Team Work:

- *Adel Mahmoud Abdelmoneim Ali Amer.*
- *Ahmed Samy Ahmed Abdelaziz.*
- *Dina Ashraf Fawzy Abdelhameed.*
- *Fatma Magdy Elsayed Albadawy Mohamed.*
- *Mahmoud Ibrahim Mohamed Ahmed.*
- *Yasmin Adel Mohamed Elhady Almahdawy.*

--..--..--..--..--..--..--..--..--..--..--..--

Department of Information Systems,
Faculty of Computers and Informatics
Zagazig University, Egypt

2018

Table of Contents

- Acknowledgment …… 3
- Learned skills…….. 4
- Abstract…….. *5*
- Chapter 1: Introduction.....6

  1.1: Background 7
  1.2: Motivation 8
  1.3: reasons for the aspect ……..... 9
  1.4: mission ……...……..... 10
  1.5: vision ……...……..... 11
  1.6: features of our project ……..... 12
  1.7: our project strength ……..... 13
  1.8: objectives of project ……..... 14
  1.9: problem areas for our user segment 15
  1.10: our methodologies ……..... 16
  1.11: phases of our project ……..... 17
  1.12: time schedule ……..... 24

- Chapter 2: System design and construction…….25

  2.1: Strategies for system design ……..….. 26
  2.2: Application architecture ……..... 27
  2.3: Database design …… 28
  2.4: Input Design …… 30

2.5: Output Design …….. 32
2.6: User interface design ……... 34
2.7: Software design 18 …….. 36
2.8: Object oriented design ……. 38
2.9 System implementation ……. 40
2.10 System support …… 42

- Chapter 3: System Design …….. 44
  - 3.1 Data Flow Diagram.....45
  - 3.2 Use Case Diagram.....47
  - 3.3 ER Diagram.....51
  - 3.4 Business Model.....52
  - 3.6 Entity details.....53
- Chapter 4: system Development……...58
  - 4.1 E-Commerce Web Application.....59
  - 4.2 Business Intelligence Analytics.....73
  - 4.3 User Tracker.....78
- Chapter 5: Sentiment Analysis……... 79
  - 5.1 Introduction.....80
  - 5.2 Sentiment Analysis.....80
  - 5.3 Approaches for Text Classification Task.....81
  - 5.4 Methodology.....82
  - 5.5 Results.....88
- Chapter 6: Conclusions and Future Work …….. 91
- Appendix …… 93
- References …….. 101

Acknowledgement

Thank " Allah " for being kind, helpful and supporting to us during our life and specially to achieve this project.
Also, we would like to prevent thankfulness to our families for giving us their attention, supporting and carefulness to accomplish this project.
We express our deep since of gratitude and sincere to our supervisors:

Dr. Ayman Mustafa.
Eng. Ahmed Samy.
Eng. Mirna Magdy.

For their guidance, valuable suggestions, generous help and careful encouragement given through this project.
Finally, we hope that our project obtain your attention and specification.
Thanks
Team work project

Learned skills

These words we wrote about personal experience that we gained in small life…

- Challenge every problem face you and don't raise white flag.
- Learn how to work with team before starting our career.
- There is always time...!

Abstract

- Our idea is to perform sentiment analysis on reviews (Comments) of agents on E-business web application to get feedback about products in the application, whether reviews about this product is (positive – Neutral - Negative) …

- This will help us give this impression about the product to its creator source to improve their product.

Chapter 1

Introduction

Background

In previous many companies were having a big amount of data which cannot be exploited in best way... especially if this data can help us improve our productivity efficiency and doing this can lead to:

- **Reduce efforts made to improve our productivity**
- **Improve customer service**
- **Achieve customer satisfaction**
- **Predict future issues in a scientific way**

Motivation

We wanted to help those peoples as we may be in their situation by making a web application (E-business) it's Data can be used to improve the website using Data Mining:

- ✓ **Allow the user to purchase products electronically**
- ✓ **Allow the user to comment on and show their opinion in the product**
- ✓ **The analysis of these comments and the production of reports and drawings give us a background on whether the general views about this product are negative or positive**
- ✓ **Send these reports to our contracted manufacturers to improve their products and meet with us all new**
- ✓ **Repeat the previous operations until the customer reaches 100% satisfaction**

Reasons for the aspect

Our competitors such as:

1. Jumia Eg.
   - Jumia Egypt The #1 online shopping in Egypt.
   - Platform specializing in electronics, fashion, home appliances and children's items.
   - Jumia, Egypt's most convenient online shopping mal.

- They ignore the user comments on products and does not analyze them.
- They does not use BI analytics to make decisions.

Mission

We try to make a Web application that is:

- ❖ at the best picture ,
- ❖ Satisfactory to the eye,
- ❖ user friendly,
- ❖ Urges the user to show his opinion in the product,
- ❖ Urges the user to Trust in our product,
- ❖ Arrives to what he wants with the least number of clicks.

Vision

We are looking forward to extending our web application that cover all of Egyptian –At first- customers who can shop online and Urge them to purchase products with high quality & to give us their opinion to get better products in the future after that our goal is to :

1. ***Cover all of the countries which can be a good customers for us.***
2. ***used as a mobile application not only a web application.***
3. ***Add more features to make our web application more usable & to use all users steps to make a purchase using a user tracker page available for the admin (we are already working on it).***

Features of our project

- *Admin Login: - Admin login's to the system using his Admin ID and password.*
- *Add Post: - Admin add products image and features.*
- *Add Keywords: - Admin add keywords in database so that system will match the comment with the keywords in database and will rate the product.*
- *User Login: - User login's to the system using his user ID and password.*
- *View Product: - User can view product images.*
- *View Product Features: - User can view features of the product to know about the product.*
- *Comment: - User will comment on the product.*
- *View Comment: - User can view comment of other user's.*
- *Rating Calculation: - System will match the comment with the keywords in database and will rate the product.*

Our project strength

- We collect advantages from other competitors application such as :
    1. Jumia Eg(offers).
- We use new trends to Monitor customer's needs and satisfaction :
    - Sentiment analysis.
    - BI analytics.

The objective of the project

World has an enormous amount of data , which in Previous times most of it was not exploited ,But With the passage of time many companies thought about how to use it's Data and Many sciences appeared that it's whole job is to make the most benefit from data …

And sentiment analysis is a field in Data mining which is one of those sciences , it performs analysis using algorithms on Text (Data) to give us information can help us to improve our work and our way of production and to tell us if our curve is Progressive or Descending .

So our idea is to perform sentiment analysis on reviews (Comments) of agents on E-commerce web application to get feedback about products in the application, whether reviews about this product is (positive – Neutral - Negative)

This will help us give this impression about the product to it's creator source to improve their product.

Problem areas for our user segment

1. Lack of user experience.
2. Internet access.
3. Payment methods.
4. Speed of service delivery.
5. User's Distrust.

Our Methodology

1. Web designing for user interface *(HTML5, CSS3, JavaScript, Bootstrap, and Ajax).*
2. Backend for development issues *(Php, Php MyAdmin).*
3. *(Rapid Miner)* for data mining issues.
4. Business intelligence analytics using *(Cognos)*.
5. Data sets for Reviews source.
6. Data sets for Data of products.

Phases of our project

1. **The Discovery Phase**

The Discovery Phase is designed to capture the detail level view of requirements from the perspective of business, creative and technical stakeholders. Fundamentally, the Discovery Phase is focused on answering the question: “What do you want your website to do?” In an ideal situation, a business case has been developed and approved that can provide a guide to the entire Discovery Phase.

❖ **Within the Discovery Phase are several steps:**

1.1. **Kicking off the Project** - It is important to conduct a formal kickoff meeting. The agenda should describe who will be involved, what topics will be discussed and what anticipated next steps look like. In most project kickoff meetings, a project manager is appointed as well as designated business, marketing and technology stakeholders who can speak to the concerns of the business.

1. 2. **Developing a Project Plan** - Once individual responsibilities for the team are defined, the project manager should develop a baseline project plan that documents the work activity tasks, dependencies, resources and timelines for the individual project elements.

1.3. **Gathering Business Requirements** - For most projects, detailed business and user experience and technical requirements are not well defined. To ensure that you have a comprehensive view of the requirements, each component of the website should be further defined through a series of business stakeholder interviews.

1.4. **Gathering Technical Requirements** - Once the core business requirements have been defined, it is important to identify third-party tools and applications that can be affected. This effort can include identifying whether the current infrastructure will support the project or whether new hardware is appropriate. As part of this effort, you should also focus on setting security requirements and reviewing licensing policies.

1.5. **Gathering User Experience Requirements** - To begin, check to determine whether or not your organization has documented web or style standards. In the event that standards exist, it will be important to align creative deliverables within these requirements.

While many people think the user experience consists of purely graphic design, the truth is that the field of information architecture (IA) is equally important. IA affects the overall hierarchy and organization of information on the website, drives the navigational model and provides a consistent labeling scheme that aids in clarity for users.

Discovery Phase Deliverables - Now that you've completed requirements gathering, the website development methodology calls for the creation of:

1- The functional requirements document

2- The information architecture document

3- The Content Management System (CMS) implementation guide

Once everything has been appropriately documented, it is time for the Implementation Phase.

2. The Implementation Phase

This is where we start building to specification. If we were to compare the building of the website to the building of a house, the Discovery Phase is where you meet with an architect, interior designer and landscaper to prototype your dream home. The implementation phase would be where we clear the land, lay the foundation and actually build the home.

❖ The steps involved in the Implementation Phase include:

2.1. **Starting Development** - Focus first on the initial setup and configuration of the three environments: development, staging and production hosting environments. It is important to note there are a number of ways the software environment can be configured. While we mentioned a traditional three-tier hosting environment, each hosting topology is different and unique to each customer's specific environment.

2.2. **Content Migration** - Once a framework of the website is complete, the next step is to begin loading content into the content management system (CMS). Depending on whether or not the project relates to a new website or a redesign of an existing website, your approach to content migration may differ. For new website properties, content is typically developed in Microsoft Word documents. For redesign projects, there are multiple approaches to consider. For example, using SQL scripts or direct APIs, you can migrate content blocks directly from the previous installation.

3. The Quality Assurance Phase

Once implementation is complete, it is time to test. The testing phase of the methodology is intended to capture and resolve any issues, bugs or problems. If there is one overlooked aspect of website development, it is typically in the testing process. As you develop use cases, be expansive and remember to not only cover the tasks described in the business requirements, but also commonly used website functions, such as search and contact forms.

❖ There are two stages of testing:

1- **System Acceptance Testing -** Using internal resources, begin the system testing phase by documenting specific test cases created using the business and functional requirements obtained during the Discovery Phase.

2- **User Acceptance Testing -** The final phase of activity before deployment is to conduct user acceptance testing. With the previous testing phase, you used IT developers and QA staff to do the testing and issue resolution activities. In this phase, you will use actual end-users to validate the site experience.

4. Deployment phase

When the testing process is complete, training delivered and a final check of system functionality, you are ready to deploy.
Remember, once the website is launched, you still have ample opportunities to tweak, modify and enhance the content, layout and functionality of the website. Use this capability to measure your expected results and make changes as appropriate. No other medium has the ability to allow you to quickly change your mind and react to how your customers perceive and interact with your company.
It is also critically important to be detailed in the tracking and reporting of the KPI metrics as defined in the Discovery Phase of activities. If the initial investment in the website is based on demonstrable business impact, these KPIs help to prove that the expected result has been attained.
Always keep in mind, although many projects count success as the mere launching of a new website, real success is measured over time and in the newfound ability to react to the marketplace.

Time Schedule

Number	Tasks	Start	End	Duration
1	*Idea*	*10/10/2017*	*15/10/2017*	*5 days*
2	*Problem definition*	*16/10/2017*	*22/10/2017*	*7 days*
3	*Problem solution*	*23/10/2017*	*27/10/2017*	*5 days*
4	*Analysis phase*	*01/11/2017*	*15/12/2017*	*45 days*
5	*Design phase*	*15/01/2018*	*15/02/2018*	*1 month*
6	*Development phase*	*20/02/2018*	*20/04/2018*	*2 months*
7	*Testing*	*01/05/2018*	*15/05/2018*	*2 weeks*
8	*Presentation*	*10/06/2018*	*15/06/2018*	*5 days*

End of chapter 1

Chapter 2

System design and construction

System design and construction

System design includes the evaluation of alternative solutions, preparation of detailed computer-based specifications that will fulfill the requirements specified during system analysis, and construction of system prototypes.

System design is the evaluation of alternative solutions and the specification of a detailed computer-based solution. It is also called **physical design**.

2.1 STRATEGIES FOR SYSTEM DESIGN

There are many popular strategies or techniques for performing system design. These techniques can be used in combination with one another.

- **Modern Structured Design**: a technique that focuses on processes.
- **Information Engineering (IE)**: a technique that focuses on data and strategic planning to produce application projects**.**
- **Prototyping:** a technique that is an iterative process involving a close working relationship between designers and users to produce a model of the new system.
- **Joint Application Development (JAD):** techniques that emphasizes participative development among system owners, users, designers, and builders. During JAD sessions for system design, the system designer takes on the role of the facilitator.
- **Rapid Application Development (RAD):** a technique that represents a merger of various structured techniques with prototyping and JAD to accelerate systems development.
- **Object-Oriented Design (OOD):** a new system strategy that follows up object-oriented analysis to refine object requirement definitions and to define new design specific objects.

2.2 APPLICATION ARCHITECTURE AND PROCESS DESIGN

Information application architecture and process design include techniques for distributing data, processes, and interfaces to network locations in a distributed computing environment.

An **application architecture** defines the technologies to be used by (and to build) one, more, or all information systems in terms of its data, process, interfaces, and network components.

- The prevailing computing model is currently **client / server** wherein networks of clients, single-user computers, are connected to and interoperates with servers, multiple-user computers that share their services. This is also called **distributed computing**.

- Centralized computing, distributed presentation, distributed data, distributed data and logic, and Internet/intranet computing are flavors of **client/server computing**.

- Client/server computing can be based on different network topologies including bus, ring, star, and hierarchical networks.

- Data storage is typically implemented using distributed relational database technology that either partitions data to different servers or replicates data on multiple servers.
- Processes are implemented using highly integrated tool kits called software development environments.
- Physical data flows diagrams model an information system's application architectures and processes. Because they show the planned implementation of all processes, data stores, and data flows, they serve as a general design or blueprint for subsequent detailed design, prototyping, and construction.
- System flowcharts are a lesser used diagram to show the same implementation features as a physical data flow diagrams. Although they are rarely drawn today, many older, legacy systems use them for documentation.

2.3 DATABASE DESIGN

Data storage is a critical component of most information systems. Some people consider it to be the critical component. The data captured by an information system is stored in files and databases. A file is a collection of records. A database is a collection of interrelated files. Many legacy systems were built with files technology because files were built for specific applications their design was optimized for those applications. This close relationship between the files and their applications made it difficult to restructure the files to meet future requirements. And because many applications use the same data, it is not uncommon to find redundant files with data values that do not always match.

- ***Database design*** is the process of translating logical data models into physical database schemas. The smallest unit of meaningful data that can be stored is called a field. There are four types of fields:

1) **A *primary* key** is a field that uniquely identifies one and only one record in a file or table.
2) A **secondary key** is a field that may either uniquely identify one and only one record in a file or table or identify a set of records with some common, meaningful characteristic.
3) **A foreign key** is a field that points to a related record in a different table.

Fields are organized into records, and similar *records* are organized into files or tables. A *database* is a collection of tables (files) with logical pointers that relate records in one table to records in a different table.
Database architecture is built around a database management system (DBMS) that provides the technology to define the database structure and then to create, read, update, and delete records in the tables that make up that structure. A DBMS provides a data language to accomplish this.

- **That language provides at least two components**:

a) **A *data definition language*** to create and maintain the database structure and rules.

b**) A *data manipulation language*** to create, read, use, update, and delete records in the database.

Today, relational database management systems are used to support the development and reengineering of the overwhelming number of information systems. Relational databases store data in a collection of tables that are related via foreign keys.

a) The data definition and manipulation languages of most relational DBMSs are consolidated into a standard language known as SQL.

b) High-end relational database management systems support triggers and stored procedures, programs that are stored with the tables and callable from other SQL-based programs.

- **Data analysis and normalization** are techniques for removing impurities from a data model as a preface to designing the database. These impurities can make a database unreliable, inflexible, and non-scalable.

Normalization involves checking each entity (table) for first, second, and third normal form impurities.

a) An entity is a first normal form if it contains no repeating attributes.

b) An entity is a second normal form if it contains no partial dependencies.

c) An entity is a third normal form if it contains no derived attributes or no transitive dependencies.

- **Database integrity** should be checked and, if necessary, improved to ensure that the business and its users can trust the stored data.

a) Key integrity ensures that every record will have a unique, non-null primary key value.

b) Domain integrity ensures that appropriate fields will store only legitimate values from the set of all possible values.
c) Referential integrity ensures that no foreign key value points to nonexistent primary key value.
A deletion rule should be specified for every relationship with another table.

2.4 INPUT DESIGN AND PROTOTYPING

Several concepts are important to input design. One of the first things you must learn is the difference between data capture and data input. Alternative input media and methods must also be understood before designing the inputs. And because accurate data input is so critical to successful processing, file maintenance, and output, you should also learn about human factors and internal controls for input design.

System designers must determine when and how to capture the data. The designer must understand the difference between the following:

- **Data capture** is the identification of new data to be input.
- ***A source document*** is a paper form used to record data that will eventually be input to a computer.
- **Data entry** is the process of translating the source document into a machine-readable format. That format may be a magnetic disk, an optical-mark form, a magnetic tape, or a floppy diskette, to name a few.
- **Data input** is the actual entry of data in a machine-readable format into the computer.

The systems analyst usually selects the method and medium for all inputs. Input methods can be broadly classified as either batch or on-line.

- **Batch input** is the oldest and most traditional input method. Source documents or forms are collected and then periodically forwarded to data entry operators, who key the data using a data entry device that translates the data into a machine-readable format.
- **On-line input** is the capture of data at its point of origin in the business and the direct inputting of that data to computer, preferably as soon as possible after the data originates.
- **Remote batch** offers on-line advantages for data that is best processed in batches. The data is input on-line with on-line editing. Remote batch is also called *deferred batch or deferred processing.*

Most new applications being developed today consist of screens having a “graphical” looking appearance. This type of appearance is referred to as a graphical user interface (GUI).

Inputs should be as simple as possible and designed to reduce the possibility of incorrect data being entered. Input controls ensure that the data input to the computer is accurate and that the system is protected against accidental and intentional errors and abuse, including fraud.

When designing input screens for an application that will contain a GUI appearance, the designer must be careful to select the proper control serves a specific purpose, has certain advantages and disadvantages, and should be used according to guidelines. Some of the most commonly used screen-based controls for inputting data include: text box, radio button, check box, list box, drop-down list, combination box, and spin box.

2.5 OUTPUT DESIGN AND PROTOTYPING

Several concepts are important to output design. One of the first things you must learn is difference between **external** and **internal outputs.** Some external outputs are designed as turnaround outputs that leave and re-enter the system later. Such outputs are usually designed for printing on specially designed manufactured forms.

Some outputs are generated to fulfill management information system requirements. These outputs can be classified as one of the following three types of reports:

- **Detailed reports:** which present information with little or no filtering or restrictions.
- **Summary reports:** which categorize information for managers who don't want to wade through details**.**
- **Exception reports:** which filter data before they are presented to the manager as information. Such reports usually report exceptions to some condition or standard.

The system designer usually selects the media and format for all outputs.

- **Medium** is what the output information is recorded on, such as paper or video display.
- **Format** is the way the information is displayed on a medium, for instance, columns of numbers or graph.

Common alternative media include the following:

- **Microfilm** is a roll of photographic film that is used to record information in a reduced size**.**
- **Microfiche** is a single sheet of film that is capable of storing many pages of reduced output.
- **Video** is the fastest-growing medium for computer outputs; it is the on-line display of information on a visual display device, such as a CRT terminal or microcomputer display.

There are several different formats you can choose for communicating information on a medium:

- **Tabular output** in which the format presents information as columns.

- **Zoned output** placed text and numbers into designed areas of a form or screen.
- **Graphic output** is the use of a graph or chart to convey information**.**
- **Narrative output** format uses sentences and paragraphs to replace standard text, numbers, and pictures.

There are many system user issues that apply to output design. The following general principles are important for output design:

- Computer outputs should be simple to read and interpret.
- The timing of computer outputs is important.
- The distribution of computer outputs must be sufficient to assist all relevant system users.
- The computer outputs must be acceptable to system users who will receive them.

The design and prototyping of computer outputs involve the following steps:

- Identify system outputs.
- Select output medium and format.
- Prototype the output for systems users**.**

2.6 USER INTERFACE DESIGN AND PROTOTYPING

User interface design is the specification of a conversation between the system user and the computer. This conversation generally results in either input or output possibly both. There are several types of user interface styles, including menu selection, instruction sets, question-answer dialogues, and direct manipulation. Traditionally these styles were viewed as alternatives. However, with recent movements toward designing systems with graphical user interfaces, a blending of all styles can be found.

- **Menu selection** is strategy of dialogue design that presents a list of alternatives or options to the user. In today's GUI designs, menu selection is commonly represented using **menu bars, pull-down menus, cascading menus, pop-up menus, and iconic menus.**
- **Instruction sets** is a strategy where the application is designed using a dialogue syntax that the user must learn. There are three types of syntax: structured English, mnemonic syntax, and natural language.
- **Question-answer dialogue strategy** is a style that was primarily used to supplement either menu-driven or syntax-driven dialogues.
- The newest and most popular of the four user interface styles allow **direct manipulation** of graphical objects appearing a screen. Essentially, this user interface style focuses on using icons, small graphic images, to suggest functions to the user.

Nowhere are human factors as important as they are in user interface design. Therefore, user interface design must be completed with human engineering consideration. System users can be broadly classified as either dedicated or casual.

- **A dedicated system user** is one who will spend considerable time using specific programs. This user is likely to become comfortable and familiar with the terminal or PC's operation.
- **The casual system user** may use a specific program only occasionally. This user may never become truly comfortable with the terminal or the program.

Given the type of user for a system, there are a number of important human engineering factors that should be incorporated into the design:

- The system user should always be aware of what to do next.
- The screen should always be formatted.
- Messages, instructions, or information should be displayed long enough to allow the system user to read them.
- Display attributes should be used sparingly.
- Default values for fields and answers to be entered by the user should be specified.
- The errors users might make should be anticipated.

With respect to the tone of dialogue, the following guidelines are offered:

- Use simple, grammatically correct sentences.
- Don't be funny or cute!
- Don't be condescending; that is, don't insult the intelligence of the system user.

With respect to the terminology used in the dialogue, the following suggestions may prove helpful:

- Don't use the computer jargon.
- Avoid most abbreviations.
- Use simple terms.
- Be consistent in your use of terminology.
- Instructions should be carefully phrased, and appropriate action verbs should be used.

The design of user interface can be enhanced or restricted by the available features of your terminal display or monitor/keyboard. The steps involved in designing and prototyping a user interface includes charting the dialogue, prototyping the dialogue, and user interface, and obtaining user feedback.

2.7 SOFTWARE DESIGN

Software design consists of two components:

a) **Modular design** is the decomposition of a program into modules. A module is a group of executable instructions with a single point of entry.
b) **Packaging** is the assembly of data, process, interface, and geography design specifications for each module.

- **Structured design** is a popular strategy for determining an optimal modular design for programs. This technique deals with the size and complexity of a program by breaking up a program into a hierarchy of modules and results in a computer program that is easier to implement and maintain.

The primary tool used in structured design is the structure chart that is used to graphically depict a modular design of a program. Structured design provides two strategies for developing structure charts:

a) **Transform analysis** is an examination of DFD to divide the processes into those that perform input and editing, those that do processing or data transformation, and those that do output.
b) **Transaction analysis** is an examination of DFD to identify processes that represent transaction centers.

Two measures of the quality of a structure chart are *coupling* and *cohesion*.

a) **Coupling** refers to the level of dependency that exists between modules.
b**) Cohesion** refers to the degree to which module's instructions are functionally related.

There are five types or levels coupling. In order of most desirable to least desirable, they are data, stamp, control, common, and content coupling.

There are seven types or levels of cohesion. In order of most desirable to least desirable, they are functional, sequential, communicational, procedural, temporal, logical, and coincidental cohesion.

Packaging results in a technical design statement that will be used during systems implementation.

2.8 OBJECT-ORIENTED DESIGN

Using object oriented techniques for designing a system is referred to as object oriented design. *Object oriented design* is concerned with identifying and classifying three object types: *interface*, *entity* and *control*. Interface and control object types are objects that are introduced as a result of implementation decisions that were made during systems design. Entity objects are identified during systems analysis and usually correspond to items in real life and contain information, known as attributes that describes the different instances of the entity. Interface objects are objects that are introduced to represent a means through which the users will interface with the system. The responsibility of the interface object is twofold: (1) it translates the user input into information that the system can understand and use to process the business event, and (2) it takes data pertaining to a business event and translates the data for

appropriate presentation to the user. Control objects are those that hold application or business rule logic controller objects serve as the traffic cop containing the application logic or business rules of the event for managing or directing the interaction between the objects.

Object responsibility is the obligation that an object has to provide a service when requested thus collaborating with other objects to satisfy the request if required object responsibility is closely related to the concept of objects being able to send and or respond to messages.

Object oriented design includes the following activities refining the use case model to reflect the implementation environment modeling object interactions and behavior that support the use case scenario, and updating the object model to reflect the implementation environment.

An ideal object model diagram is used to document objects and their interactions an ideal object model diagram includes symbols to represent actors interface control and entity objects and arrows that represent messages or communication between the objects.

A popular tool for documenting the behaviors and collaboration for an object is the class responsibility collaboration **(CRC)** card.
An interaction diagram is used to document the detailed interaction between objects.

2.9 SYSTEMS IMPLEMENTATION

Systems implementation is the construction of the new system and the delivery of that system into production (meaning day to day operation) unfortunately, systems development is a common synonym.

The purpose of the construction phase of systems implementation is twofold to build **and test a functional system that fulfills business and design requirements and to** implement the interface between the new system and existing production systems.

The construction phase consists of four activities build and test networks build and write and test new programs.

There are three levels of testing performed on new programs:

a) **Stub testing** is the test performed on individual modules, whether they are main program, subroutine, sub program, block, or paragraph.
b**) Unit or program testing** is a test whereby all the modules that have been coded and stub tested are tested as an integrated unit.
c) **System testing** ensures that application programs written in isolation work properly when they are integrated into the total system.

The purpose of the delivery phase is to smoothly convert form the old system to the new system. The delivery phase of system implementation consists of the following activities: conducting a system test, preparing a systems conversion plan, installing databases, training system users, and converting form the old system to the new system.

There are several commonly used strategies for converting from an existing to a new production information system, including:

- **Abrupt cut over** on a specific date the old system is terminated and the new system is placed into operation.
- **Parallel conversion** both the old and new system is operated for some time period to ensure that all major problems in the new system are solved before the old system is discarded.
- **Location conversion** when the same system will be used in numerous geographical locations, it is usually converted at one location and following approval farmed to the other sites.
- **Staged conversion** each successive version of the new system is converted as it is developed. Each version may be converted using the abrupt, parallel, or location strategies.

The system acceptance test is the final opportunity for end users, management, and information system operation management to accept or reject the system. A system acceptance test is a final system test performed by end users using real data over an extended period. It is an extensive test that addresses three levels of acceptance testing: verification testing, validation testing.

- **Verification testing** runs the system in a simulated environment using simulated data
- **Validation testing** runs the system in a live environment using real data this is sometimes called beta testing.

2.10 SYSTEM SUPPORT

Systems support: is the ongoing maintenance of a system after it has been placed into operation. This includes program maintenance and system improvements.

System support involves solving different type of problems with the system there are several different types of system support system maintenance, system recovery end user assistance system enhancement and reengineering.

Regardless of how well designed, constructed, and tested a system or application may be errors or bugs will inevitably occur, the corrective action that must be taken is called systems maintenance.

From time to time a system failure is inevitable. It generally results in an aborted or hung program (also called an ABEND or crash) and possible loss of data the system analyst often fixes the system or acts as intermediary between the users and those who can fix the system; this is referred to recovery.

Another relatively routine ongoing activity of system support is routine end user assistance. No matter how well users have been trained or how well documentation has been written, users will require additional assistance. The system analyst is generally on call to assist users with the day to day use of specific application. In mission critical application the analyst must be on call day and night.

Most adaptive maintenance is in response to new business problems, new information requirements, or new ideas for enhancement. It is reactionary in nature-fix it when it breaks or when users make a request. We call this **system enhancement**. The objective of system enhancement is to modify or expand the application system in response to constantly changing requirements. Another type of reactionary maintenance deals

with changing technology. Information system staffs have become increasingly reluctant to wait until systems break. Instead, they choose to analyze their program libraries to determine which applications and programs are costing the most to maintain. These systems might be adapted to reduce the costs of maintenance. The preceding examples of adaptive maintenance are classified as *reengineering*.

Thanks End of chapter2

Chapter 3

System Design and Database Models

3.1 **Data Flow Diagram:**

>>>Context Level DFD:

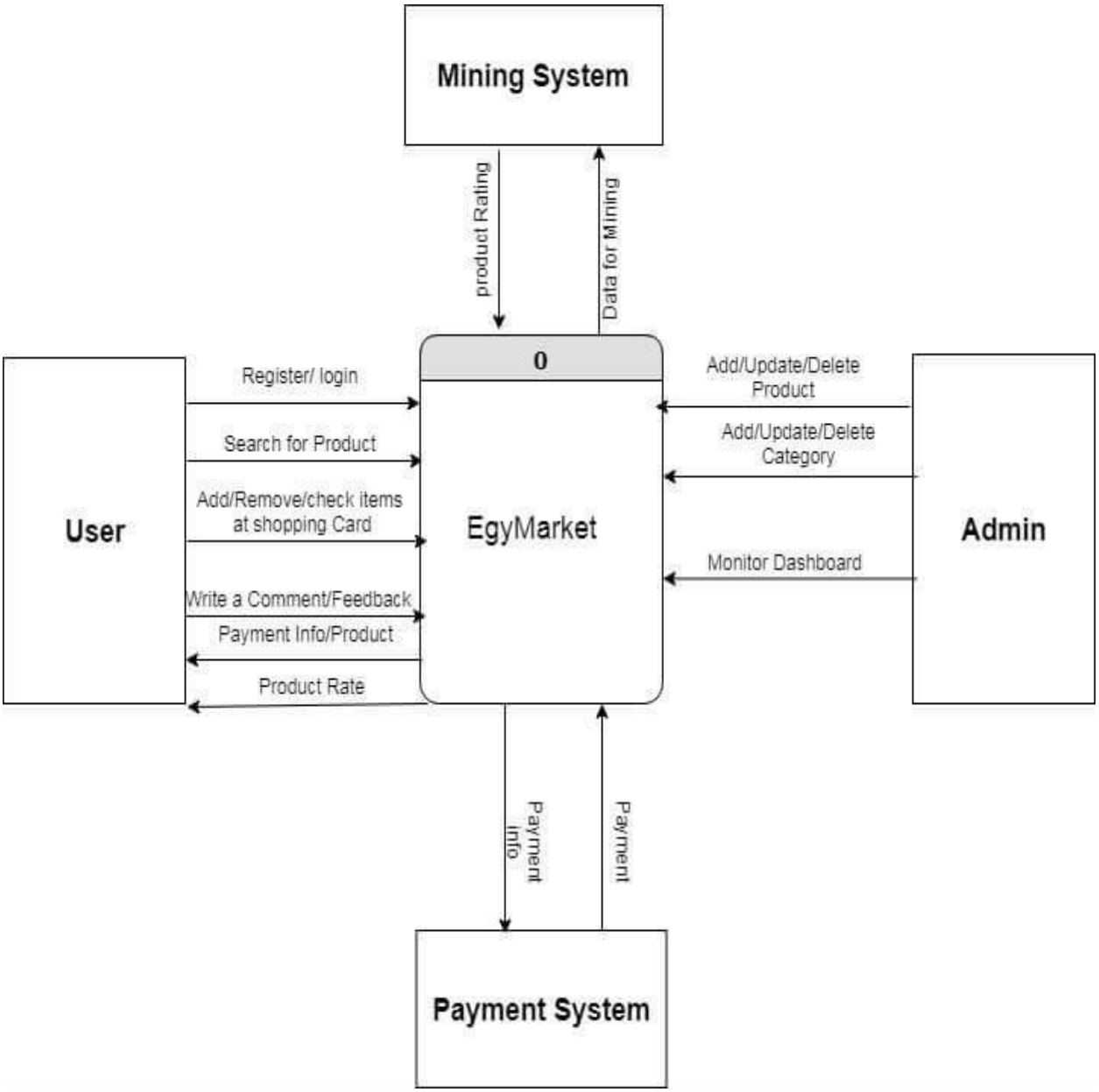


Figure 3.1 context level DFD

>>> Level (1) one DFD:

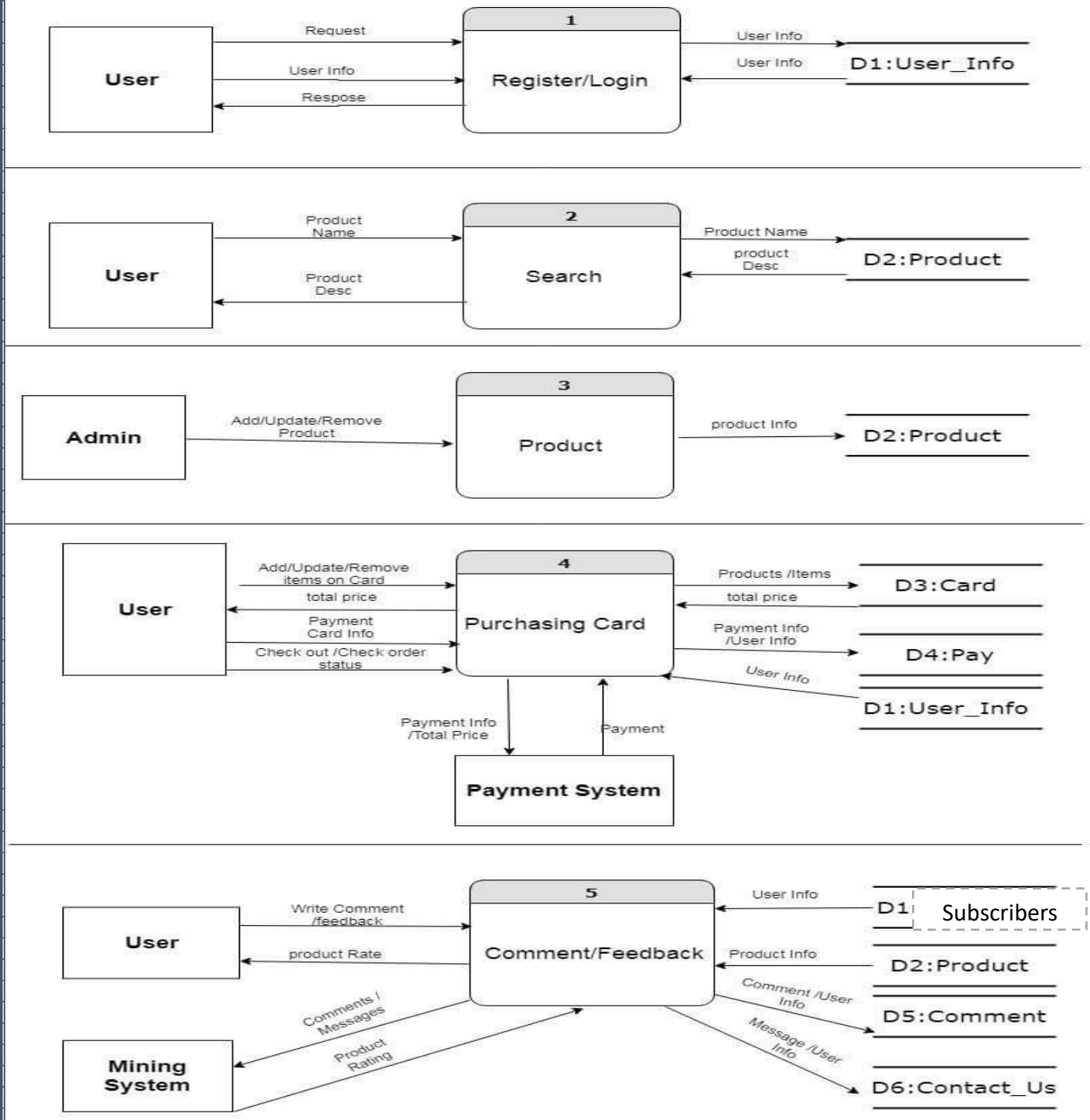


Figure 3.2 level (1) one DFD

3.2 Use Case Diagram:

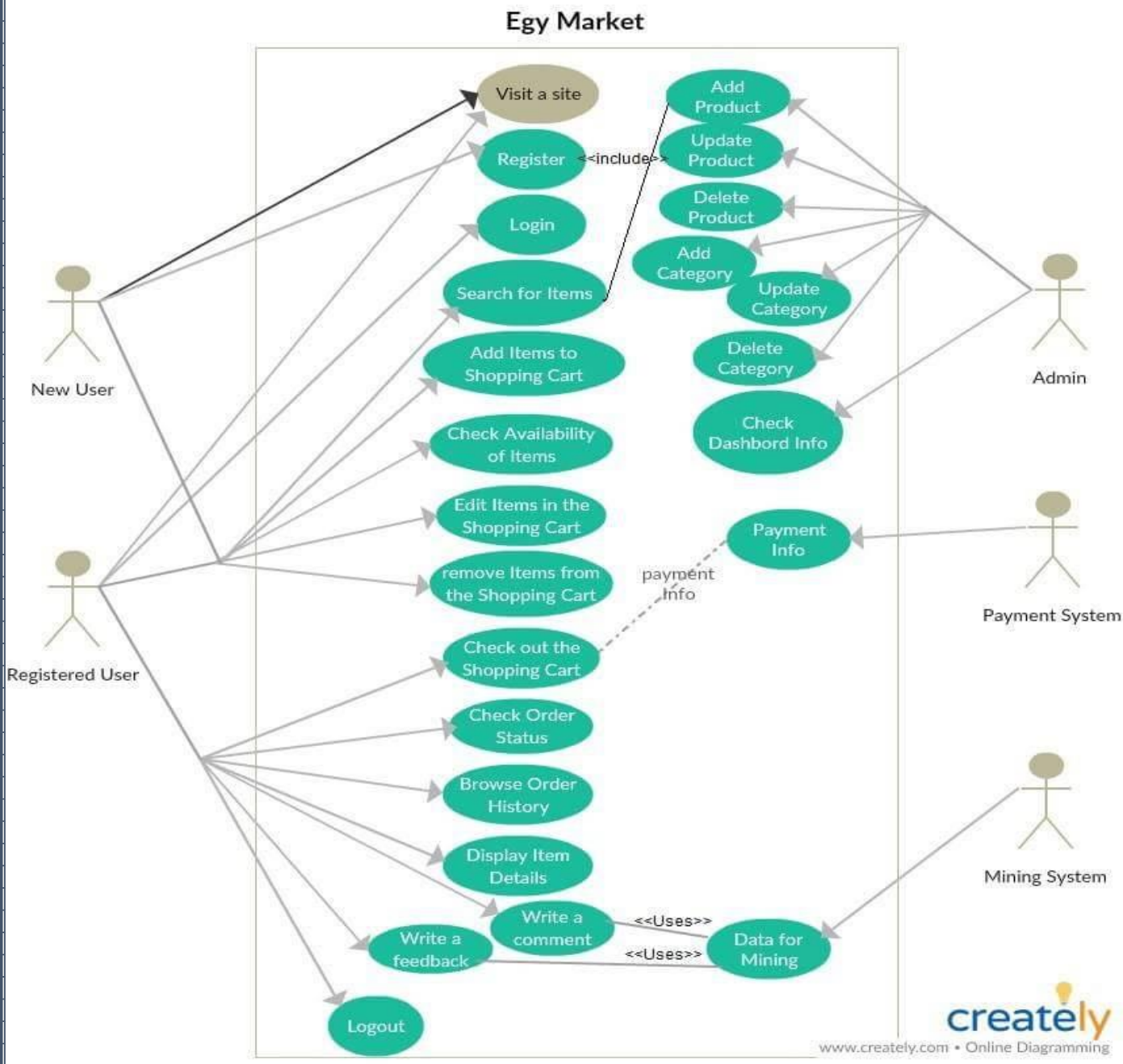


Figure 3.3 Use Case Diagram

A-Functional Requirements:

1-Search and Browse:

- The system will allow customers to search for categories or products.
- The system will allow customers to browse products and read Specifications for each one.
- The system will allow customers to add products to purchasing card.
- The system will allow customers to remove products from purchasing card.
- The system will allow customers to edit products in purchasing card.
- The system will allow customers to check products availability.

2-Purchase:

- The system will enable the customer to create a customer account to save his information.
- The system will enable the customer to check out the shopping card.
- The system will enable the customer to check order status.
- The system will enable the customer to pay online or cash.
- Sentiment Analysis:
- The system will analyze the customer's comments on products and produce the product rate.

3-Sentiment Analysis:

- The system will analyze the customer's comments on products and produce the product rate.

4-Business Intelligence Analytics:

- Use transaction data for making reports and dashboards.

5-Administration:

- The system will allow admin to add, remove, or update products or categories.
- The system will allow admin to monitor changes by a dashboard.

B- Nonfunctional Requirements:

1-Operational:

- The system will run on any Web browser.
- The system will enable the customer to write comments on products.
- The system will display the most visited products.

2-Performance:

- Speed: All transaction should take a few time.
- Capacity: Automated system should perform large number of transaction and Perform operations quickly.

- Reliability: Automated system should be available for use 24 hours per day and 365 days per year.

3-Security:

- Customer information will be secured.
- Payment information will be secured.
- Only software manager can do the operational tasks in the system.
- All users have different account and password.

4-Cultural and political:

- No special cultural and political requirements are expected.

3.3 ER Diagram:

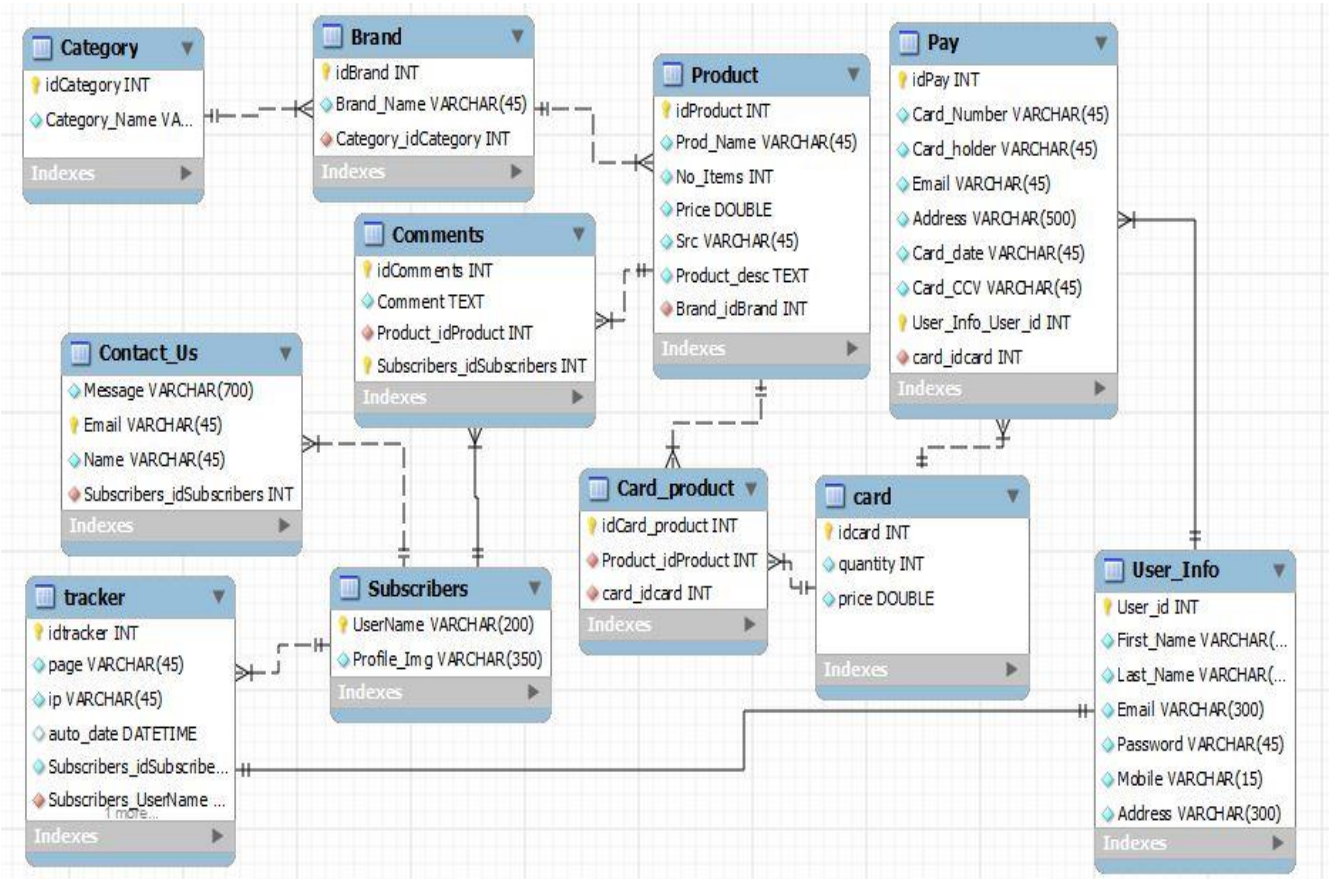


Figure 3.4 Entity-Relationship Diagram

3.4 Business Model:

EgyMarket Business Model

Company Name:
EgyMarket

Key Partners
Who are our Key Partners?
Who are our Key Suppliers?
Which Key Resources are we acquiring from partners?
Which Key Activities do partners perform?

- **Content Providers** (website owner who need their data to be analyzed).
- **Data sets** about A specific product or Service.

Key Activities
What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?

- App development
- Improve text mining algorithm.
- Use clients actions.

Key Resources
What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

- Product or service to be rated.
- Text about this product or service to be analyzed.
- Website need our service.

Value Proposition
What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?

- Client can browse EgyMarket searching for a product to buy .
- Give analyzed reports to other websites owners .
- Quick service.

Customer Relationships
What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

- Cost are stable not overrated
- Services are better as product is improved.

Channels
Through which Channels do our Customer Segments want to be reached?
How are we reaching them now? How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

- online sales
- EgyMarket (website)

Customer Segments
For whom are we creating value?
Who are our most important customers?

- Systems owners.
- clients on EgyMarket

Cost Structure
What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?

- Web hosting costs.
- Marketing & Sales.
- Website Development .

Revenue Streams
For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?

- Contact with systems owners who need our services.
- Sold products by EgyMarket.
- Traffic happened on EgyMarket.

Figure 3.5 Business Model.

3.5 List of Entities:
3.6 Entity details:

>> ***Preparation to the Physical Design:***

This step was including Data Description and Validation Rules, We performed this step by fulfilling forms of file designing that through fulfilling the following data:

1. **The name of the file (in English language).**
2. **Application name.**
3. **The field name (English).**
4. **Determination of symbolic data.**
5. **Determination of field lengths.**
6. **Determination of field types.**
7. **Description of validation rules.**
8. **Items entity.**

Entity- Category:

Server: 127.0.0.1 » Database: market » Table: category

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**category_id**	int(50)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**category_name**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Brand:

Server: 127.0.0.1 » Database: market » Table: brand

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**brand_id**	int(50)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**brand_name**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity: Product:

Server: 127.0.0.1 » Database: market » Table: product

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**product_id**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**product_name**	varchar(200)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**no_items**	int(50)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**price**	int(50)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	**src**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	**category_name**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	**product_desc**	text	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
8	**product_brand**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Pay:

Server: 127.0.0.1 » Database: market » Table: pay

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**card_number**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**card_holder**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**email**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**address**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	**card_date**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	**card_ccv**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	**product_id**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Comments:

Server: 127.0.0.1 » Database: market » Table: comment

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**comment_id**	int(11)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**comment**	text	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**userId**	int(11)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**product_id**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Subscribers:

Server: 127.0.0.1 » Database: market » Table: subscribers

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**userId**	int(11)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**userName**	varchar(255)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**profile_img**	varchar(255)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**regstatus**	int(11)			No	0		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Card:

Server: 127.0.0.1 » Database: market » Table: cart2

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**id**	int(50)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**product_id**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**qty**	int(50)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**user_id**	varchar(100)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	**ip_add**	varchar(100)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Contact Us:

Server: 127.0.0.1 » Database: market » Table: contactt

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**id**	int(11)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**email**	varchar(100)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**name**	varchar(100)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**message**	text	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity: User Info

Server: 127.0.0.1 » Database: market » Table: user_info

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**user_id**	int(10)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**first_name**	varchar(100)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**last_name**	varchar(100)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	**email**	varchar(300)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	**password**	varchar(300)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	**mobile**	varchar(10)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	**address1**	varchar(300)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
8	**address2**	varchar(11)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Admin:

Server: 127.0.0.1 » Database: market » Table: admin

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**user_name**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**password**	varchar(50)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Entity- Tracker:

Server: 127.0.0.1 » Database: market » Table: tracker

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**page**	varchar(64)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	**IP**	varchar(15)	utf8_general_ci		No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	**date_auto**	int(50)		UNSIGNED	No	*None*		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Thanks End of chapter 3

Chapter 4

System development (Implementation phase)

User Side:

- ⦿ **Header & slider**
- ⦿ **Sign up & log in**
- ⦿ **Products Gallery**
- ⦿ **User CART**
- ⦿ **Purchase process**
- ⦿ **Comment (Give review about product)**
- ⦿ **footer**

Admin side (Dashboard):

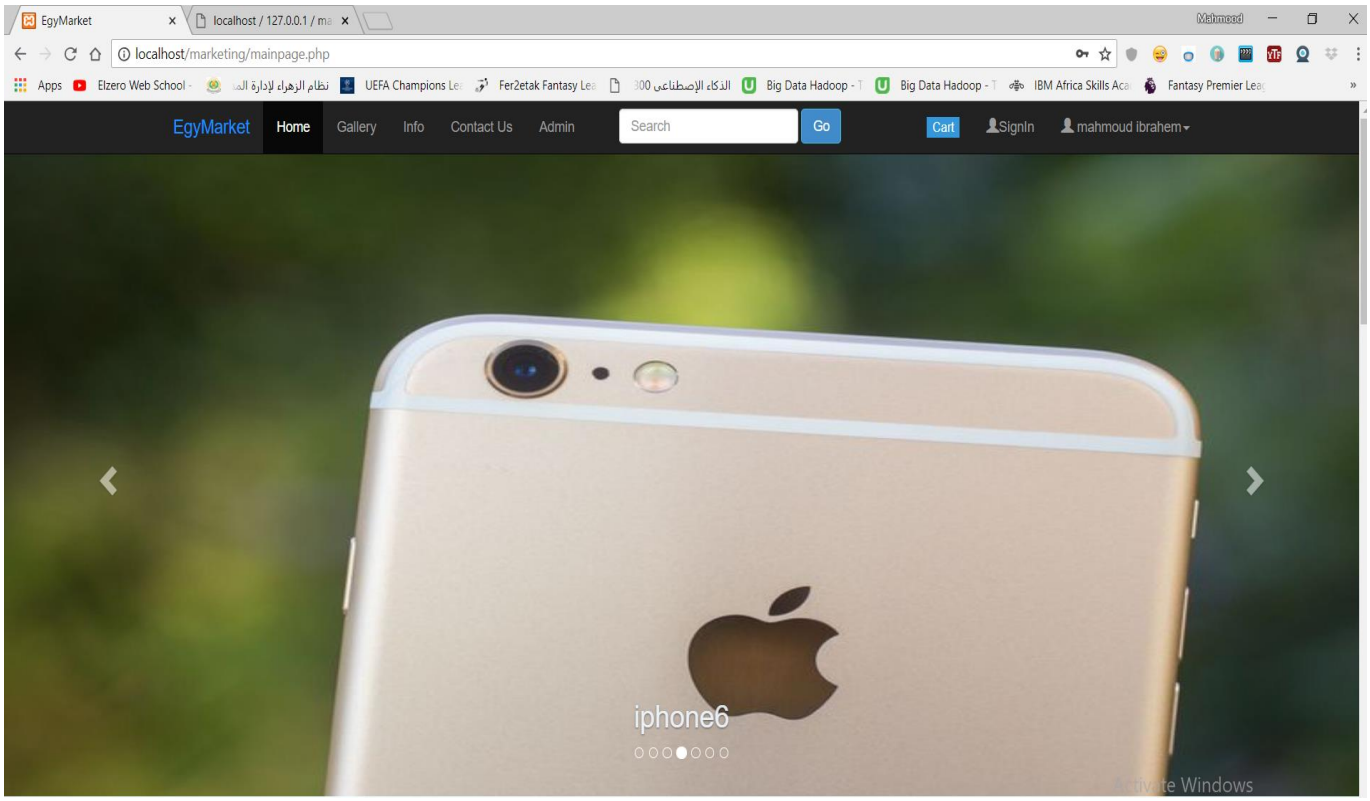
- ⦿ **Add new products**
- ⦿ **Update & Delete products from gallery.**
- ⦿ **Log in as admin**
- ⦿ **Add Category**
- ⦿ **Analyze comments**

Project screen flow (Result)

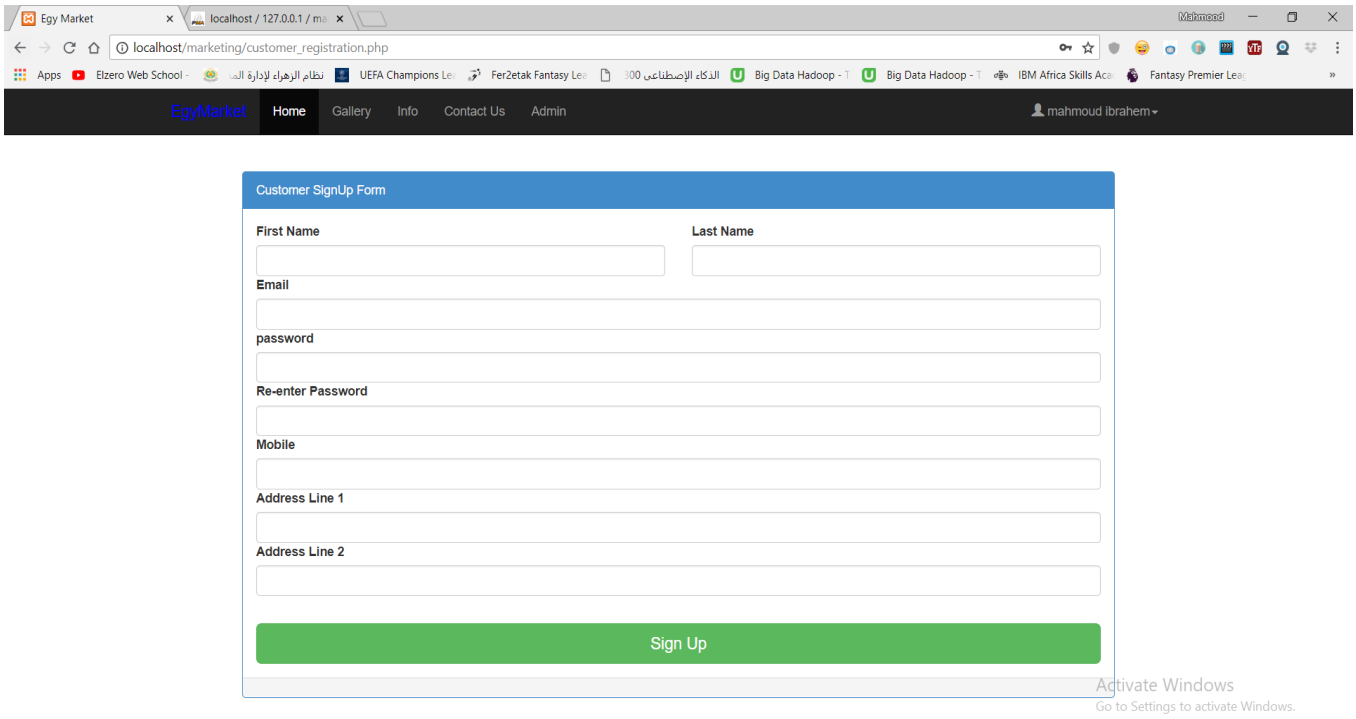
4.1 Introduction:

The project screen flow indicates the final result or the final shape of the project. In this chapter we will indicate by screen the followed steps for the users for using it, and also screen indicates the approach by which the user will navigate and use it.

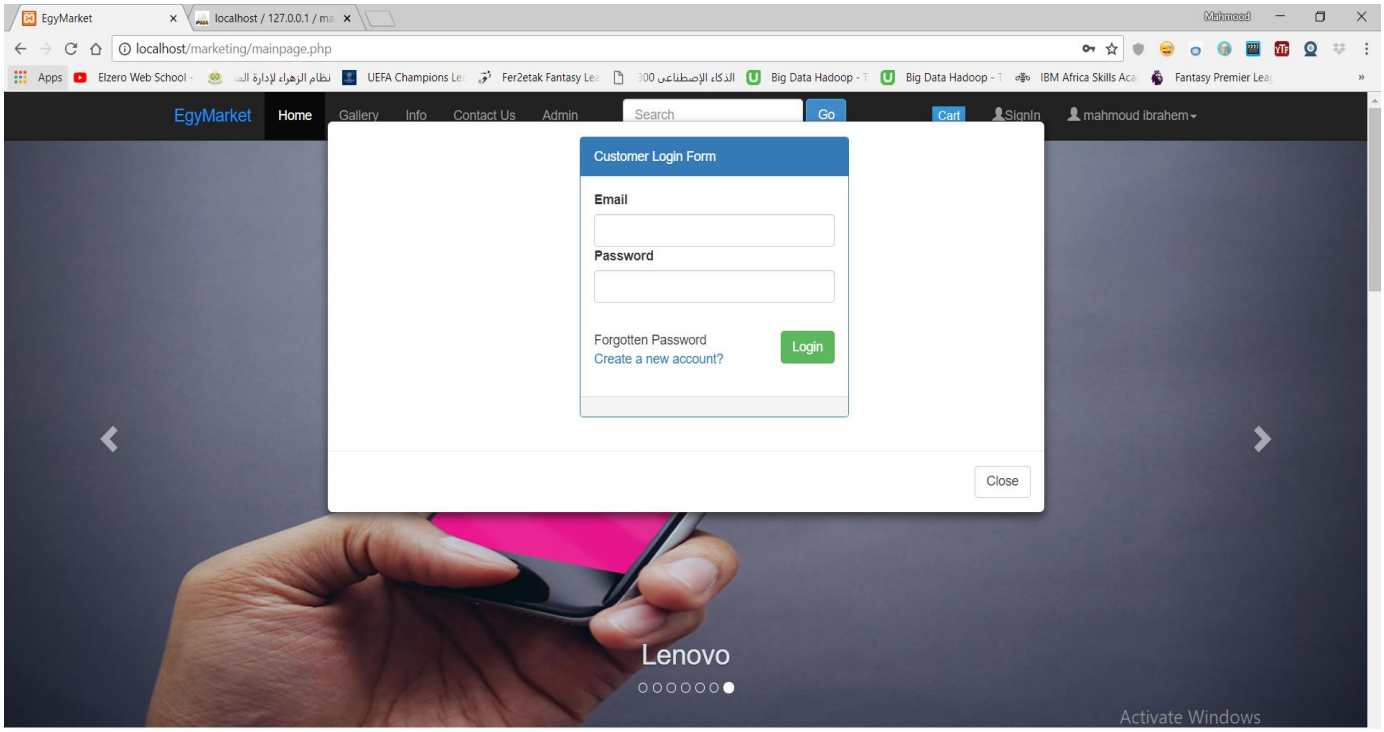
4.2 Home Page:



4.3 Customer Sign Up Form:

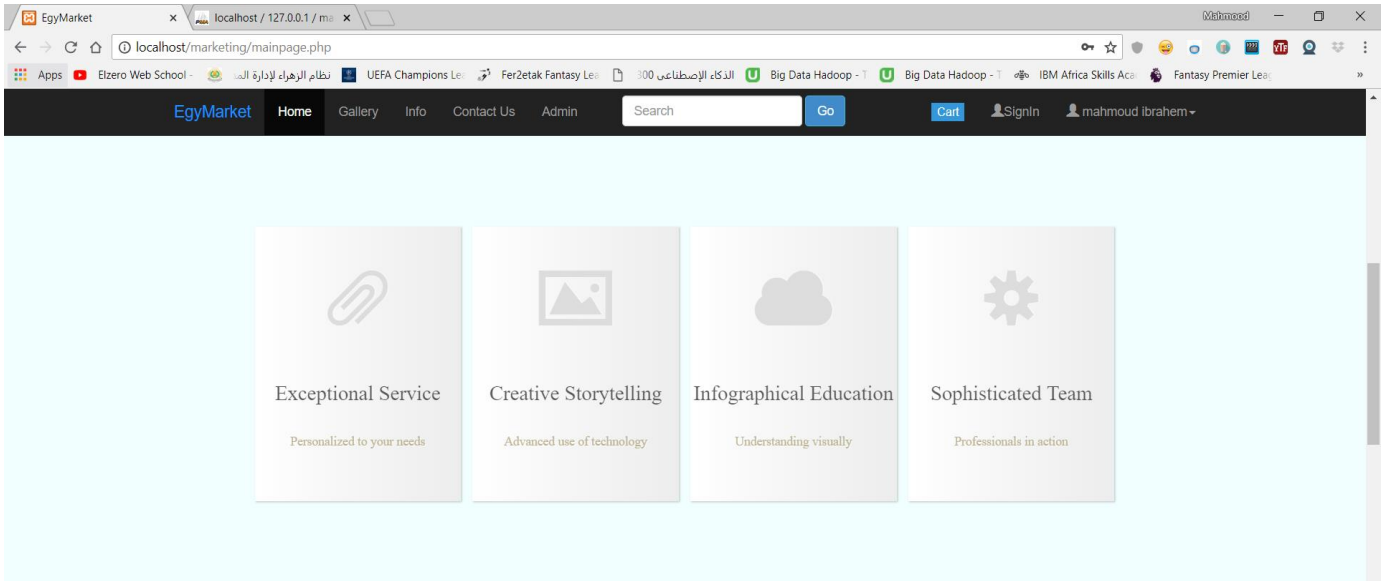


4.3 Customer Login Form:

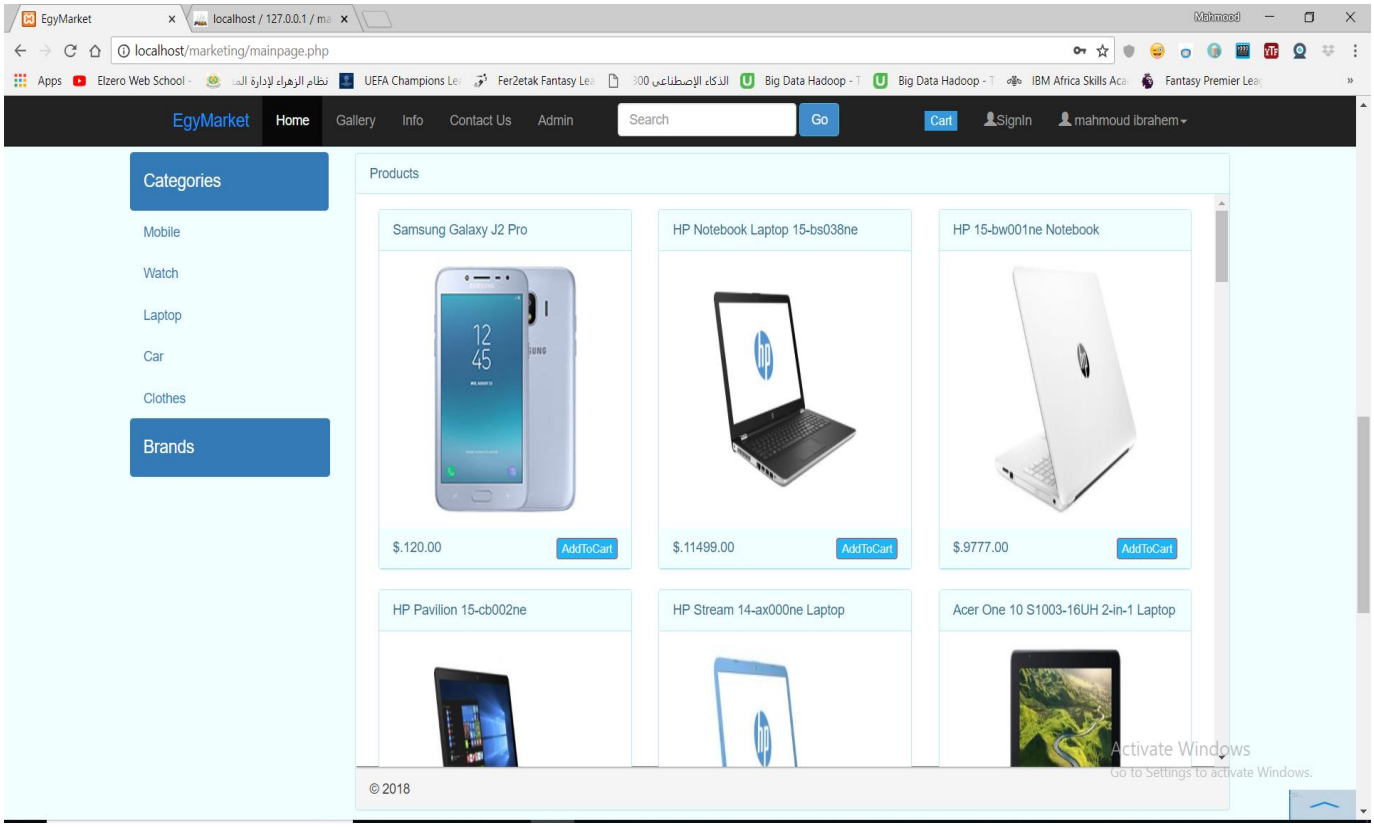


From this form users logged in to the system. They entered user name and password. If he entered wrong username or password an error message will appear “invalid username or password”.

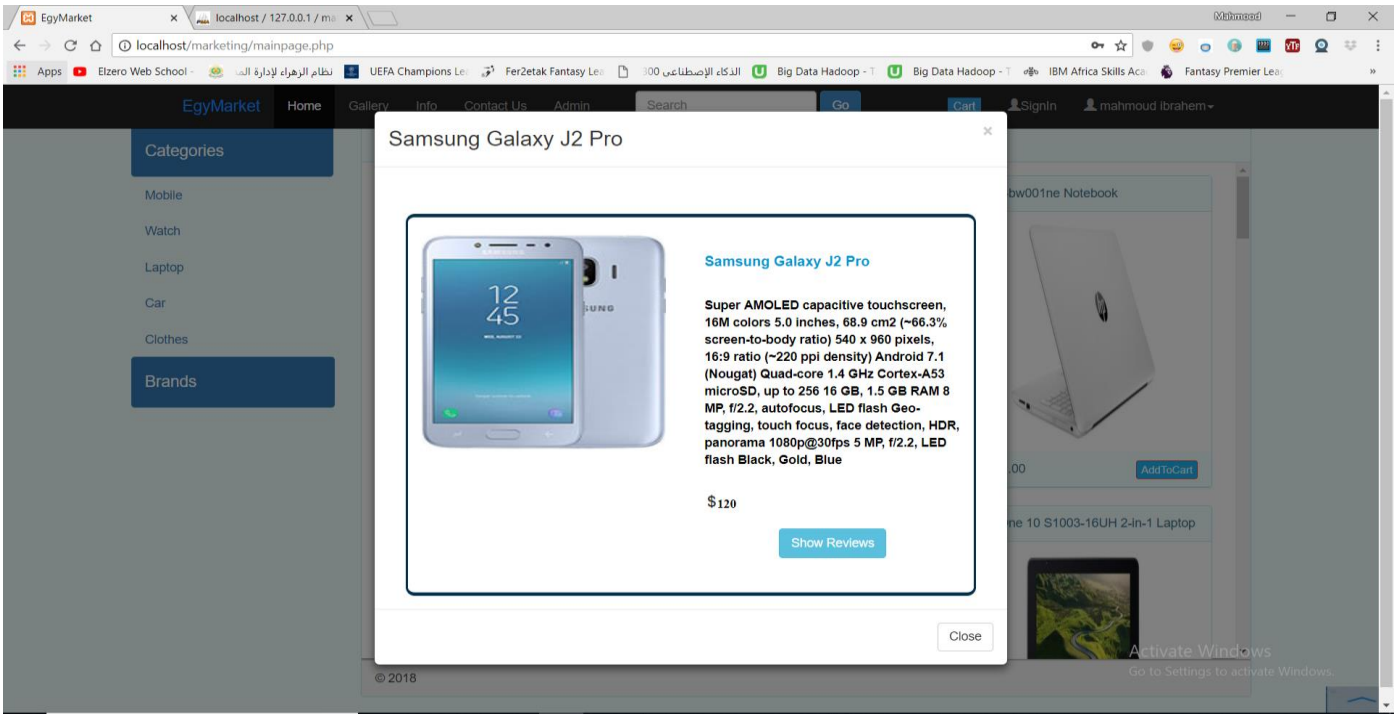
4.4 Info



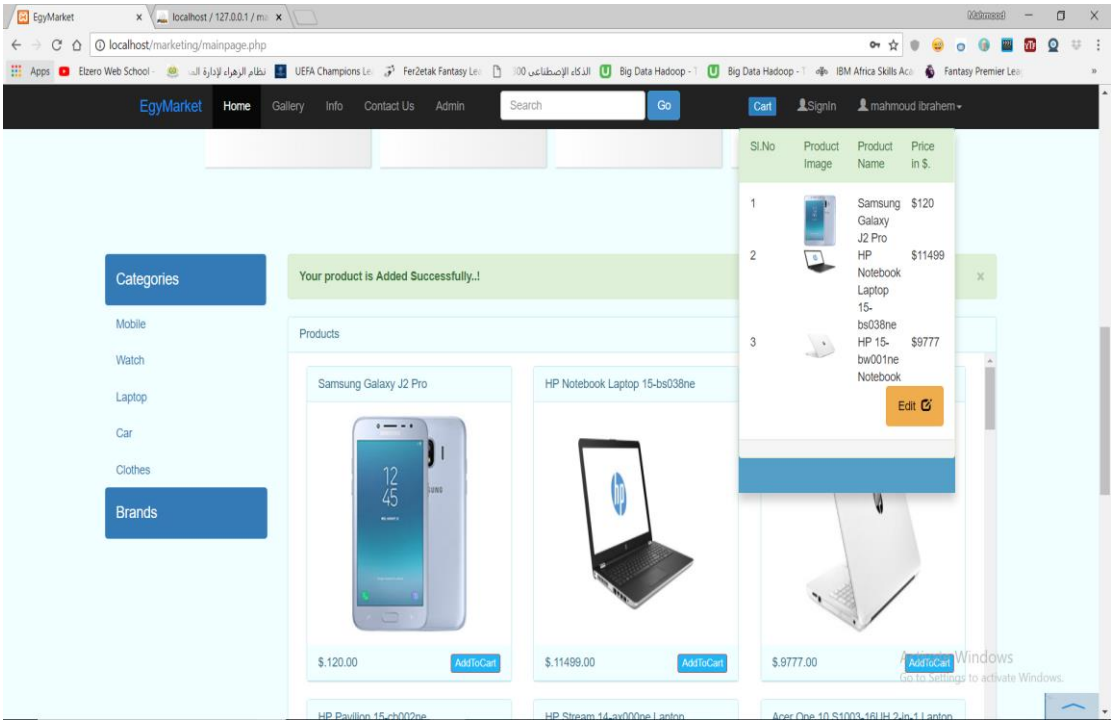
4.4 Gallery



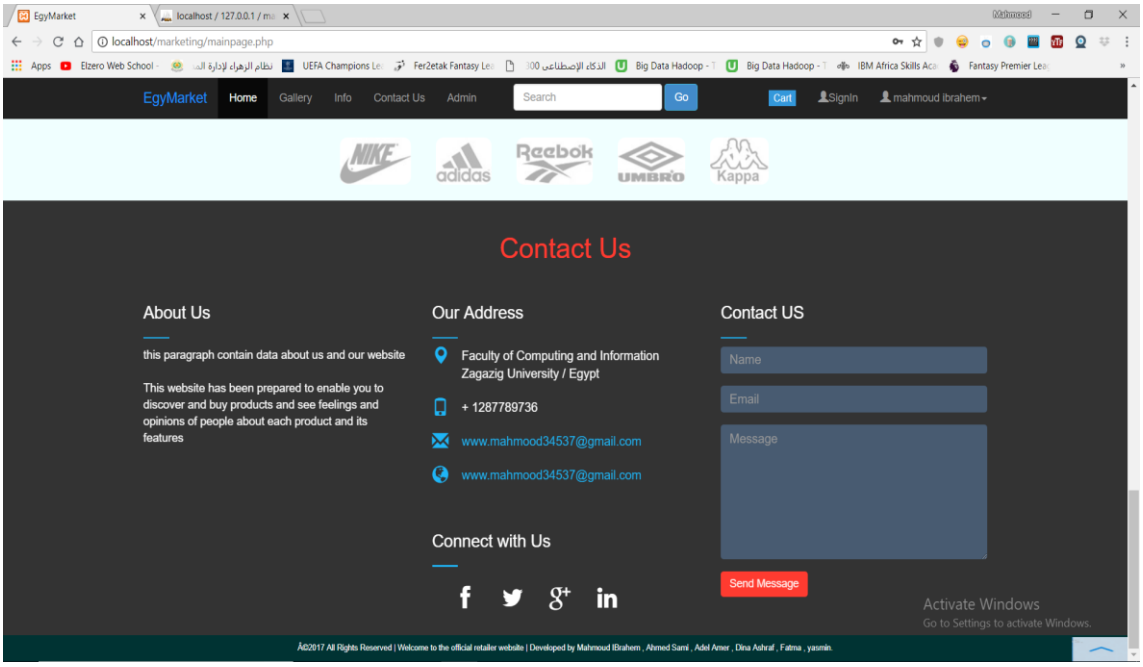
4.5 Product Description



4.6 Brand Selection



4.7 Footer



This page enables the user to do the following:

-User can navigate through the system and see the latest versions of products of different categories.

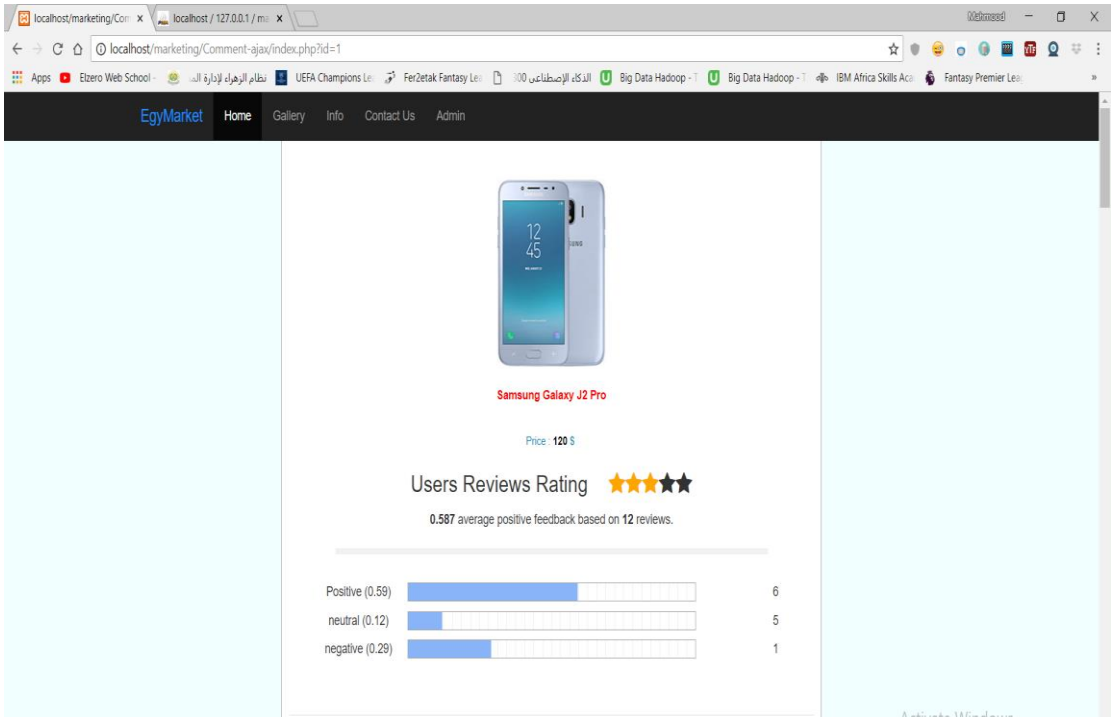
- User can view product images.

- User can view features of the product to know about the product.

- User will comment on the product.

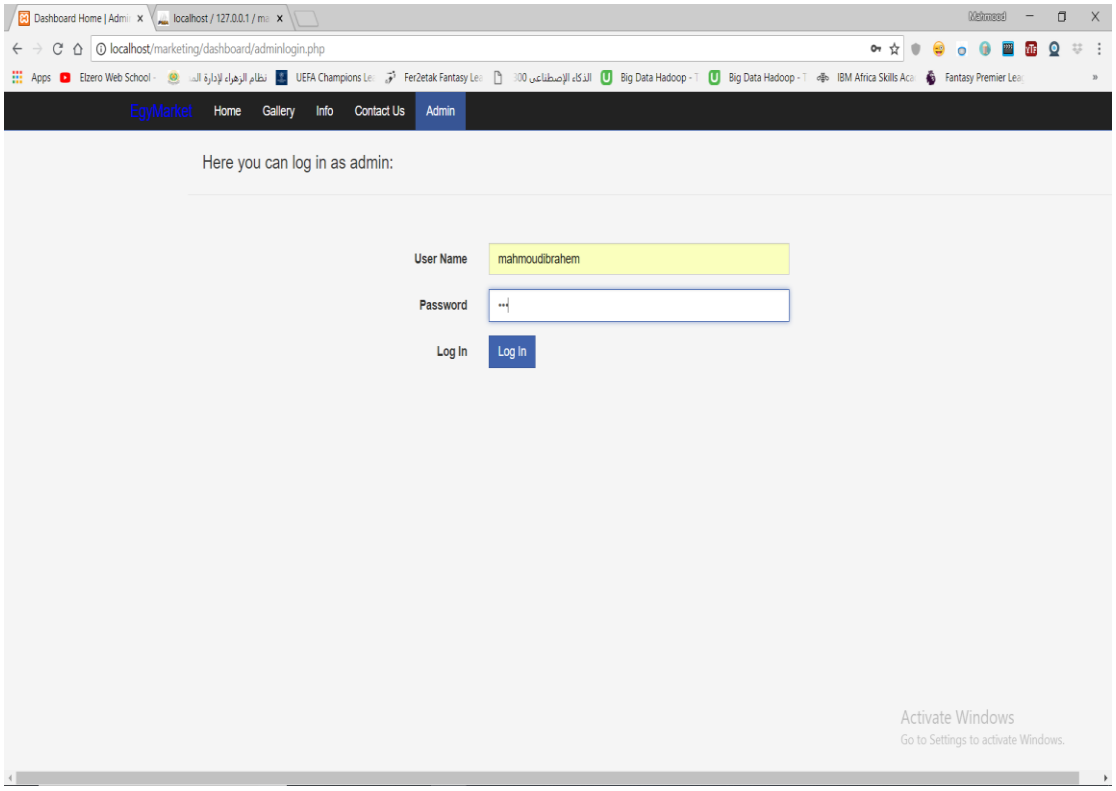
- User can view comment of other user's.

4.8 Rated comment on a product



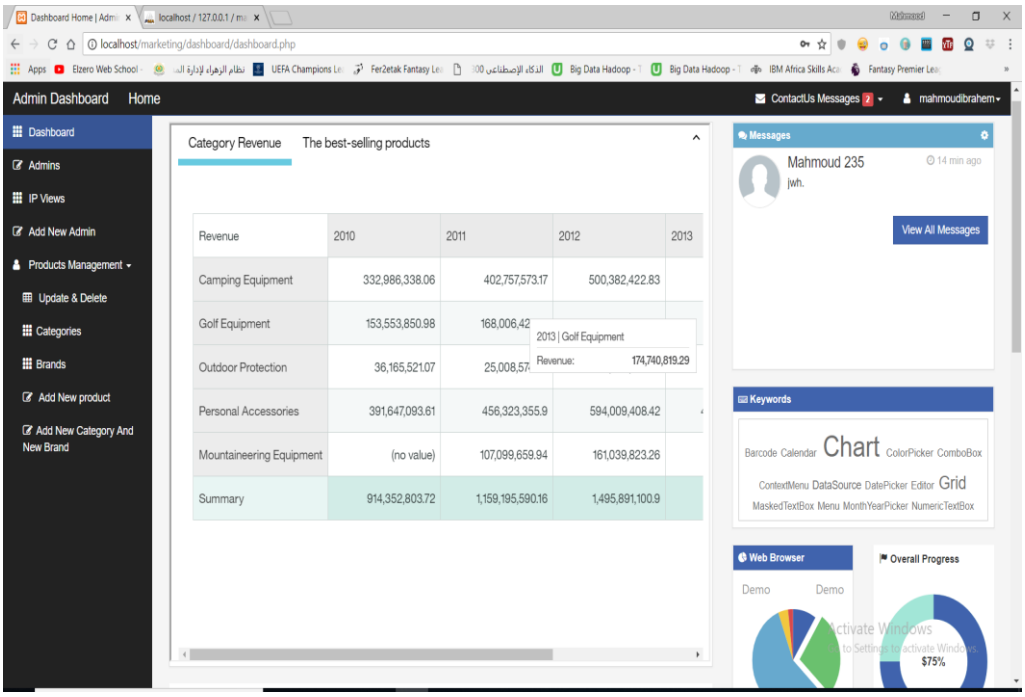
The user can add any number of comments on any product explaining the features he liked in this product or any negative manners he didn't like.

4.9 Admin Login

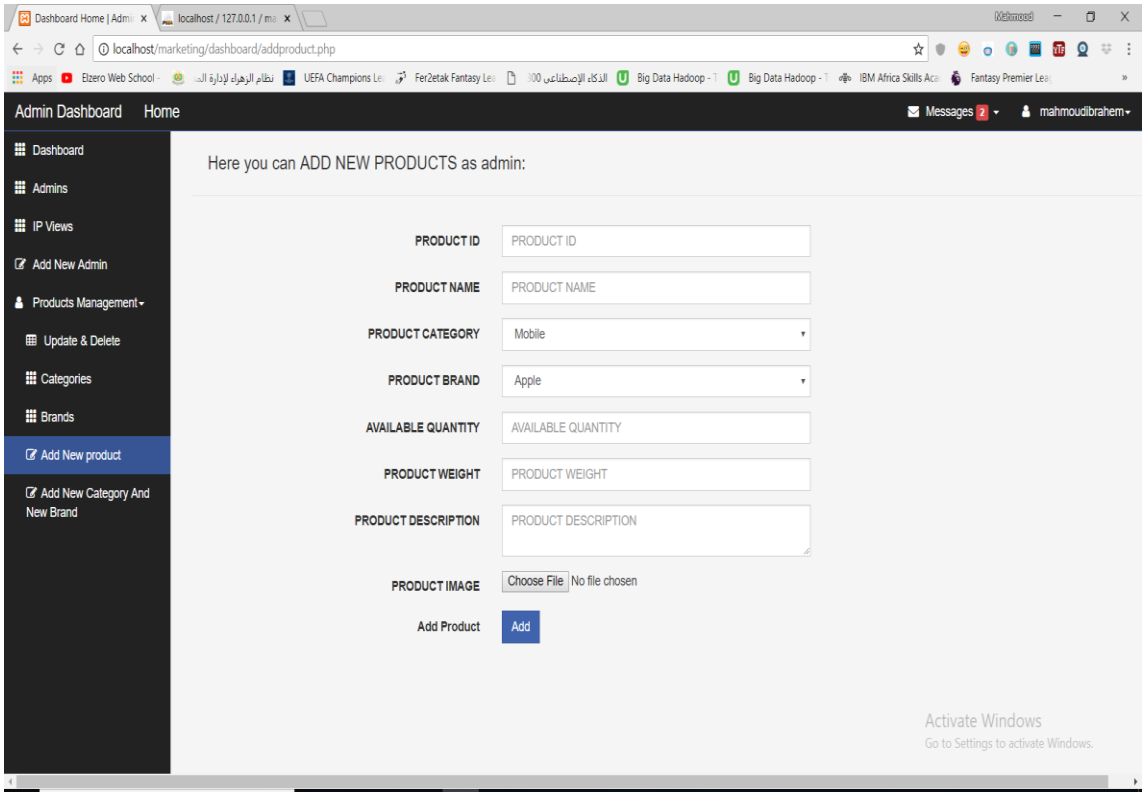


-Through this page the Admin can login to the system using his ID and password..

4.10 Dashboard

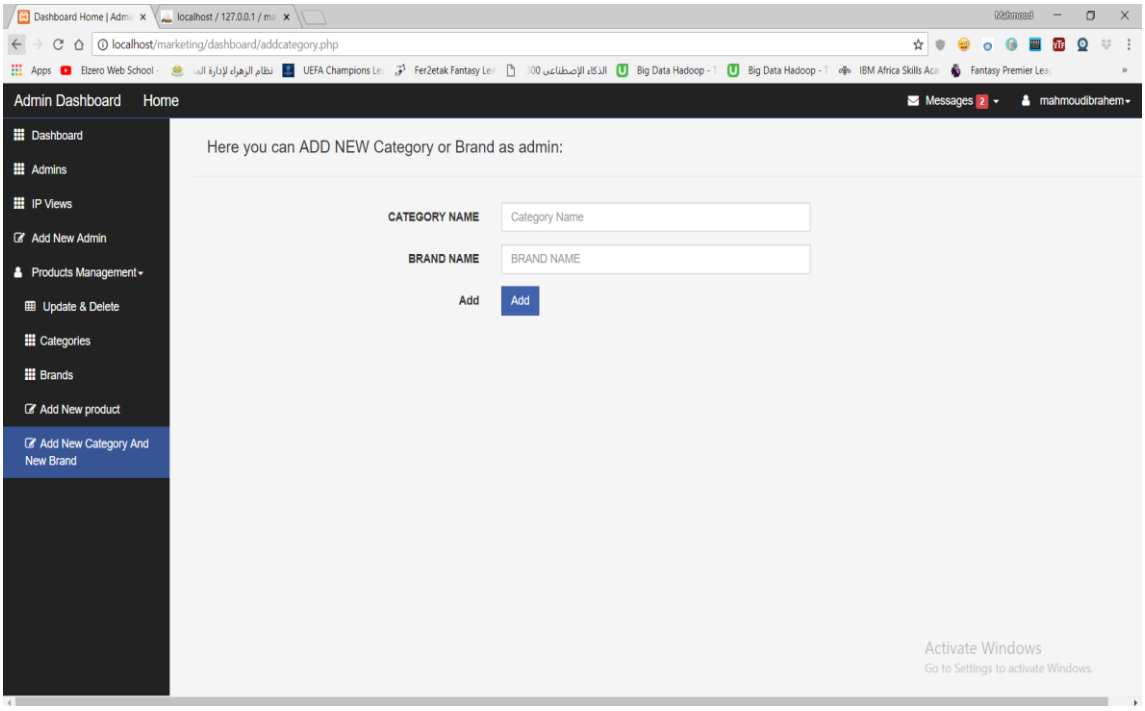


4.11 Add New Product by Admin

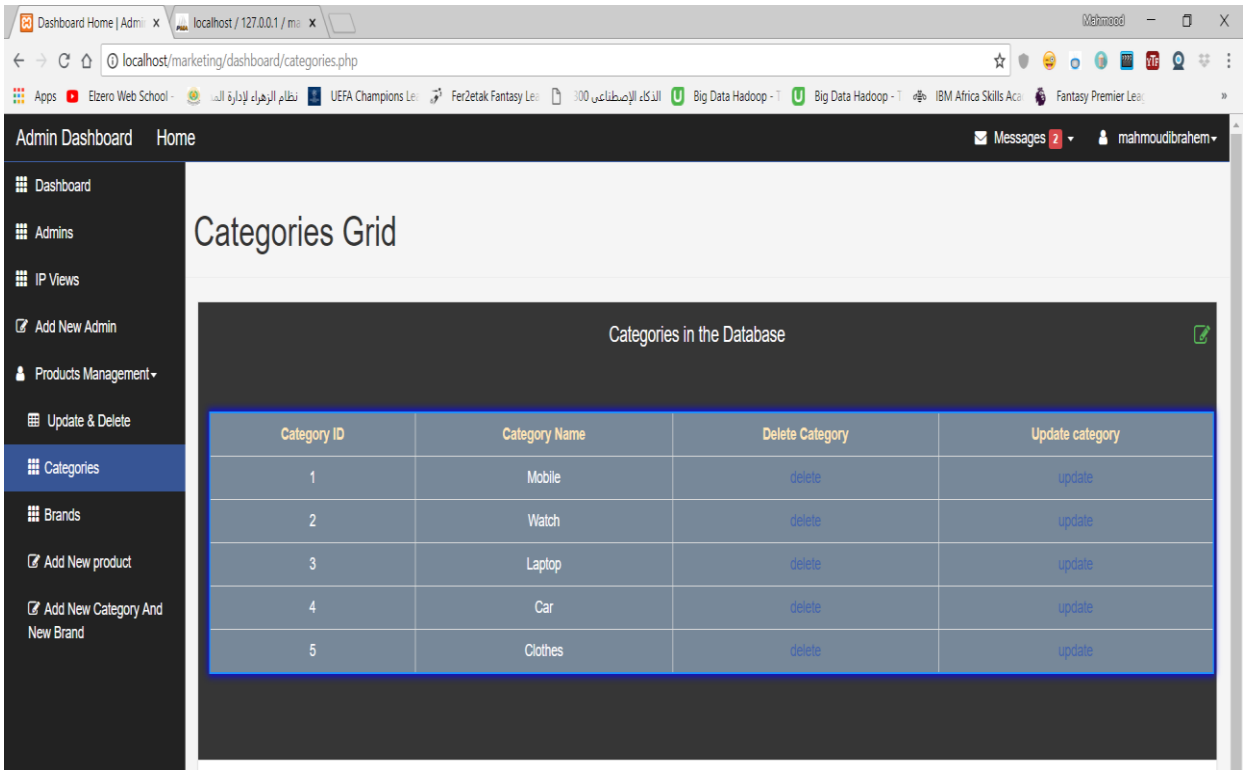


-in this page, Admin add products image and their features.
-Admin add keywords in database so that system will match the comment with the keywords in database and will rate the product.

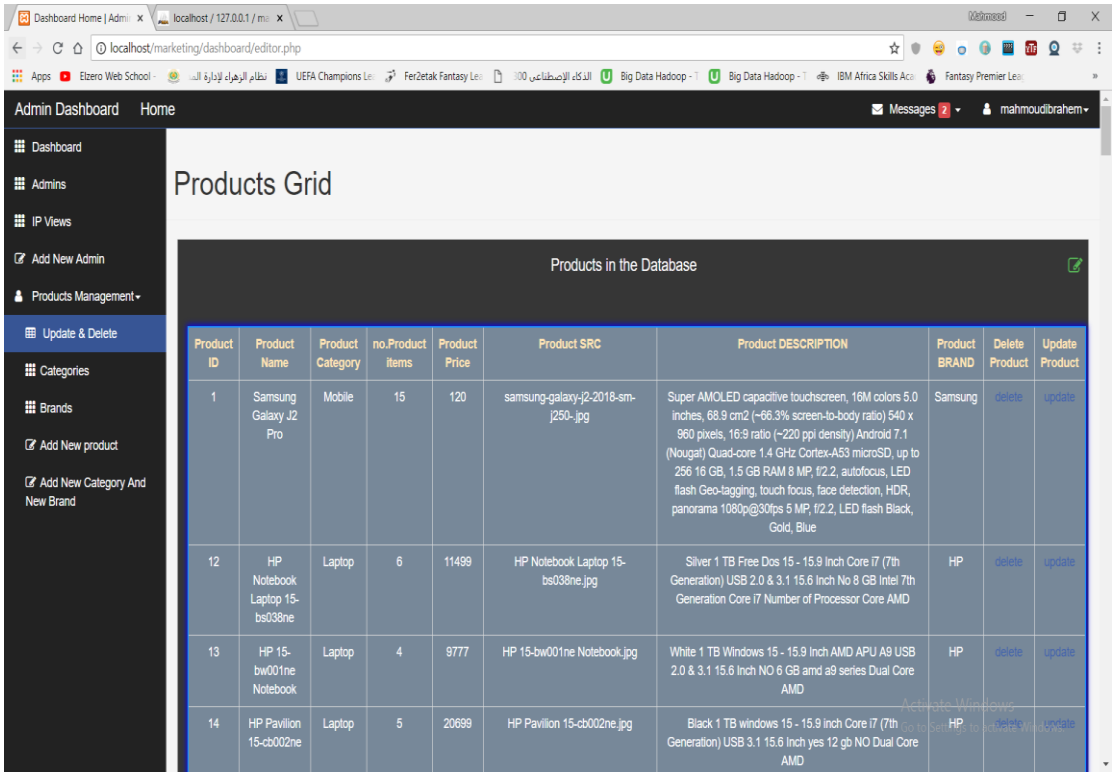
Admin Dashboard (add Brand or Category):



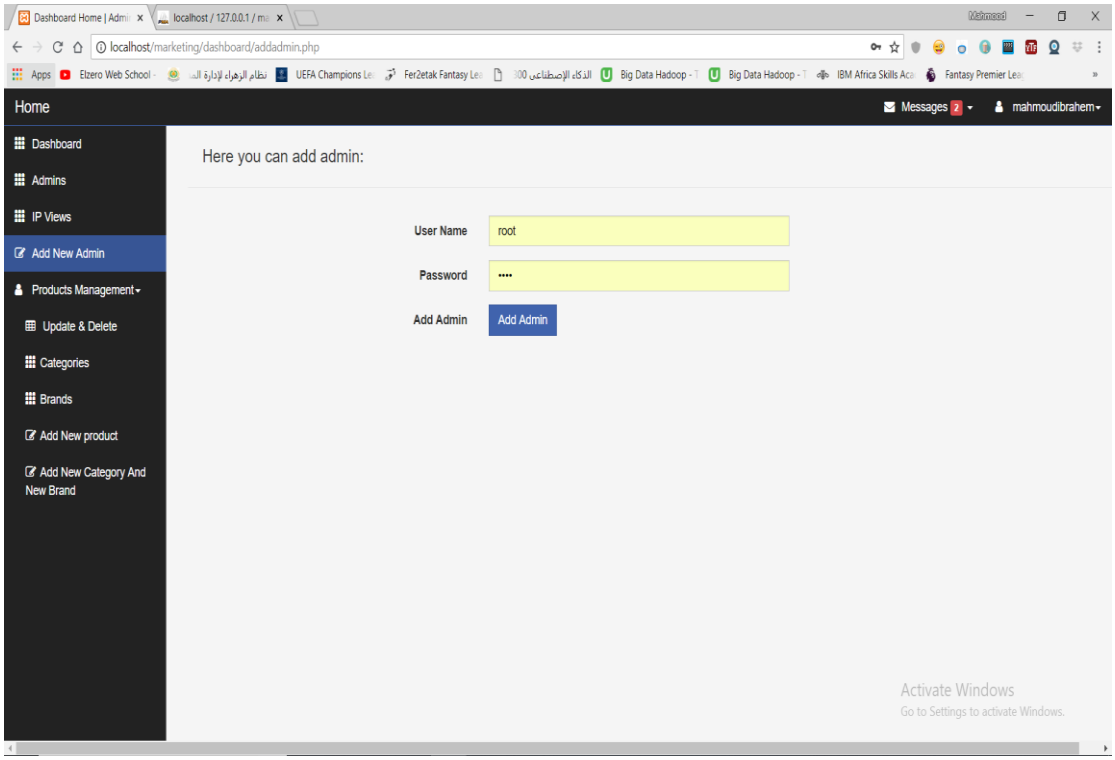
Admin Dashboard cont`d:(categories)



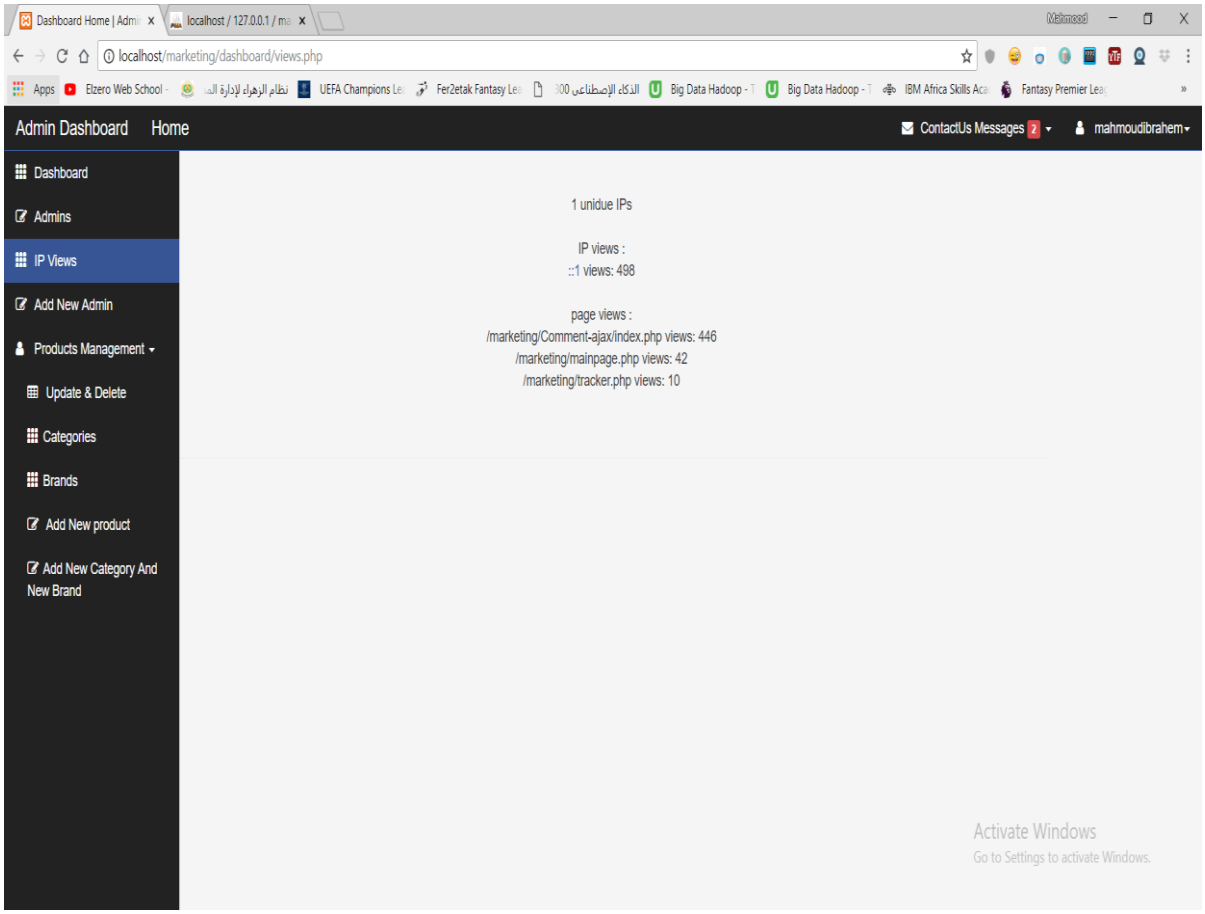
Admin Dashboard cont`d:(update&delete)



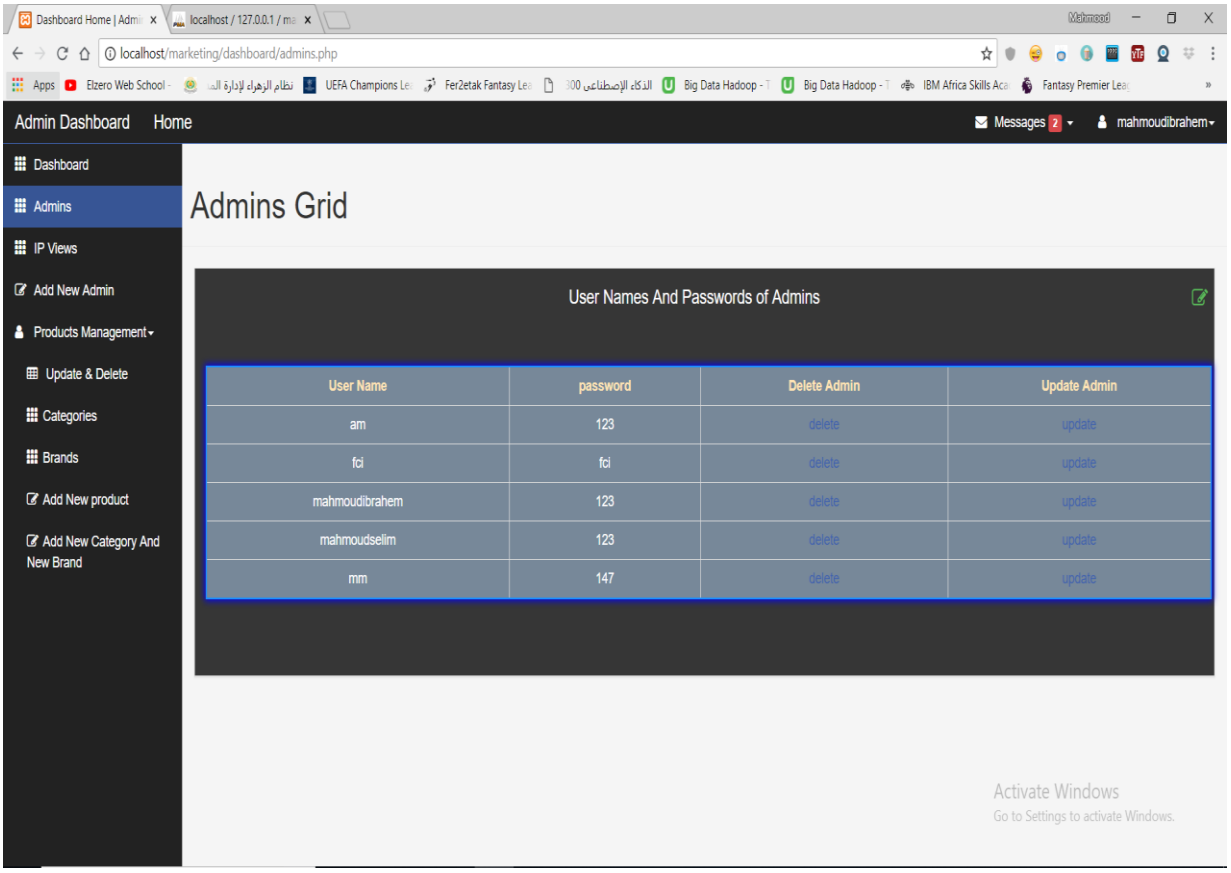
Admin Dashboard cont`d:(add new admin)



Admin Dashboard cont`d:(ip views)



Admin Dashboard cont`d:



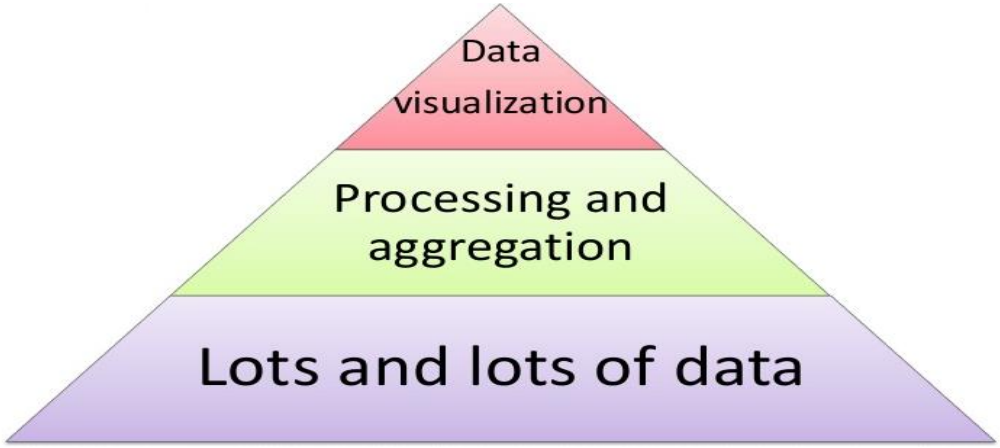
Data Model for sentiment analysis using Rapidminer:

Business intelligence analytics using Cognos :

Users tracking to improve webApp efficiency:

Business Intelligence Analytics:

Business intelligence (BI) is a set of theories, methodologies, architectures, and technologies that transform raw data into meaningful and useful information for business purposes.



Information Delivery

- Dashboards
- Reporting
- Microsoft Office integration
- Ad hoc report/query
- Mobile BI

Analysis

- Interactive visualization
- Search-based data discovery
- Geospatial and location intelligence
- Online analytical processing (OLAP)
- Embedded advanced analytics

Integration

- BI infrastructure and administration
- Collaboration
- Metadata management
- Embeddable analytics
- Development tools
- Support for big data sources

Scope of Business Analytics

- Descriptive analytics
  - uses data to understand past and present
- Predictive analytics
  - analyzes past performance
- Prescriptive analytics
  - uses optimization techniques

Data for Business Analytics

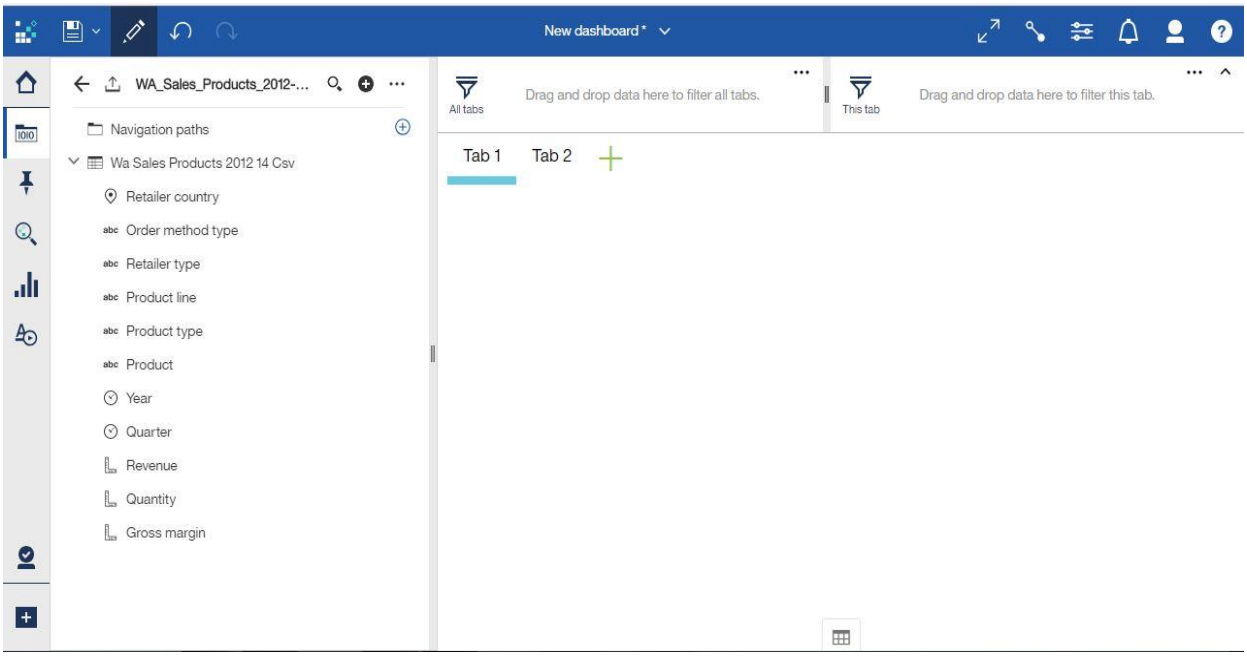
- DATA
  - collected facts and figures
- DATABASE
  - collection of computer files containing data
- INFORMATION
  - comes from analyzing data

Business Pressures–Responses–Support Model

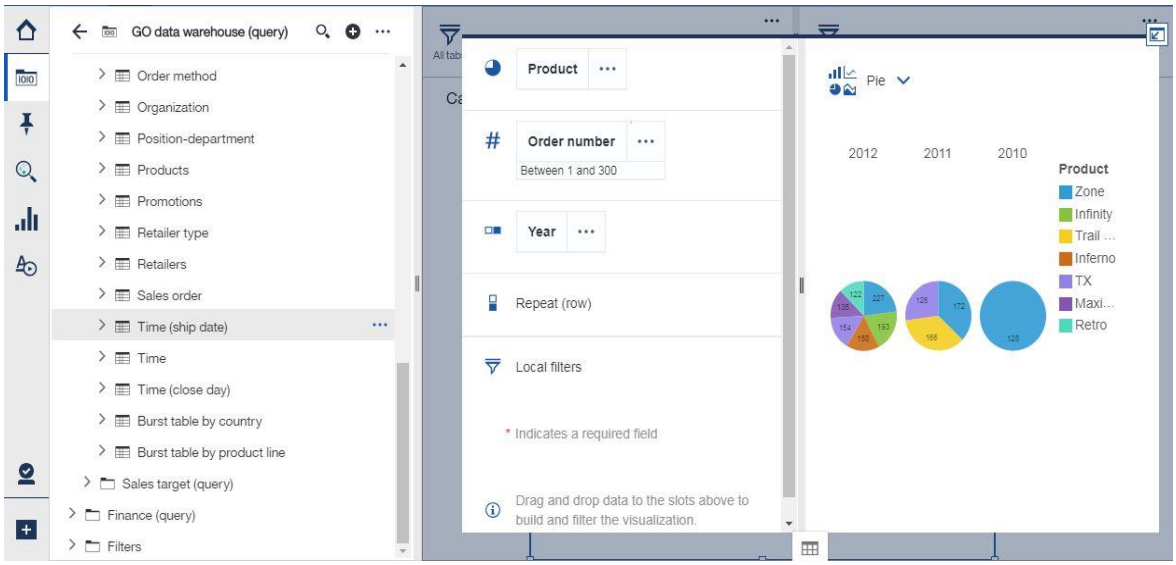
- Business pressures:
  - Result of today's competitive business climate.
- Responses:
  - To counter the pressures.
- Support:
  - To better facilitate the process.

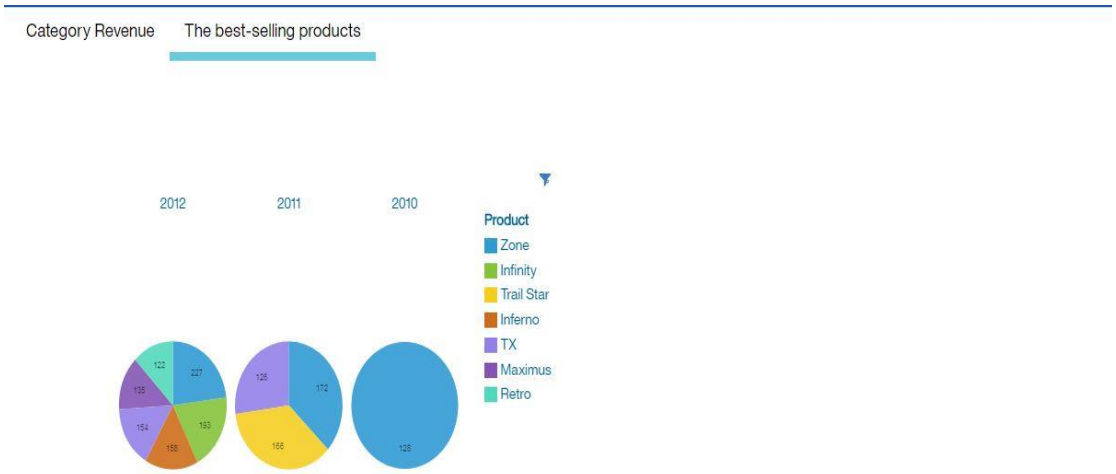
IBM Cognos BI Implementation:

1. New Dashboard

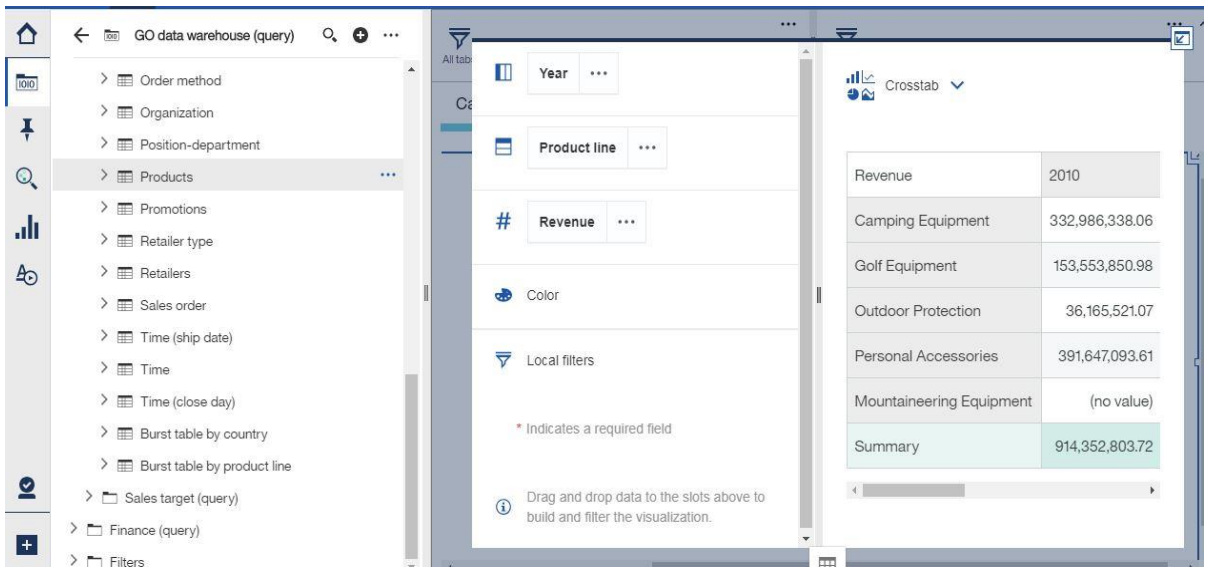


2. Pie Chart for The best selling product:





3. Crosstab for Category Revenue



Category Revenue | The best-selling products

Revenue	2010	2011	2012	2013	Summary
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97	1,589,036,664.03
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29	726,411,367.89
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	4,471,025.26	75,994,296.25
Personal Accessories	391,647,093.61	456,323,355.9	594,009,408.42	443,693,449.85	1,885,673,307.78
Mountaineering Equipment	(no value)	107,099,659.94	161,039,823.26	141,520,649.7	409,660,132.9
Summary	914,352,803.72	1,159,195,590.16	1,495,891,100.9	1,117,336,274.07	4,686,775,768.85

User Tracker:

Log file for user tracking: In this log file Admin can track user's actions what pages he visited & what products he purchased, when did he visited this page, URL of visited pages, how many times this page has been visited by different users, And finally Ip of those users. We can in future exploit these data to facilitate actions user can do to achieve benefit from our website , and also we can prioritize pages based on most visited pages (products) or most purchased products , and many other benefits may achieved from this log file…let's make it a challenge for us to improve our projects.

Users Tracker

pages viewed by ::1

Page URL	Date
/marketing/tracker.php	02:16:37 04/14/2018
/marketing/tracker.php	02:18:05 04/14/2018
/marketing/tracker.php	02:21:53 04/14/2018
/marketing/tracker.php	02:22:09 04/14/2018
/marketing/tracker.php	02:22:10 04/14/2018
/marketing/tracker.php	02:22:12 04/14/2018
/marketing/mainpage.php	02:25:58 04/14/2018
/marketing/mainpage.php	02:26:05 04/14/2018
/marketing/mainpage.php	02:26:07 04/14/2018
/marketing/tracker.php	02:26:14 04/14/2018
/marketing/tracker.php	02:26:15 04/14/2018
/marketing/mainpage.php	02:26:19 04/14/2018
/marketing/tracker.php	02:28:11 04/14/2018

Thanks End of chapter 4

Chapter 5

Sentiment Analysis

5.1 Introduction

Sentiment Analysis is an important application of Natural Language Processing. It is a process of finding people's opinion, attitude, views and detecting emotions towards any entity. The entity can be individuals, products, events or topics. These topics are mostly covered by reviews. Sentiment analysis is a process of identifying the sentiment expressed in the form of text and then analyzing it to get beneficial information. Document-level, sentence-level, and aspect-level are three main classification levels. In document-level sentiment analysis aims to classify an input document as a positive or negative sentiment. The whole document is considered as an input unit in document-level sentiment analysis. Sentence-level sentiment analysis has an aim to classify sentiments expressed in each sentence and then identifying the sentence as subjective or objective. If the sentence is subjective, sentence-level sentiment analysis will determine whether the sentence contains positive or negative sentiments. Aspect-level sentiment analysis aims to classify the sentiments with respect to the specific aspects of entities and then identifying the opinion regarding entities.

5.2 Sentiment Analysis

In Natural Language Processing, the field of sentiment analysis is a computational task for automatically detecting and classifying sentiment from text, document or from sentences to finding it's ''polarity'' or ''orientation''. The polarity of the documents can be positive, negative or neutral. Some more fine-grained in polarity including very positive, very negative or intensity levels from 1 to 5 scales have also been considered.

5.3 Approaches for Text Classification Task

Machine learning based classification need training datasets and test datasets. Training datasets are prepared to learn the different characteristics of data. Test datasets are used to validate the automatic classifier performance. A various machine learning approaches have been available to classify content. Different kind of machine learning approaches is Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM) that have been gained great success in text domain. Some another mostly used machine learning approaches in the natural language processing fields are winnow classifier, K-nearest neighborhood, N-gram model, ID3, C5, centroid classifier.

5.3.1 Naive Bayes is a simple machine learning approach. It is based on probabilistic classifier approach. It uses Bayes theorem due to strong (naive) independence assumptions. The reason behind the success of Naïve Bayes approach is that it's working quality for text domain because the evidence is vocabularies/dictionaries or words/terms appearing in texts or documents, and the size of the vocabularies/dictionaries are commonly in the range of thousands. The large size of the corpus, evidence, and vocabularies will make this model best and work best for text classification problems. The Naive Bayes approach has widely used for content present in Indian Language for document classification.

5.3.2 A statistical classification approach is a support vector machine approach. It used to classify the training data into two different classes and makes decisions usually depends upon the support vectors that are used to choosing only effective elements of the training datasets. A support vector machine is a discriminative classification approach and it is considered as the best text classification approach. The K-nearest neighbor well known with short form KNN is a typical example-based classification approach that does not makes explicit decisions and declarative representation of the sentiments categories but it relies on the categories labels that allocated in the training documents similar to the test documents for comparison. For example, give a test document

as an input to the system to finds the k nearest neighbors from the training documents and getting optimal classification results.

5.3.3 All supervised machine learning approaches provide a reasonable amount of accuracy but it can be obtained subjective only if it fulfills the requirement that validates the test data that should be similar to training data. When moving towards supervised machine learning approaches text domain classification to another specific domain it would require the collection of annotated data for the new specified domain and retained data for the classification process. This dependency on annotated training datasets is one of the major shortcomings of all supervised machine learning approaches.

5.3.4 Unsupervised learning machine learning approach is used to build a set of fully annotated data that helps to train a classifier with the use of machine learning approaches. This classification approach is then used to separate novel incoming data.

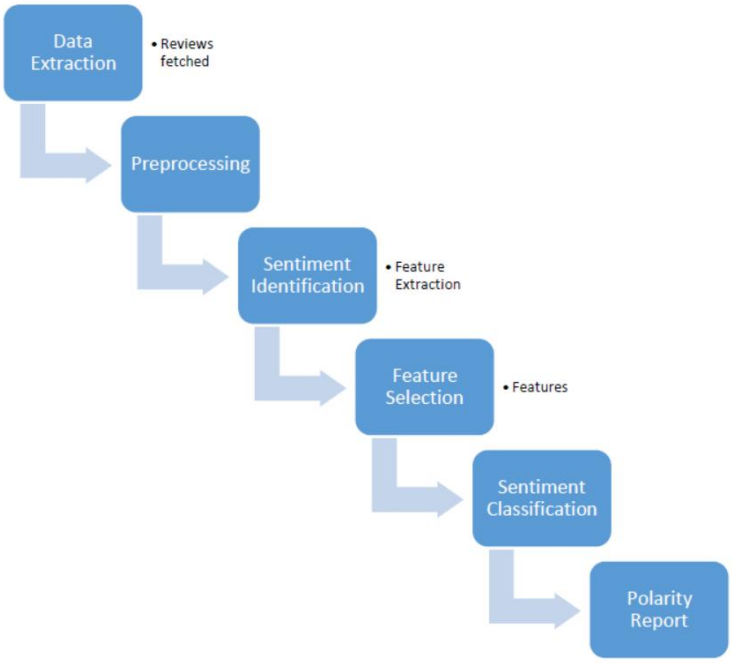
5.4 Methodology

In this thesis work, a piece of end-to-end research on sentiment labeling, using supervised learning techniques is performed. This will involve preprocessing corpora, making choices about features extraction to include in text representations, training classifiers and evaluating performance.

Support vector machine is a classification approach that receives input information during its training process and then building a hypothesis function for systems input and output that used for predicting future results. Support vector machine is a supervised machine learning approach with various learning methods. Support vector machine algorithm is used to analyze data and it uses recognize patterns and regression methods for classification. A support vector machine is based on natural learning instead of heuristics or analogies techniques and the results are based on statistical learning techniques. Support vector

machine performs an implicit embedment of data into a high-frequency feature space. It used nonlinear rules, geometry, and linear algebra for classification in input spaces.

The data sets which are used is an important part of Sentiment Analysis. The main sources of data are from the product reviews. The reviews given by the users gives insight into product reception and quality, which can be used to make important business related decisions. The reviews sources are mainly review sites. With the explosive growth of user generated messages, Twitter has become a social site where millions of users can exchange their opinion. Sentiment Analysis on Twitter data has provided an economical and effective way to expose public opinion. There have been some research work focusing on assessing the relations between online public sentiment and real-life events (e.g., consumer confidence, stock market [1], polls [2]). It is reported that events in real life indeed have a significant and immediate effect on the public sentiment in Twitter. Based on such correlations, some other work [3], [4] made use of the sentiment signals in blogs and tweets to predict movie sales and elections. The pioneering work of figuring out application and challenges in the field of Sentiment Analysis was presented by Pang and Lee [5] and Liu [6]. They mentioned the techniques used to solve each problem in Sentiment Analysis.



5.4.1 Tokenization

Given input as form of text, documents to the system, tokenization is a process of breaking text up into small unit of pieces known as tokens and deleting some unique characters such as punctuation marks and special symbols at the same time. A token is a small unit of characters, and a sequentially grouped characters that makes a sensible sentence and it useful for semantic processing. Remove punctuation marks, special symbols, numerical values from the text documents.

5.4.2 Stop Word Removal

A stop list is a list of commonly used words. Stop words usually language specific, although every language may contain many stop words. Natural language processing systems and many search engines having a variety of stop words depend upon languages, or it can be used a single multilingual stop-list. By the ignorance of functional words, the meaning of terms can be clearer, when processing the natural language content. Hence it is a process of removing words which do not support for information analysis task and appear too often in the documents. To remove these words from the text documents, proposed system has built a list of Punjabi stop words, which has been manually analyzed and identified stop words.

5.4.3 Stemming

Stemming is a task to reduce a derived word from their root value or stem value.

5.4.4 Part of Speech Tagging

POS tagging is a task that used for allotment of correct tags to the word from a number of available tags. Here the tag means grammatical

information of the word. It is well known that a computer will understand the language and process the language if the meaning of each and every word of that language is known or well defined.

5.4.5 Transformation

The weight value of each and every word from the corpus is determined with the use of term frequency and inverse document frequency (TF-IDF). TF-IDF defines the weight values of each and every word in a document using the formula:-

wd = fw, d* log(|Doc| fw,Doc)

w denotes words in an individual document, Doc is a collection of documents, d is single document belongs to Doc, |Doc| is the size of the corpus, fw,d is a number of times word appears in a document, fw,Doc is a number of documents in which word appears in Doc. And the highest valued TF-IDF Punjabi words are candidates for feature selection in this step.

5.4.6 Feature Selection

Feature Selection helps to build a classification more effective by deleting the quantity of content to be analyzed and selecting related features to be considered for the classification process.

5.4.7 Classification

Text classification is used to classifying data into predefined classes and the classes can be positive and negative or it can be used some other classes based on their needs. The first step of machine learning approaches is transforming documents means converting a string format into a suitable string format. In proposed system used supervised learning approach for text classification. Each word corresponding to features takes its weight value.

5.4.8 Lexicon-Based Approach

The lexicon-based approach involves calculating orientation for a document from the semantic orientation of words or phrases in the document. Dictionaries for lexicon-based approaches can be created manually, as authors describe in this article or automatically, using seed words to expand the list of words. Much of the lexicon-based research has focused on using adjectives as indicators of the semantic orientation of text. First, a list of adjectives and corresponding Sentiment Orientation (SO) values is compiled into a dictionary. According to previous study, adjectives are good indicators of SO. Then, for any given text, all adjectives are extracted and annotated with their SO value, using the dictionary scores. The SO scores are in turn aggregated into a single score for the text. However, although an isolated adjective may indicate subjectivity, there may be insufficient context to determine semantic orientation. As pointed out by Turney P., the adjective „brutal‟, „insane‟ may have negative orientation in a pet review, in a phrase such as “brutal and insane breed of dog”, but it could have largely a positive orientation in a movie review, in a phrase like “brutal and insane action sequence”. Therefore the algorithm extracts two consecutive words. The first member is an adverb or an adjective while the second word provides the context.

A) Dictionary-based approach involves using a dictionary which contains synonyms and antonyms of a word. Thus, a simple technique in this approach is to use a few seed sentiment words to bootstrap based on the synonym and antonym structure of a dictionary. Specifically, this method works as follows: A small set of sentiment words (seeds) with known positive or negative orientations is first collected manually, which is very easy. The algorithm then grows this set by searching in any online available dictionary for their synonyms and antonyms. The seed list will be added with the new found words. The process iteratively keeps on adding the words until no more new words are found. Manual inspection can be used to clean up the list at last.

5.4.9 Sentiment Analysis

Sentiment analysis is a task that changed unstructured data into beneficial information. When the analysis process has been done, the results are represented in the form of graphs for examples pie chart, line graphs, and bar chart.

5.4.10 Applications of Sentimental Analysis

SA can be used in diverse fields for various purposes. This section discusses some of the Common ones.

A) Online Commerce The most general use of SA is in e-commerce enterprise. Websites allow their users to submit their experience about shopping and reviewing their thoughts, opinions and their take on product qualities. They provide summary for the product and different features of the product by assigning ratings or scores. For example, htttp://www.flipkart.com is an online shopping website where in users rate the products they have bought critically.

B) Voice of Customers (VOC) and Voice of Market (VOM) VOC is a market research technique to describe the in-depth process of capturing a customer's intentions, desires, antipathies and expectations while VOM means that you would be surveying not only your own customers but those of key competitors as well. Detection of such information as early as possible helps in direct and target key marketing campaigns.

C) Brand Reputation Management (BRM) BRM helps in finding how public perception of a certain brand changes positively or negatively. The variation after an event can be analyzed using SA. A novel work in interpreting the change in sentiment variation has been done by Tan S et al.

D) Recommendation Systems By classifying the people's opinion into positive and negative, the system can say Which one should get recommended and which one should not get recommended.

E) Policy Making Using SA, policy makers can go through the public sentiment towards a policy, and use it to make the policies which are in demand by the public at large. [28].

5.5 Results

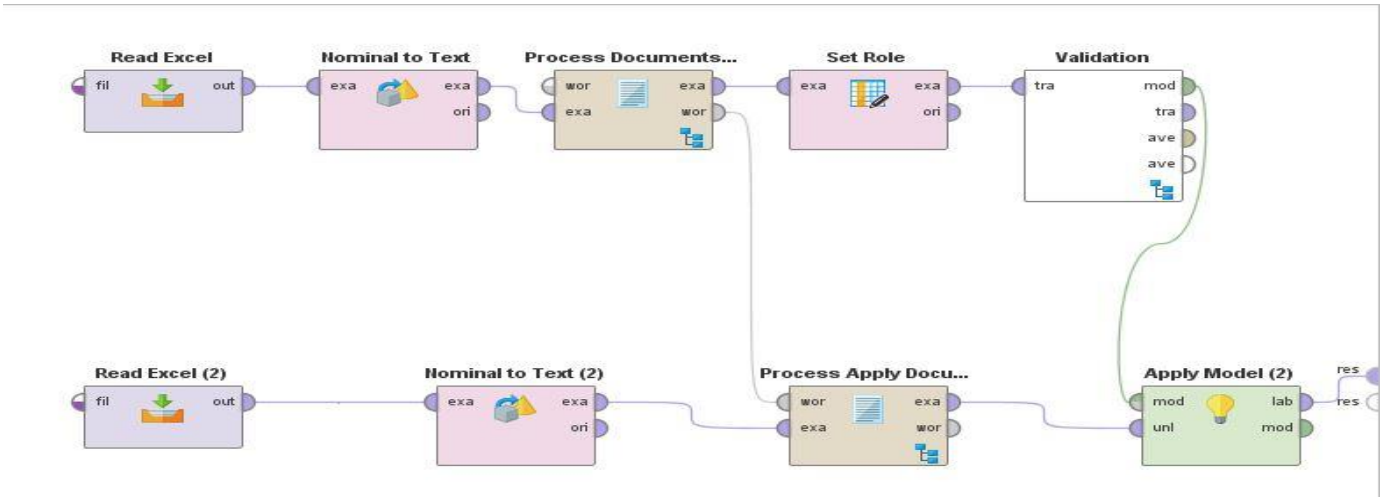
The proposed sentiment analysis system used only support vector machine approach for the implementation, and the advantage of this method is that it can further applied to virtually any language in this world.

- **Accuracy:**

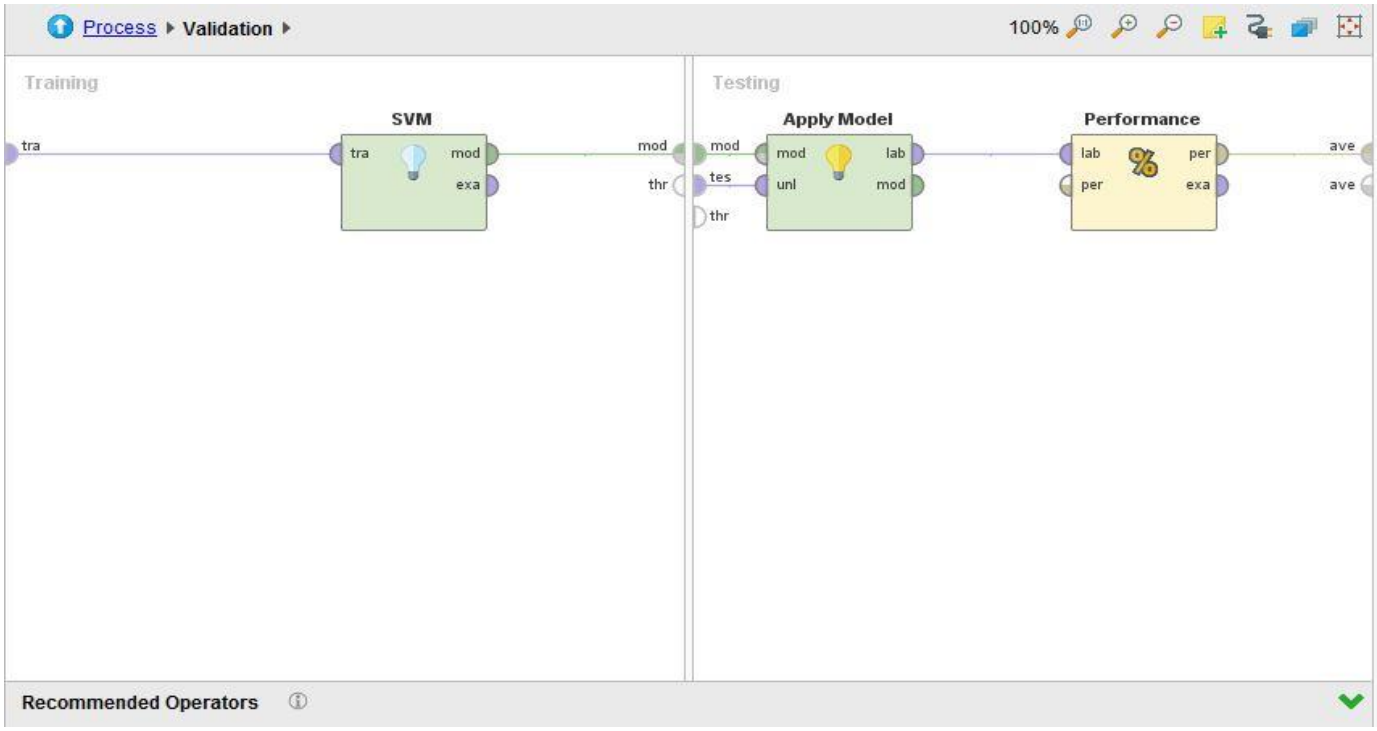
accuracy: 63.50% +/- 5.94% (mikro: 63.50%)

	true negative	true positive	class precision
pred. negative	26	4	86.67%
pred. positive	69	101	59.41%
class recall	27.37%	96.19%	

- **Sentiment Model:**



- **Validation Model:**



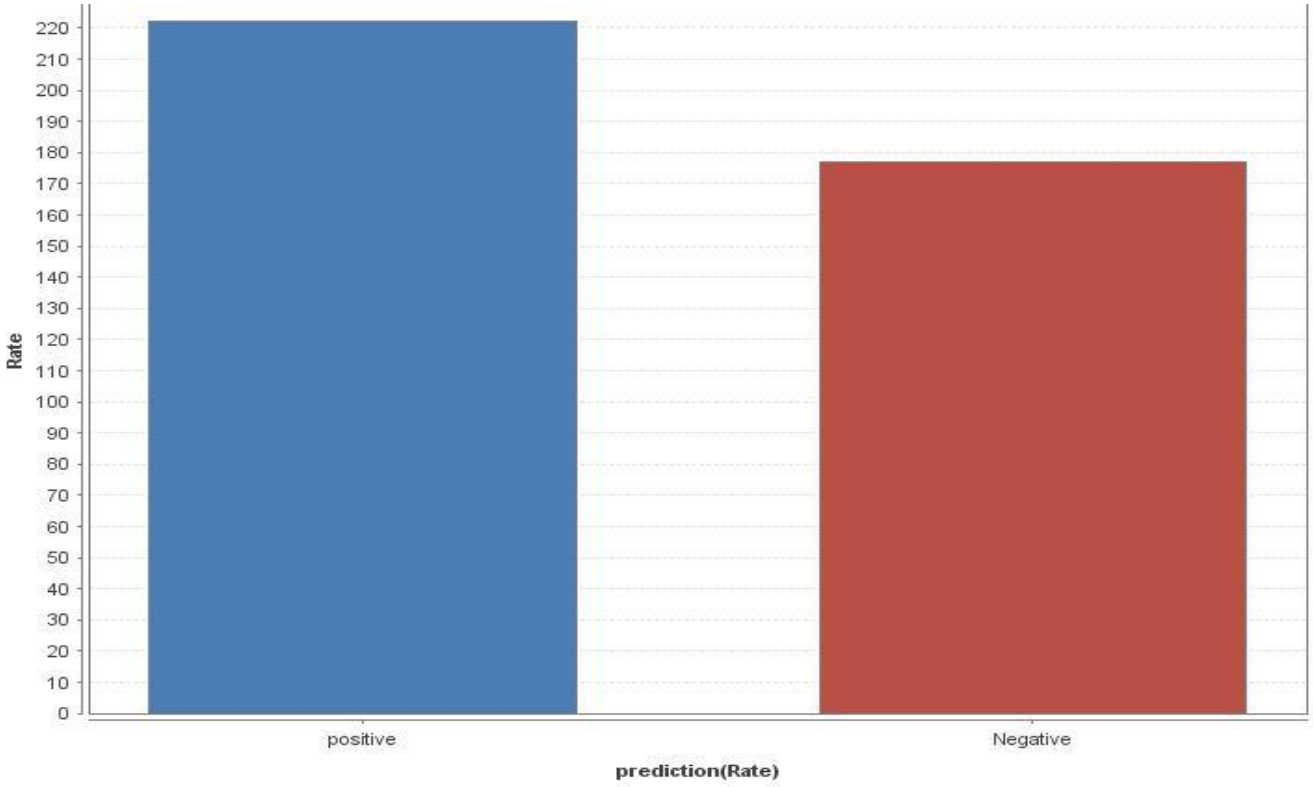
- **Tabular Result:**

ExampleSet (399 examples, 4 special attributes, 1310 regular attributes)

Filter (399 / 399 examples): all

Row No.	Rate	prediction(R...	confidence(...	confidence(...	A	ABLE	ABRACADA...	ABSORBED	ABU
1	?	positive	0.655	0.345	0	0	0	0	0
2	?	positive	0.655	0.345	0	0	0	0	0
3	?	Negative	0.269	0.731	0	0	0	0	0
4	?	positive	0.655	0.345	0	0	0	0	0
5	?	Negative	0.269	0.731	0	0	0	0	0
6	?	Negative	0.157	0.843	0	0	0	0	0
7	?	Negative	0.269	0.731	0	0	0	0	0
8	?	positive	0.655	0.345	0	0	0	0	0
9	?	positive	0.655	0.345	0	0	0	0	0
10	?	positive	0.655	0.345	0	0	0	0	0
11	?	Negative	0.157	0.843	0	0	0	0	0
12	?	Negative	0.269	0.731	0	0	0	0	0
13	?	positive	0.655	0.345	0	0	0	0	0
14	?	Negative	0.269	0.731	0	0	0	0	0
15	?	positive	0.655	0.345	0	0	0	0	0

- **Graph Result:**



Thanks End of chapter 5

Chapter6
Conclusions and Future Work

6.1 Conclusion:

You can now give us your data of your organization which you think it's useless and we will give you information will make your organization better by some computational processes you don't need to know, finally:

- As a business owner do you think our service can help you? If it can. What can we do to achieve more success in the future?
- As a user of our site do you think this will reflect on the products you buy positively? If not, why not

6.2 Future work:

The future work of our project is divided into two parts:

Technical side:

1- Improve our algorithm of sentiment analysis to achieve the most accuracy can be reached.

2- Make most benefit from data in log file of user tracking.

3- Build a Data Warehouse System to Store Data to be analyzed.

4- sentiment analysis on features of each category of the products.

Market side:

5- Search for systems owners in the market who have data of their work but don't know how to make use of it or how to improve production, sales, and profitability of their product or service.

Appendix

```php
class Sentiment {

    /**
     * Location of the dictionary files
     * @var str
     */
    private $dataFolder = '';

    /**
     * List of tokens to ignore
     * @var array
     */
    private $ignoreList = array();

    /**
     * List of words with negative prefixes, e.g. isn't, arent't
     * @var array
     */
    private $negPrefixList = array();

    /**
     * Storage of cached dictionaries
     * @var array
     */
    private $dictionary = array();

    /**
     * Min length of a token for it to be taken into consideration
     * @var int
     */
    private $minTokenLength = 1;

    /**
     * Max length of a taken for it be taken into consideration
     * @var int
     */
    private $maxTokenLength = 15;

    /**
     * Classification of opinions
     * @var array
     */
    private $classes = array('pos', 'neg', 'neu');
```

Activate Windows
Go to Settings to activate Windows.


```php
    private $maxTokenLength = 15;

    /**
     * Classification of opinions
     * @var array
     */
    private $classes = array('pos', 'neg', 'neu');
```


```php
    /**
     * Token score per class
     * @var array
     */
    private $classTokCounts = array(
            'pos' => 0,
            'neg' => 0,
            'neu' => 0
    );

    /**
     * Analyzed text score per class
     * @var array
     */
    private $classDocCounts = array(
            'pos' => 0,
            'neg' => 0,
            'neu' => 0
    );

    /**
     * Number of tokens in a text
     * @var int
     */
    private $tokCount = 0;

    /**
     * Number of analyzed texts
     * @var int
     */
    private $docCount = 0;
```


```php
    private $tokCount = 0;

    /**
     * Number of analyzed texts
     * @var int
     */
    private $docCount = 0;
```


```php
    /**
     * Implication that the analyzed text has 1/3 chance of being in either of the 3 categories
     * @var array
     */
    private $prior = array(
            'pos' => 0.333,
            'neg' => 0.333,
            'neu' => 0.334,
    );

    /**
     * Class constructor
     * @param str $dataFolder base folder
     * Sets defaults and loads/caches dictionaries
     */
    public function __construct($dataFolder = false) {

            //set the base folder for the data models
            $this->setDataFolder($dataFolder);

            //load and cache directories, get ignore and prefix lists
            $this->loadDefaults();
    }
```

```php
/**
 * Get scores for each class
 *
 * @param str $sentence Text to analyze
 * @return int Score
 */
public function score($sentence) {

        //For each negative prefix in the list
        foreach ($this->negPrefixList as $negPrefix) {

                //Search if that prefix is in the document
                if (strpos($sentence, $negPrefix) !== false) {
                        //Reove the white space after the negative prefix
                        $sentence = str_replace($negPrefix . ' ', $negPrefix, $sentence);
                }
        }

        //Tokenise Document
        $tokens = $this->_getTokens($sentence);
        // calculate the score in each category

        $total_score = 0;

        //Empty array for the scores for each of the possible categories
        $scores = array();

        //Loop through all of the different classes set in the $classes variable
        foreach ($this->classes as $class) {

                //In the scores array add another dimention for the class and set it's value to 1. EG $scores->neg->1
                $scores[$class] = 1;

                //For each of the individual words used loop through to see if they match anything in the $dictionary
                foreach ($tokens as $token) {

                        //If statement so to ignore tokens which are either too long or too short or in the $ignoreList
                        if (strlen($token) > $this->minTokenLength && strlen($token) < $this->maxTokenLength && !in_array($token, $this->ignoreList)) {
                                //If dictionary[token][class] is set
                                if (isset($this->dictionary[$token][$class])) {
                                        //Set count equal to it
                                        $count = $this->dictionary[$token][$class];
                                } else {
                                        $count = 0;
                                }

                                //Score[class] is calcumeted by $scores[class] x $count +1 divided by the $classTokCounts[class] + $tokCount
                                $scores[$class] *= ($count + 1);
                        }
                }

                //Score for this class is the prior probability multiplyied by the score for this class
                $scores[$class] = $this->prior[$class] * $scores[$class];
        }

        //Makes the scores relative percents
        foreach ($this->classes as $class) {
                $total_score += $scores[$class];
        }

        foreach ($this->classes as $class) {
                $scores[$class] = round($scores[$class] / $total_score, 3);
        }

        //Sort array in reverse order
        arsort($scores);

        return $scores;
}
```

```
/**
 * Get the class of the text based on it's score
 *
 * @param str $sentence
 * @return str pos|neu|neg
 */
public function categorise($sentence) {

        $scores = $this->score($sentence);

        //Classification is the key to the scores array
        $classification = key($scores);

        return $classification;
}

/**
 * Load and cache dictionary
 *
 * @param str $class
 * @return boolean
 */
public function setDictionary($class) {
        /**
         *  For some people this file extention causes some problems!
         */
        $fn = "{$this->dataFolder}data.{$class}.php";

        if (file_exists($fn)) {
                $temp = file_get_contents($fn);
                $words = unserialize($temp);
        } else {
                echo 'File does not exist: ' . $fn;
        }

        //Loop through all of the entries
        foreach ($words as $word) {

                $this->docCount++;
                $this->classDocCounts[$class]++;

                //Trim word
                $word = trim($word);

                //If this word isn't already in the dictionary with this class
                if (!isset($this->dictionary[$word][$class])) {

                        //Add to this word to the dictionary and set counter value as one. This function ensures th
                        $this->dictionary[$word][$class] = 1;
                }//Close If statement

                $this->classTokCounts[$class]++;
                $this->tokCount++;
        }//Close while loop going through everyline in the text file

        return true;
}
```

```
/**
 * Set the base folder for loading data models
 * @param str  $dataFolder base folder
 * @param bool $loadDefaults true - load everything by default | false - just change the directory
 */
public function setDataFolder($dataFolder = false, $loadDefaults = false){
        //if $dataFolder not provided, load default, else set the provided one
        if($dataFolder == false){
                $this->dataFolder = __DIR__ . '/data/';
        }
        else{
                if(file_exists($dataFolder)){
                        $this->dataFolder = $dataFolder;
                }
                else{
                        echo 'Error: could not find the directory - '.$dataFolder;
                }
        }

        //load default directories, ignore and prefixe lists
        if($loadDefaults !== false){
                $this->loadDefaults();
        }
}

/**
 * Load and cache directories, get ignore and prefix lists
 */
private function loadDefaults(){
        // Load and cache dictionaries
        foreach ($this->classes as $class) {
                if (!$this->setDictionary($class)) {
                        echo "Error: Dictionary for class '$class' could not be loaded";
                }
        }

        if (!isset($this->dictionary) || empty($this->dictionary))
                echo 'Error: Dictionaries not set';

        //Run function to get ignore list
        $this->ignoreList = $this->getList('ign');

        //If ingnoreList not get give error message
        if (!isset($this->ignoreList))
                echo 'Error: Ignore List not set';

        //Get the list of negative prefixes
        $this->negPrefixList = $this->getList('prefix');

        //If neg prefix list not set give error
        if (!isset($this->negPrefixList))
                echo 'Error: Ignore List not set';
}

/**
```

```php
/**
 * Break text into tokens
 *
 * @param str $string   String being broken up
 * @return array An array of tokens
 */
private function _getTokens($string) {

        // Replace line endings with spaces
        $string = str_replace("\r\n", " ", $string);

        //Clean the string so is free from accents
        $string = $this->_cleanString($string);

        //Make all texts lowercase as the database of words in in lowercase
        $string = strtolower($string);
        $string = preg_replace('/[[:punct:]]+/', '', $string);

        //Break string into individual words using explode putting them into an array
        $matches = explode(' ', $string);

        //Return array with each individual token
        return $matches;
}

/**
 * Load and cache additional word lists
 *
 * @param str $type
 * @return array
 */
public function getList($type) {
        //Set up empty word list array
        $wordList = array();

        $fn = "{$this->dataFolder}data.{$type}.php";
        ;
        if (file_exists($fn)) {
                $temp = file_get_contents($fn);
                $words = unserialize($temp);
        } else {
                return 'File does not exist: ' . $fn;
        }

        //Loop through results
        foreach ($words as $word) {
                //remove any slashes
                $word = stripcslashes($word);
                //Trim word
                $trimmed = trim($word);

                //Push results into $wordList array
                array_push($wordList, $trimmed);
        }
        //Return $wordList
        return $wordList;
}
```

```
    /**
     * Function to clean a string so all characters with accents are turned into ASCII characters. EG: ‡ - a
     *
     * @param str $string
     * @return str
     */
    private function _cleanString($string) {

        $diac =
                /* A */ chr(192) . chr(193) . chr(194) . chr(195) . chr(196) . chr(197) .
                /* a */ chr(224) . chr(225) . chr(226) . chr(227) . chr(228) . chr(229) .
                /* O */ chr(210) . chr(211) . chr(212) . chr(213) . chr(214) . chr(216) .
                /* o */ chr(242) . chr(243) . chr(244) . chr(245) . chr(246) . chr(248) .
                /* E */ chr(200) . chr(201) . chr(202) . chr(203) .
                /* e */ chr(232) . chr(233) . chr(234) . chr(235) .
                /* Cc */ chr(199) . chr(231) .
                /* I */ chr(204) . chr(205) . chr(206) . chr(207) .
                /* i */ chr(236) . chr(237) . chr(238) . chr(239) .
                /* U */ chr(217) . chr(218) . chr(219) . chr(220) .
                /* u */ chr(249) . chr(250) . chr(251) . chr(252) .
                /* yNn */ chr(255) . chr(209) . chr(241);

        return strtolower(strtr($string, $diac, 'AAAAAAaaaaaaOOOOOOooooooEEEEeeeeCcIIIIiiiiUUUUuuuuyNn'));
    }

}
```

```
public static function totalDominantScore()
{
    $db = new Comments();
    $score=0;
    if(@$_SESSION['id']!=null)
    $p=$_SESSION['id'];
    else
    $p=1;
    $poscount=0;
    $neucount=0;
    $negcount=0;
    $strings =  $db->getRows("select * from comment  WHERE product_id = ? order by comment_id desc" , [$p]);
    $count = count($strings);
    require_once __DIR__ . '/../../../phpInsight-master/autoload.php';
    $sentiment = new \PHPInsight\Sentiment();
    $pos=0;
    $neu=0;
    $neg=0;
    foreach ($strings as $string)
    {

      // calculations:
     $scores = $sentiment->score($string["comment"]);
     $class = $sentiment->categorise($string["comment"]);
    $pos+=$scores['pos'];
    $neg+=$scores['neg'];
    $neu+=$scores['neu'];
    if($scores['pos']>$scores['neg']&&$scores['pos']>$scores['neu'])
        $poscount+=1;
    if(($scores['neu']>$scores['neg']&&$scores['neu']>$scores['pos'])||($scores['pos']==$scores['neg']))
        $neucount+=1;
    if($scores['neg']>$scores['pos']&&$scores['neg']>$scores['neu'])
        $negcount+=1;

    }
    /*if($count!=0){
    return '[pos]=> '.$pos/$count.' [neu]=> '.$neu/$count.' [neg]=> '.$neg/$count;}
    else {return '[pos]=> 0 [neu]=> 0 [neg]=> 0 ';}*/
    $std = array();
    if($count!=0)
    {
    $std['pos'] = $pos/$count;
    $std['neu'] = $neu/$count;
    $std['neg'] = $neg/$count;
    $std['count']=$count;

    }
    else
    {
    $std['pos'] = 0;
    $std['neu'] = 0;
    $std['neg'] = 0;
    $std['count']=$count;

    }

    $std['poscount'] = $poscount;
    $std['neucount'] = $neucount;
    $std['negcount'] = $negcount;


    return $std;

}
```

References

- http://www.saedsayad.com/data_mining_map.htm
- https://www.kaggle.com/
- https://www.ibm.com/analytics/
- https://www.w3schools.com/
- https://www.creately.com/
- https://pdfs.semanticscholar.org/5b71/a71d2cf8f6046570f7a1513a44523f0839a8.pdf
- http://positivewordsresearch.com/list-of-positive-words/
- https://rapidminer.com/
- http://www.enchantedlearning.com/wordlist/negativewords.shtml
- http://www.enchantedlearning.com/wordlist/positivewords.shtml