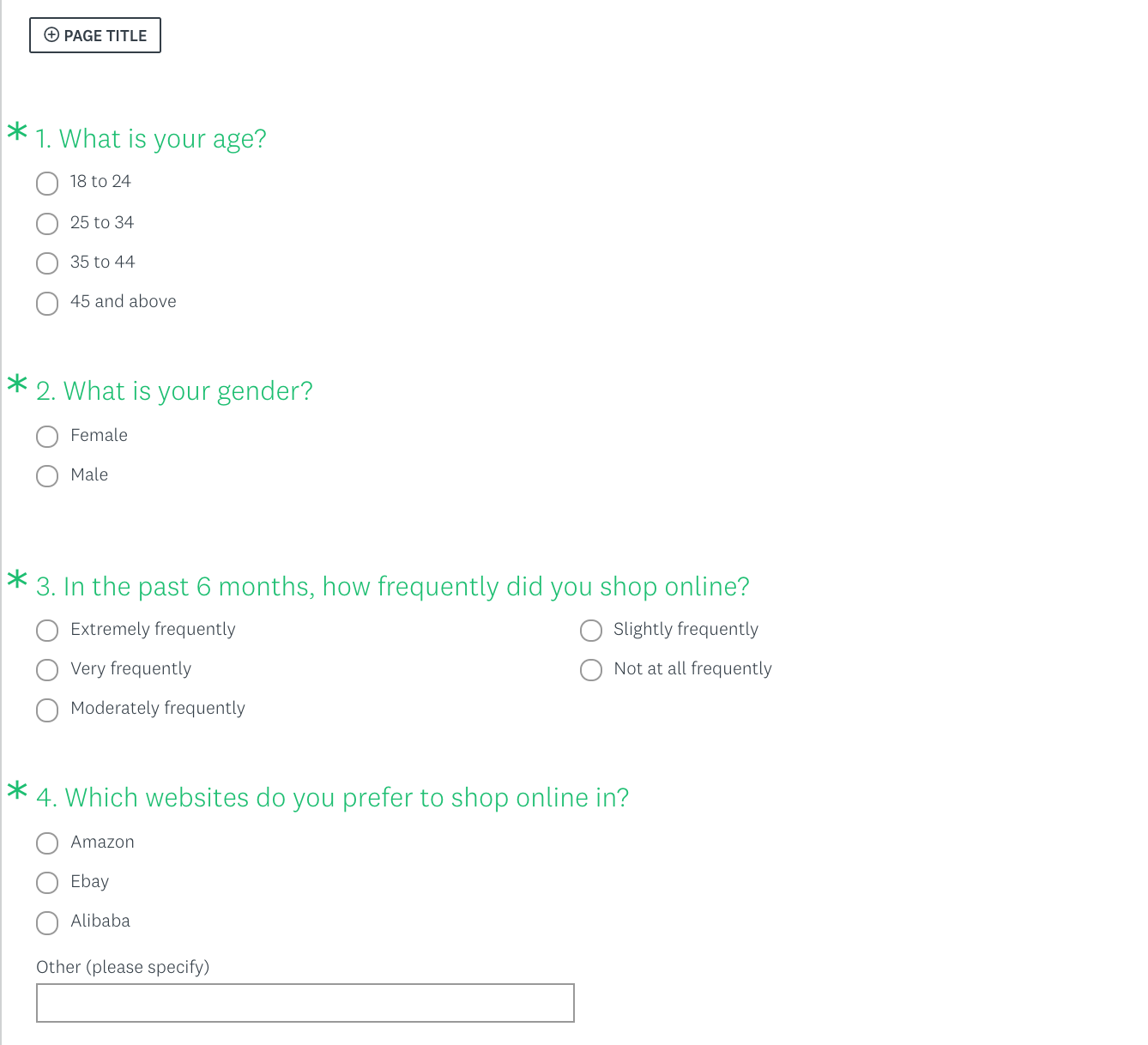
|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**07** | Display rule item | |
| **Brief description** | Display item that satisfy the selected rule | |
| **Primary actors** | User | |
| **Preconditions:**  User select rule | | |
| **Post-conditions:**  Item displayed | | |
| **Main success scenario:** | | |
| **Actor action** | | **System response** |
| 1. User select desired rule | | 2. Display items that is satisfy the rule |
| 3. User select an item | | 4.Perform Display item use case. |
| **Alternative flows:** | | |
| **Special requirements:**  Easy to use interface, performance | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**08** | Update tree | |
| **Brief Description** | Update tree based on the facet/facets that the user selects | |
| **Primary actors** | User | |
| **Preconditions:**  Facet selected | | |
| **Post-conditions:**  Tree updated | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User select facet/s | | 2.link root node with a child node which is the facet selected |
|  | | 3. Get the attributes for the facet from the database |
|  | | 4. Link facet node with its attributes |
|  | | 5. Retrieve matched items from database (See 5.a. For alternative flow) |
|  | | 6. Link items with its parent |
| **Alternative flows:**  5.a. If items not found, display a massage “don’t found” | | |
| **Special Requirements:**  Easy to use interface, performance | | |

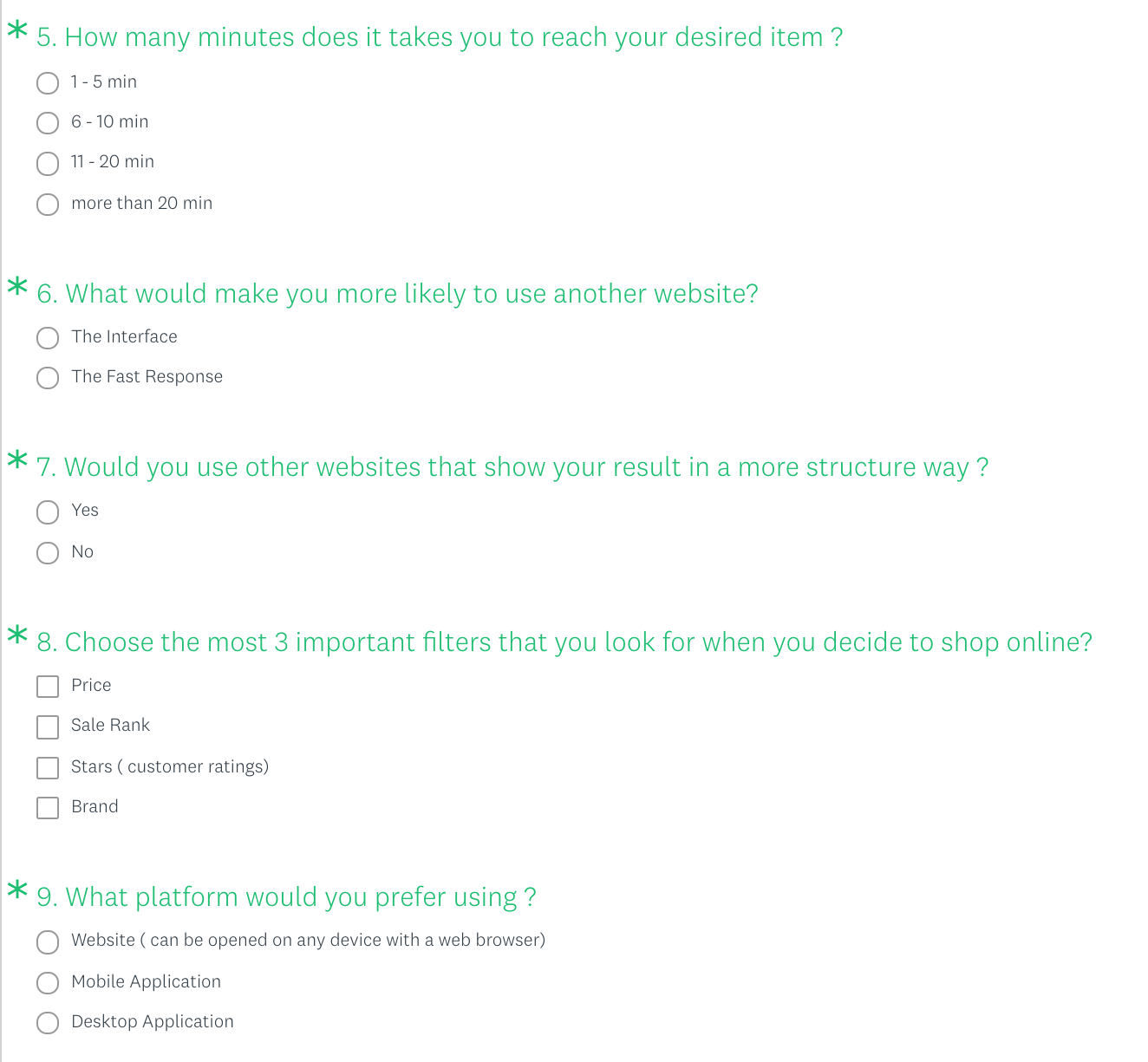
|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC**09** | Clear all facets | |
| **Brief Description** | Clear all user’s previous selection | |
| **Primary actors** | User | |
| **Preconditions:**  There is at least one facet already chosen | | |
| **Post-conditions:**  No facet selected | | |
| **Main Success Scenario:** | | |
| **Actor Action** | | **System Response** |
| 1. User clicks on button clear facet | | 2. Remove the facet/facets selection |
|  | | 3. Remove from the tree from the page |
|  | | 4. Get keyword tree. |
|  | | 5. Show tree to the user. |
| **Alternative flows:** | | |
| **Special Requirements:**  Easy to use interface, performance | | |

# Appendix B – Survey and Survey Analysis

* Survey Questions:

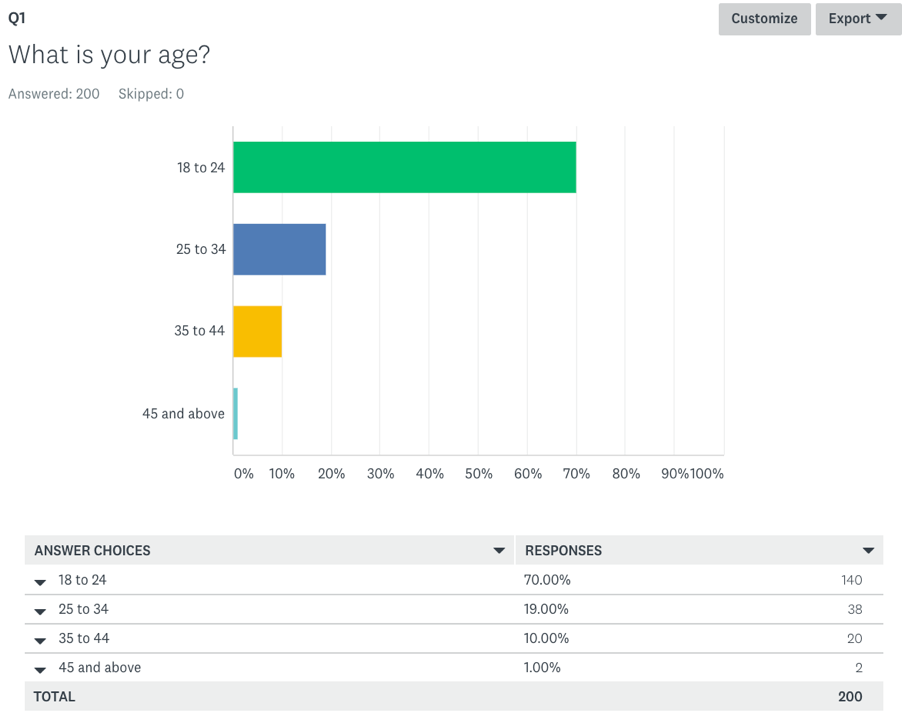


##### Figure 13. Survey questions #1.



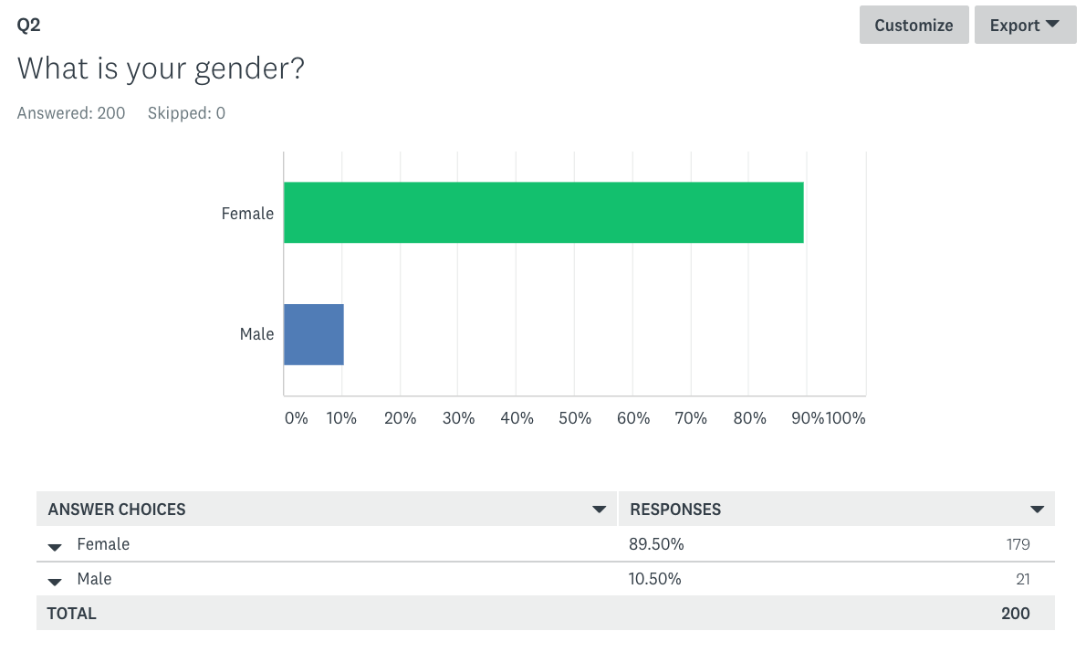
##### Figure 14. Survey questions #2.

* Questions Analysis:
* Analysis of Question # 1:



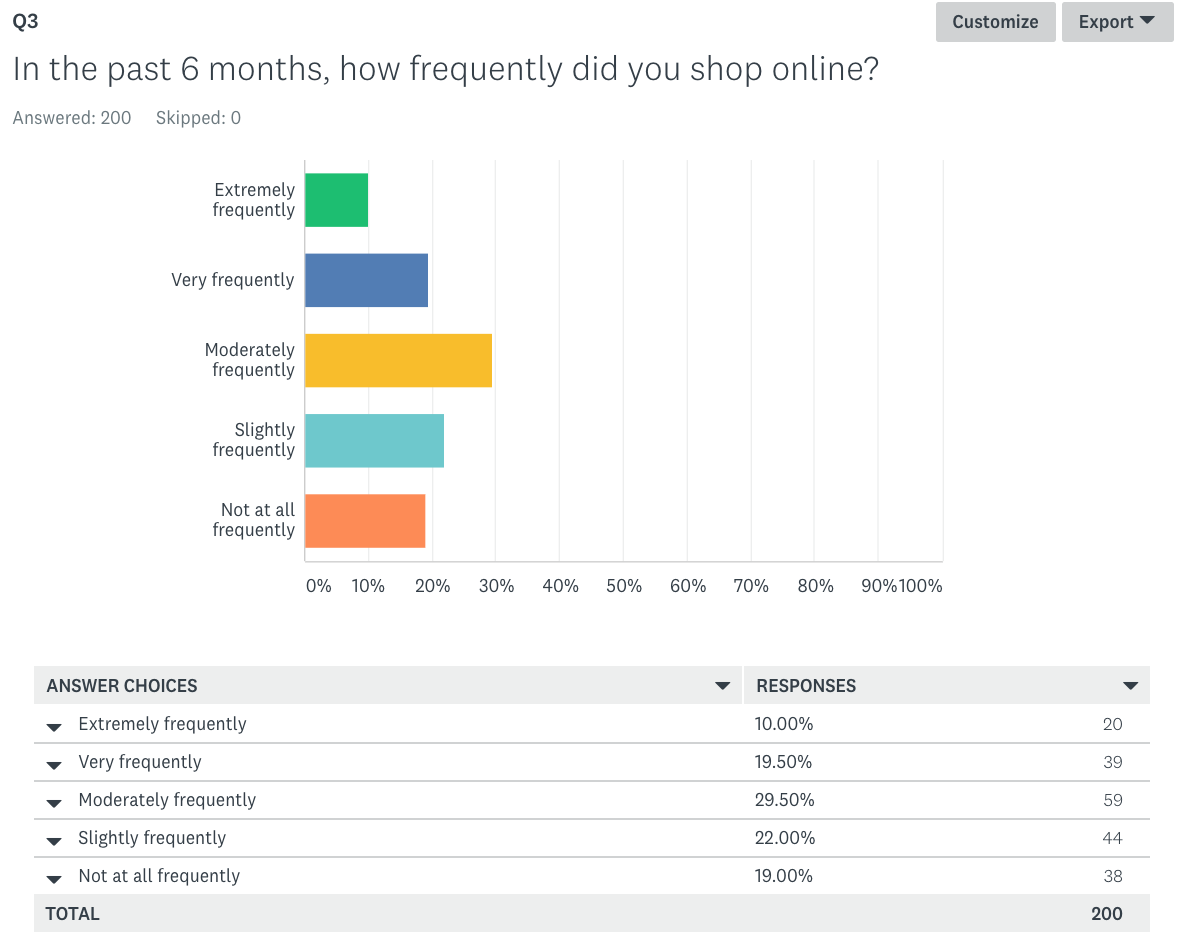
##### Figure 15. Analysis of Question # 1.

* Analysis of Question # 2:



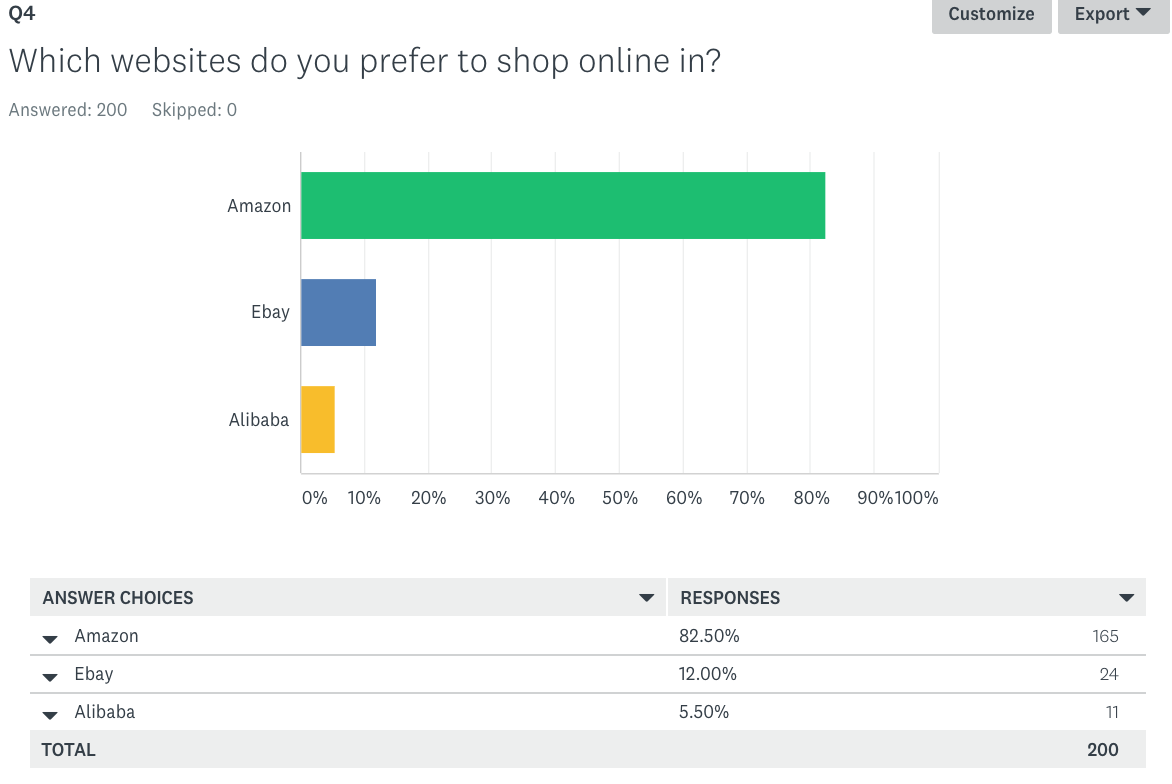
##### Figure 16. Analysis of Question # 2.

* Analysis of Question # 3:



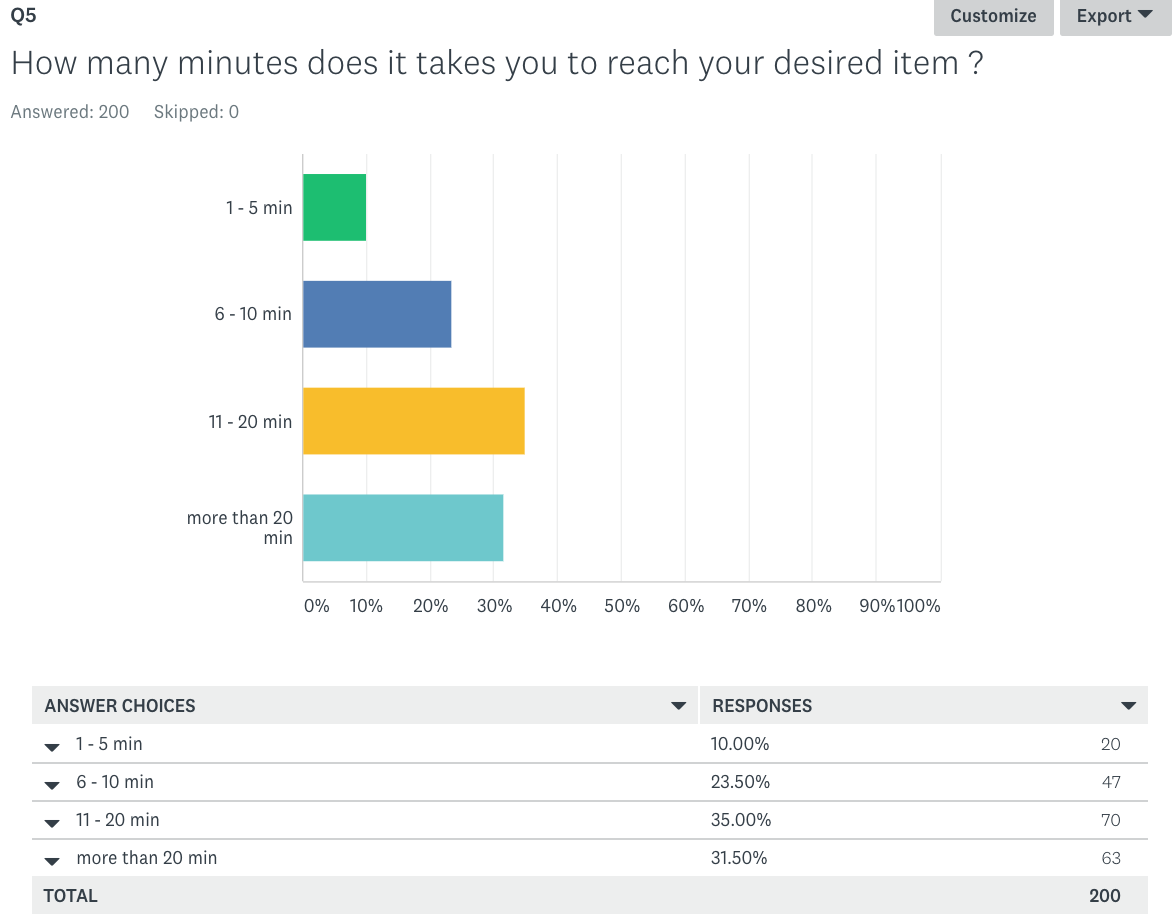
##### Figure 17. Analysis of Question # 3.

* Analysis of Question # 4:



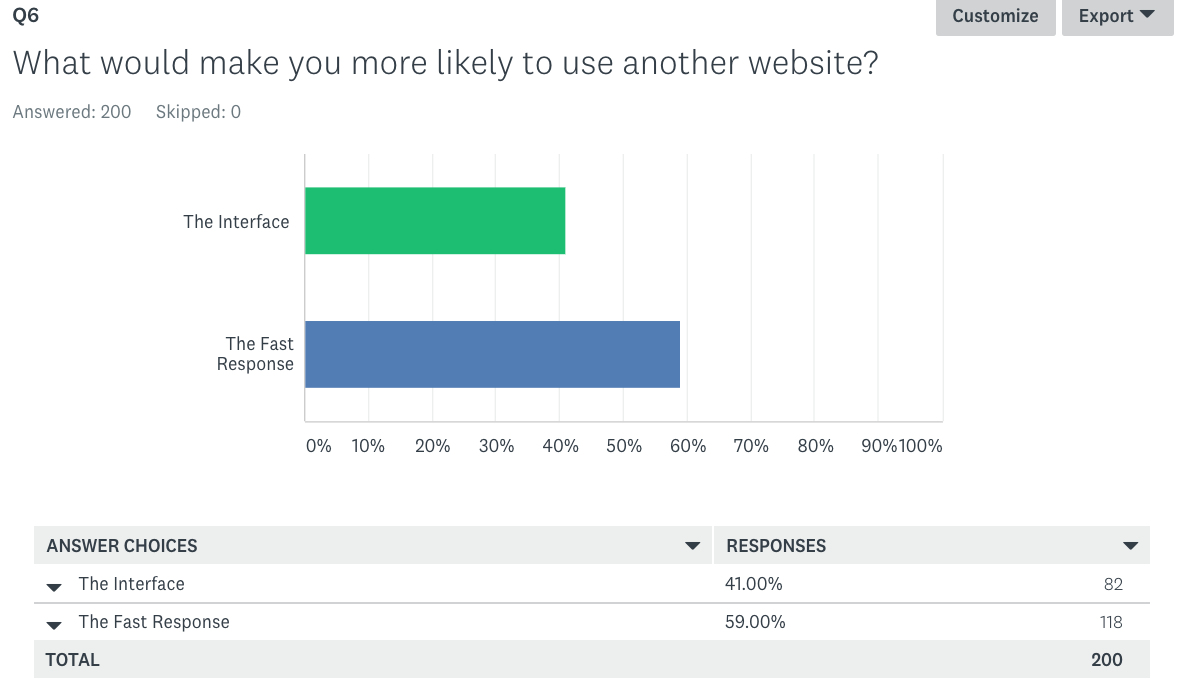
##### Figure 18. Analysis of Question # 4.

* Analysis of Question # 5:



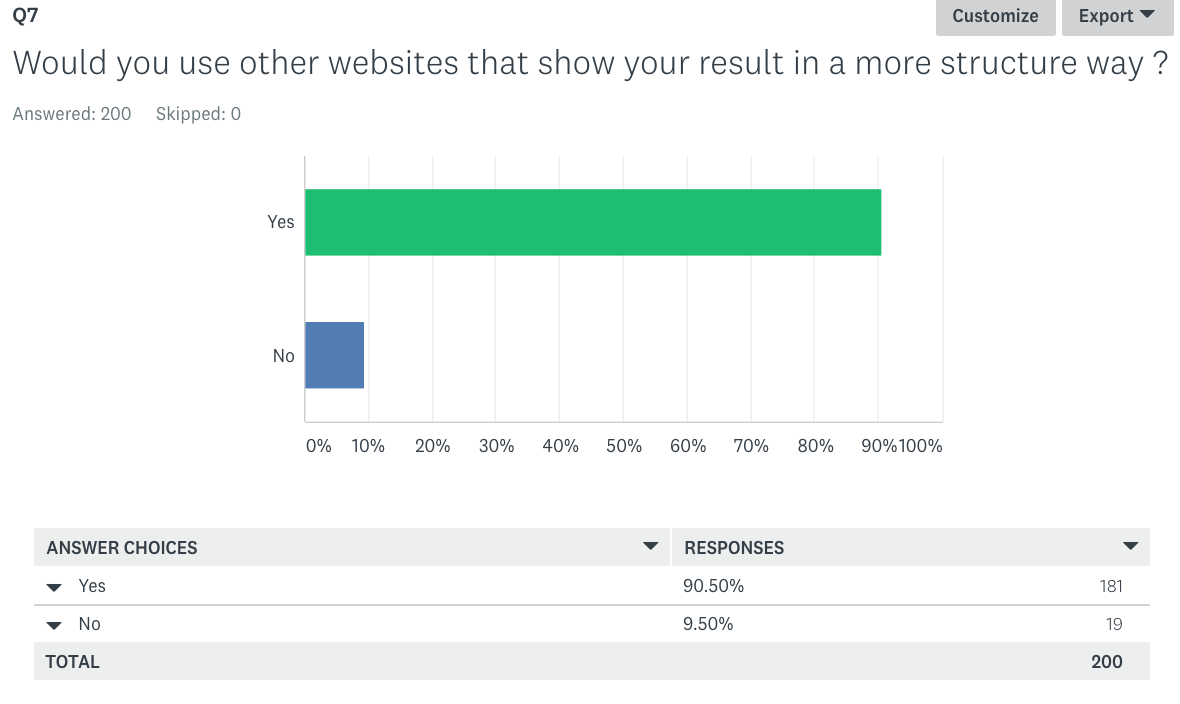
##### Figure 19. Analysis of Question # 5.

* Analysis of Question # 6:



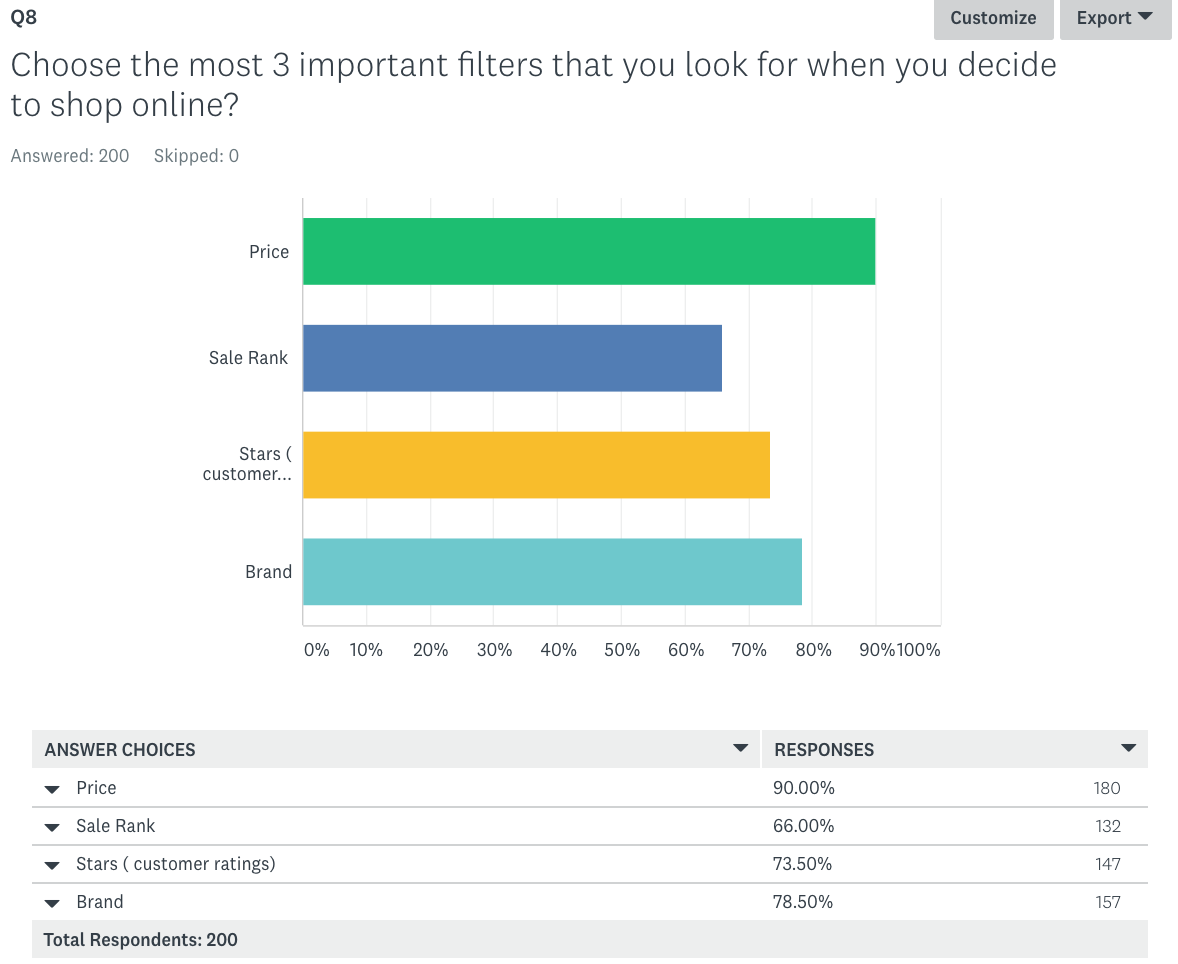
##### Figure 20. Analysis of Question # 6.

* Analysis of Question # 7:



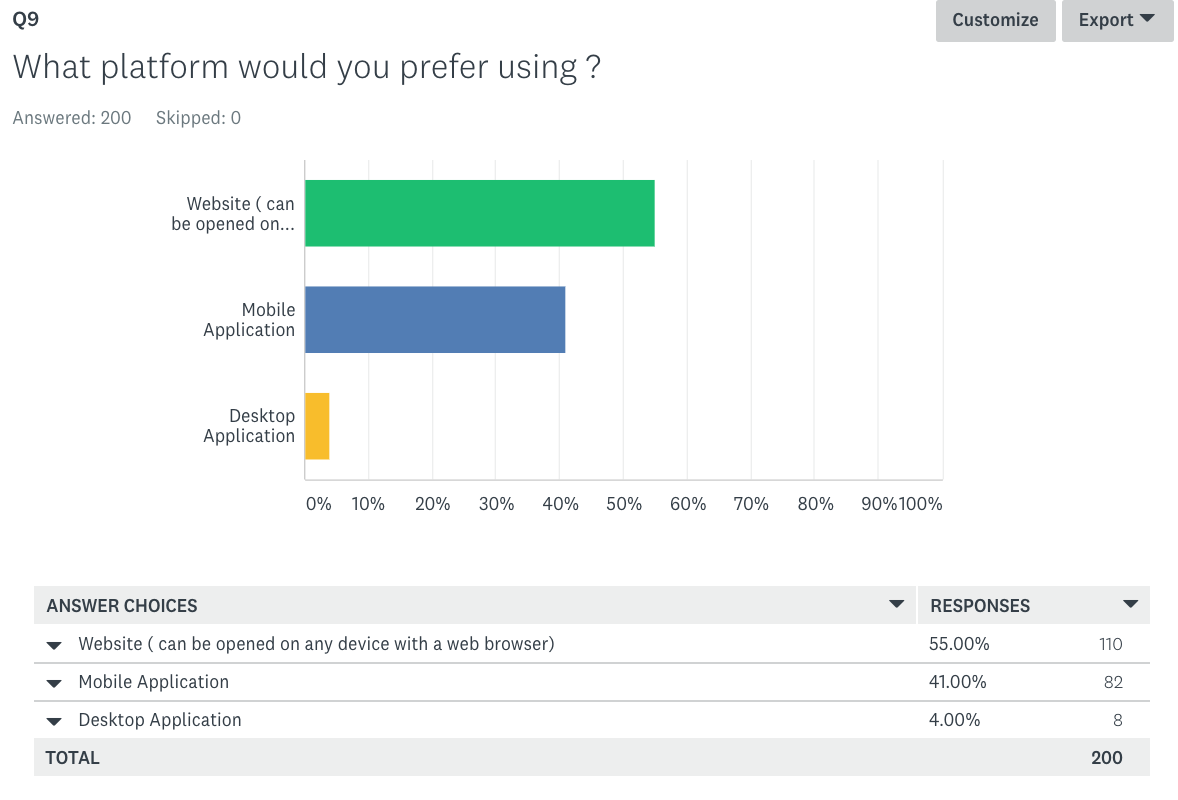
##### Figure 21. Analysis of Question # 7.

* Analysis of Question # 8:



##### Figure 22. Analysis of Question # 8.

* Analysis of Question # 9:



##### Figure 23. Analysis of Question # 9.