QATAR UNIVERSITY

COLLEGE OF ENGINEERING

MOBILE APP FOR HIDDEN DATA ANALYTICS OF ONLINE MARKETPLACE SYSTEMS

BY

ABOUBAKR AQLE

A Project Submitted to the Faculty of

College of Engineering

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

January 2016

Members of the Committee's Approval

The members of the Committee approve the dissertation of Aboubakr Aqle defended on 17th January 2016.

Dr. Abdelaziz Bouras

Committee Member

Dr. Mohammad Saleh

Committee Member

Dr. Tamer Elsayed

Committee Chair

Prof. Ali Jaoua

Dissertation Supervisor

Approved:

Dr. Rashid Alammari, Dean, College of Engineering

Abstract

In this project, an extensive analysis and evaluation of the existing e-marketplaces is performed. The aim of this analysis is to improve the experience of end-users through an Android application that is capable of summarizing multiple heterogeneous hidden data sources and unify received responses to one single, structured and homogenous source.

The proposed Android application is based on the multi-level conceptual analysis and modeling strategy. In which, the data is analyzed in a way that helps discovering the main entities of any unknown dataset captured from hidden web sources.

Several experiments have been conducted that depend on static data analytics for discovering entities. The results showed that query results analysis and re-structuring the output before displaying to the end-user in conceptual multilevel mechanism are reasonably effective in response time to the user interaction with minimal number of screens and clicks. The proposed application can also predict user requirements from the initial query that built on the results obtained from different e-commerce marketplaces.

Based on the proposed intelligent application that predict user required products, the interface is minimized to only two navigation screens, and the approximated time needed is 8 seconds to reach the targeted product. This solution is faster and easier to use than the current available application solutions by comparing the response time and the user interaction for the obtained results that met user requirements.

Table of Contents

List of Tables

List of Figures

Acknowledgment

First of all, I would like to thank Allah for giving me this opportunity and the ability to finish this research for the Master Degree.

This project would not have been possible without the help of few people that in a way or another contributed to my work.

I would like to express my deep gratitude to Professor Ali Jaoua, my research supervisor, for his continuous guidance, support, encouragement, and patience throughout the work of this research.

I am grateful to my family for their unconditional love, support, encouragement, prayers and confidence in me.

Also I am using this opportunity to express my gratitude to all my friends who supported me during this research.

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

Chapter 1

Introduction

E-business is Electronic business technique to facilitate performing business by taking advantage of internet technology that incorporates website, computer and internet. Nowadays, one of the most important types of e-businesses is the transactional business models called e-marketplaces, also known as online marketplaces. The first e-marketplace is the eBay platform, first used in 1995 [1].

Business online is more flexible as the parties involved are sitting behind PCs and trade online with no need to actual physical move from one place to another. Internet provides useful tools for marketing, application for orders, and handling private information of the customers. As a niche, m-business starts to grow with more companies implementing mobile platforms for their business. Therefore, no need to be near a PC, a smartphone with internet connection is suffice.

This project aim to show the impact of an innovative mobile application based on new analysis and presentation concept that can predict users' needs from initial input query which improves the interface. Many Human Computer Interaction HCI concepts of look and feel design are taken in consideration. We will illustrate how online marketplaces are working today and how we can get the advantage of their Application Program Interfaces (APIs). We are looking for better solutions to serve the end users to find their target products from different marketplaces.

1.1. Quick Overview of e-Marketplaces

Online marketplace systems are present on the web for more than 20 years and there are a huge number of players working on a global scale producing more than $50 billion dollar every year. This type of commerce has become widely used due to uncontested advantages over traditional model in online commerce. When going online there are no limitations for opening hours or business locations. Beside this, a buyer can get in contact with great numbers of sellers from which he can chose to buy a certain specific product [2]. This is very advantageous for the buyer because the competition between suppliers can lead to lower purchasing prices of products. On the other hand, a seller will always be grateful for the wide network of clients from all over the globe. This is very advantageous for the business as the sell process and the profit is greatly influenced by the number of customers [3].

Other advantages of e-marketplaces are:

- Online marketplaces offer a much wider area of services.
- Quick access to the search services.
- Comparison between products.

Even searching for a product on Amazon or eBay saves time; but there is still a room for improvements that can rely on hidden databases for better and faster solutions that can be integrated with different online marketplaces.

1.2. Online Systems with Hidden Data

Hidden database are data that cannot be reached by current search engines. They are much more abundant than direct public data that we could find by querying current search engines. The hidden web contains huge data repositories on the Web. A statistic from 2014 states that the hidden web comprises of approximately 550 billion documents and content greater of the web by 2000 time’s [4]. Most hidden data are accessible through direct queries to online systems or through particular provided APIs Extracting information from hidden databases can be in 3 general steps:

1. Formulation of search method.
2. Search sources which are relevant to the task.
3. Submit search form to source to extract results of each relevant resource.

Figure 1: Hidden Web Step by Step

1.3. Problem Statement and Challenges

Searching different online marketplace systems for a certain product or service is time consuming for the end users and doesn’t provide comparison capability between different items distributed among different systems. In order to overcome this gap and allow the user to have a pleasant experience, our research investigates the following primary research question:

- **How Mobile Application could assist the end user to find his target while searching different online marketplaces for certain product within minimal number of clicks?**

This is the main target of this project, to have a unified, integrated and interactive solution built as one mobile application that works as an intelligent interface for multiple different online marketplace systems through conceptual extraction and discovery of the available products along with their features. This kind of data analysis and modeling provides the end user with a multilevel browsing environment that can predict the end users' needs exactly and reaching his target product within minimal number of clicks and navigation screens.

In order to answer and respond to the primary research question, the research seeks to find answers to the following related sub-questions:

- **How can we find and extract products hidden data from different online marketplaces?**
- **How can we structure the captured hidden data to shorten the query path to the target products?**

Hidden data are retrieved through specific APIs that help extracting sufficient amount of data for our analytical purposes. These data sets from different sources are unified in one structure that will be fed to our data analysis and modeling component based on automatic entity extraction and conceptual browsing techniques. These techniques are smart enough to be domain independent and can adapt to any online marketplaces and products searching.

1.4. Aim and Objectives

The aim of this research is to design and develop one integrated mobile application interface capable to output in one place the searching results from multiple different online marketplace systems. Our proposed new presentation methodology depends on entities extraction and concepts methodologies.

The aim of the research can be met through the following specific objectives:

- Investigate existing online marketplaces alternatives, as Amazon and eBay.
- Discover entities by static data analytics.
- Summarize findings of the research.
- Implement methodology framework for mobile applications on Android OS based smartphones.
- Make different test cases for the implemented framework.
- Present test results in detail.
- Provide lessons learned, recommendations and future work.

1.5. Proposed Solution

This project proposes a solution of one mobile application interface for multiple different online marketplaces systems. For that purpose, providing APIs feature can be used to customize the search process by optimizing the query results with minimum number of clicks to reach the target products. The online systems to be included for this search process are: Amazon, eBay and can have as additional Add-on or plugin feature to integrate with new online marketplace systems in the future that allow application program interface API to their systems.

The design of such solution will depend on many Human Computer Interaction concepts of look and feel design and system usability. On the other hand, semantic web concepts that allow the exchange of information between different systems are also involved.

The solution developed and presented in this research allows:

- Search multiple online marketplaces in one search Mobile Application.
- Unify in one structure the relevant pages found from different sources.
- Predict user needs.
- Browse the results in more intelligent way.
- Reach target product within minimal number of clicks.

1.6. Contribution

Our main contribution can be specified as follows;

(1) The main focus is on the customer and not on the seller

(2) The mobile application will provide one structure results from different online marketplaces.

(3) The application is built on the concept of user requirements prediction, so the user will input only the initial query. This can be considered as automatic query reformulation using only the interface.

Concerning the first point, focus on the customer rather than the seller, is an innovative approach as most existing studies focus on how to provide the seller with a solution that can help him sell more products. Rather than following the same path, this report explains a solution that has as goal to help the customer to find easily and fast the product(s) he is looking for.

The present study proposes to search for the product in an automatic way on all reached hidden databases and bring together, in one window, all the results. Up to now, the client had to perform a searching process on each online market (for example, Amazon and eBay) and then to analyze one by one each output of the query. This is a time consuming and error prone process as it requires many clicks and navigation between many tabs opened on different online marketplaces. Consequently, it was decided to bring all the results in one tree-structured view that helps the user to easily navigate between different search criteria and automatically filters the initial search query to achieve the target product. At the end of query reformulation, the user can go directly to the required product page on the source marketplace (which at the moment of the search is unknown) rather than surfing on each marketplace, by turns.

With this mobile application a user can get a product much faster because the user's requirements are being guessed even before he mentions them. All results will come up in a tree structure where products will be listed in folders based on common characteristics. For example, when a customer looks for an iPhone 6S the mobile application will arrange results in folders like Gold, Rose, Gray and so on. Thus, for client it is much easier to avoid searching process on different places that is time consuming.

This report is organized as follows: first present a general background on online marketplace systems, hidden data and formal concept analysis with comprehensive review of related work in Chapter 2. Chapter 3 describes the hidden data and entity extraction from online marketplaces. Chapter 4 presents Mobile App design and development phases. Comprehensive results are discussed in Chapter 6, followed by validation and evaluation of the results at the same chapter. At the end, the conclusion and guidelines for future work are summarized in Chapter 6.

Chapter 2

Background and Related Works

This chapter provides a brief background and related works about online marketplaces, business models, Formal Concept Analysis, automatic entity extraction and functional dependencies.

2.1. Background

A. Business Models behind Online Marketplaces

Online marketplace system has a variety of names, including web or virtual marketplace, e-marketplace, market space, market maker and E-hub [5]. In this paper it was adopted the definition given by Hanson and Kalyanam (2007): an e-marketplace is defined as a virtual marketplace based on the internet where numerous companies execute economic transactions [6].

According to Both (2001) online marketplaces can be split into vertical and horizontal systems. The vertical dimensions are specialized in computers, electronics, vehicles, chemicals or steel. These markets provide vertically integrated services unique to each industrial sector [7]. The horizontal dimensions are specialized in items not tied to a particular industry.

Table 1: Vertical and Horizontal online marketplaces

Vertical online marketplaces	Horizontal online marketplaces
Uber	Amazon.com
Airbnb	eBay.com
HomeAway	Alibaba.com

Customers compare online marketplaces and start transactions with the most suitable ones. When a level of trust has built – up, the transaction with that online marketplace is continued. Trust is formed when the buyer feels the seller has the ability to satisfy with success its needs [8].

Putting the business to the web has the benefit of offering personalized services and good communication with customers. According to Baltzan and van Phillips (2009), there are few approaches to be used for facilitating business on the web. B2B, B2C and C2C are the most typical business models adopted by online marketplaces [9]. Moreover, in time some of these businesses use more business models. For example, eBay first was used as a B2B but in time it becomes a B2C model [10].

B. Online Systems with Hidden Data and Mobile Apps

Hidden database are much more abundant than direct public data that we could find by querying current search engines. Most hidden data are accessible through direct queries to online systems or through particular provided APIs. Currently, many studies are carried out for improving the extraction of information from hidden databases and some proposals have been already made: manual approach, wrapper generation and automatic extraction [11]. In this project, a hybrid model of wrapping and automatic extraction methods have been considered: the forms collected are being reorganized and resized to keep important features like link, id of product and product description as will be discussed in more details at chapter 3.

C. Binary Relation , Rectangle and Maximal Gain

In general a binary relation **R** is subset of the Cartesian product **E** × **F** where **E** is a set of arguments and **F** set of images.

Non empty rectangle or rectangular relation of binary relation **R** is any Cartesian product **A** × **B** included in **R** where, **A**≠Ø, **B**≠Ø, **A**⊆**E** and **B**⊆**F**. **A** is the domain and **B** is the codomain or range of the rectangle.

The gain that is associated with a given rectangular relation **RE** = **A** × **B** is calculated by:

$$\mathbf{g(RE)} = (\|\mathbf{A}\| \times \|\mathbf{B}\|) - (\|\mathbf{A}\| + \|\mathbf{B}\|) \qquad (1)$$

where ||**A**|| denotes the cardinality of the set **A**.

A rectangle **RE** = **A** × **B** that contains an element (a, b) of a binary relation **R** is said to be optimal if it realizes the maximal gain **g(RE)** among all rectangles that contain (a, b).

Optimal rectangle represents most important data associations involving (a, b) [12].

D. Formal Concept Analysis

Formal concept analysis (FCA) is a data analysis technique that defines relations between specific set of objects and specific set of attributes. FCA creates two types of output from the input data: the concept lattice which is a set of all formal concepts in the data that are ordered hierarchically by a super-concept and sub-concept relation, and a set of attributes implications that define a valid dependency in the data [13].

Logical attributes in a table can be represented by a triplet (**G**, **M**, **I**) where **I** is the relation between **G** and **M**. Components of **G** are called objects and represent tuples in the table, components of **M** are called attributes and represent columns in the table, if we have g1∈ **G** and m1∈ **M**, (o1, attr1) ∈ **I**, then it shows that object o1 has attribute attr1, while (o1, attr1) ∉ **I**, shows that object o1 does not have attribute attr1. The equivalent table (**G**, **M**, **I**) is given by **G** = {o1, o2, o3}, **M** = {attr1, attr2, attr3} and we have (o1, attr1) ∈ **I**, (o2, attr3) ∉ **I** as shown in Table 2.

"A formal context is a triplet (**G**, **M**, **I**) where **G** and **M** are non-empty sets and **I** is a binary relation between **G** and **M**, **I** ⊆ **G** × **M**" [13].

Table 2: Formal Context Example

	attr1	**attr2**	**attr3**
o1	x	x	x
o2	x	x	
o3		x	x

2.2. Related Work

A. Automatic Entity Extraction from Database Instance

Discovering concepts or entity types from database instance is useful for decomposition.

Information system decomposition process starts from concrete hidden database on the system structure coming from several tables or n-ary relations. Each table contains several tuples and each tuple is a vector of values of some attributes.

Analyzer has to extract the information system entities, and some general dependencies that must be valid for any instance of the system. Functional dependencies can be between some subsets of attributes or between some previously discovered entities [14].

Jaoua et al. [15] proposed rectangular decomposition of n-ary relation for database decomposition and extracting entity types. Any rectangle RE represents two clusters of fully associated information: domain (RE) and codomain (RE).

Let's have one instance r(R) of an n-ary relation. In the first step, we map r(R) to an equivalent binary relation R2. In the second step, we find optimal rectangle (REopt) of R2 that provides the maximal gain. (REopt) is composed of two clusters domain (REopt) and codomain (REopt) that are the most fully associated. The entity sets for the two clusters are disjoint, in another word there is no entity type with its set of attributes is shared between the two clusters. Then we continue for the same decomposition process for the two lower levels of the domain and codomain to discover lower sets of entities till we find the minimal number of entities that make coverage for the whole relation.

We can take one example of n-ary relation for the decomposition process that automates the entity discovery and extraction scenario. Example of one table that has 24 records as rows and 13 attributes as columns is given in Table 3 below [15]:

Table 3: Database Instance [15]

SNO	SNAME	STATUS	SCITY	PNO	PNAME	COLOR	WEIGHT	PCITY	JNO	JNAME	JCITY	QUANTITY
S1	Smith	20	London	p1	Nut	Red	12	London	J1	Sorter	Paris	200
S1	Smith	20	London	p1	Nut	Red	12	London	J4	Console	Athens	700
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J1	Sorter	Paris	400
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J2	Punch	Rome	200
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J3	Reader	Athens	200
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J4	Console	Athens	500
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J5	Collator	London	600
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J6	Terminal	Oslo	400
S2	Durand	10	Paris	p3	Screw	Blue	17	Rome	J7	Tape	London	800
S2	Durand	10	Paris	p5	Cam	Blue	12	Paris	J2	Punch	Rome	100
S3	Dupont	30	Paris	p3	Screw	Blue	17	Rome	J1	Sorter	Paris	300
S3	Dupont	30	Paris	p4	Screw	Red	14	London	J2	Punch	Rome	500
S4	Clark	20	London	p6	Cog	Red	19	London	J3	Reader	Athens	300
S4	Clark	20	London	p6	Cog	Red	19	London	J7	Tape	London	300
S5	Kurt	30	Athens	p1	Nut	Red	12	London	J4	Console	Athens	1000
S5	Kurt	30	Athens	p2	Bolt	Green	17	Paris	J4	Console	Athens	100
S5	Kurt	30	Athens	p2	Bolt	Green	17	Paris	J2	Punch	Rome	200
S5	Kurt	30	Athens	p3	Screw	Blue	17	Rome	J4	Console	Athens	1200
S5	Kurt	30	Athens	p5	Cam	Blue	12	Paris	J5	Collator	London	500
S5	Kurt	30	Athens	p5	Cam	Blue	12	Paris	J4	Console	Athens	400
S5	Kurt	30	Athens	p5	Cam	Blue	12	Paris	J7	Tape	London	100
S5	Kurt	30	Athens	p4	Cam	Red	14	London	J4	Consle	Athens	800
S5	Kurt	30	Athens	p6	Cam	Red	19	London	J2	Punch	Rome	200
S5	Kurt	30	Athens	p6	Cam	Red	19	London	J4	Console	Athens	500

We can translate n-ary relation to binary relation by taking attribute-values in pairs; all pairs of any attribute value are as pair of this format (Value1.AttributeOrder1, Value2.AttributeOrder2) as Figure 2 example for Table 3:

{(S 1.00,Smith.01);(S 1.00,20.02);(S 1.00,London.03);(S 1.00,pl.04);(S 1.00,Nut.05);

(S1.00,Red.06);(S 1.00,12.07);(S 1.00,London.08);(S 1.00,J1.09);(S 1.00,Sorter. 10);

(S 1.00,Paris. 11) ;(S 1.00,200.12) ;($mith.01,S 1.00);(Smith.01,20.02) ;(Smith.01,London.03);

(Smith.01,p 1.04) ;($mith.01,Nut.05) ;(Smith.01,Red.06) ;(Smith.01,12.07) ;(Smith.01,London.08);

Figure 2: Binary Relation as attribute-values in pairs [15]

Once we translate this relation to binary relation, we can find the first optimal rectangle according the maximal gain g(RE), which splits the relation to two disjoint sets of domain and codomain as below binary-table Figure 3:

	JNAME<->	JCITY<->A	QUANTITY	SNO<->S2	SNAME<->Durand	STATUS<->10	SCITY<->Paris	PNO<->p3	PNAME<->Screw	COLOR<->Blue	WEIGHT<->17	PCITY<->Rome	QUANTITY	JNO<->J2
PNO<->p3	1	1	0	1	1	1	1	0	1	1	1	1	1	1
PNAME<->Screw	1	1	0	1	1	1	1	1	0	1	1	1	1	1
COLOR<->Blue	1	1	0	1	1	1	1	1	1	0	1	1	1	1
WEIGHT<->17	1	1	0	1	1	1	1	1	1	1	0	1	1	1
PCITY<->Rome	1	1	0	1	1	1	1	1	1	1	1	0	1	1
QUANTITY<->400	1	1	0	1	1	1	1	1	1	1	1	1	0	0
JNO<->J2	0	0	0	1	1	1	1	1	1	1	1	1	0	0
JNAME<->Punch	0	0	0	1	1	1	1	1	1	1	1	1	0	1
JCITY<->Rome	0	0	0	1	1	1	1	1	1	1	1	1	0	1
JNO<->J3	0	1	0	1	1	1	1	1	1	1	1	1	0	0
JNAME<->Reader	0	1	0	1	1	1	1	1	1	1	1	1	0	0
QUANTITY<->500	1	1	0	1	1	1	1	1	1	1	1	1	0	1
JNO<->J5	0	0	0	1	1	1	1	1	1	1	1	1	0	0
JNAME<->Collator	0	0	0	1	1	1	1	1	1	1	1	1	0	0
JCITY<->London	0	0	0	1	1	1	1	1	1	1	1	1	0	0
QUANTITY<->600	0	0	0	1	1	1	1	1	1	1	1	1	0	0
JNO<->J6	0	0	0	1	1	1	1	1	1	1	1	1	1	0
JNAME<->Terminal	0	0	0	1	1	1	1	1	1	1	1	1	1	0
JCITY<->Oslo	0	0	0	1	1	1	1	1	1	1	1	1	1	0
JNO<->J7	0	0	0	1	1	1	1	1	1	1	1	1	0	0
JNAME<->Tape	0	0	0	1	1	1	1	1	1	1	1	1	0	0
QUANTITY<->800	0	1	0	1	1	1	1	1	1	1	1	1	0	0
PNO<->p5	1	1	0	1	1	1	1	0	0	1	0	0	1	1
PNAME<->Cam	1	1	0	1	1	1	1	0	0	1	0	0	1	1

Figure 3: Binary Relation

Then we make projection on the same relation for the domain attributes to make the decomposition, and the same for codomain attributes.

After doing the same steps for the Domain and Codomain recursively, will end up with the final entities that discovered from Figure 3 as below Figure 4:

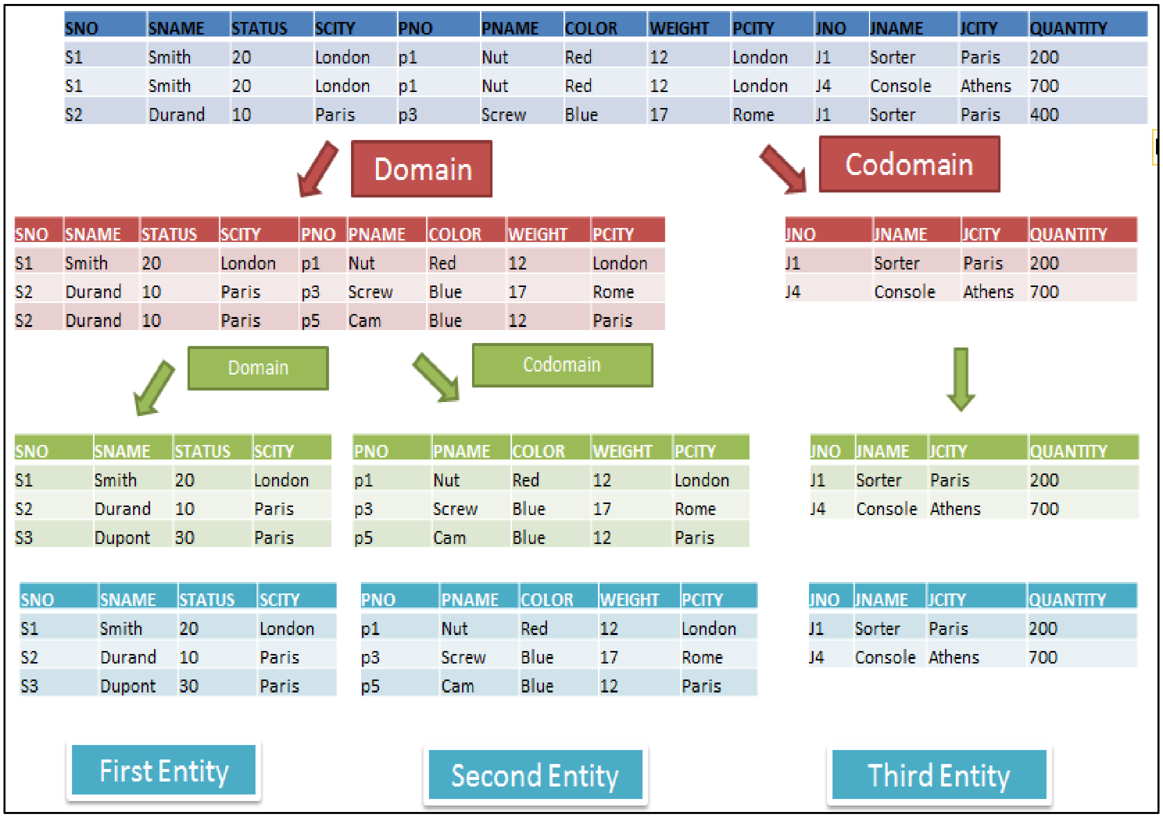


Figure 4: Decomposition Process

The algorithm can continue for the decomposition till it reaches single attribute level, but that is not practical to reach our purpose for the minimal number of entities. For that reason, we proposed a termination condition to stop the decomposition. Decomposition should stop if the number of records in the new domain and codomain matched the number of the records of the original n-ary table. That means, after decomposition the number of records may not be reduced because all attributes belongs to the same entity and no need for further decomposition process to be carried out.

B. Design of Conceptual Browser Tree

Formal concept analysis defines methods for data clustering. Formal concept associated pattern is a set of objects sharing the maximum set of properties.

We can illustrate the proposed method for selecting optimal concepts to minimize information representation [16, 17]. That is text summarization process to generate the optimal concepts with word similarity method. The idea is to extract summary from document and association between contained keywords in the document. First, creating binary context where objects are sentences and properties are words. List of empty words is consulted to avoid these words from the text. Then from this binary context, we find optimal concept for all sentences in this domain concept to generate the summary keywords. Decomposition process depends on the maximum weight of the words associated to the text.

For example for our research problem from conceptual browser point of view, assume we get results from e-marketplaces APIs for our search context. These results considered as objects O for the retrieved records {o1, o2, o3, o4, o5, o6} and its features description as properties or attributes P {IPhone6s, smartphone, HTC, M9, M10} which considered as keywords.

Retrieved results:

1. IPhone6s smartphone original
2. IPhone6s smartphone used
3. HTC smartphone new
4. HTC M9 smartphone brand
5. HTC M9 old
6. M10

Then we can build the formal context by the following binary-table Table 4:

Table 4: Formal Context

	IPhone6s	smartphone	HTC	M9	M10
o1	1	1	0	0	0
o2	1	1	0	0	0
o3	0	1	1	0	0
o4	0	1	1	1	0
o5	0	0	1	1	0
o6	0	0	0	0	1

Next, we can find the formal concepts of the above formal context, where concept is from two parts: extension that is all objects belonging to a concept, and its intension that is all attributes shared by these objects. That can be visualized as concept lattice that represents structure set of concepts as Figure 5 below:

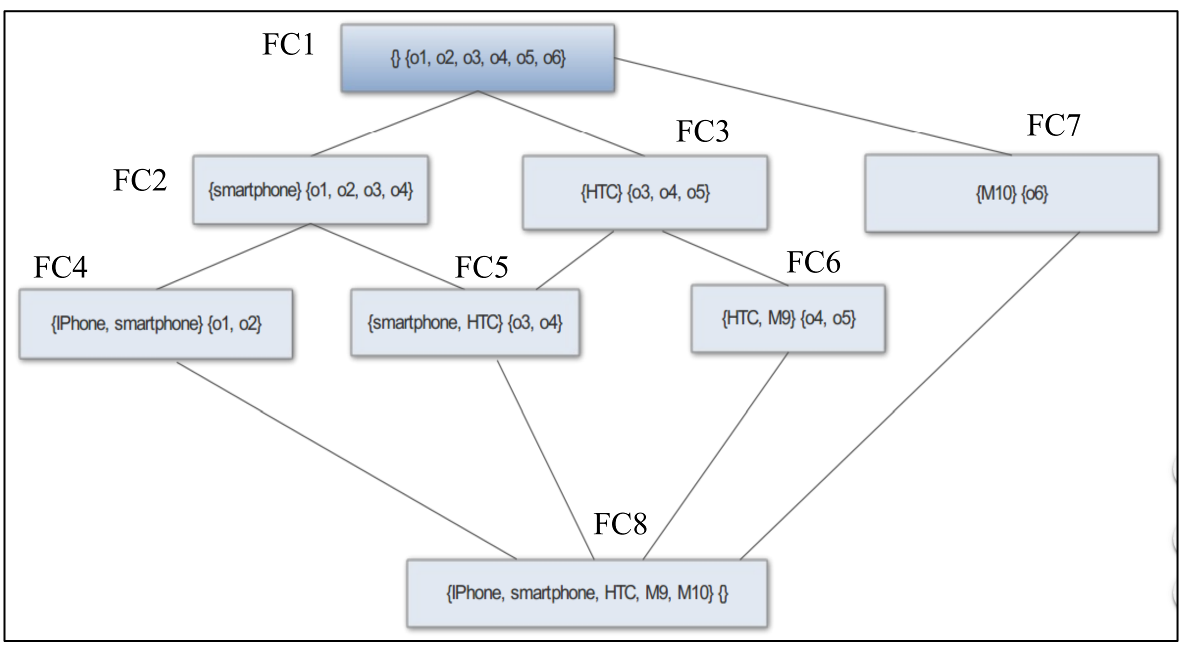


Figure 5: Concept Lattice

Then, make coverage of the context by a minimal number of concepts as below Figure 6:

	IPhone6s	smartphone	HTC	M9	M10
o1	1	1	0	0	0
o2	1	1	0	0	0
o3	0	1	1	0	0
o4	0	1	1	1	0
o5	0	0	1	1	0
o6	0	0	0	0	1

Figure 6: Coverage of the Context

Coverage of context formed by:

- FC4 concept of ({o1, o2}, {IPhone6s, smartphone})
- FC5 concept of ({o3, o4}, {smartphone, HTC})
- FC6 concept of ({o4, o5}, {HTC, M9})
- FC7 concept of ({o6}, {M10})

Each concept associated with a significant title that is a word with a maximum weight selected from the domain of the concept that is keywords. Finally organizing the words into a heap that is complete binary tree where words with greater weight appear at higher level in the tree as below Figure 7:

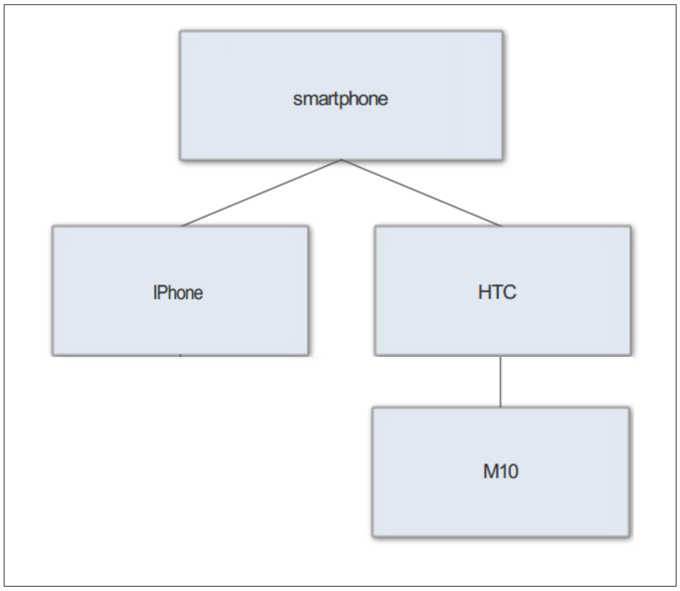


Figure 7: Conceptual Browser Tree

This method can be considered as unsupervised classifier to generate summary keywords for the domain text.

Chapter 3

Entity Extraction from Hidden Data

This chapter provides descriptive analysis of the methodology used to collect the required hidden data from different online marketplaces. This includes methods and tools used for the data collection, analytics and entity discovery.

3.1. Hidden Data

Online marketplaces have a lot of information for the related products on e-commerce websites. These data can be displayed to the end users with proper arrangement of information categories or be hidden from the end users and used for the management of the online marketplace system.

Hidden data is important factor for the data analysis process while it provides more information for the study subject. The ability of having direct access to these hidden data depends on the business strategy of the online marketplace and on the services that can be provided for the customers or for other business parties.

The most famous online marketplaces as eBay and Amazon provide this kind of service to access their repository database for the products by registered Product Advertising APIs to send the request for searching certain items and receive the respond in XML (Extensible Markup Language) format to view information about these stored items.

XML response file contains a list of products and information of each product as attributes that describe each product from the hidden database that can be accessed only by registered API. Product Advertising API is encoded by Web Service Description Language (WSDL) that is an XML document that defines the operations, parameters, requests, and responses used in web service interactions.

For this study, we registered for eBay API authentication to send our text query and get the matched products list with all related information of the attributes of these products in XML that contains header for the request parameters submitted through HTTP GET method. Product Advertising API returns a well-defined set of tags for each kind of request.

The HTTP GET call method passes values in the URL using name-value (NV) pairs as below for eBay:

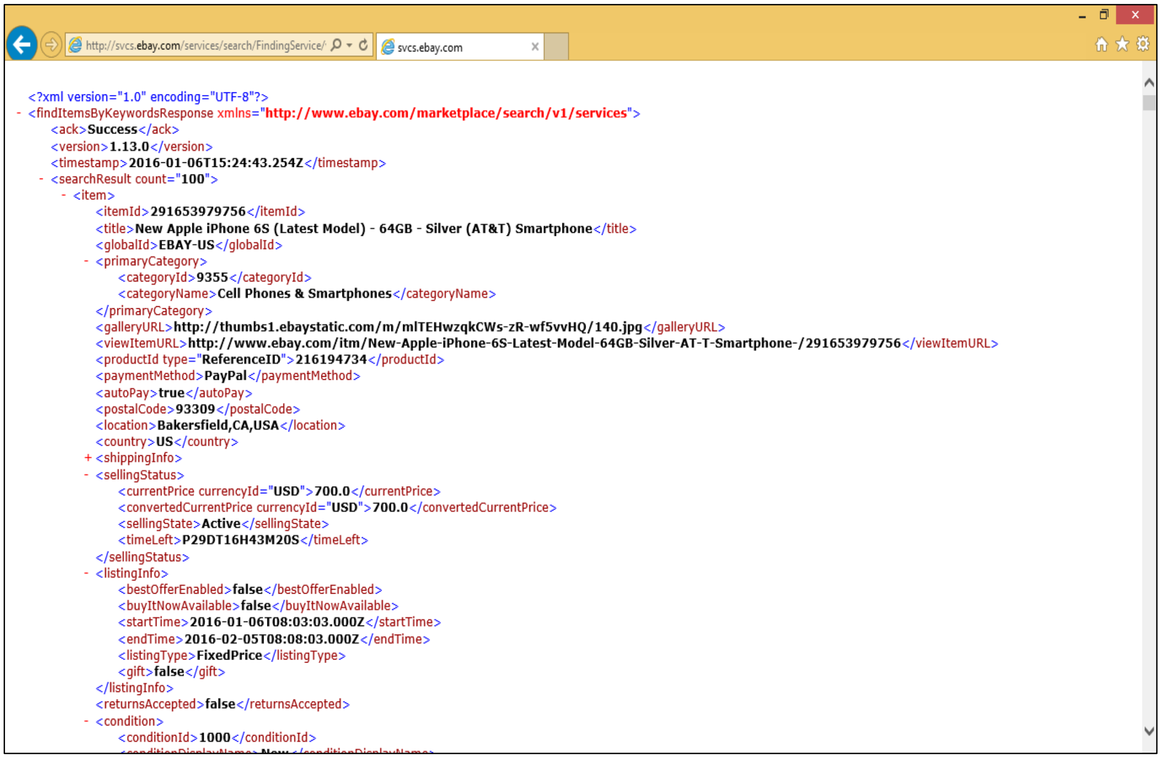
- http://svcs.ebay.com/services/search/FindingService/v1?OPERATION-NAME=findItemsByKeywords&SERVICE-VERSION=1.0.0&SECURITY-APPNAME=QUab7f54c-a457-49ad-b3d0-5a66db6f9f3&RESPONSE-DATA-FORMAT=XML&keywords=iphone%206s

The request consists from the following end point URL and named-value pairs:

- End point URL to receive the request: http://svcs.ebay.com/services/search/FindingService/v1?
- Operation-Name: findItemsByKeywords
- Service-Version: 1.0
- Security-AppName: which given to our registered authenticated API
- Response-Data-Format: XML

- Keywords: the search text

The response XML file as below Figure 8:




```
<?xml version="1.0" encoding="UTF-8"?>
- <findItemsByKeywordsResponse xmlns="http://www.ebay.com/marketplace/search/v1/services">
    <ack>Success</ack>
    <version>1.13.0</version>
    <timestamp>2016-01-06T15:24:43.254Z</timestamp>
  - <searchResult count="100">
    - <item>
        <itemId>291653979756</itemId>
        <title>New Apple iPhone 6S (Latest Model) - 64GB - Silver (AT&T) Smartphone</title>
        <globalId>EBAY-US</globalId>
      - <primaryCategory>
          <categoryId>9355</categoryId>
          <categoryName>Cell Phones & Smartphones</categoryName>
        </primaryCategory>
        <galleryURL>http://thumbs1.ebaystatic.com/m/mlTEHwzqkCWs-zR-wf5vvHQ/140.jpg</galleryURL>
        <viewItemURL>http://www.ebay.com/itm/New-Apple-iPhone-6S-Latest-Model-64GB-Silver-AT-T-Smartphone-/291653979756</viewItemURL>
        <productId type="ReferenceID">216194734</productId>
        <paymentMethod>PayPal</paymentMethod>
        <autoPay>true</autoPay>
        <postalCode>93309</postalCode>
        <location>Bakersfield,CA,USA</location>
        <country>US</country>
      + <shippingInfo>
      - <sellingStatus>
          <currentPrice currencyId="USD">700.0</currentPrice>
          <convertedCurrentPrice currencyId="USD">700.0</convertedCurrentPrice>
          <sellingState>Active</sellingState>
          <timeLeft>P29DT16H43M20S</timeLeft>
        </sellingStatus>
      - <listingInfo>
          <bestOfferEnabled>false</bestOfferEnabled>
          <buyItNowAvailable>false</buyItNowAvailable>
          <startTime>2016-01-06T08:03:03.000Z</startTime>
          <endTime>2016-02-05T08:08:03.000Z</endTime>
          <listingType>FixedPrice</listingType>
          <gift>false</gift>
        </listingInfo>
        <returnsAccepted>false</returnsAccepted>
      - <condition>
          <conditionId>1000</conditionId>
```


Figure 8: XML Response File

3.2. Data Collection and Analysis

API of eBay returns 24 attributes for each product, these attribute names as below Figure 9 from the response XML file in table format:

ID	_IDsearchResult	autoPay	condition	country	galleryPlusPictureURL	galleryURL	globalId	isMultiVariationListing	item	itemId	listingInfo	location	paymentMethod	postalCode	primaryCategory	returnsAccepted
1	1	false	0	CN	http://galleryplus.ebayimg.cc	http://thumbs1	EBAY-US	false	0	161219431396	0	China	PayPal		0	true
2	1	false	0	JP	http://galleryplus.ebayimg.cc	http://thumbs1	EBAY-US	false	0	121775370476	0	Japan	PayPal		0	true
3	1	false	0	CN	http://galleryplus.ebayimg.cc	http://thumbs2	EBAY-US	false	0	281787685121	0	China	PayPal	201101	0	true
4	1	false	0	JP	http://galleryplus.ebayimg.cc	http://thumbs2	EBAY-US	false	0	221910559201	0	Japan	PayPal		0	true
5	1	false	0	CN	http://galleryplus.ebayimg.cc	http://thumbs1	EBAY-US	false	0	221794575416	0	China	PayPal		0	true
6	1	true	0	HK	http://galleryplus.ebayimg.cc	http://thumbs1	EBAY-US	false	0	160981881908	0	Hong Kong	PayPal		0	true
7	1	true	0	US	http://galleryplus.ebayimg.cc	http://thumbs2	EBAY-US	false	0	161660289901	0	Rochester,I	PayPal	14625	0	true
8	1	false	0	JP	http://galleryplus.ebayimg.cc	http://thumbs1	EBAY-US	false	0	252120892932	0	Japan	PayPal		0	true
9	1	false	0	JP	http://galleryplus.ebayimg.cc	http://thumbs2	EBAY-US	false	0	252120896785	0	Japan	PayPal		0	true
10	1	false	0	HK	http://galleryplus.ebayimg.cc	http://thumbs3	EBAY-US	false	0	231669653738	0	Hong Kong	PayPal	0000	0	true
11	1	true	0	US	http://galleryplus.ebayimg.cc	http://thumbs2	EBAY-US	false	0	121670564261	0	Dallas,GA,U	PayPal	30132	0	true
12	1	false	0	JP	http://galleryplus.ebayimg.cc	http://thumbs3	EBAY-US	false	0	262088623986	0	Japan	PayPal		0	true
13	1	true	0	JP	http://galleryplus.ebayimg.cc	http://thumbs4	EBAY-US	false	0	281808858899	0	Japan	PayPal		0	true
14	1	false	0	CN	http://galleryplus.ebayimg.cc	http://thumbs3	EBAY-US	false	0	311061032322	0	China	PayPal		0	true
15	1	false	0	CA		http://thumbs4	EBAY-ENCA	false	0	262100022819	0	Canada	PayPal	L8L7L6	0	false
16	1	true	0	CN	http://galleryplus.ebayimg.cc	http://thumbs1	EBAY-US	false	0	191561518692	0	China	PayPal		0	true
17	1	false	0	CN	http://galleryplus.ebayimg.cc	http://thumbs4	EBAY-US	false	0	281570855967	0	China	PayPal		0	true
18	1	false	0	JP	http://galleryplus.ebayimg.cc	http://thumbs4	EBAY-US	false	0	121790661195	0	Japan	PayPal		0	true

secondaryCategory	sellingStatus	shippingInfo	title	topRatedListing	viewItemURL	startTime
0	0	0	One Piece Anime The New V	false	http://www.ebay	2014-02-07T06:2
0	0	0	F/S MegaHouse Variable Ac	false	http://www.ebay	2015-10-01T11:0
0	0	0	Anime One Piece Roronoa Z	false	http://www.ebay	2015-08-31T15:4
0	0	0	ONE PIECE Dramatic Showca	false	http://www.ebay	2015-10-11T09:4
0	0	0	Anime One Piece Roronoa Z	false	http://www.ebay	2015-06-07T00:2
0	0	0	Bandai Super One Piece Sty	true	http://www.ebay	2013-02-26T16:5
0	0	0	One Piece Jolly Roger Metal	true	http://www.ebay	2015-04-02T14:4
0	0	0	One Piece manga anime figu	false	http://www.ebay	2015-10-11T13:1
0	0	0	One Piece anime figures Cre	false	http://www.ebay	2015-10-11T13:1
0	0	0	One Piece Battle Ver. RORO	false	http://www.ebay	2015-08-29T04:5
0	0	0	Roronoa Zoro's Shusui One	true	http://www.ebay	2015-06-05T20:3
0	0	0	One Piece anime figures Ror	false	http://www.ebay	2015-10-11T13:1
0	0	0	MegaHouse Variable Action	false	http://www.ebay	2015-09-24T12:3
0	0	0	**One Piece Roronoa Zoro N	true	http://www.ebay	2014-08-22T07:5
0	0	0	Roronoa Zoro, Variable Acti	false	http://www.ebay	2015-10-18T08:5
0	0	0	5"/13cm One Piece Roronoa	false	http://www.ebay	2015-04-18T11:2
0	0	0	One Piece POP Zoro Ear Clip	false	http://www.ebay	2015-01-21T06:2
0	0	0	MegaHouse Variable Heroes	false	http://www.ebay	2015-10-18T01:2

Figure 9: Records Attributes from eBay XML Response File

We recognized that some of these attributes can be serial number of the record for the obtained results, blank, or with single value for all the records. After eliminating these attributes we ended up with total of 9 attributes and most of these attributes are URL link to the product page, URL to product image, or true/false values. While only one attribute for product description which called title as below Figure 10 for the nine attributes: country, galleryURL, itemId, location, title, topRatingListing, niewItemURL and startTime.

country	galleryURL	globalId	itemId	location	title	topRatedListing	viewItemURL	startTime
CN	http://thumbs1.ebaystatic.com/m/	EBAY-US	161219431396	China	One Piece Anime The New World	false	http://www.ebay.com/itm/One-Pie	2014-02-07T06:28:15.000Z
JP	http://thumbs1.ebaystatic.com/m/	EBAY-US	121775370476	Japan	F/S MegaHouse Variable Action H	false	http://www.ebay.com/itm/F-S-Meg	2015-10-01T11:06:57.000Z
CN	http://thumbs2.ebaystatic.com/m/	EBAY-US	281787685121	China	Anime One Piece Roronoa Zoro B	false	http://www.ebay.com/itm/Anime-C	2015-08-31T15:49:55.000Z
JP	http://thumbs2.ebaystatic.com/m/	EBAY-US	221910559201	Japan	ONE PIECE Dramatic Showcase :	false	http://www.ebay.com/itm/ONE-PIE	2015-10-11T09:47:41.000Z
CN	http://thumbs1.ebaystatic.com/m/	EBAY-US	221794575416	China	Anime One Piece Roronoa Zoro 1(	false	http://www.ebay.com/itm/Anime-C	2015-06-07T00:21:03.000Z
HK	http://thumbs1.ebaystatic.com/m/	EBAY-US	160981881908	Hong Kong	Bandai Super One Piece Styling F	true	http://www.ebay.com/itm/Bandai-S	2013-02-26T16:58:00.000Z
US	http://thumbs2.ebaystatic.com/m/	EBAY-US	161660289901	Rochester,NY,USA	One Piece Jolly Roger Metal Keyc	true	http://www.ebay.com/itm/One-Pie	2015-04-02T14:41:31.000Z
JP	http://thumbs1.ebaystatic.com/m/	EBAY-US	252120892932	Japan	One Piece manga anime figures R	false	http://www.ebay.com/itm/One-Pie	2015-10-11T13:13:16.000Z
JP	http://thumbs2.ebaystatic.com/m/	EBAY-US	252120896785	Japan	One Piece anime figures Creator x	false	http://www.ebay.com/itm/One-Pie	2015-10-11T13:17:57.000Z
HK	http://thumbs3.ebaystatic.com/m/	EBAY-US	231669653738	Hong Kong	One Piece Battle Ver. RORONOA	false	http://www.ebay.com/itm/One-Pie	2015-08-29T04:53:04.000Z
US	http://thumbs2.ebaystatic.com/m/	EBAY-US	121670564261	Dallas,GA,USA	Roronoa Zoro's Shusui One Piece	true	http://www.ebay.com/itm/Roronoa	2015-06-05T20:34:50.000Z
JP	http://thumbs3.ebaystatic.com/m/	EBAY-US	262088623986	Japan	One Piece anime figures Roronoa	false	http://www.ebay.com/itm/One-Pie	2015-10-11T13:18:49.000Z
JP	http://thumbs4.ebaystatic.com/m/	EBAY-US	281808858899	Japan	MegaHouse Variable Action Heroe	false	http://www.ebay.com/itm/MegaHo	2015-09-24T12:38:06.000Z
CN	http://thumbs3.ebaystatic.com/m/	EBAY-US	311061032322	China	**One Piece Roronoa Zoro New W	true	http://www.ebay.com/itm/One-Pie	2014-08-22T07:56:00.000Z
CA	http://thumbs4.ebaystatic.com/m/	EBAY-ENCA	262100022819	Canada	Roronoa Zoro, Variable Action Her	false	http://www.ebay.com/itm/Roronoa	2015-10-18T08:59:23.000Z
CN	http://thumbs1.ebaystatic.com/m/	EBAY-US	191561518692	China	5"/13cm One Piece Roronoa Zoro	false	http://www.ebay.com/itm/5-13cm-(	2015-04-18T11:20:48.000Z
CN	http://thumbs4.ebaystatic.com/m/	EBAY-US	281570855967	China	One Piece POP Zoro Ear Clip Allo	false	http://www.ebay.com/itm/One-Pie	2015-01-21T06:21:49.000Z
JP	http://thumbs4.ebaystatic.com/m/	EBAY-US	121790661195	Japan	MegaHouse Variable Heroes One	false	http://www.ebay.com/itm/MegaHo	2015-10-18T01:20:28.000Z
JP	http://thumbs3.ebaystatic.com/m/	EBAY-US	121752343470	Japan	Roronoa Zoro DXF Figure One Pie	false	http://www.ebay.com/itm/Roronoa	2015-09-06T05:05:09.000Z
JP	http://thumbs3.ebaystatic.com/m/	EBAY-US	252129596706	Japan	MegaHouse excellent Model POP	false	http://www.ebay.com/itm/MegaHo	2015-10-17T14:47:17.000Z

Figure 10: Records from eBay API with 9 attributes only

Finally, we can use these sample records for entity extraction process to identify entities that can be represented by which attributes.

3.3. Entity Discovery

The exported table from the XML response file in the previous section is considered as the database instance from eBay products database.

Discovering entity types or concepts from database instance is a decomposition process that preserves all important information of the original relation, so there is no loss of information while decomposition. Each element of the relation represents attribute-value pairs linking different entity types that are unknown. Attribute set can be broken into one or several attribute subsets.

In this section, we will discuss information system decomposition and discovery of conceptual knowledge in n-ary relation or database instance.

Unknown n-ary relation can consists of many entity types, and by decomposition process that make conversion to a binary relation can split the relation to two parts as domain and codomain of the original table.

Entity discovery and extraction process:

1. Read n-ary relation (table)
2. Create binary relation from n-ary relation
3. Find optimal rectangle from binary relation
4. Extract domain and codomain from max. rectangle results
5. Redo step 2 through step 4 for the domain and codomain
6. Stop when all max rectangles of the results (domain and codomain) analyzed

For entity discovery and extraction purpose, we built java application to take as input the relation table to make the binary relation for attribute-value matrix and find the maximum rectangle that split the table to domain and codomain recursively till the final minimal concept as below Figure 11:

	galleryURL<->http://thumbs1.ebaystatic.com/m/m:	title<->One Piece Anime The New World Roronoa	viewItemURL<->http://www.ebay.com/	startTime<->2014-02-07T(
galleryURL<->http://thumbs1.ebaystatic.com	0	1	1	1
title<->One Piece Anime The New World Ror	1	0	1	1
viewItemURL<->http://www.ebay.com/itm/	1	1	0	1
startTime<->2014-02-07T06:28:15.000Z	1	1	1	0
galleryURL<->http://thumbs1.ebaystatic.com	0	0	0	0
title<->F/S MegaHouse Variable Action Hero	0	0	0	0
viewItemURL<->http://www.ebay.com/itm/	0	0	0	0
startTime<->2015-10-01T11:06:57.000Z	0	0	0	0
galleryURL<->http://thumbs2.ebaystatic.com	0	0	0	0
title<->"Anime One Piece Roronoa Zoro Battl	0	0	0	0
viewItemURL<->http://www.ebay.com/itm/	0	0	0	0
startTime<->2015-08-31T15:49:55.000Z	0	0	0	0
galleryURL<->http://thumbs2.ebaystatic.com	0	0	0	0
title<->ONE PIECE Dramatic Showcase 2nd se	0	0	0	0
viewItemURL<->http://www.ebay.com/itm/	0	0	0	0
startTime<->2015-10-11T09:47:41.000Z	0	0	0	0
galleryURL<->http://thumbs1.ebaystatic.com	0	0	0	0
title<->"Anime One Piece Roronoa Zoro 10th	0	0	0	0
viewItemURL<->http://www.ebay.com/itm/	0	0	0	0
startTime<->2015-06-07T00:21:03.000Z	0	0	0	0
galleryURL<->http://thumbs1.ebaystatic.com	0	0	0	0
title<->Bandai Super One Piece Styling Film Z	0	0	0	0
viewItemURL<->http://www.ebay.com/itm/	0	0	0	0
startTime<->2013-02-26T16:58:00.000Z	0	0	0	0
galleryURL<->http://thumbs2.ebaystatic.com	0	0	0	0

Figure 11: Binary Table for Automatic Entity Extraction

Final result for automatic entity extraction has title and galleryURL for the domain and rest attributes for the codomain. For the domain galleryURL will be the URL address of the matched title, and codomain will have start time of the advertisement and view item URL which will be the image URL. Therefore, we can consider the title as the main attribute to be taken into consideration for our application to define the entities and to be used for our Mobile Application process for conceptual discovery and browsing for the matched entities that will be discussed further in Chapter 4.

We conducted another set of experiments for the final result, we got from automatic entity extraction for functional dependency to evaluate this result.

A functional dependency exists in case of one or more attributes in a relation uniquely determines another attribute(s). That idea will be useful for our case to determine which attributes will be centric for other attributes and we can focus our search criteria on these centric attributes.

Open source java application called "ConExp" used in our case study for analyzing attribute object table as formal context in Formal Concept Analysis (FCA) that is exploring different dependencies, that exists between attributes.

For the same example instance we obtained from eBay, it was used as input for this experiment to determine the functional dependency that can be noticed from the output as below result:

- [title] → [country, itemId, topRatedListing, viewItemURL, globalId, location, startTime, galleryURL]
- [country, globalId, title] → [itemId, topRatedListing, viewItemURL, location, startTime, galleryURL]
- [country, globalId, title, viewItemURL] → [itemId, topRatedListing, location, startTime, galleryURL]
- [country, globalId, title, topRatedListing] → [itemId, viewItemURL, location, startTime, galleryURL]
- [country, globalId, location, title] → [itemId, topRatedListing, viewItemURL, startTime, galleryURL]

We can notice from this experiment that "title" attribute is the most key attribute that most of the other attributes depend on. As a result, we can take into consideration title attribute to be our focal attribute to determine the search of our query for the matched records.

As a final result for the entity discovery and extraction, we can end up with the three attributes: itemId, title and viewItemURL. Title is the text description of the product, itemId is the unique identifier for this item and viewItemURL is for the product page URL address.

Chapter 4

Mobile App Design and Development

This chapter describes the development lifecycle chosen for the Mobile App, system workflow and screen layout design with implementation.

4.1. Development Lifecycle

Nowadays, agile development methodology is widely used for mobile application development for its simplicity and flexibility.

Agile methodology for software development is an alternative to traditional project management that helps responding to unpredictable events through incremental and iterative work process known as sprints as Figure 12. Agile development methodology provides opportunities to assess project direction throughout development lifecycle.

Since it is an innovative methodology, agile was preferred over the traditional waterfall development scheme for our project. Moreover, agile method breaks down the software development into small phases, and quality testing is performed at the end of each step.

Some of agile based principles are the following:

- Early Customer satisfaction and continuous delivery of valuable software.
- Welcome changing requirements, even in late development.
- Working software is delivered frequently (weeks rather than months).
- Working software is the principal measure of progress.

- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstance.

Generally, Apps can be of two types: goal driven Apps or entertainment Apps. Since our proposal for this research is goal driven App, we considered two features to be taken in consideration that agile methodology support:

- Usefulness: reach fast a target product
- User friendly: few buttons, simple graphic interface and fluent navigation

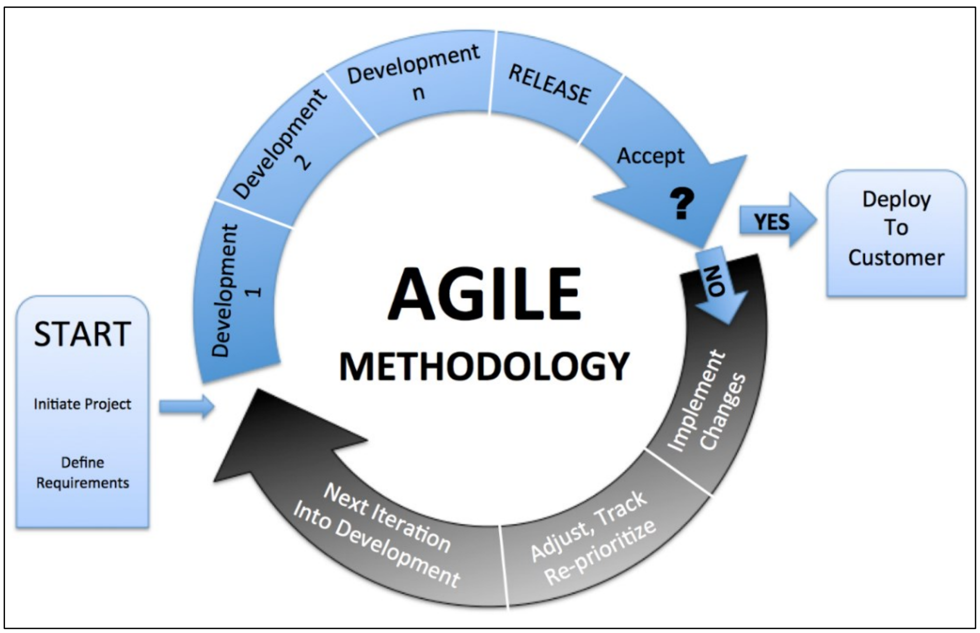


Figure 12: Agile Methodology

Agile methodology used for our project where each full iterative cycle focused on one aspect. First iteration focused on delivering Java application using a signed registered API (Amazon) to send predefined product as text and receive the response XML file from one marketplace. After the successful completion of first cycle, changes applied to test the second marketplace registered API (eBay). Third iteration was for migration from Java to Android Studio Integrated Development Environment (IDE). Forth iteration was to deliver user input text for the product parameter to be sent to the registered marketplaces. Fifth and final iteration was to customize and enhance the search, retrieve and conceptual browser processes of the solution to deliver the final solution.

Each iteration time frame was between one to three weeks to deliver the final Mobile App solution as the redefined requirements at each iteration.

Furthermore, agile methodology adopted in our research recommends the use of competitive analysis as a tool for assessing the App's performance as will be discussed in further more details in chapter 5 for experimental results.

4.2. Workflow

It is an important part from the design point of view, to create the system workflow. That is the system infrastructure for defined sequence of tasks arranged as workflow which will provide a complete description for the series of activities that are necessary to complete the required tasks.

The App workflow can be summarized by the following sub-tasks:

1. Get user input for the search product.
2. Fetch item information from registered online marketplaces APIs that are eBay and Amazon APIs for this research.
3. Remove duplicated items.
4. Summarize results for both Amazon and eBay.

5. After summarization, make tree structure that display main concepts as keywords in expandable levels.
6. When user click on any top level, show the next keyword levels.
7. When user click on keyword view button, it starts extracting matched records and display them in List View.
8. After extracting records matched, save them to Record Item for each record that keeps Item ID, Title, and URL address of the product page.
9. When click List View's item, redirect to the advertisement URL address for the item within web browser.

These steps are represented graphically in Figure 13. There, it is very clear how the App works and how the steps succeed one by one till the list with results is returned to the end user.

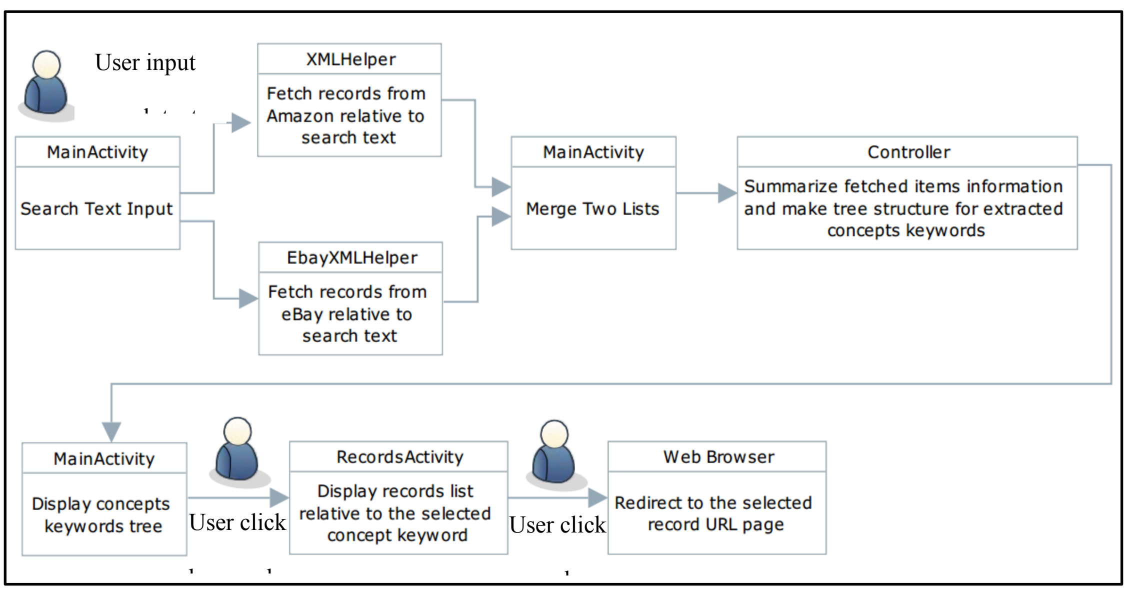


Figure 13: App Workflow

App workflow makes a visual and detailed description for the required tasks to achieve the target job to find the related products to user search text and identifies the needed screens for either user input or display process outcome to the end user.

The first sub-task to find the target products can display the retrieved records from online marketplaces on the same screen of the user search input to minimize the number of screens that user need to navigate between back and forth. This idea is taken in consideration for the next phase for screen prototype design.

4.3. Screen Design

We started the design development by designing sketch of the screens from HCI perspective in terms of the App look and feel and ease of use for the end user.

These screens will be total of two designed screens, first screen to capture user input for his target product, and the same screen will display all the matched results from Amazon and eBay with hierarchy levels from top conceptual level till the most down level concept as conceptual browser for the products as their descriptions.

User can expand each level by clicking on the top node to display its children nodes and the same will be for the children nodes. If user wants to check the matched records at any level, user can click on the right side of the target node to go for the second screen.

The second screen, will display matched records in list view, each list item will display the description of the product and will have a link to the specific product page URL.

These products list view will be designed as objects not as texts with hyperlinks, in other words, these objects will contain all its attributes, as its item ID, item description and item URL address. Once the user click on any product from this list, the mobile App will navigate to the product page, the below screenshots Figures 14 through 17 will illustrate the sketch design for each screen as prototype and mockup for the App.

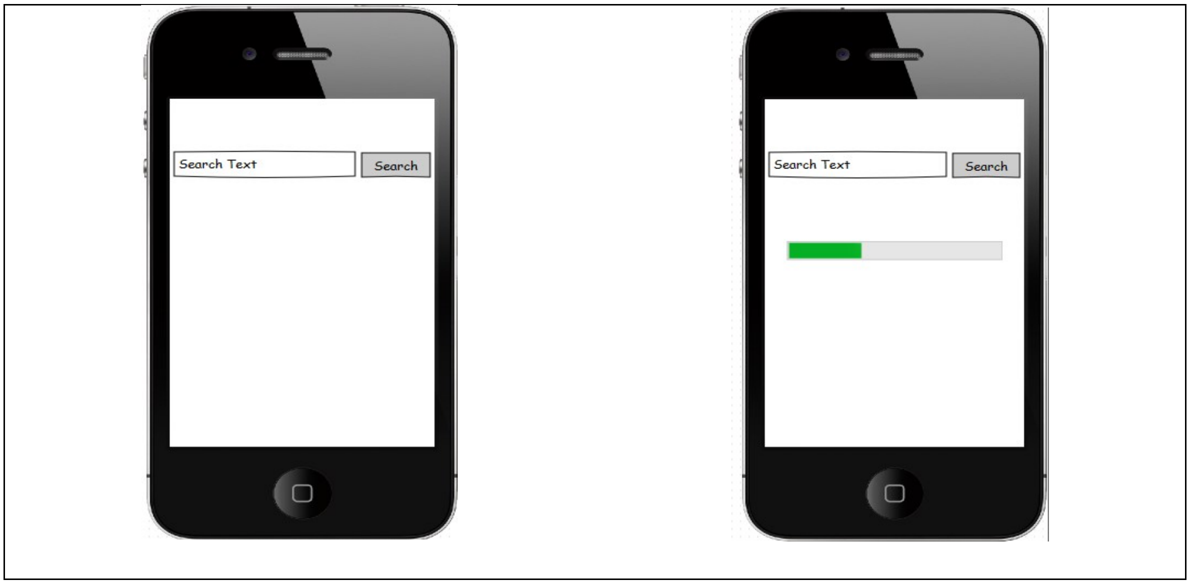


Figure 14: First Screen Sketch

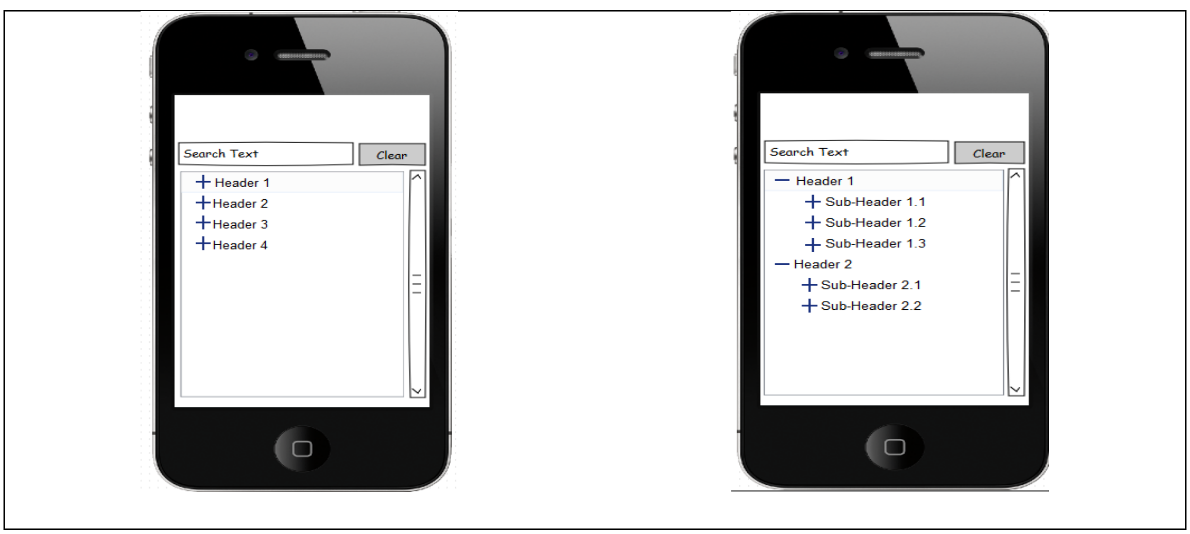


Figure 15: Search Results 1st and 2nd Level

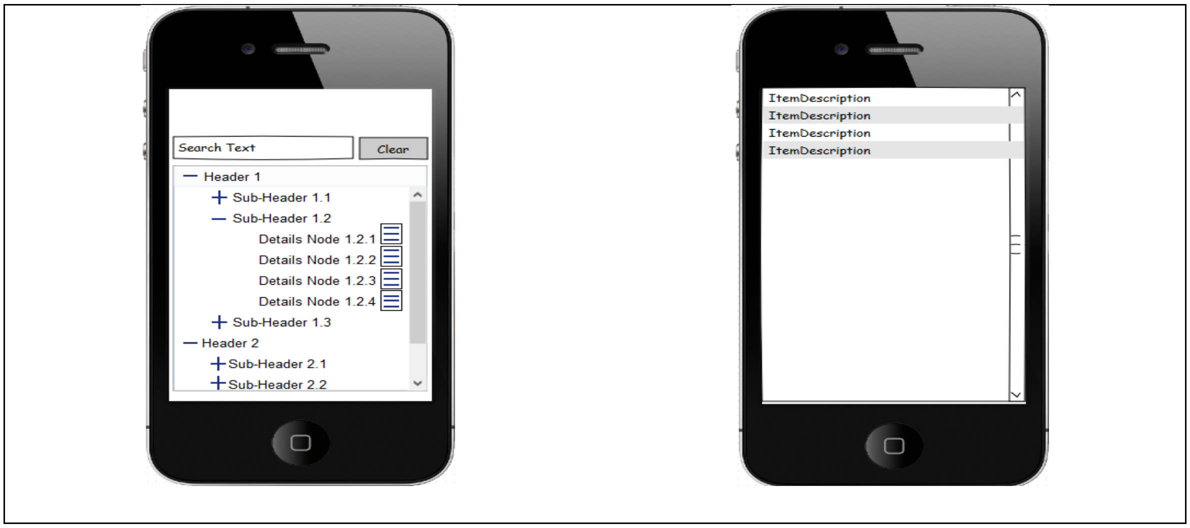


Figure 16: Search Results 3rd Level and Item Description Page

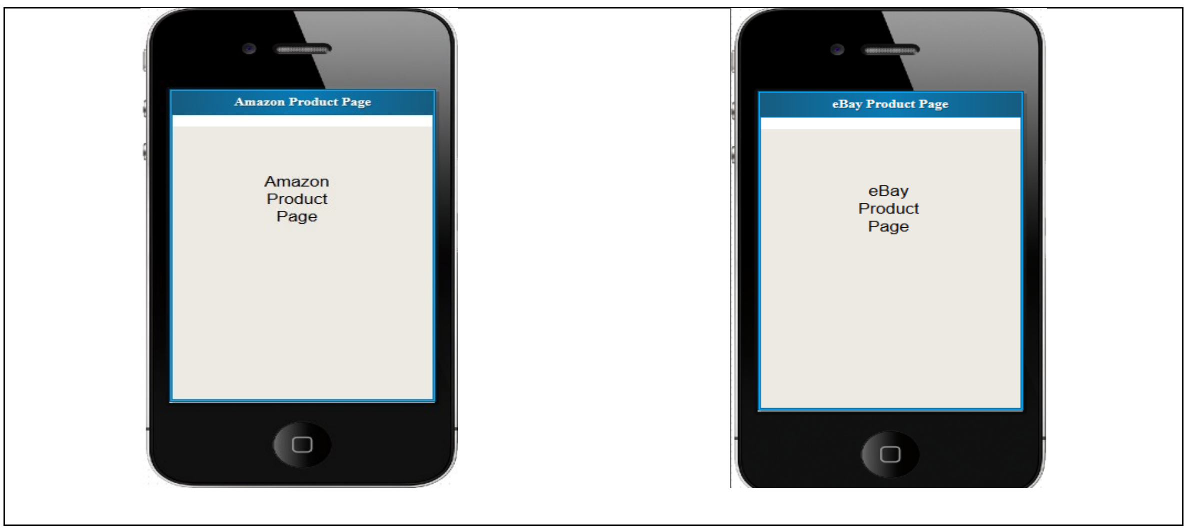


Figure 17: Product Page

4.4. Implementation

We named our proposed App in this project as QU Marketplaces Search App and it was developed on Android Studio Integrated Development Environment (IDE) using Java programming language. All responses are received in XML format.

The required API authentication account created for Amazon and eBay to have the security access key to enable our App to send requests for their services and get the required XML response for the available products.

The most important thing when developing such applications is to keep a simple interfaces in such way that is not distract the client from the main goal of the application, that is to find an easy and fast process for the target searched product. This application is indeed simple in appearance and in use.

Figure 18 shows Android Studio development environment as it was designed at the early stage with a simple graphic user interface, easy to use and without any elements of distractions.

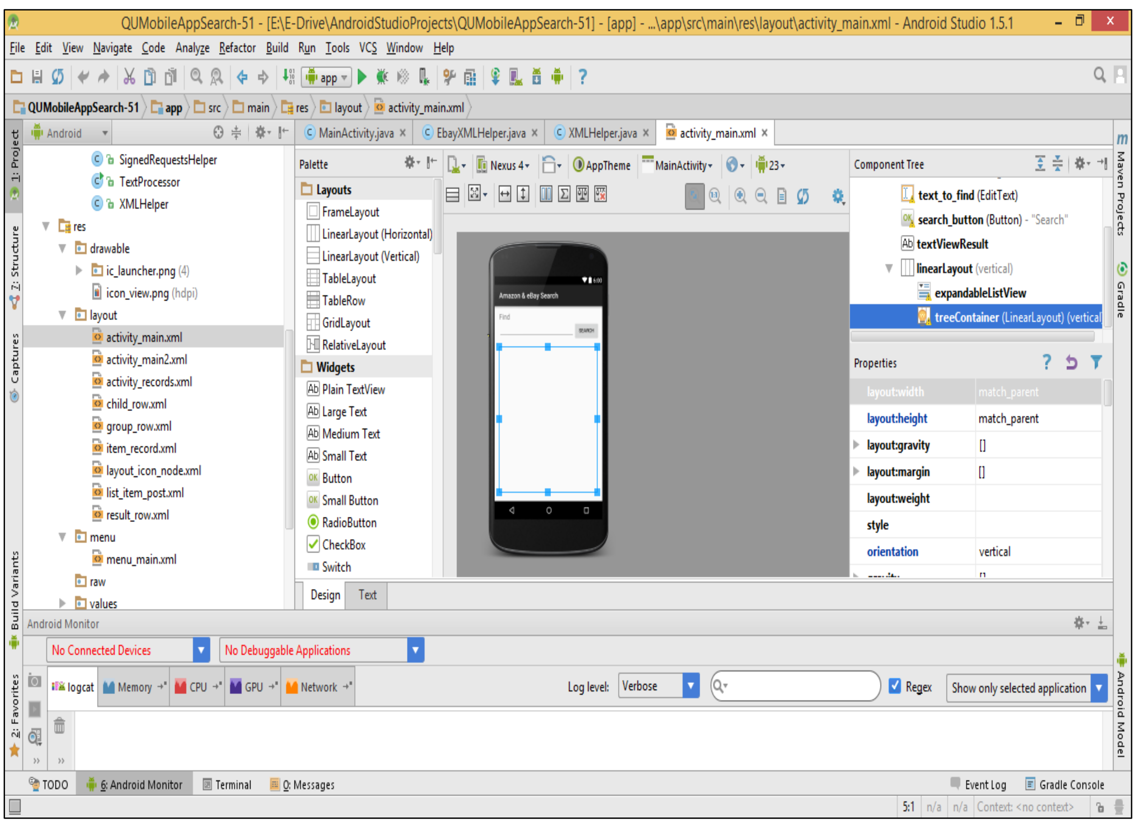


Figure 18: Android Studio IDE

All Product Advertising API requests are based on HTTP GET method and all responses are based on XML. Product Advertising API returns a well-defined set of tags for each kind of request. Then we use parsing mechanisms to pull out the XML responses for the data we will process in the conceptual discovery before displaying to the end user. The next three figures: Figure 19 through Figure 21 show how exactly APIs of Amazon and eBay were implemented in the code of the mobile App.

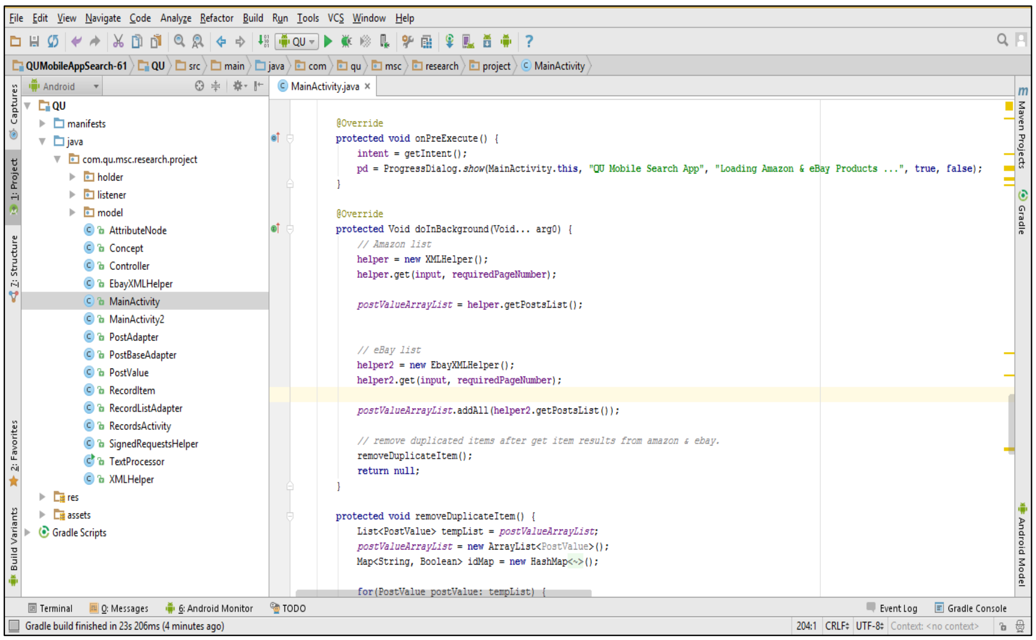


Figure 19: Main Activity class calling Amazon and eBay classes

```java
 * Method to read XML from {link XMLHelper#URL_MAIN}
 */
public void get(String in, int pagenumber) {
    /*
     * Set up the signed requests helper
     */
    SignedRequestsHelper helper;
    try {
        helper = SignedRequestsHelper.getInstance(ENDPOINT, AWS_ACCESS_KEY_ID, AWS_SECRET_KEY);
    } catch (Exception e) {
        e.printStackTrace();
        return;
    }
    String requestUrl = null;
    /* The helper can sign requests in two forms - map form and string form */
    /*
     * Here is an example in map form, where the request parameters are stored in a map.
     */
    Map<String, String> params = new HashMap<~>();
    params.put("Service", "AWSECommerceService");
    params.put("Version", "2009-03-31");
    params.put("ResponseGroup", "Small");
    params.put("AssociateTag", "q09d6-20");
    params.put("Operation", "ItemSearch");
    params.put("Keywords", in);
    params.put("SearchIndex", "All");
    // for pages retrieval
    params.put("ItemPage", ""+pagenumber);
    requestUrl = helper.sign(params);
    System.out.println("Amazon Signed Request is \"" + requestUrl + "\"");
```

Figure 20: Signed Request sent to Amazon

```java
String ebayURL = "http://svcs.ebay.com/services/search/FindingService/v1?paginationInput.entriesPerPage=10&OPERATION-NAME=findItems
int totalResults = 0;
Boolean startItemTag = false;
Boolean currTag = false;
String currTagVal = "";
private PostValue post = null;
private ArrayList<PostValue> posts = new ArrayList<~>();

public ArrayList<PostValue> getPostsList() { return this.posts; }

public  ArrayList<PostValue> fetchPostsListByType(ArrayList<String> type){...}

public int getTotalResults() { return totalResults; }

/**
 * Method to read XML from {link XMLHelper#URL_MAIN}
 */
public void get(String in, int pageNumbers) {
    String requestUrl = null;
    try {
        requestUrl = ebayURL+pageNumbers+"&keywords="+URLEncoder.encode(in, "UTF-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    System.out.println("eBay Request is \"" + requestUrl + "\"");
    try {
        SAXParserFactory factory = SAXParserFactory.newInstance();
        SAXParser mSaxParser = factory.newSAXParser();
        XMLReader mXmlReader = mSaxParser.getXMLReader();
        mXmlReader.setContentHandler(this);
        InputStream mInputStream = new URL(requestUrl).openStream();
        mXmlReader.parse(new InputSource(mInputStream));
```

Figure 21: Signed Request sent to eBay

Figure 22 presents the appearance of the final App version for user interface

Figure 22: Final Home Page User Interface

Chapter 5

Experimental Results, Validation and Evaluation

In this chapter, we first will provide walkthrough testing to evaluate the proposed App in general, then will present out evaluation setup with details on the datasets for different Mobile Apps. In the last part of the chapter we will present and discuss in details, the results of the evaluation of prediction in response to the research questions mentioned in Chapter 1.

5.1. Test Environment

It was planned to make the test on real smartphones instead of emulator to make actual measurement of the Mobile App.

Next, it was decided to make the test on two different smartphones and with different Android operating systems in order to see if there is any significant tool dependence when it comes to speed of processing a search activity. The first smartphone was HTC One E9 Plus with Android Lollipop 5.0 operating system. Second smartphone was Sony xperia Z3 Compact with Android Jelly Bean 4.3 operating system. Each test case was searched on both phones and through our App, MySmartPrice and Smartprix Apps (selected Mobile Apps for searching marketplaces). Basically, it was intended to perform the tests in similar conditions. In order to work properly, the mobile Apps need the smartphone to be connected to Internet. In this case two Wi-Fi connections at different speed were checked, 25 MB and 10 MB.

List of products to be searched with the Mobile Apps was prepared. The products on this list have a wide variety, from smartphones to sport items (bicycle). This list can be found in Appendix A.

We conducted two types of testing, one as walkthrough testing to evaluate the proposed Mobile App in general to make observation on the system output with its strength and weaknesses for further improvement.

Second testing as comparative point of view related to other online marketplaces' Apps to illustrate what our research can add to the end users.

5.2. Walkthrough Testing

Walkthrough testing purpose is to run the App multiple times and to notice the results and the performance to the end user.

Walkthrough testing will start after the user enter his search text as "IPhone 6s" and start searching to get the matched records by clicking on search button at first screen, at this moment a search operation on different e-marketplaces (Amazon and eBay) started The App fetches first patch of items' information from the integrated online marketplaces' APIs. The items' information are retrieved with xml files. These xml files include metadata such as link to item, title, item id, and other features list. The application will keep only the id and the title for summarizing results of the search to be displayed as keywords with different levels. Then the user can click on the keyword that matched to his search criteria, and either he can go for the next level or next keyword. The search process screenshots as below Figure 23 through Figure 27:

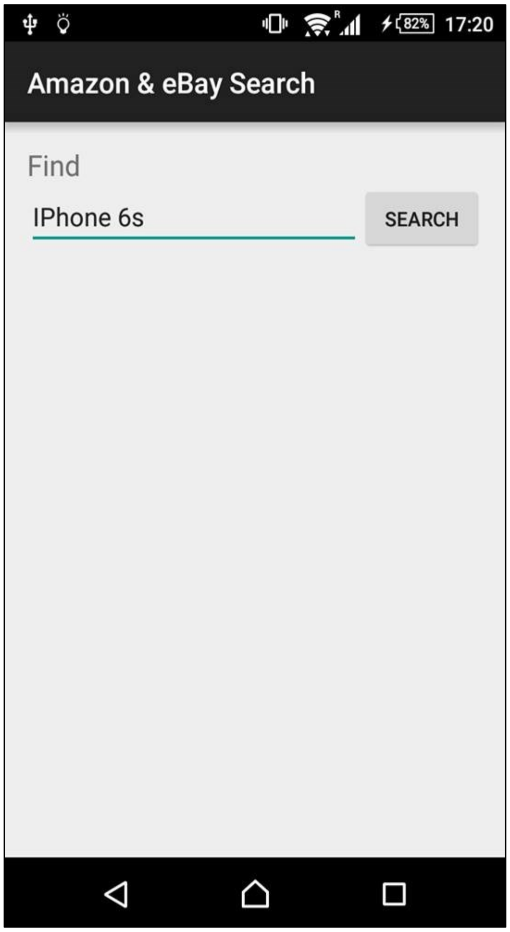


Figure 23: User Search Query

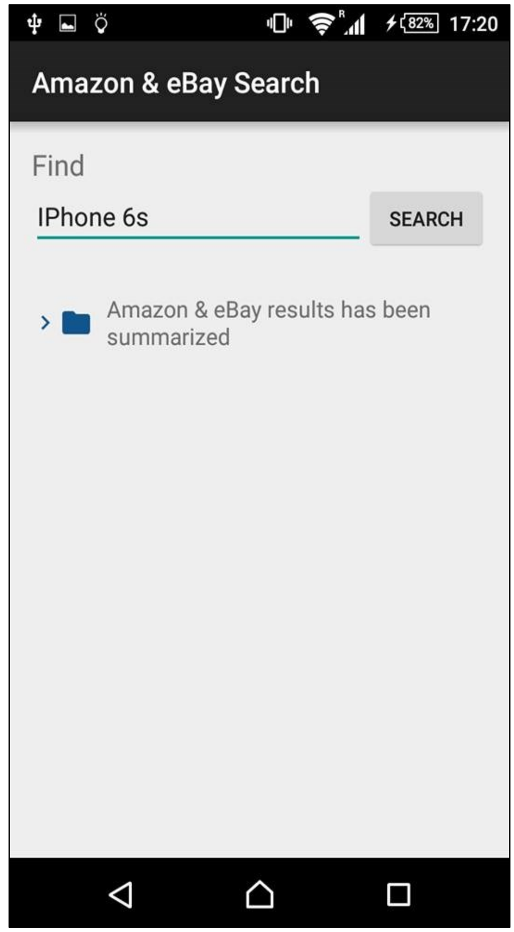


Figure 24: Search Result

The same screen will display the results in tree structure view for the extracted keywords:

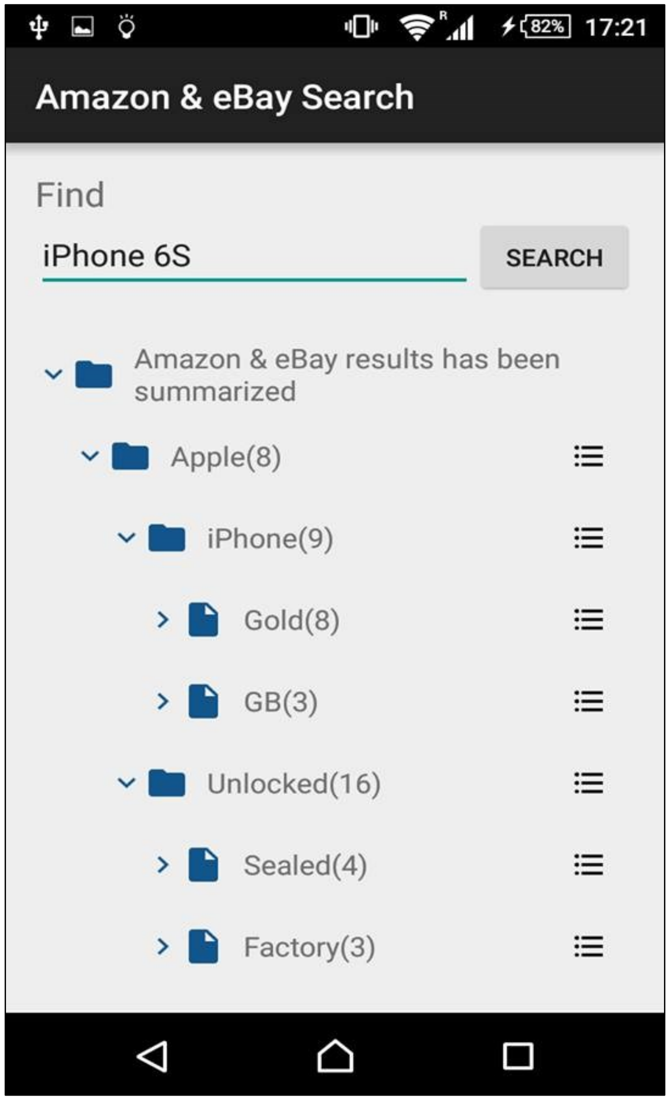


Figure 25: Tree Structure List View

A second screen will be a list view for the records that match the keyword hit by the user. In this case, we want to search for Gold iPhone 6S. Each record of this list view will contain the link to the target product; once the end user clicks on one of the entries the App will navigate him to the source that met all conditions of the tree. In other words, the user will be directed to Amazon or eBay, which are the only e-marketplaces included up to this moment. Clicking on the last record of the list will redirect the user to eBay.

Second screen displaying all the matched products with link to the source of the product page as below Figure 26 and Figure 27:

Figure 26: Products List View

Figure 27: Product Home Page

We noticed that results met user expectation for the App output results, but the first process of fetching the target products and re-organizing the results output to the user is taking quite long time that is almost 70-90 seconds which considered as long process for Mobile App.

We split the first process to two sub-tasks, first sub-task to fetch the products from online marketplaces, and second sub-task to build the conceptual tree for the user as Figures: 28 and 29. We recognized that fetching the results is talking normal time, while building the conceptual tree from the summarization process is talking the longest time to finish the process. Therefore, we checked the XML file generated from the App and noticed the file contains 110 matched records for each call to the APIs which make the summarization process to take that much time.

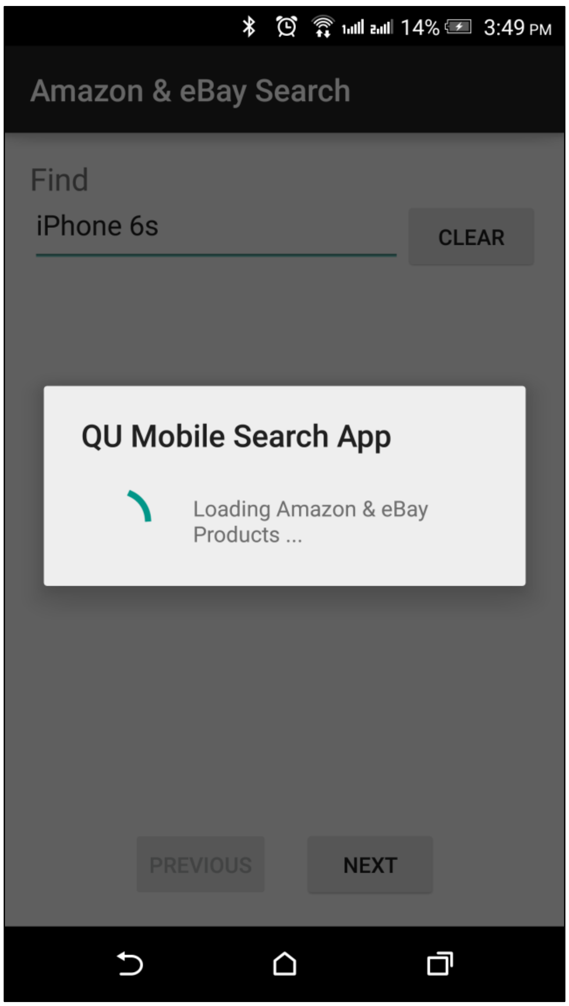


Figure 28: Loading sub-task process

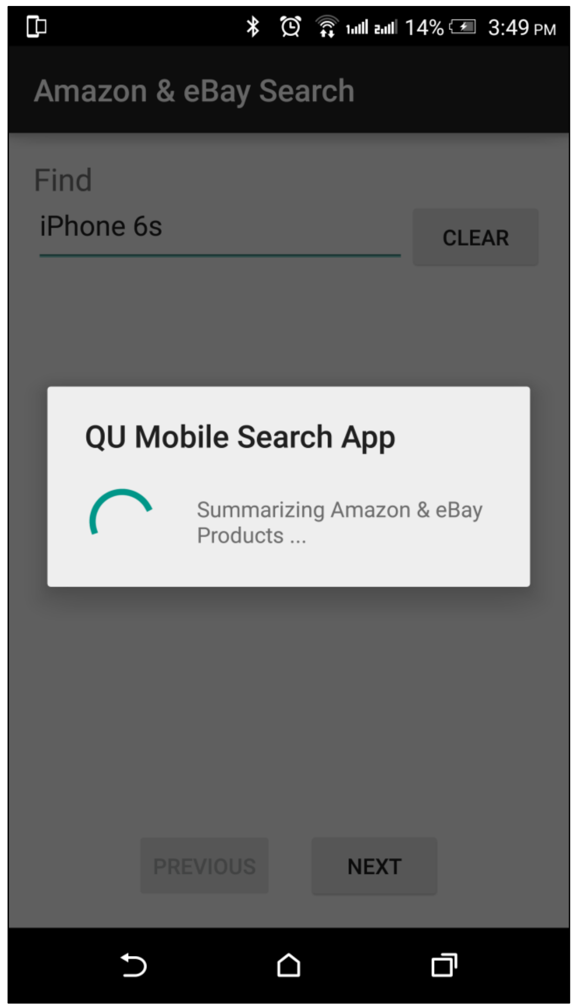


Figure 29: Summarizing sub-task process

Amazon API returning by default first 10 records matched to the sent request call, while eBay returning the first 100 records. We customized our App API calls to return the first 10 records from each online marketplace to minimize the summarization process on these results and provided a next button to fetch the next 10 records for each marketplace.

After customizing our App API calls, the execution time for the process dropped from 70-90 seconds to 5-7 seconds which increased the performance of the process.

Finally, after we customized the proposed solution by fetching records from the marketplaces by 10 results for each page, we can proceed now for comparison with other marketplaces' Apps.

5.3. Comparison with e-marketplace Apps

This competitive analysis will make a comparison between the proposed App, MySmartPrice and Smartprix dedicated Apps to be carried out. This analysis considered as quality measurement with the help of three metrics: number of clicks to reach the target product, time spent to reach the target product and number of screens navigation for the list of products as Figure 30:

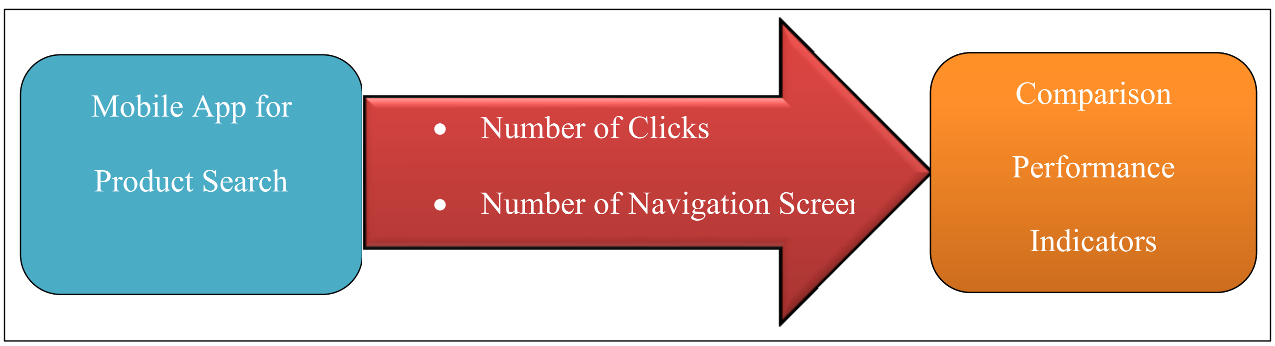


Figure 30: Quality Performance Measurement

The comparison built on three quality index factors as below:

A. Number of Clicks

Number of clicks counted for each search item in Table 5 below for each App, keywords between brackets not entered at the search text and it is kept as user preference item additional information which he is looking for.

Table 5: Number of Clicks to reach target product

Items	Proposed App	MySmartPrice App	Smartprix App
iPhone 6S (Gold)	3	4 to 8 *	4 to 6 *
iPad mini (2nd generation)	3	4 to 6 *	6 to 8 *
LG TV (HD)	3	6 to 8 *	6 to 8 *
Canon (EOS SLR)	3	4 to 6 *	4 to 6 *
Nikon (DSLR)	3	3 to 4 *	3 to 5 *
Calvin Klein watch (men)	3	5 to 7 *	3 to 5 *
Trek (road bicycle)	3	5 to 7 *	6 to 8 *

(*) for number of clicks may vary depends on options checked in filtering option

B. Number of Navigation Screens

Navigation guides users between different parts of the App. The App's structure was organized according to the content. Moreover, the navigation through the App should be intuitive and predictable. To determine the type of navigation best suited to the App, we identify typical paths they might take through the App as the results returned.

On MySmartPrice and Smartprix Apps when users move from one view to the next, they observe scenes containing imagery, actions, and new contents. Thus, they can get distracted very easily. When these two are compared to our App, an important difference can be noticed: the simplicity of our design, without any elements of distractions. Below Figures 31 and 32 clearly shows how easy is to get distracted when using MySmartPrice and Smartprix Apps compared to our App.

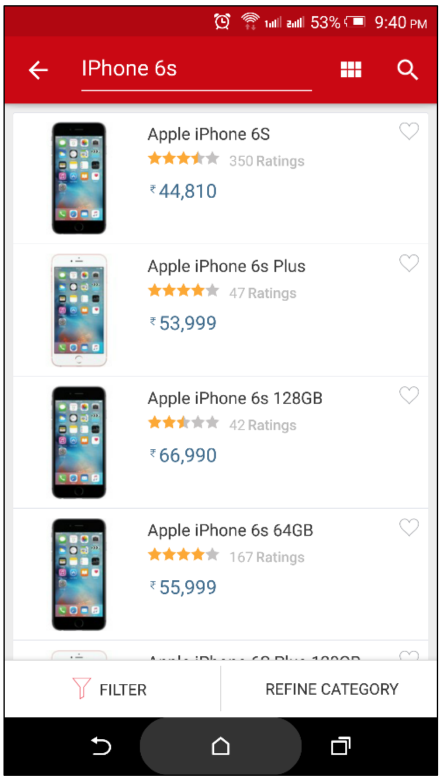


Figure 31: MySmartPrice Search Result Screen

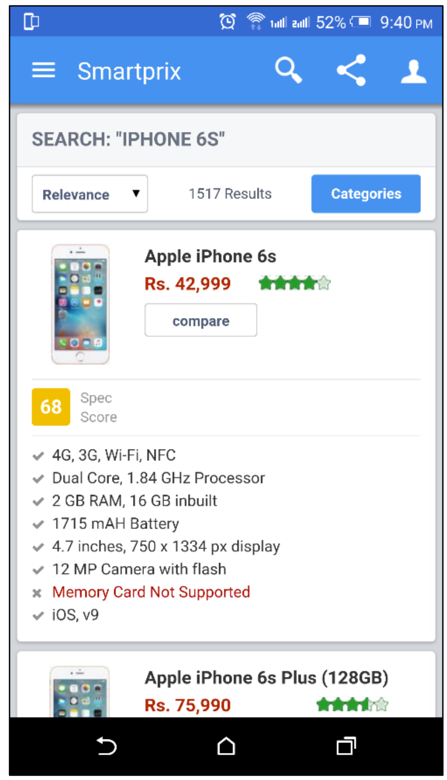


Figure 32: Smartprix Search Result Screen

In Table 6 the results of a comparison between proposed App, MySmartPrice and Smartprix Apps concerning the number of screens a user needs to navigate before reaching the target product is presented. It is easy to notice that the proposed App is much simpler and consists in only two screens to navigate through before reaching the target product. The same thing cannot be said about MySmartPrice and Smartprix Apps: in average they carry the user through average 3 screens, based on the different options selected in the filters. A sample can be seen in Figure 33 where some filters enabled in MySmartPrice and Smartprix Apps are exposed.

Table 6: Number of Navigation Screens

Items	Proposed App	MySmartPrice App	Smartprix App
iPhone 6S (Gold)	2	3	3
iPad mini (2nd generation)	2	3	3
LG TV (HD)	2	3	3
Canon (EOS SLR)	2	3	3
Nikon (DSLR)	2	3	3
Calvin Klein watch (men)	2	3	3
Trek (road bicycle)	2	3	3

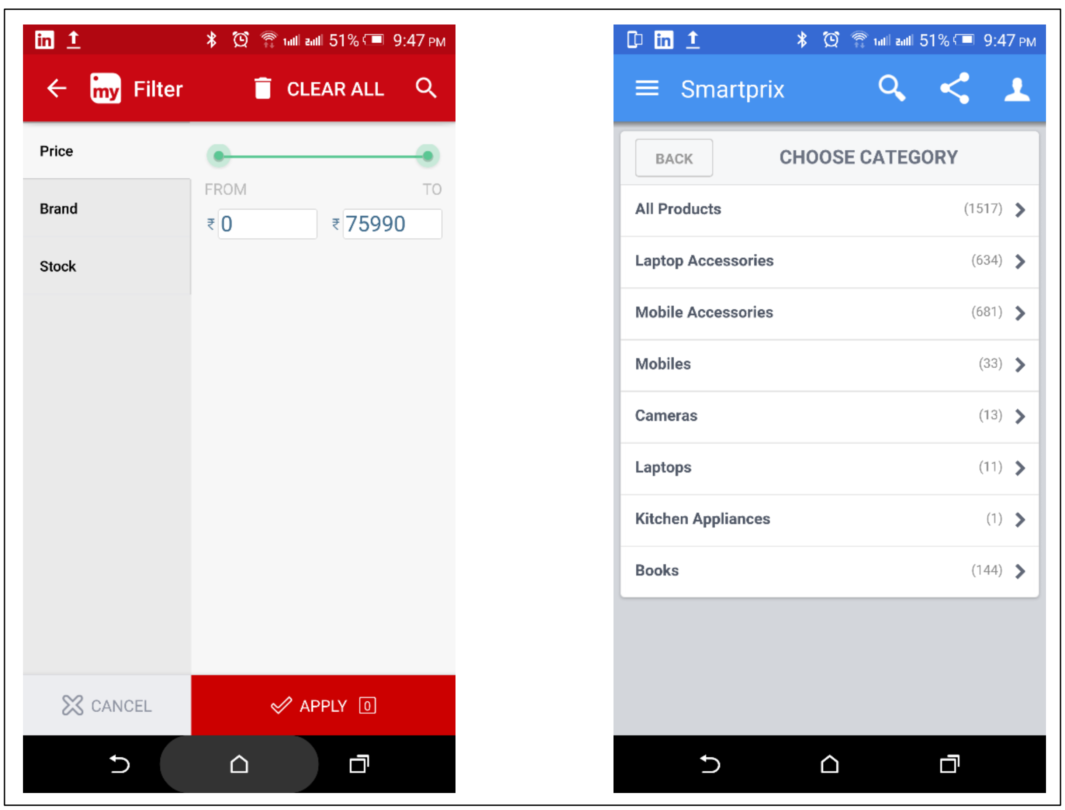


Figure 33: Filter Types in Mobile Apps

MySmartPrice and Smartprix Apps are not capable to guess user requirements but they give to user many options to choose from. Although this could be very useful, many times it proves to be a distraction and time consuming as the user has to click more times and go through available filters. On the other hand, our application is very simple and is capable of guessing client's requirements from the initial query, by obtaining the matched results from different marketplaces with other keywords that are entered by user and are united to his input. Then, the results are presented in a tree structure grouped based on features like color, model and other options.

C. Time Consuming

Time is defined as the total number of seconds needed to reach the target product that can be found. Time index is strictly connected to navigation screen index, the higher number of screens a user has to navigate through; the higher will be the time to reach the target product. This is another reason why our App has just two screens. Below Table 7 for time results of the Apps:

Table 7: Time Index in seconds

Items	Proposed App	MySmartPrice App	Smartprix App
iPhone 6S (Gold)	8 seconds	20 seconds	25 seconds
iPad mini (2^{nd} generation)	10 seconds	35 seconds	30 seconds
LG TV (HD)	9 seconds	45 seconds	35 seconds
Canon (EOS SLR)	8 seconds	20 seconds	35 seconds
Nikon (DSLR)	6 seconds	26 seconds	32 seconds
Calvin Klein watch (men)	7 seconds	27 seconds	23 seconds
Trek (road bicycle)	5 seconds	55 seconds	55 seconds

5.4. Evaluation

The results obtained for MySmartPrice and Smartprix Apps perform very similar. The app developed and presented in this project is much faster and easier to use than both applications, especially when products were searched based on their category as illustrated in Figure 34 and Table 8 for the number of clicks, navigation screens and time taken to reach the end user target product. For example, we can have a better look at the last entry, Trek. The end user wanted to buy Trek road bicycle and if he wants to perform the search with the help of this App, it is enough to write just ‘trek’ in the search field. On the other hand, when the search is performed on MySmartPrice and Smartprix Apps the process is much slower because these app returns results such as candy bars, star trek movie, and items related to trekking but no sign of bicycles as Smartprix. In order to have an efficient output, user has to write also ‘bike’ into the search field. The same for MySmartPrice, since it has different approach to reach the products, it display all bicycles for the first result, but user need to search for the required type of bicycles for mountain or road.

Therefore the proposed App concept proved to be faster and much user friendly since it created automatically a folder called bike and added it to the tree structure as below Figure 34:

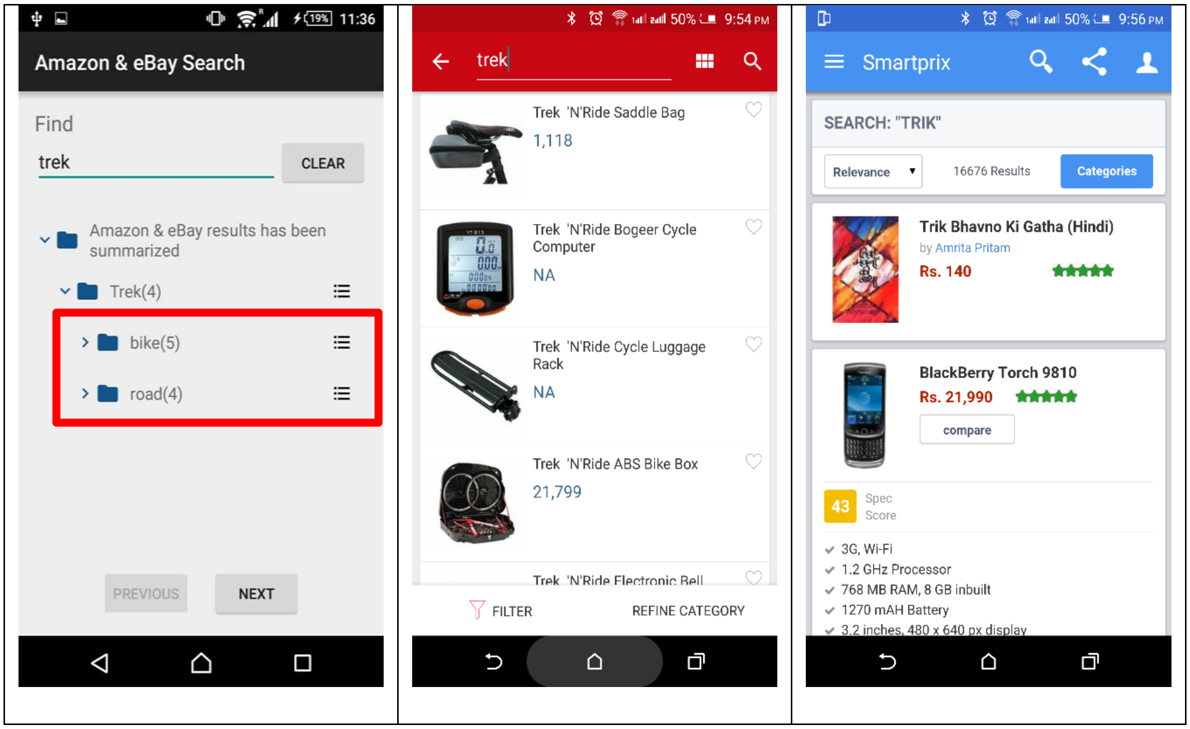


Figure 34: Comparison between Proposed App and MySmartPrice/Smartprix Apps

In Table 8: walkthrough testing results as below, the complete experimental results can be observed and other test cases provided in the appendix:

Table 8: Walkthrough Testing Results

Index	Proposed App	MySmartPrice App	Smartprix App
Number of navigation screens	2	3	3
Number of clicks	3	5 to 7 *	5 to 7 *
Time in seconds (average)	8	33	34

(*) for number of clicks may vary depends on options checked in filtering option

Chapter 6

Conclusion and Future Work

This chapter outlines the main conclusions obtained based on the research conducted in the last months, and proposes recommendations for future work related to improvement of the proposed App.

6.1. Conclusions

Nowadays, one of the most important types of e-businesses is the transactional models called e-marketplaces. It incorporates a website, a computer and Internet but in this project the computer was replaced with mobile devices (smartphones).

Based on the results of the study case presented in this research two main conclusions can be drawn regarding the difference between the proposed App on one side, MySmartPrice and Smartprix Apps on the other side:

- The App will search simultaneously and retrieve results from different e-marketplaces in one place. This means a user will have the advantage of saving time because the search is not performed on each marketplace alone and to make comparison and user will have more products from different sources.

- The algorithm implemented in the proposed App has a goal to find the target product with minimal number of clicks, which is ensured by entity discovery and conceptual browser methodologies.

Conceptual browser is domain independent and can be adapted to any other unknown domains. This research adapted conceptual browser to validate hidden data, shorten the path of the query and automatically reformulate the query by user selections from interface

6.2. Future Work

This study can be improved in the future for the following:

- Additional plugins and add–on like adding extra e-marketplace APIs is recommended. In the future Alibaba.com API and other Middle East online marketplaces can be introduced by implementing XMLHelper abstract class for each marketplace integrated API.
- The proposed App can be enhanced with additional features like sort, or order functionalities and other possibilities provided by e-marketplace APIs. That can be combined with our App by filtering the request call to the marketplaces' engines, and that will not have effect to the proposed Mobile App performance.
- Introducing more attributes for conceptual browser like: product group and category description. That will provide more choices to the end users to make a specific category selection for retrieving results that match his category.

Bibliography

[1] Gordijn, J., Akkermans, H. and van Vliet, Designing and Evaluating E-business Models Submission, 12th International Conference on Knowledge Engineering and Knowledge Management, Juan-les-Pins, France, 2000.

[2] Hatch, D A, Barnes, J and Fisher, S.M, Creating your own E-business in 10 easy steps, LSU AgCenter, LA, 2003.

[3] L Turban, E., E-marketplaces: structures, Mechanisms, Economics, and Impacts. Pearson Prentice Hall. Chapter 2, p.37; p. 50 – 55, 2008.

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

[5] Campbell, C., Ray, G. and Muhanna, W. A. 2005, Search and Collusion in Electronic Markets, Management Science, Vol 51 (3), p. 497 – 507.

[6] Hanson, W. and Kalyanam, K., Internet Marketing and e-Commerce. Mason, OH, Thomson, 2007.

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

[8] Jarvenpaa, S. L., Tractinsky, N. and Vitale, M. 2000, Consumer Trust in an Internet Store. Information Technology and Management, Vol. 1 (1 – 2).

[9] Baltzan, P and van Phillips, A, Essentials of Business Driven Information Systems. McGraw-Hill Irwin, New-York, New-York, 2009.

[10] Ameen, S., Difference between B2B and B2C Explained Thoroughly, 2015.

[11] Custora, Holiday 2013 E – commerce recap, 2013.

[12] G.Schmidt and S. Ströhlein, Relations and Graphs, Springer-Verlag, 1989.

[13] Belohl´avek, R.: Introduction to formal context analysis. Tech. rep., Dept of Computer science. Palack y University, Olomouk, Czech Republic 2008.

[14] A. Jaoua, N. Belkhiter, J. Desharnais, and T. Moukam, Proprietes des dependances difonctionnelles dans les bases de donnees relationnelles, Information Systems Operational Research J. 30(3):297-316 (1992).

[15] Ali Jaoua, Habib Ounalli, Nadir Belkhiter, Automatic Entity Extraction From an N-Ary Relation: Toward a General Law for Information Decomposition. Inf. Sci. 87(1-3): 153-169 (1995).

[16] 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).

[17] Structured Conceptual Meta-Search Engine, A. Jaoua, M. Lazhar Saidi, Ahmed Hasnah, jihad M. Al-Jaam, Sahar Ahmed, Baina Salem, Noura Rashid Shereen Shareef, Suad Zaghlan. Fourth International Conference on Concept Lattices and Their Applications, Hammamet-Tunisia 2006, 30/10 au 1/11/2006.

Appendix

Appendix A: Test Cases

Search context as below, note between brackets not entered

1. Sunglasses (Polarized)
2. Trekking Bag
3. Wind Chime
4. Car Charger (Samsung)
5. Whiteboard Marker (Color)
6. Deodorant (Dove)
7. Axe (Sports)
8. Transformers (Figure)
9. HDMI (cable)
10. Alarm clock
11. Cake Tray (Mold)
12. Microwave (Stainless)
13. Sennheiser (Headphones)
14. Dumbbells (Adjustable)
15. Stapler (Swingline)
16. Table Fan (Clip)
17. Door Handles (Wave Style)
18. Picnic Table (Folding)
19. Denim's (Jeans)
20. Batteries (Duracell)

Appendix B: Number of Clicks

Table 9: Extended Test Cases – Number of Clicks

Items	Proposed App	MySmartPrice App	Smartprix App
Sunglasses (Polarized)	4	6 to 8 *	5 to 7 *
Trekking Bag	4	4 to 6 *	4 to 6 *
Wind Chime	4	6 to 8 *	7 to 9 *
Car Charger (Samsung)	5	4 to 6 *	5 to 7 *
Whiteboard Marker (Color)	5	4 to 6 *	4 to 6 *
Deodorant (Dove)	4	5 to 7 *	5 to 7 *
Axe (Sports)	3	6 to 8 *	6 to 8 *
Transformers (Figure)	3	5 to 7 *	4 to 6 *
HDMI (Cable)	3	4 to 6 *	4 to 6 *
Alarm Clock	3	5 to 7 *	6 to 8 *
Cake Tray	4	2 to 4 *	2 to 4 *
Microwave (Stainless)	5	3 to 5 *	3 to 5 *
Sennheiser (Headphones)	3	2 to 4 *	3 to 5 *
Dumbbells (Adjustable)	4	3 to 4 *	3 to 4 *
Stapler (Swingline)	3	3 to 5 *	2 to 5 *
Table Fan (Clip)	3	4 to 6 *	5 to 7 *
Door Handles (Wave Style)	4	4 to 6 *	5 to 7 *
Picnic Table (Folding)	4	4 to 6 *	5 to 7 *
Denim’s (Jeans)	3	4 to 6 *	5 to 7 *
Batteries (Duracell)	3	4 to 6 *	5 to 7 *

*number of clicks depends on options checked in filtering option

Appendix C: Number of Navigation Screens

Table 10: Extended Test Cases – Number of Navigation Screens

Items	Proposed App	MySmartPrice App	Smartprix App
Sunglasses (Polarized)	2	3	3
Trekking Bag	2	3	3
Wind Chime	2	3	3
Car Charger (Samsung)	2	3	3
Whiteboard Marker (Color)	2	3	3
Deodorant (Dove)	2	3	3
Axe (Sports)	2	3	3
Transformers (Figure)	2	3	3
HDMI (Cable)	2	3	3
Alarm Clock	2	3	3
Cake Tray	2	3	3
Microwave (Stainless)	2	3	3
Sennheiser (Headphones)	2	3	3
Dumbbells (Adjustable)	2	3	3
Stapler (Swingline)	2	3	3
Table Fan (Clip)	2	3	3
Door Handles (Wave Style)	2	3	3
Picnic Table (Folding)	2	3	3
Denim's (Jeans)	2	3	3
Batteries (Duracell)	2	3	3

Appendix D: Time Consuming

Table 11: Extended Test Cases – Time in seconds for reach target product

Items	Proposed App	MySmartPrice App	Smartprix App
Sunglasses (Polarized)	5 seconds	15 seconds	12 seconds
Trekking Bag	8 seconds	10 seconds	15 seconds
Wind Chime	10 seconds	30 seconds	25 seconds
Car Charger (Samsung)	13 seconds	30 seconds	25 seconds
Whiteboard Marker (Color)	10 seconds	20 seconds	10 seconds
Deodorant (Dove)	6 seconds	20 seconds	40 seconds
Axe (Sports)	8 seconds	40 seconds	60 seconds
Transformers (Figure)	3 seconds	5 seconds	10 seconds
HDMI (Cable)	2 seconds	5 seconds	10 seconds
Alarm Clock	6 seconds	15 seconds	5 seconds
Cake Tray	8 seconds	20 seconds	5 seconds
Microwave (Stainless)	8 seconds	20 seconds	10 seconds
Sennheiser (Headphones)	5 seconds	5 seconds	15 seconds
Dumbbells (Adjustable)	11 seconds	20 seconds	16 seconds
Stapler (Swingline)	13 seconds	15 seconds	10 seconds
Table Fan (Clip)	7 seconds	25 seconds	25 seconds
Door Handles (Wave Style)	10 seconds	6 seconds	60 seconds
Picnic Table (Folding)	8 seconds	15 seconds	15 seconds
Denim's (Jeans)	8 seconds	7 seconds	5 seconds
Batteries (Duracell)	10 seconds	10 seconds	5 seconds

Appendix E: Average Number (Clicks, Screens, Time)

The average of experimental results can be observed in table below:

Table 12: Extended Test Cases – Average Number of (Clicks, Screens and Time)

Index	Proposed App	MySmartPrice App	Smartprix App
Number of navigation screens	2	3	3
Number of clicks	3.7	4.1 to 5.55 *	4.35 to 6.4 *
Time in seconds (average)	8	11.7	18.9

* number of clicks depends on options checked in filtering option