**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Laptop Reviews**

|  |  |
| --- | --- |
| **Group 1** | |
| **Group member** | Vo Thi Minh Chau – Team Leader – SE60931  Nguyen Van Hon – Team Member - 60390  Dinh Huu Toan – Team Member - SE60871  Nguyen Manh Khuong – Team Member - 60455 |
| **Supervisor** | Mr. Kieu Trong Khanh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | LRA |

-Ho Chi Minh City, ***06/2015***-

*This page is intentionally left blank*

***ACKNOWLEDGEMENTS***

We wish to thank various people for their contribution to this project: Our teachers for their advice and participation in the final review, our friends for their valuable technical support.

Special thanks should be given to Mr.Kiều Trọng Khánh, our research supervisor for his professional guidance and the useful, constructive recommendations throughout the course of this project.

# Table of Contents

[Table of Contents 4](#_Toc423987316)

[List of Tables 5](#_Toc423987317)

[List of Figures 6](#_Toc423987318)

[Definitions, Acronyms, and Abbreviations 7](#_Toc423987319)

[Report No.4 Software Design Description 8](#_Toc423987320)

[1. Design Overview 8](#_Toc423987321)

[2. System Architectural Design 9](#_Toc423987322)

[3. Component Diagram 10](#_Toc423987323)

[4. Detailed Description of Components 11](#_Toc423987324)

[4.1 Class Diagram 11](#_Toc423987325)

[4.2 Class Diagram Explanation 11](#_Toc423987326)

[4.3 Sequence Diagram 16](#_Toc423987327)

[5. User Interface Design 18](#_Toc423987328)

[6. Database Design 18](#_Toc423987329)

[6.1 Logical Diagram 18](#_Toc423987330)

[6.2 Data Dictionary 19](#_Toc423987331)

[7. Algorithms 21](#_Toc423987332)

[7.1 Analyze Comment 21](#_Toc423987333)

[7.2 String Comparison 23](#_Toc423987334)

# List of Tables

[Table 1: Entity Data Dictionary 18](#_Toc369941379)

[Table 2: Attribute Data Dictionary 19](#_Toc369941380)

# List of Figures

[Figure 1: MVC Architecture 9](#_Toc422480854)

[Figure 2: Component Diagram 10](#_Toc422480855)

[Figure 3: Class Diagram 11](#_Toc422480856)

[Figure 4: Force Parse Data Sequence Diagram 15](#_Toc422480857)

[Figure 5: Import Excel Sequence Diagram 15](#_Toc422480858)

[Figure 27: Logical Diagram 17](#_Toc422480859)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| LRA | Laptop Review |

# Report No.4 Software Design Description

## Design Overview

* This document describes the technical and user interface design of Laptop Reviews System using web. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe an ERD with logical diagram.

## System Architectural Design



Figure 1: MVC Architecture

**(http://www.w3schools.com/aspnet/mvc\_intro.asp)**

**The Model** is the part of the application that handles the logic for the application data. Often model objects retrieve data (and store data) from a database.

**The View** is the parts of the application that handles the display of the data. Most often the views are created from the model data.

**The Controller** is the part of the application that handles user interaction. Typically controllers read data from a view, control user input, and send input data to the model.

## Component Diagram

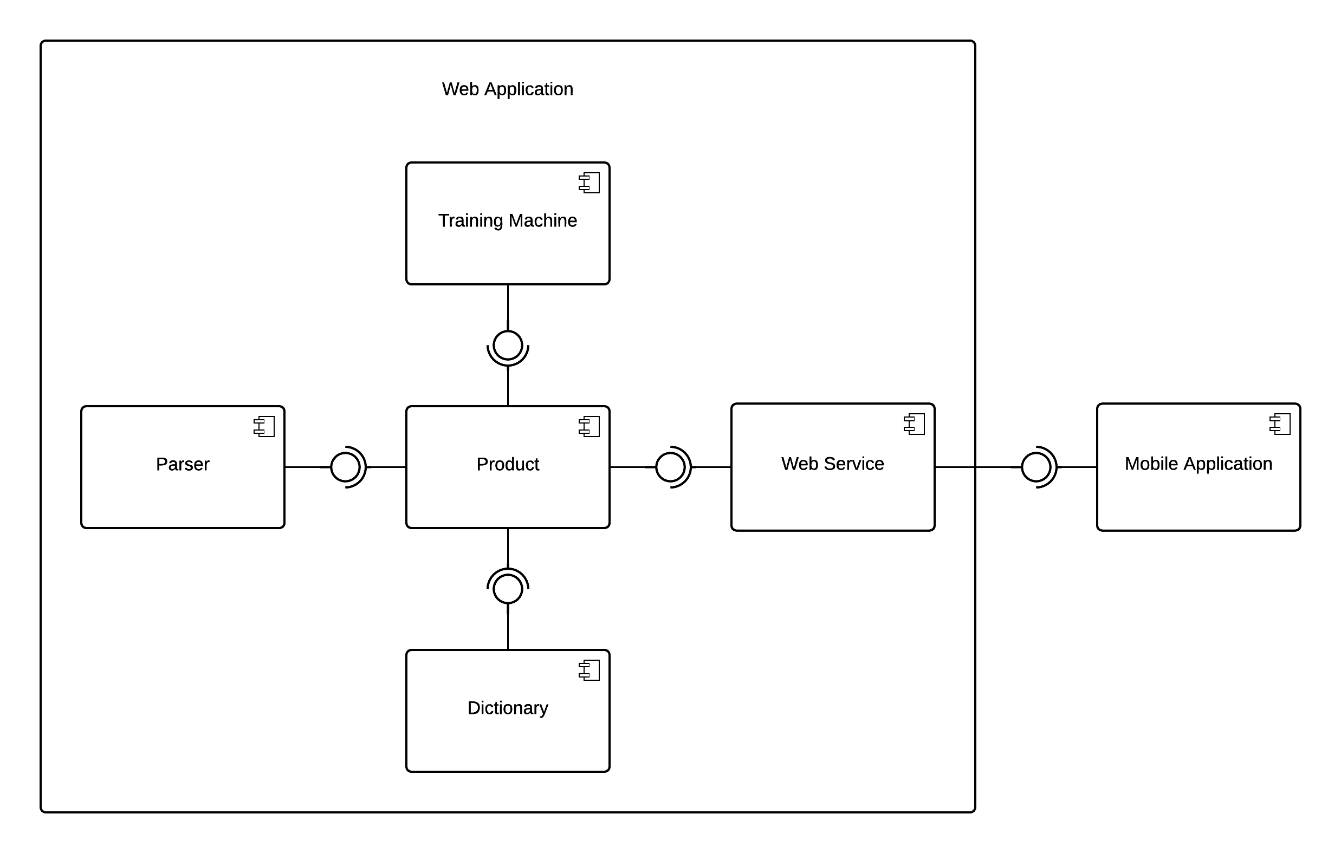


Figure 2: Component Diagram

## Detailed Description of Components

### Class Diagram

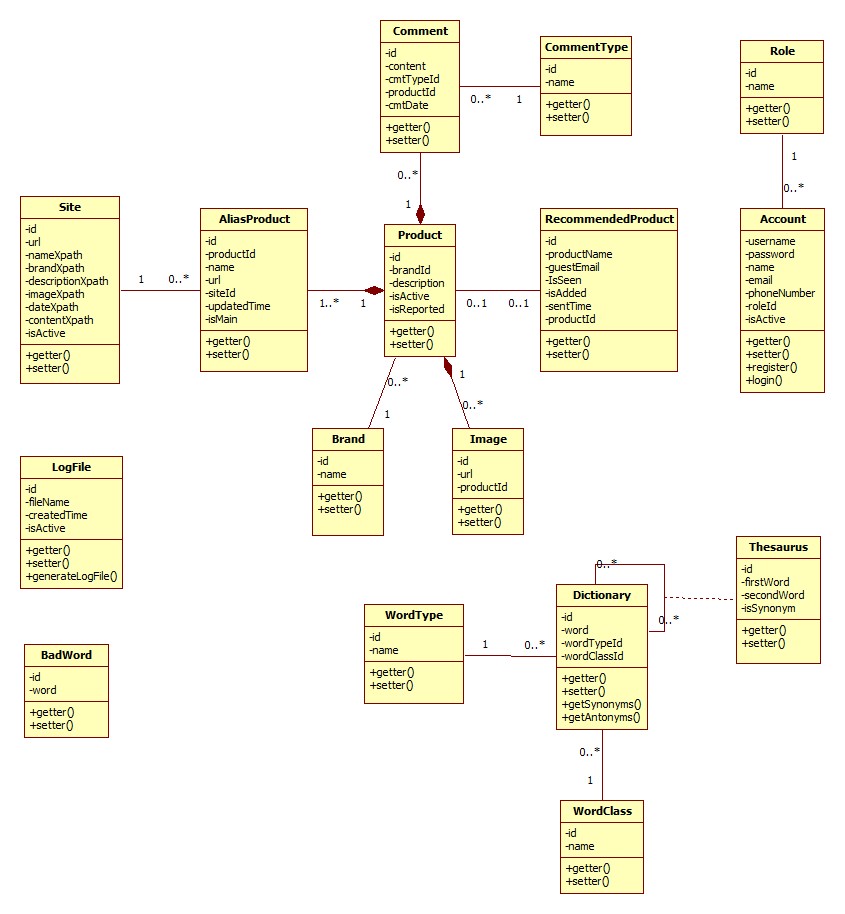


Figure 3: Class Diagram

### Class Diagram Explanation

#### Site

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each site |
| url | string | Public | Link to the site |
| nameXpath | string | Public | Xpath that defines product’s name location in the website |
| brandXpath | string | Public | Xpath that defines product’s brand location in the website |
| descriptionXpath | string | Public | Xpath that defines product’s description location in the website |
| imageXpath | string | Public | Xpath that defines product’s image location in the website |
| dateXpath | string | Public | Xpath that defines product’s submitted date location in the wesite |
| contentXpath | string | Public | Xpath that defines product’s content location in the website |
| siteId | integer | Public | Id of site |
| isActive | boolean | Public | Status of site |

#### AliasProduct

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each alias product |
| productId | integer | Public | Id of product |
| name | string | Public | Name of alias product |
| url | string | Public | Link to website contains alias product |
| siteId | int | Public | Id of site |
| updatedTime | datetime | Public | Day that alias product is updated |
| isMain | boolean | Public | Use to check for main product |

#### Product

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each product |
| brandId | integer | Public | Id of brand |
| description | string | Public | Description of product |
| isActive | boolean | Public | Status of product |
| isReported | int | Public | State how many times the product is reported |

#### Image

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each image |
| url | string | Public | Link to website contains image |
| productId | integer | Public | Id of product |

#### Comment

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each comment |
| content | string | Public | Content of comment |
| cmtTypeId | integer | Public | Id of comment type |
| cmtDate | datetime | Public | The day the comment was submitted |

#### CommentType

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each comment type |
| name | string | Public | Name of comment type |

#### Brand

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each brand |
| name | string | Public | Name of Brand |

#### RecommendedProduct

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each recommended product |
| productName | string | Public | Name of recommended product |
| guestEmail | string | Public | Email of guest |
| isSeen | Boolean | Public | Indicate whether staff have seen the request or not |
| isAdded | Boolean | Public | Indicate whether staff have executed request or not |
| sentTime | datetime | Public | Day the recommended product is submitted |
| productId | integer | Public | Id of Product |

#### Role

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each role |
| name | string | Public | Name of comment role |

#### Account

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | string | Public | Unique username of each account |
| password | string | Public | Password used to access each account |
| name | string | Public | Name of account owner |
| email | string | Public | Email of account owner |
| phoneNumber | string | Public | Phone number of account owner |
| roleId | integer | Public | Id of role |
| isActive | Boolean | Public | Status of account |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| register | Boolean | Public | Register new account |
| login | Boolean | Public | Login to the system |

#### Dictionary

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each word in dictionary |
| word | string | Public | Word of dictionary |
| wordTypeId | integer | Public | Id of word type |
| wordClassId | integer | Public | Id of word class |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getSynonyms | Dictionary | Public | Get word’s synonyms |
| getAntonyms | Dictionary | Public | Get word’s antonyms |

#### WordType

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each word type |
| name | string | Public | Name of word type |

#### Thesaurus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each thesaurus |
| firstWord | integer | Public | Id of word which role is main word |
| secondWord | integer | Public | Id of word which role is synonym or antonym of main word |
| isSynonym | Boolean | Public | Indicate whether the relationship of 2 words is synonym or not |

#### WordClass

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each word class |
| name | string | Public | Name of word class |

#### BadWord

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each bad word |
| word | string | Public | word of bad word |

#### LogFile

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | integer | Public | Unique identifier of each log file |
| filename | string | Public | Name of log file |
| createdTime | datetime | Public | Date the log file is created |
| isActive | Boolean | Public | Status of log file |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| generateLogFile | Boolean | Public | Generate log file for each time system runs parser |

### Sequence Diagram

#### Force Parse Data

Figure 4: Force Parse Data Sequence Diagram

#### Import Excel

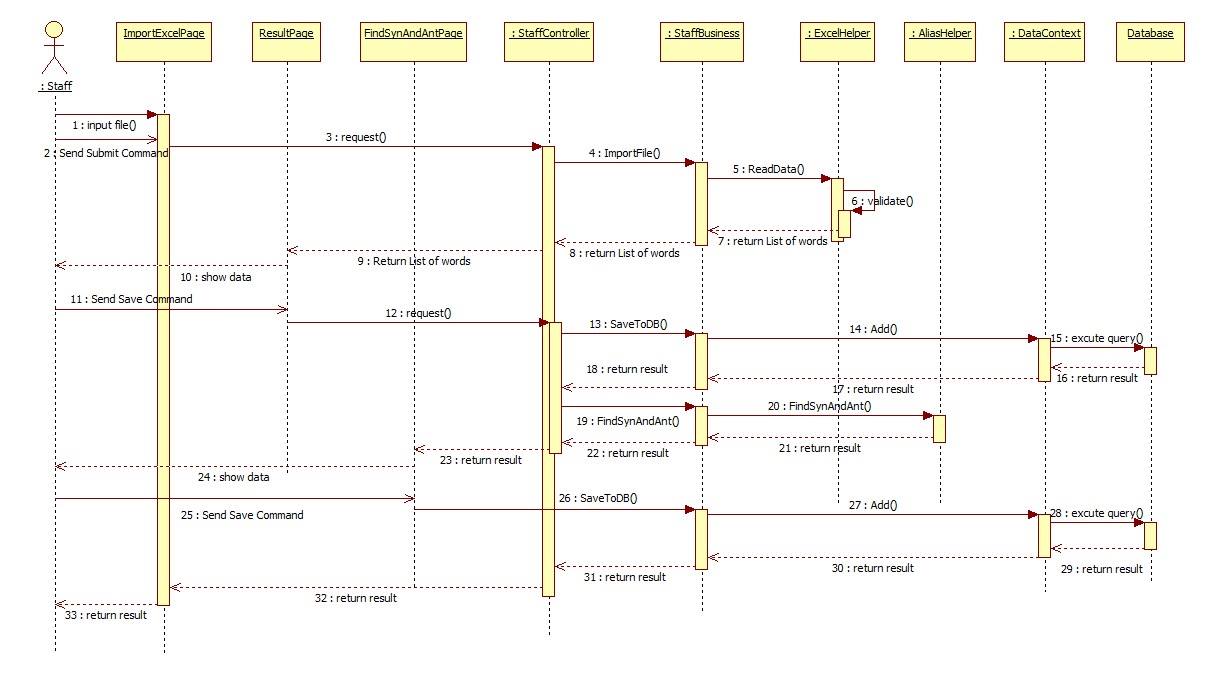
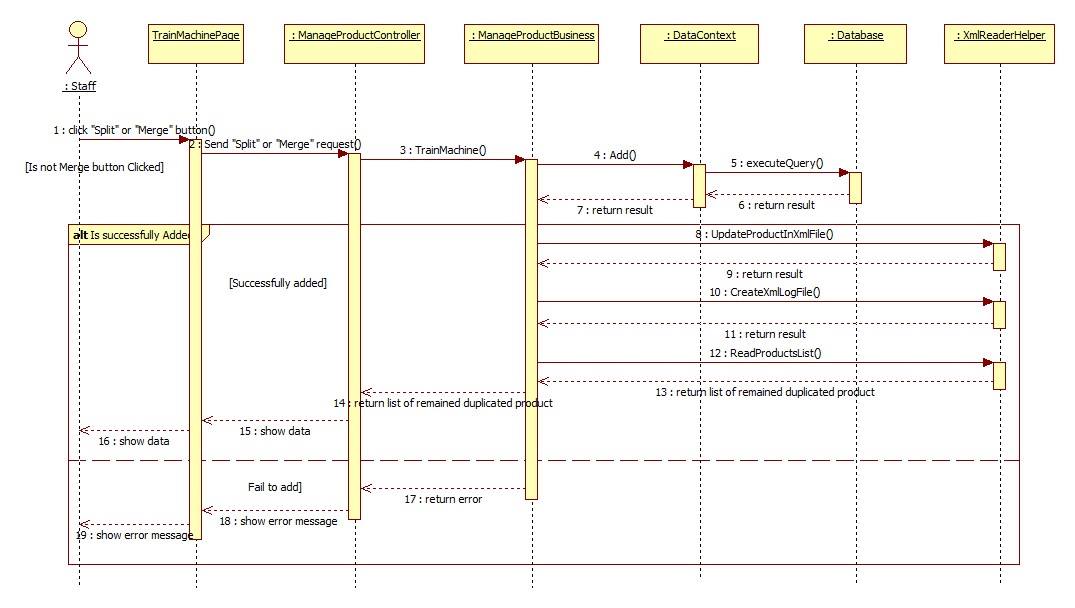


Figure 5: Import Excel Sequence Diagram

#### Search Laptop

#### Import Dictionary

#### Train Machine



## User Interface Design

## Database Design

### Logical Diagram

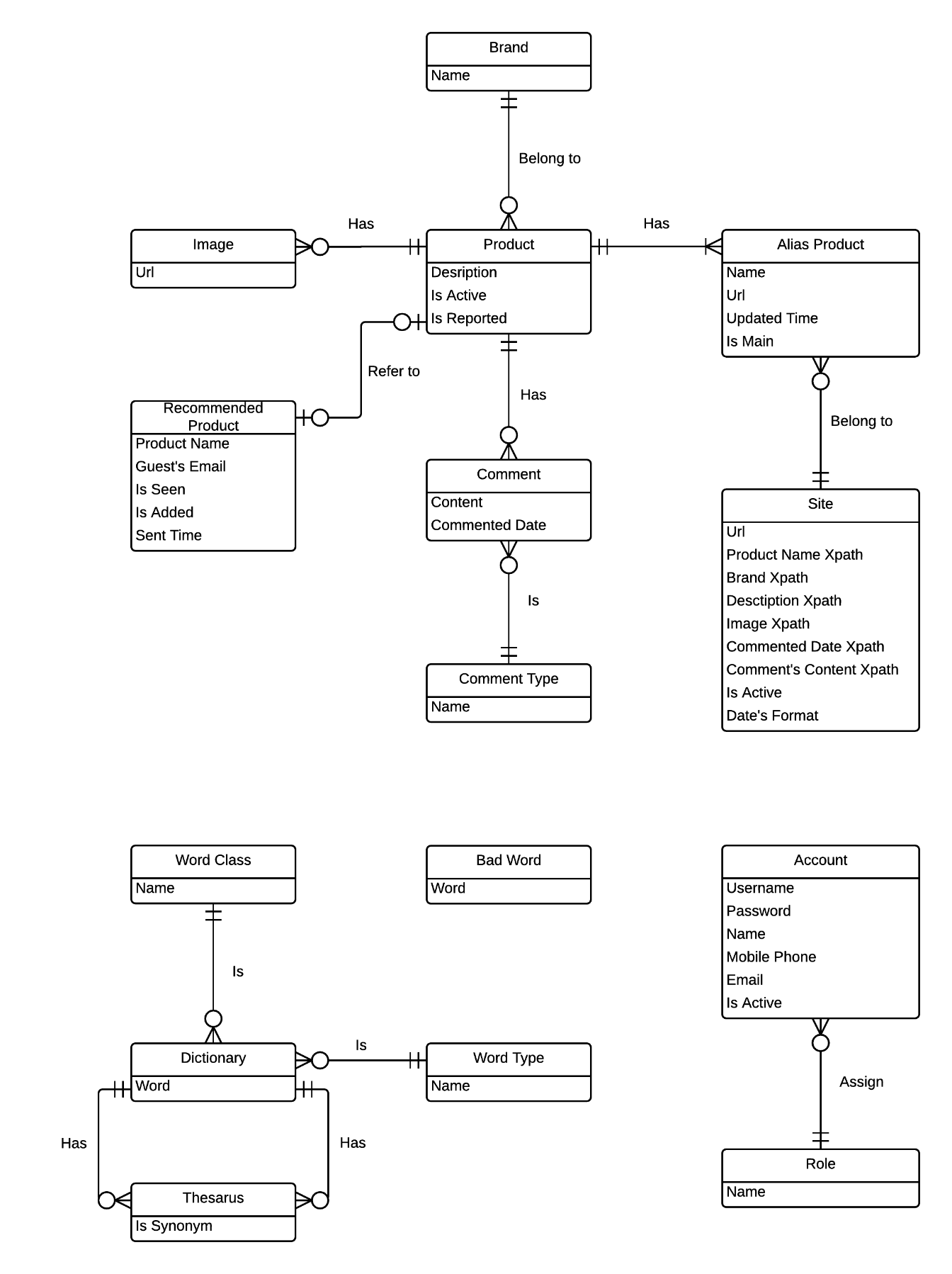


Figure 27: Logical Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe contents of all entities** | |
| **Entity Name** | **Description** |
| Image | Describe all image links that product is included. |
| Product | Describe all products in the system. |
| Alias Product | Describe all alias names of product in the system. |
| Recommend Product | Describe all products that recommend by user. |
| Brand | Describe all brands that product is included. |
| Comment | Describe all comments of products. |
| Comment Type | Describe all types of comments. |
| Site | Describe all site parse data. |
| Dictionary | Describe all words |
| Word Class | Describe all classes of words |
| Bad Word | Describe all bad words |
| Word Type | Describe all types of words |
| Thesaurus | Describe all words’ synonyms/antonyms |
| Account | Describe all user accounts |
| Role | Describe all accounts’ roles |

Table 1: Entity Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| Image | Url | Describe all image link that product is included. | nvarchar(255) | No |
| Product | Description | Describe all descriptions of all products | nvarchar(max) | Yes |
| Is Active | Describe whether products are active or not | bit | No |
| Is Reported | Describe whether products are reported or not | int | Yes |
| Alias Product | Name | Describe names of alias products | nvarchar(200) | No |
| Url | Describe links to websites contain alias products | nvarchar(max) | No |
| Updated Time | Describe time alias products are updated | datetime | Yes |
| Is Main | Describe whether alias products are main products or not | bit | No |
| Recommend Product | Product’s Name | Describe names of recommended products | nvarchar(200) | No |
| Guest’s Email | Describe emails of guests who suggest the products | nvarchar(50) | No |
| Is Seen | Describe whether staff have seen the guests’ requests or not | bit | No |
| Is Added | Describe whether staff have collected the recommend products information or not | bit | No |
| Sent Time | Describe the time the guests’ requests were sent | datetime | No |
| Brand | Name | Describe names of brands | nvarchar(50) | No |
| Comment | Content | Describe contents of comments | nvarchar(max) | No |
| Comment Date | Describe date of the comments | datetime | No |
| Comment Type | Name | Describe name of types of comments. | nvarchar(15) | No |
| Site | Url | Describe sites’ url information | nvarchar(200) | No |
| Product Name Xpath | Describe Xpath that define products names in the websites | nvarchar(255) | No |
| Brand Xpath | Describe Xpath that define products brands in the websites | nvarchar(255) | No |
| Description Xpath | Describe Xpath that define products descriptions in the websites | nvarchar(255) | No |
| Image Xpath | Describe Xpath that define products images in the websites | nvarchar(255) | No |
| Commented Date Xpath | Describe Xpath that define products commented date in the website | nvarchar(255) | No |
| Comment’s Content Xpath | Describe Xpath that define products comments content in the website | nvarchar(255) | No |
| Is Active | Describe whether sites are active or not | bit | No |
| Date’s Format | Describe date’s format in sites. | nvarchar(30) | No |
| Dictionary | Word | Describe words of dictionary | nvarchar(20) | No |
| Word Type | Name | Describe names of word type. | nvarchar(20) | No |
| Word Class | Name | Describe names of word class. | nvarchar(20) | No |
| Bad Word | Word | Describe bad words. | nvarchar(20) | No |
| Thesaurus | Is Synonyms | Describe whether words in Dictionary are synonyms or not | bit | No |
| Account | Username | Describe username information | varchar(35) | No |
| Password | Describe password information | varchar(35) | No |
| Name | Describe account owner names | nvarchar(35) | Yes |
| Mobile Phone | Describe account owner mobile phone numbers | varchar(15) | Yes |
| Email | Describe account owner Emails | nvarchar(50) | No |
| Is Active | Describe whether account owner is active or not | bit | No |

Table 2: Entities’ Attributes Dictionary

## Algorithms

### Analyze Comment

#### Define Problem

Given a sentence then the system will check whether that sentence has positive or negative or neutral meaning

#### Requirement

All sentence must have correct grammar. They must also have sufficient length. Moreover, all sentences that contains impolite or meaningless words will not be counted.

#### Solution

* Manually prepare 10 lists of lower-cased words by reading first 100 comments:
* A list contains all words which meanings are totally Pros
* A list contains all adjectives and adverbs which meanings are Pros (these words’ positive meaning is not as strong as words in “Totally Pros” list)
* A list contains all adjectives and adverbs which meanings are Neutral
* A list contains all adjectives and adverbs which meanings are Cons
* A list contains all adjectives and adverbs which meanings are totally Cons (these words’ negative meaning is not as strong as words in “Totally Cons” list)
* A list contains all nouns and verbs which meanings are Pros (these words’ positive meaning is not as strong as words in “Totally Pros” list)
* A list contains all nouns and verbs which meanings are Neutral
* A list contains all nouns and verbs which meanings are Cons (these words’ negative meaning is not as strong as words in “Totally Cons” list)
* A list contains all nouns and verbs which meanings are totally Cons
* A list of negative words such as not, no, do not, does not …
* Lower case the whole sentence and break it into a list of words, then lower case all the words.
* With a list of words, we will check how many words of that list belong to the 10 lists above, then we divide into these cases:
* Case 1: List of words contains word(s) which belong to “Totally Pros” word list: In this case, the sentence will be Positive sentence.
* Case 2: List of words contains word(s) which belong to “Totally Pros” word list but it also contains word(s) which belong to “Negative” word list: In this case, the sentence will be Negative sentence.
* Case 3: We will check the adjectives and adverbs fist. So if list of words contains adjectives, adverbs and belongs to adjectives/adverbs’ “Pros”, “Cons” or “Neutral” lists, we will have these sub-cases:
* If there are more “Pros” words than “Cons” words 🡪 The sentence is positive (1)
* If there are more “Cons” words than “Pros” words 🡪 The sentence is negative (2)
* With those 2 above sub-cases, if there are words belong to “Negative” list, then the sentence will be negative with sub-case (1) and positive with sub-case (2)
* If the sentence contains no adjectives/adverbs that belongs to “Pros”, “Cons” and has words belong to “Neutral”, that sentence is neutral. If the sentence has same number of “Pros” and “Cons” adjectives/adverbs and has no “Neutral” adjectives/adverbs, we will check in Case 4.
* If there is no “Pros”, “Cons” and “Neutral” adjectives/adverbs in that sentence, we will check in Case 4.
* Case 4: After checking for adjectives, adverbs, we will check verbs and nouns in that sentence. We have these sub-cases
* If there are more “Pros” words than “Cons” words 🡪 The sentence is positive (1)
* If there are more “Cons” words than “Pros” words 🡪 The sentence is negative (2)
* With those 2 above sub-cases, if there are words belong to “Negative” list, then the sentence will be negative with sub-case (1) and positive with sub-case (2)
* If the sentence contains no verbs/nouns that belongs to “Pros”, “Cons” and has words belong to “Neutral”, that sentence is neutral. Similarly, if the numbers of verbs/nouns belongs to “Pros” and “Cons” are the same, that sentence is Neutral
* If there is no “Pros”, “Cons” and “Neutral” verbs in that sentence, it will be unidentified and will be decided later by staff

#### Example

Giving the sentence: “*This Mac is fast, and combined with Mavericks I am now getting some great battery life.”*

* Assume that we already have “Totally Pros” words list which contains “fast” and “great”.
* Lower case the whole sentence:

+ This Mac is fast, and combined with Mavericks I am now getting some great battery life.🡪 this mac is fast, and combined with mavericks i am now getting some great battery life.

* Split sentence into list words:

+ this mac is fast, and combined with mavericks i am now getting some great battery life 🡪 {this, mac, is, fast, and, combined, with, mavericks, i, am, now, getting, some, great, battery, life}

* We will check for “Totally Pros” words first. In this case, we have 2 words: fast and great. These 2 words belongs to “Totally Pro” list, so this sentence is Positive.

### String Comparison

#### Define Problem

Given 2 strings. Calculate their matching percent.

#### Requirement

* A robustness to changes of word order: two strings which contain the same words, but in a different order, should be recognized as being similar.
* Language independence: the algorithm should work not only in English, but in many different languages.

#### Solution

* If a string contains many words, break it into a list of words.
* For each word, we find out how many adjacent character pairs are contained in it.
* Create a function *pairs(s)* which returns a list of adjacent character pairs of string *s*.
* Then, we use below formula to calculate matching percent.

#### Example

Calculate the matching percent of 2 strings: France and French.

* Upper case 2 strings:
  + MacBook Air 2015 MACBOOK AIR 2015.
  + MacBook Air 2015 Retina MACBOOK AIR 2015 RETINA
* Break string into list of adjacent character pairs:
  + MACBOOK AIR 2015
  + MACBOOK AIR 2015 RETINA
* Calculate its matching percent.