

E-COMMERCE WEBSITE AND MOBILE APPLICATION  
WITH SOCIAL FEATURES



E077973

KRISADA KAEWJUNCHAI

KANPITCHA INAUIM

ISARAPORN KOSIDBENJAPOL

เลขหมู่.....  
เลขทะเบียน **077973**  
วัน,เดือน,ปี...จ.ค. 2559

b.12808167  
i.....

BACHELOR OF ENGINEERING PROGRAM IN SOFTWARE ENGINEERING  
INTERNATIONAL COLLEGE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
2013

**Thesis – Academic Year 2013**

B.Eng. in Software Engineering

International College, King Mongkut's Institute of Technology Ladkrabang

**Title:** E-commerce Website and Mobile Application with Social Features

**Authors:**

1. Mr. Krisada Kaewjunchai      Student ID: 53090001
2. Ms. Kanpitcha Inauim      Student ID: 53090002
3. Ms. Isaraporn Kosidbenjapol      Student ID: 53090031

Approved for submission :



(Dr.Natthapong Jungteerapanich)

INTERNATIONAL COLLEGE

Date 29/11/2014

# **E-commerce Website and Mobile Application**

## **with Social Features**

Mr. Krisada Kaewjunchai 53090001

Ms. Kanpitcha Inauim 53090002

Ms. Isaraporn Kosidbenjapol 53090031

Dr. Natthapong Jungteerapanich Advisor

Academic Year 2013

### **ABSTRACT**

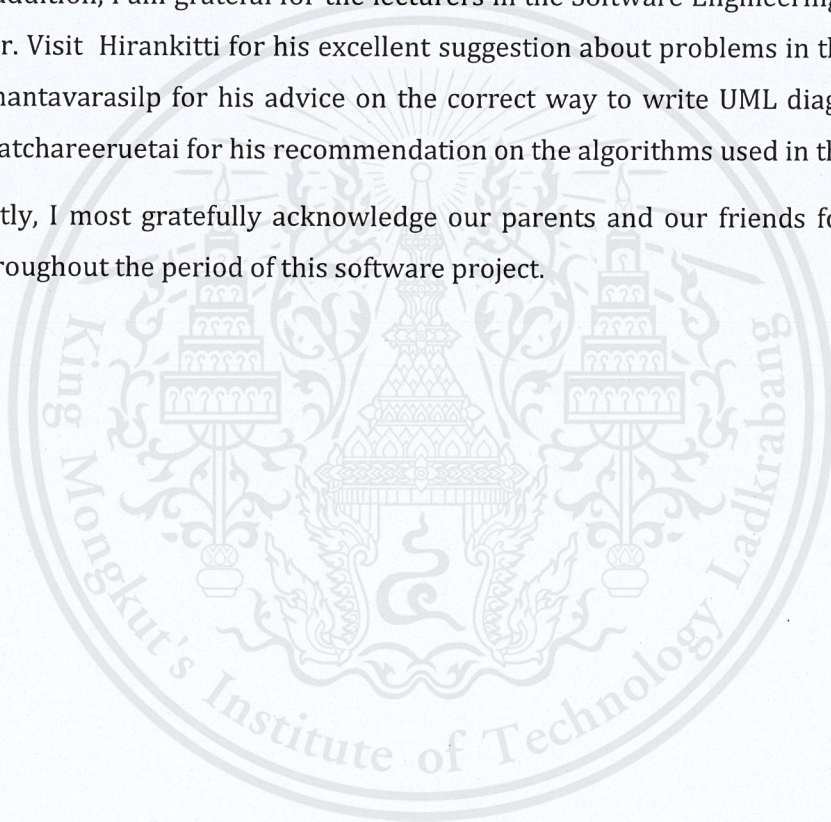
The primary aim of this project is to explore the integration of e-commerce with online social networking and develop a prototype of an online marketplace system with a web-based front end and a mobile application. Within this project, we studied various techniques for developing a recommender system commonly used on e-commerce websites, including collaborative recommendation techniques and content-based recommendation techniques. We adapted and extended such techniques by utilizing relationships among the users and the information obtained from the users' profiles on Facebook to create an enhanced recommender system which not only provides recommendations on products to the user but also suggests "friends", the other users who are expected to share similar interest with the user. Regarding the implementation, the server side of the software was developed using the Django framework, while the web-based front end made extensive uses of HTML 5 features (including web push notifications). The mobile application was developed for the iOS platform.

# Acknowledgements

First of all, we would like to express my sincere thanks to our project advisor, Dr. Natthapong Jungteerapanich for his invaluable help and constant encouragement throughout the course of this software project. We are most grateful for his teaching and advice, not only the project methodologies but also many other methodologies in life. I would not have achieved this far and this report would not have been completed without all the support that I have always received from him.

In addition, I am grateful for the lecturers in the Software Engineering program: Asst.Prof.Dr. Visit Hirankitti for his excellent suggestion about problems in this project, Dr.Isara Anantavarasilp for his advice on the correct way to write UML diagrams, and Dr.Ukrit Watchareeruetai for his recommendation on the algorithms used in this project.

Lastly, I most gratefully acknowledge our parents and our friends for all their support throughout the period of this software project.



# Table of Contents

<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 MOTIVATION.....	1
1.2 OBJECTIVES.....	1
1.3 SCOPE OF WORK.....	2
1.4 CONTRIBUTIONS.....	2
1.5 PROCEDURE.....	2
1.6 STRUCTURE OF THIS THESIS.....	4
<b>CHAPTER 2 PROBLEM DESCRIPTION AND RELATED WORK.....</b>	<b>5</b>
2.1 PROBLEM DESCRIPTION.....	5
2.2 REVIEW OF RELATED WORKS.....	5
<b>CHAPTER 3 BACKGROUND KNOWLEDGE.....</b>	<b>11</b>
3.1 COLLABORATIVE RECOMMENDATION.....	11
3.2 CONTENT-BASED RECOMMENDATIONS.....	13
<b>CHAPTER 4 REQUIREMENT ANALYSIS.....</b>	<b>16</b>
4.1 REQUIREMENTS FOR THE WEB SITE.....	16
4.2 REQUIREMENTS FOR IOS APPLICATION.....	17
4.3 USE CASE DIAGRAM.....	18
<b>CHAPTER 5 SOFTWARE DESIGN.....</b>	<b>30</b>
5.1 SYSTEM ARCHITECTURE.....	30
5.2 ER DIAGRAM.....	31
5.3 CLASS DIAGRAM FOR THE IOS APPLICATION.....	33
<b>CHAPTER 6 DEVELOPMENT.....</b>	<b>34</b>
6.1 TECHNIQUES.....	34
6.2 IMPLEMENTATION.....	43
<b>CHAPTER 7 RESULTS.....</b>	<b>45</b>
7.1 WEBSITE.....	45
7.2 MOBILE APPLICATION.....	49
<b>CHAPTER 8.....</b>	<b>55</b>
8.1 SUMMARY.....	55
8.2 PROBLEMS AND OBSTACLES.....	55

8.3 FURTHER WORK ..... 55

REFERENCE ..... 57



# Table of Figures

Figure 2.1: eBay website.....	5
Figure 2.2: Product recommendation on eBay.....	6
Figure 2.3: Amazon website .....	6
Figure 2.4: Product rating on Amazon.com .....	7
Figure 2.5: Wanelo website .....	7
Figure 2.6: User profile on Wanelo.com.....	8
Figure 2.7: Bahtor.com website.....	8
Figure 2.8: User comment on the Bahtor website.....	9
Figure 2.9: Lazada.com website.....	9
Figure 2.10: Sharing products through the user's Facebook or Google Plus accounts.....	10
Figure 3.1: Basic concept of collaborative recommendation .....	11
Figure 3.2: Comparing Alice with two other users .....	13
Figure 3.3: Basic concept of content-based recommendation.....	14
Figure 4.1: Use case diagram .....	18
Figure 4.2: Use case diagram for iOS application .....	26
Figure 5.1: System architecture.....	30
Figure 5.2: ER diagram.....	32
Figure 5.3: Class diagram for iOS application .....	33
Figure 6.1: The product's attributes to be provided when creating a new product.....	34
Figure 6.2: New user creation.....	35
Figure 6.3: Adding categories.....	36
Figure 6.4: Adding relative categories.....	36
Figure 6.5: Categories in our website.....	37

Figure 6.6: Sample code of collaborative recommendation.....	39
Figure 6.7: To illustrate customer's behavior.....	40
Figure 6.8: Sample code of content-based recommendation.....	42
Figure 6.9: Sample code of social recommendation.....	42
Figure 6.10: Back-end system.....	43
Figure 7.1: Product page.....	45
Figure 7.2: Recommendation function.....	46
Figure 7.3: Detail of product page.....	47
Figure 7.4: Notification and friend request.....	47
Figure 7.5: Menu bar.....	48
Figure 7.6: Social feature and buy function.....	48
Figure 7.7: Recommendation function.....	48
Figure 7.8: Home page.....	49
Figure 7.9: Register page.....	49
Figure 7.10: Log in page.....	50
Figure 7.11: Log in by using Facebook.....	50
Figure 7.12: Product page.....	51
Figure 7.13: List of products.....	51
Figure 7.14: List of products.....	52
Figure 7.15: List of products.....	52
Figure 7.16: More pages.....	53
Figure 7.17: Search pages.....	53
Figure 7.18: Detail of product page.....	54

# Chapter 1

## Introduction

### 1.1 Motivation

Nowadays, there are a large number of e-commerce websites, for example, E-bay, Amazon, etc. The popularity of social networking sites, such as Facebook, Instagram, etc., has steadily increased over the past few years. Many companies have been utilizing social network as a channel for communication with their customers. We believe that social networking and online shopping can be integrated more tightly and more extensively. By doing so, we believe online shopping can be made a joyous social activity.

### 1.2 Objectives

The primary objective of this project is to explore the integration of e-commerce with online social networking and develop a prototype of an online marketplace system with a web-based front end and a mobile application. As in most e-commerce websites, for our system to be effective, it must be able to help the user search for the products he/she is looking for, as well as provide the user with recommendations on related products, which the user might be interested in.

Therefore, as another objective of this project, we aim to study various techniques for developing a recommender system and extend the existing techniques by utilizing the relationships among the users and the information obtained from the users' profiles on social networks to create an enhanced recommender system which not only provides recommendations on products but also bringing the users who share similar interest together.

### **1.3 Scope of work**

The scope of this project can be summarized as follows:

- Study the theory and techniques of recommender systems.
- Develop an online marketplace system which incorporates social features using the Django framework and develop a web-based front end and a mobile application on the iOS platform.

### **1.4 Contributions**

In this project, we have developed a prototype of an online marketplace system, which incorporates social features. Our work demonstrates various ideas and possibilities in combining social networking with online shopping. We hope that the prototype we have developed will provide an insight into the advantages and the disadvantages of social features in e-commerce.

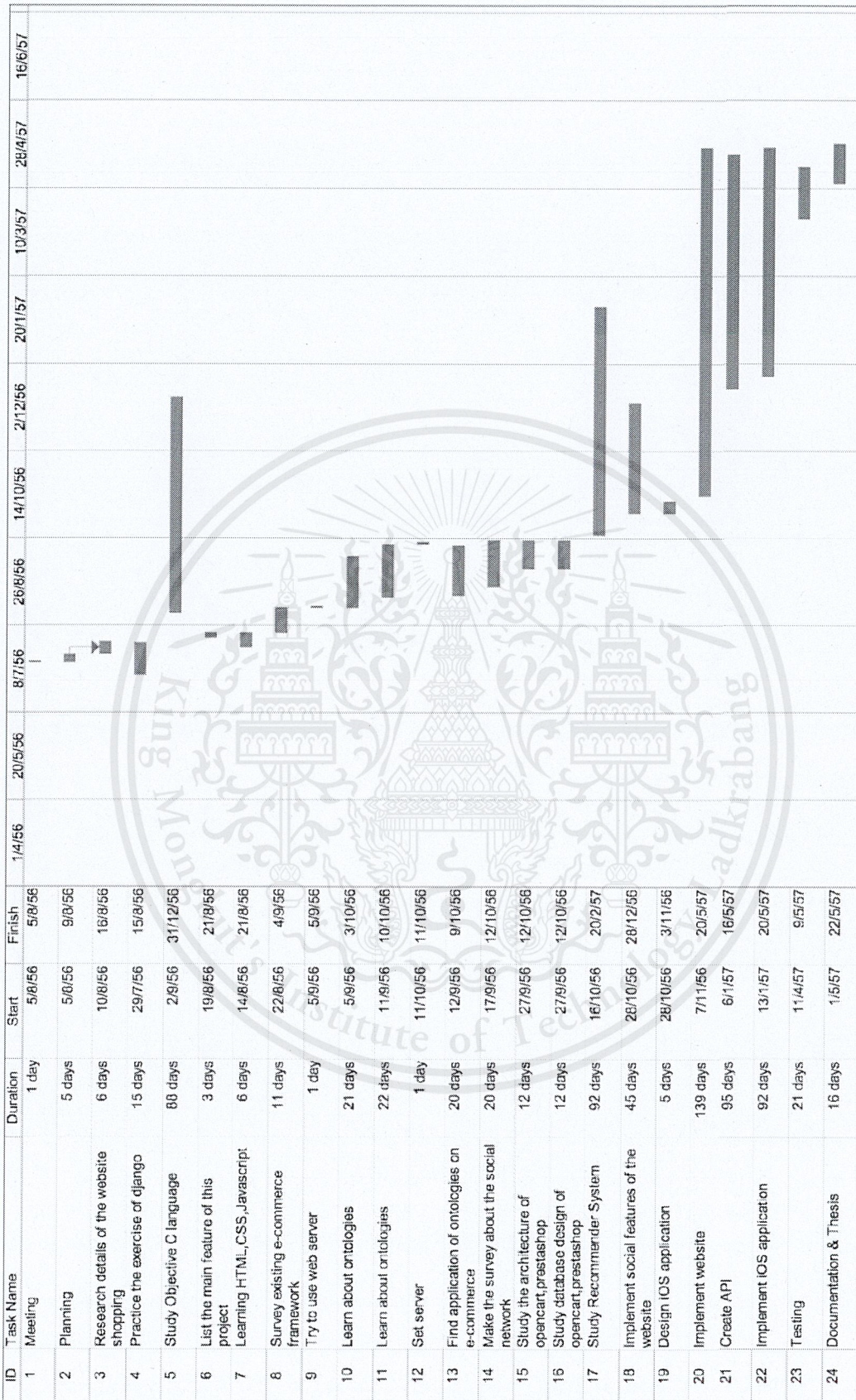
### **1.5 Procedure**

We separate the work into five phases.

- 1<sup>st</sup> phase: Plan the project.
- 2<sup>nd</sup> phase: Research on existing e-commerce and social networking websites.
- 3<sup>rd</sup> phase: Learn Objective-C, Django, and web programming.
- 4<sup>th</sup> phase: Analyze and design the architecture of the system and each component of the system, namely, the back end, the web-based front end and the mobile application.
- 5<sup>th</sup> phase: Implement the system.

## 1.5.1 Gantt chat

S



## 1.6 Structure of this thesis

Chapter 2 of this thesis describes the problem description and a review of related work.

Chapter 3 explains the theory and other knowledge that are relevant to the project.

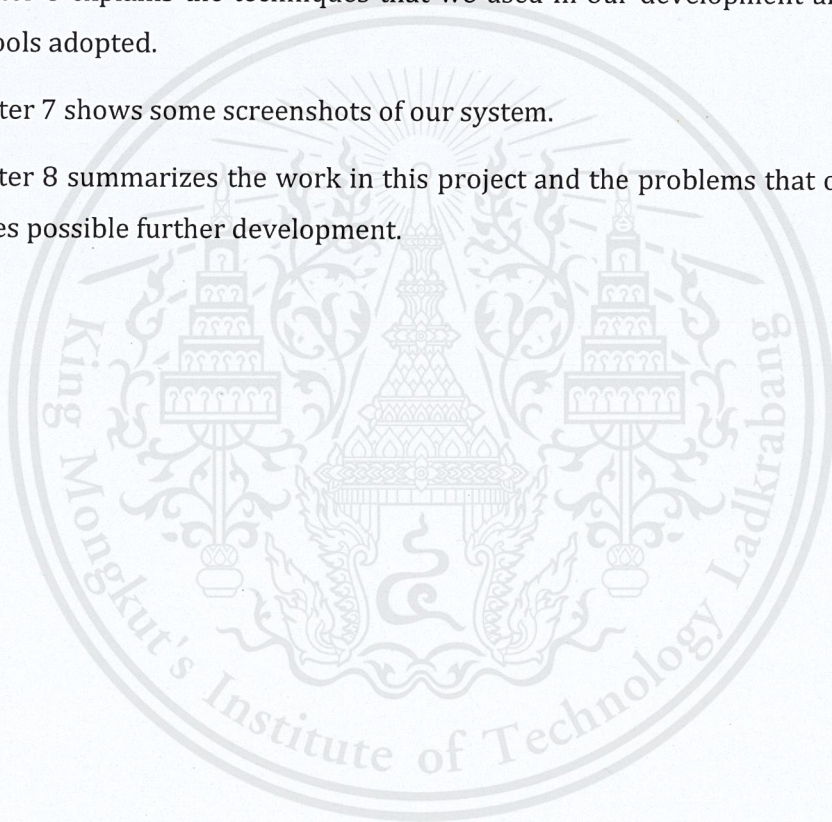
Chapter 4 explains the overall requirements and the architecture of the system.

Chapter 5 focuses on the system design and the design of each component of the system including the ER diagram and the class diagram.

Chapter 6 explains the techniques that we used in our development and also the software tools adopted.

Chapter 7 shows some screenshots of our system.

Chapter 8 summarizes the work in this project and the problems that occurs, and also outlines possible further development.



# Chapter 2

## Problem Description and Related Work

### 2.1 Problem description

The popularity of social networking and e-commerce sites has been steadily increasing over the past few years. However, the current e-commerce websites do not provide functionalities that allow users to conveniently communicate with each other. The social networking websites do not aim at facilitating online shopping. The goal of this project is to eliminate these problems by developing a web-based social e-commerce system which offers a unified experience in online shopping and social networking.

### 2.2 Review of related works

Currently, there are countlessly many e-commerce websites. To gain understanding of the features and the limitations of those websites, we selected and studied some popular e-commerce websites. Below is the summary of our study.

#### 2.2.1 eBay.com

eBay is the most popular online auctioning website. This site is a place for the buyer or seller to buy or sell products. Figure 2.1 shows the home page of the eBay website.

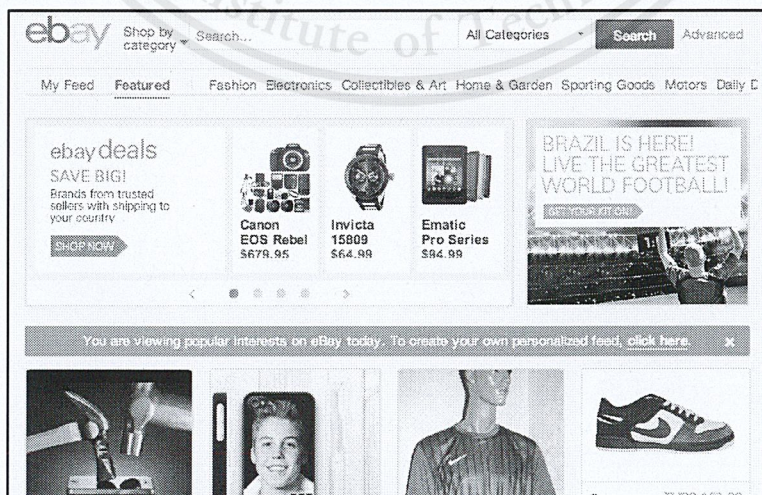


Figure 2.1: eBay website

The site provides recommendations for the products that the users are most likely to purchase. Further, the site allows users to share a product on social network. Figure 2.2 illustrates the recommendation feature in website.

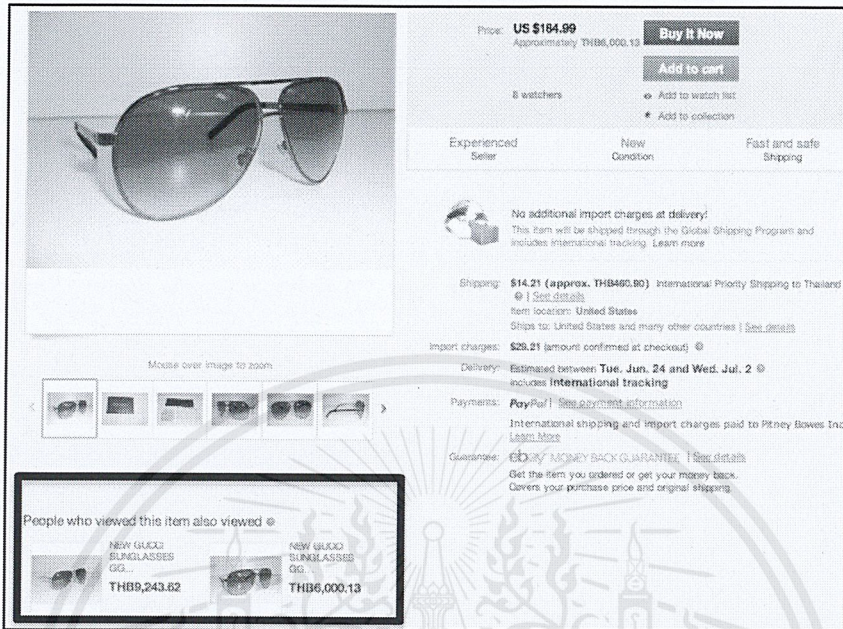


Figure 2.2: Product recommendation on eBay

## 2.2.2 Amazon.com

Amazon.com is one of the most popular e-commerce websites. Figure 2.3 shows the homepage of Amazon website.



Figure 2.3: Amazon website

The site provides features such as rating of the product, product review, product recommendation and sharing on social network. Figure 2.4 illustrates the rating of a product.



Figure 2.4: Product rating on Amazon.com

### 2.2.3 Wanelo.com

Launched in 2012, Wanelo.com is the new e-commerce website. The website boasts an attractive design a la the popular image sharing website pinterest.com. Figure 2.5 shows the home page of Wanelo site.

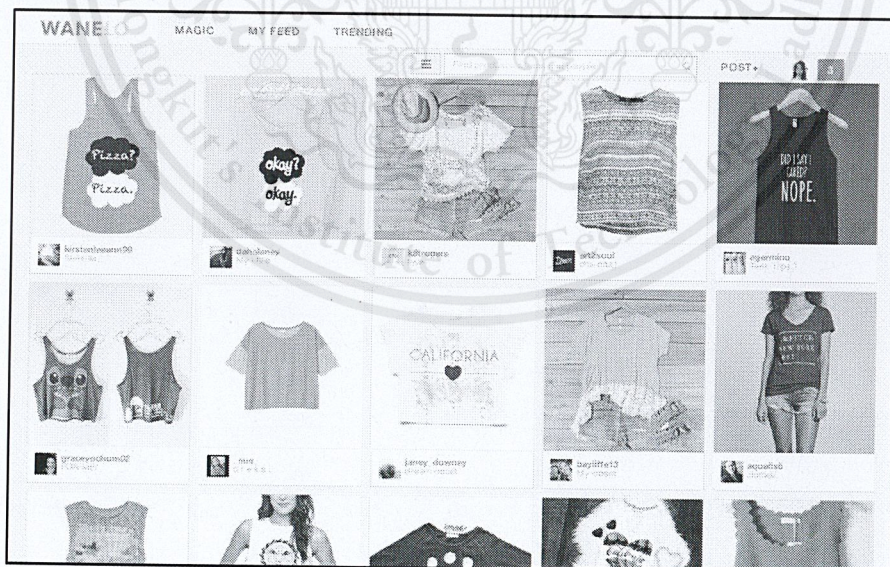


Figure 2.5: Wanelo.com website

The site provides rich social features such as profile, comment wall, rating of the product, product recommendation and sharing on social network. Figure 2.6 illustrates the profile of a user.

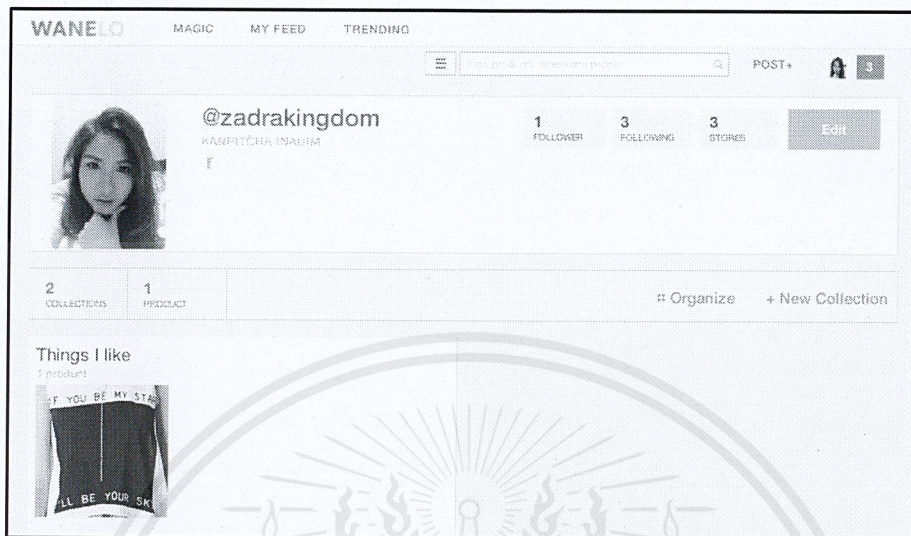


Figure 2.6: User profile on Wanelo.com

## 2.2.4 Bahtor.com

Bahtor.com is a Thai e-commerce website with emphasis on social features. The site acts like a marketplace where users can buy and /or sell their products to other users. Figure 2.7 shows the home page of Bahtor.com.



Figure 2.7: Bahtor.com website

The website provides social features through the integration with facebook.com.

Figure 2.8 illustrates the comment feature on the Bahtor website.

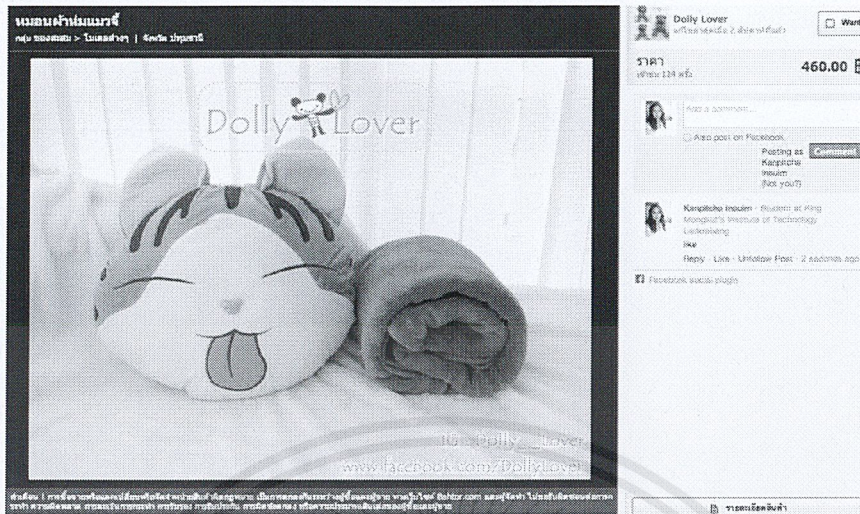


Figure 2.8: User comment on the Bahtor website

## 2.2.5 Lazada.com

Lazada.com is one of the most popular e-commerce website in Thailand. The website focuses on lowering product prices and marketing campaigns, such as vouchers and various forms of extra discounts. Figure 2.9 shows the home page of Lazada.com.



Figure 2.9: Lazada.com website

The site provides features such as product rating, product review and product sharing on Facebook and Google Plus. The website also has companion applications on iOS and Android devices. Figure 2.10 illustrates the share feature in Lazada website.



Figure 2.10: Sharing products through the user’s Facebook or Google Plus accounts

## 2.2.6 Summary

From our survey of current e-commerce websites, we can summarize the key features as described in Table 2.1.

	Product recommend	Rating of product	Review	Profile	Comment wall	Share on Social network
1.E-bay	x			x		x
2.Amazon	x	x	x	x		x
3.Wanelo	x	x		x	x	x
4.Bahtor				x	x	x
5.Lazada		x	x			x

Table 2.1: E-commerce Websites

# Chapter 3

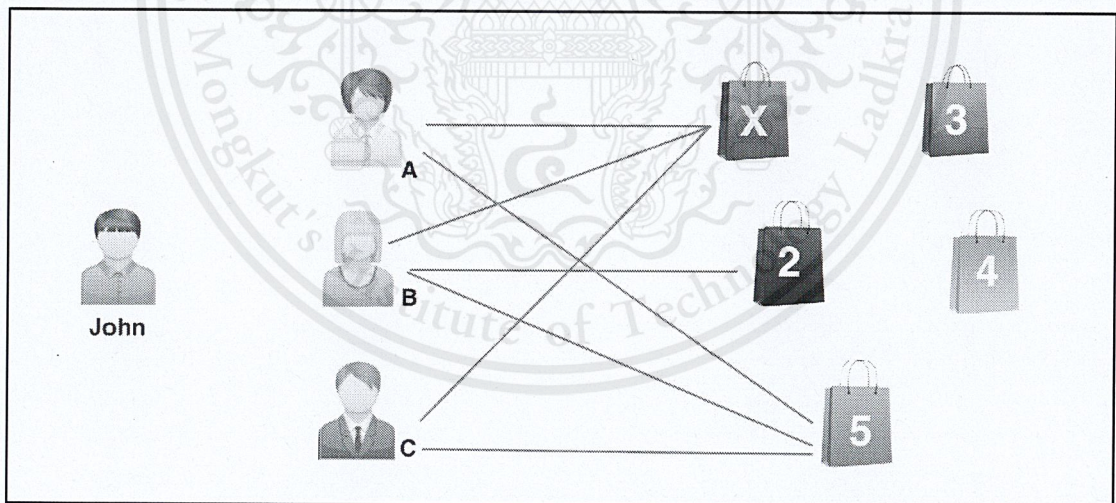
## Background Knowledge

This chapter describes the background knowledge and the theory of recommender systems that is crucial in the development of our social e-commerce website. In e-commerce, a recommender system is a software system that determines which products should be recommended to a particular visitor. There are two main techniques which are commonly used in developing a recommender system, namely, collaborative recommendation and content-based recommendation.

### 3.1 Collaborative recommendation

#### 3.1.1 Basic concept

The basic ideas of these systems are based on purchases made by similar users. The basic idea of these systems is that if users shared the same interests in the past they will also have similar interest in the future. This can be seen in Figure 3.1.



**Figure 3.1:** Basic concept of collaborative recommendation

From Figure 11, it can be seen that many customers who bought product X also bought product 5. If John is buying Product X, then he is likely to be interested to buy Product 5.

### 3.1.2 User-based nearest neighbor recommendation

Table 3.1 shows a database of product ratings by Alice and some other users. We would like to predict whether Alice will like or dislike Item 5. The basic idea is to search for a user whose taste is most similar to Alice's and then take the ratings of this group for Item 5.

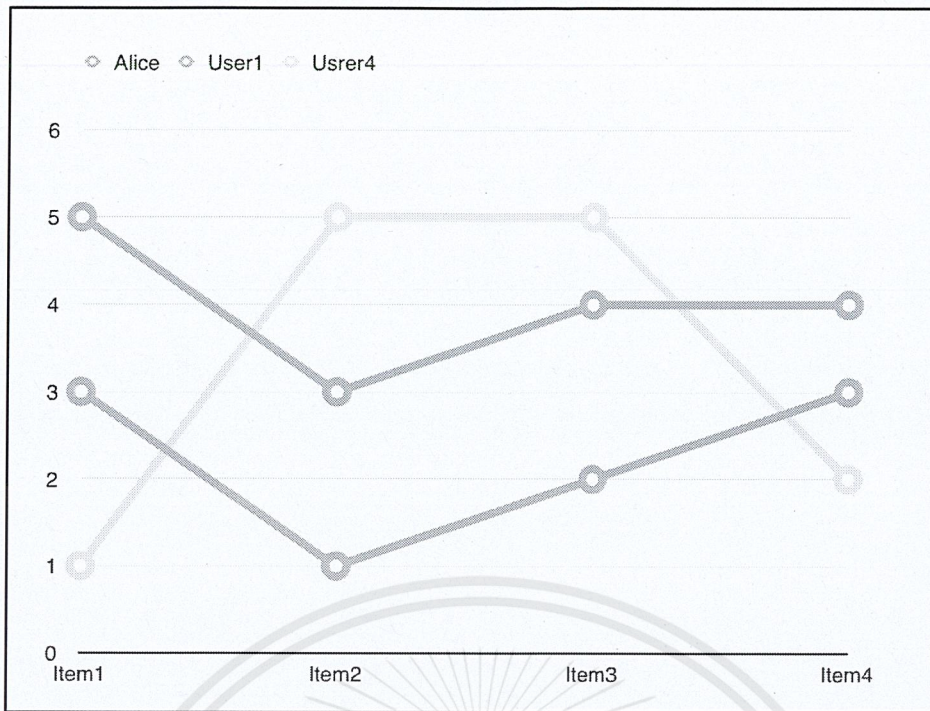
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

**Table 3.1:** Rating database for collaborative recommendation

The ratings in the above table can be displayed graphically in Figure 3.2. It is clear from this graph that Alice's preferences on products are more similar to User 1's than User 4's. We can calculate a degree of similarity of two users using Formula 1 below.

- $\text{sim}(a,b)$  denotes the degree of similarity of users  $a$  and  $b$ .
- The symbol  $r_a$  corresponds to the average rating of user  $a$ .

$$\text{sim}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}} \quad (1)$$



**Figure 3.2:** Comparing Alice with two other users

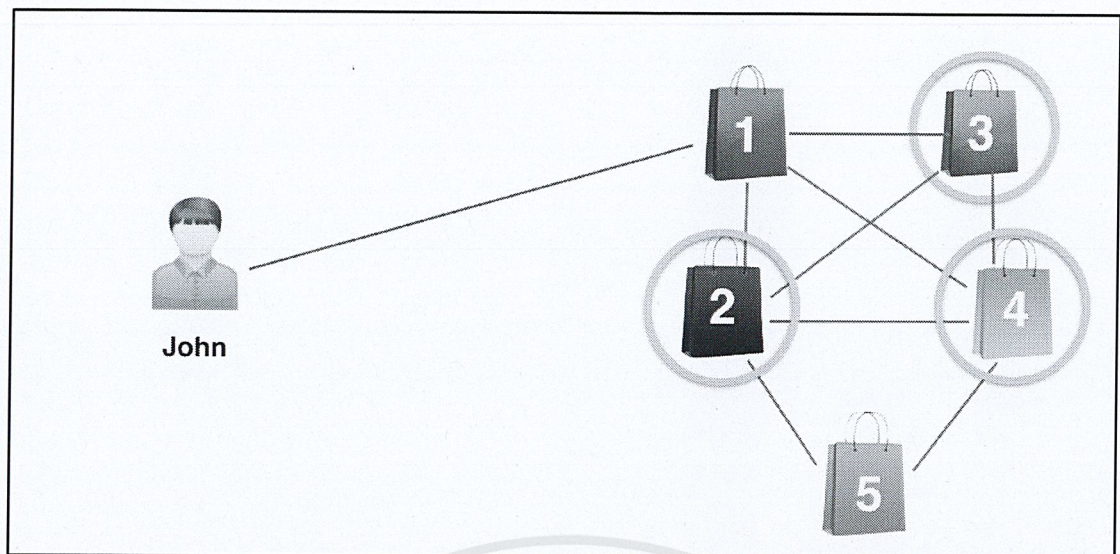
After obtaining the similarity, a possible formula for computing a prediction for rating of user  $a$  for item  $p$  is shown in Formula (2).

$$pred(a,p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a,b)} \quad (2)$$

## 3.2 Content-based recommendations

### 3.2.1 Basic concept

The basic ideas of these systems are based on the content of the item viewed. For instance, a content-based system may recommend other science fictions to users who viewed a science fiction film. This can be seen in Figure 3.3.



**Figure 3.3:** Basic concept of content-based recommendation

If John is buying Product 1, he might be interested to buy Products 2, 3 or 4 which are related to Product 1.

### 3.2.2 TF-IDF

TF-IDF stands for term frequency-inverse document frequency. Term frequency describes how often important word appears in a document. Formula (3) can be used to calculate term frequency.

$$TF(i, j) = \frac{freq(i, j)}{maxOthers(i, j)} \quad (3)$$

where

- $TF(i, j)$  is the value of keyword  $i$  in document  $j$
- $freq(i, j)$  is the absolute number of occurrences of  $i$  in  $j$ .
- $maxOthers(i, j)$  is the maximum frequency of other keyword.

Inverse document frequency is aims to reduce the weight of keywords that appear very often in all documents. Formula (4) can be used to calculate Inverse document frequency.

$$IDF(i) = \log \frac{N}{n(i)} \quad (4)$$

So in TF-IDF, The weight is a statistical measure used to evaluate how important a word is to a document in a collection.



# Chapter 4

## Requirement Analysis

### 4.1 Requirements for the web site

#### 4.1.1 For accessing to system

- The system allows guests to register.
- The system allows users to login.

#### 4.1.2 For shopping

- The system allows users to buy and sell products in the web site.
- The system allows users to browse products by categories.
- The system allows users to add and delete products and store.
- The system allows users to click “like” the product.
- The system allows users to comment about the product.
- The system allows users to create their wish list.
- The system allows users buyers to check status of products.
- The system allows sellers to change status of products.

#### 4.1.3 For search

- The system allows guests and users to search product and store.
- The system allows guests and user to search product by hashtag.
- The system allows users to search friends.

#### 4.1.4 For notification

- The system can notify the user when other people comment, buy and like the user products.
- The system can notify when the user has a friend request and message.

#### 4.1.5 For social network

- The system provides the social network functions: indicate that he/she like products, comment, adds friend, share product, select circle (group of same preference), and create their own wish list.

#### 4.1.6 For administration

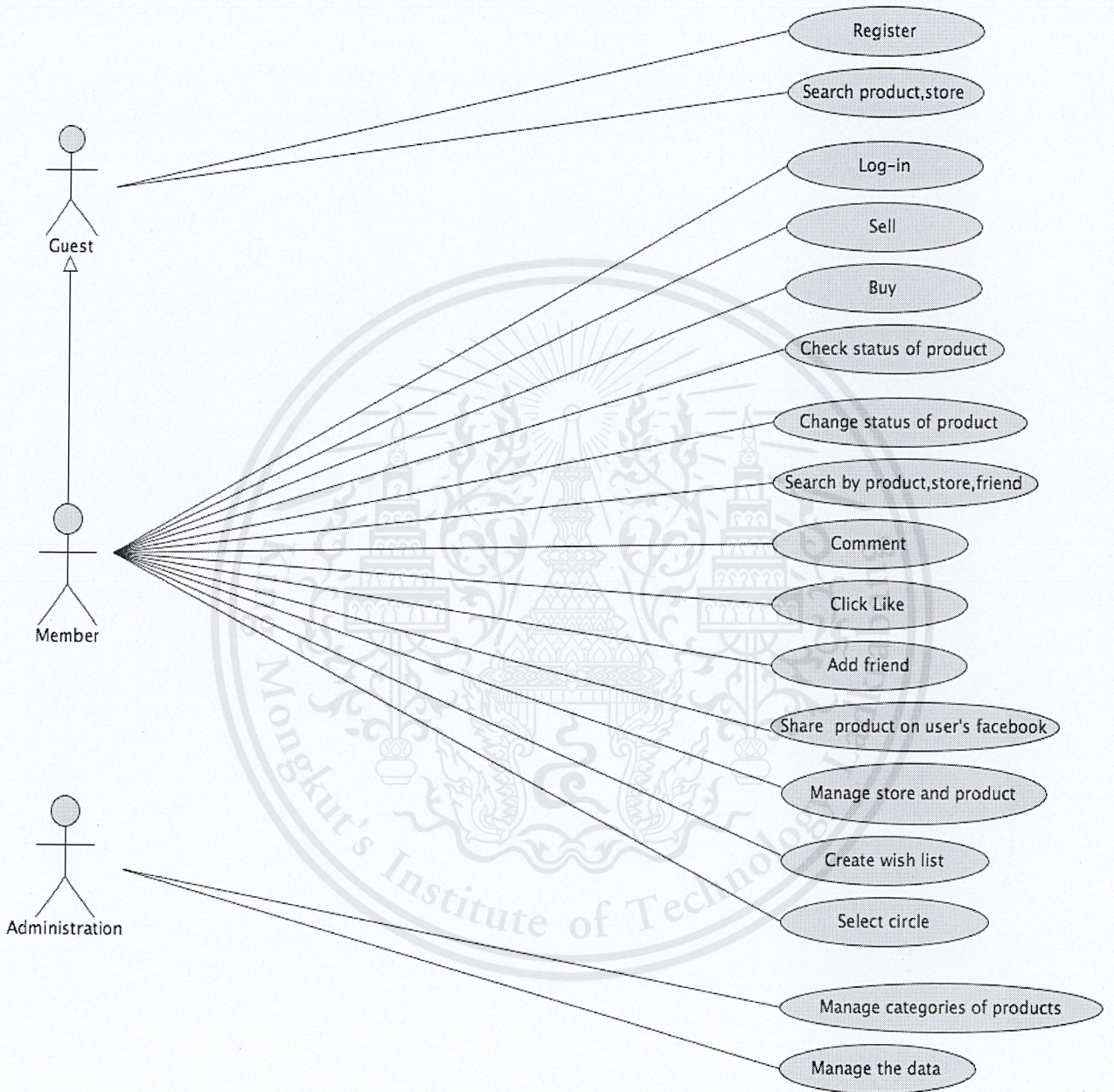
- The system allows administration to manage categories of products.
- The system allows administration to manage the data in the system.

### 4.2 Requirements for iOS application

- The system allows users to log in by username and password.
- The system allows users to log in by Facebook
- The system allows user to search products by name and hashtag of product.
- The system allows users to buy products in the web site.
- The system allow user to check the order.
- The system provides the social network functions: indicate that he/she like and comment products.

## 4.3 Use case diagram

### 4.3.1 Use cases for the website



**Figure 4.1:** Use case diagram

#### 4.3.1.1 Use case description for website

- **Expanded description of the “Register” use case**

**Use case:** Register

**Actors:** Guest

**Goal:** To be member of the system

**Overview:** The guests have to register by filling their information such as last name, first name, and email address. This system also provides registration by Facebook.

**Typical course of events:**

Actor action	System response
1. Guest accesses in the system.	
2. Guest fills personal information.	
3. Guest click register.	4.The system records guest information and change from guest to member.
	5. The system allows member for using the feature.

- **Expanded description of the “search product” use case**

**Use case:** search product

**Actors:** guest

**Goal:** To search product in the system

**Overview:** When the guest come into the website and then they can search the product of store.

**Typical course of events:**

Actor action	System response
1. Guest accesses in the system.	
2. Guest search the product or store.	3.The system shows the product.

- **Expanded description of the “Login” use case**

**Use case:** Login

**Actors:** Member

**Goal:** To access to the system

**Overview:** When the member would like to access the system. They have to log in by using username and password or their Facebook.

**Typical course of events:**

Actor action	System response
1. Member accesses in the system.	
2. Member click log in button.	3. The system provide the blank to fill username and password.
4. Member fills the username and password.	5. The system allows member to access in the system.

**Alternative course of events:** When the member fills wrong username and password.

- **Expanded description of the “Buy” use case**

**Use case:** Buy

**Actors:** Member

**Goal:** To buy product in the website.

**Overview:** When the member would like to buy the product, they have to select the product and click buy button.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. The member chooses the product.	
2. The member click buy button.	3.The system shows the shopping cart.
4. The member chooses the quantity, payment and then clicks the order button.	5. The system shows the list of member history.

- **Expanded description of the “Sell” use case**

**Use case:** Sell

**Actors:** Member

**Goal:** To sell product in the website.

**Overview:** When the member would like to sell the product, they have to add product into their store.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. The member click add product button	2. The system shows the product pages.
3. The member fills the information of the product.	
4. The member click create button.	5. The system records the product in the system.

- **Expanded description of the “Check and Change status of product ” use case**

**Use case:** Check and Change status of product

**Actors:** Member

**Goal:** To check and change status of product that the user sell and purchase.

**Overview:** When the member sells the product, they can change status of the product such as waiting, shipping status. When the member buys the product, they can check status of the product.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. When the product is sold. The member click changes the status of the product.	2. The system shows the status of the product.
3. When the product is bought. The member click checks the status of the product.	4. The system shows the status of the product.

- **Expanded description of the “search product” use case**

**Use case:** search product

**Actors:** member

**Goal:** To search product in the system

**Overview:** When the member come into the website and then they can search the product of store.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. Member accesses in the system.	
2. Member searches the product or store.	3. The system shows the product.

- **Expanded description of the “Comment and Like” use case**

**Use case:** Comment and Like

**Actors:** Member

**Goal:** To comment and like of the product.

**Overview:** When the member views the product and would like to comment and like the product.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. The member clicks like button.	2. The system updates the number of likes.
3. The member types the comment and click sends comment.	4. The system shows the comment on the screen.

- **Expanded description of the “Add friend” use case**

**Use case:** Add friend

**Actors:** Member

**Goal:** To be friend with another one.

**Overview:** When the member would like to be friend with someone, they can click add friend and waiting for respond.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. Click add friend button.	2. The system sends request to that person.
	3. The system waiting for responds from that person.
	4. The system records those person are friend.

- **Expanded description of the “Manage store and product” use case**

**Use case:** Manage store and product

**Actors:** Member

**Goal:** To edit, add and delete store or product.

**Overview:** When the member would like to edit, add and delete the store or product.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1.The member clicks edit store or product.	2. The system allows member to edit the product.
3. The member clicks ok.	4. The system updates the latest information.

- **Expanded description of the “Create wish list” use case**

**Use case:** Create wish list

**Actors:** Member

**Goal:** To create wish list that member want to buy in the future.

**Overview:** When the member favors the product, they can click wish list to record the product into their list.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. Click the wish list button.	2. The system records the product in member list and shows in the screen.

- **Expanded description of the “Select Circle” use case**

**Use case:** Select Circle

**Actors:** Member

**Goal:** To select group that most likely the same kind of product.

**Overview:** The system try to group the same kind of member together.

**Typical course of events:**

Actor action	System response
	1. The system suggests the circle to the member.
2. The member decides to be part of that circle.	
3. The member joins the circle.	4. The system records joining.

- **Expanded description of the “Manage categories of products” use case**

**Use case:** Manage categories of products

**Actors:** Administrator

**Goal:** To manage categories of products.

**Overview:** When the administrator would like to add, delete or edit the categories of products.

**Typical course of events:**

Actor action	System response
1. The administrator clicks admin page.	2. The system shows admin page.
3. The administrator clicks categories button.	4. The system shows the categories page.
5. The administrator manages the categories.	6. The system updates the information.

- **Expanded description of the “Manage the data” use case**

**Use case:** Manage the data

**Actors:** Administration

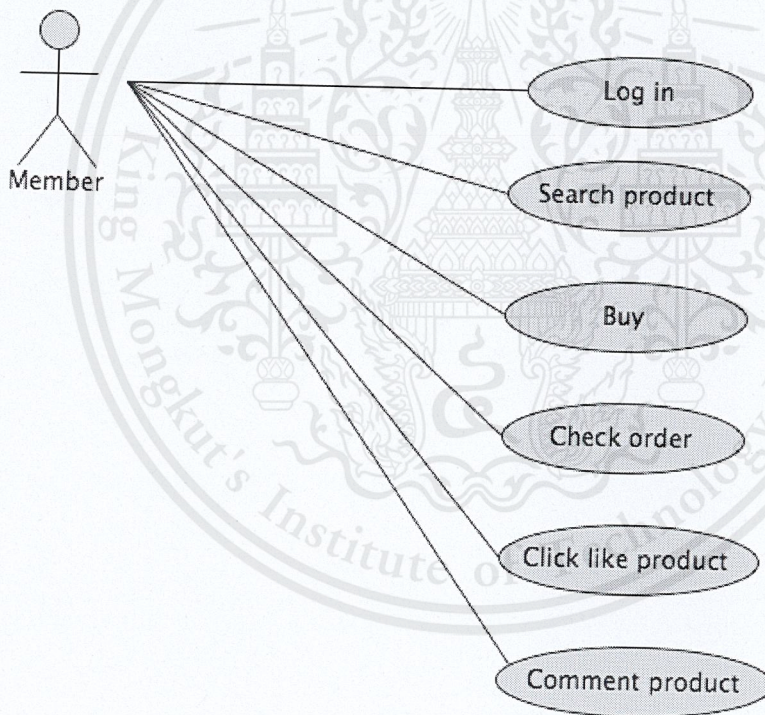
**Goal:** To manage the data of the product.

**Overview:** When the administrator would like to add, delete or edit data.

### Typical course of events:

Actor action	System response
1. The administrator clicks admin page.	2. The system shows admin page.
3. The administrator clicks the data that want to manage.	
4. The administrator clicks save.	5. The system updates the data.

### 4.3.2 Use cases for the iOS application



**Figure 4.2:** Use case diagram for iOS application

#### 4.3.2.1 Use case description

- **Expanded description of the “Login” use case**

**Use case:** Login

**Actors:** Member

**Goal:** To access to the system in mobile application.

**Overview:** When the member would like to access the system. They have to log in by using username and password or their Facebook.

**Typical course of events:**

Actor action	System response
1. Member accesses in the system.	
2. Member click log in button.	3. The system provides the blank to fill username and password.
4. Member fills the username and password.	5. The system allows member to access in the system.

**Alternative course of events:** When the member fills wrong username and password.

- **Expanded description of the “search product” use case**

**Use case:** search product

**Actors:** Member

**Goal:** To search product in the system

**Overview:** When the member comes into the application and then they can search the product.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. Member accesses in the mobile application.	
2. Member searches the product.	3.The system shows the product.

- **Expanded description of the “Buy” use case**

**Use case:** Buy

**Actors:** Member

**Goal:** To buy product in the mobile application.

**Overview:** When the member would like to buy the product, they have to select the product and click buy button.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. The member chooses the product.	
2. The member click add to cart button.	3.The system records in the shopping cart.
4. The member clicks check out button.	5. The system records the order.

- **Expanded description of the “Check status of product ” use case**

**Use case:** Check status of product

**Actors:** Member

**Goal:** To check status of product that the user sell and purchase.

**Overview:** When the member sells the product, they can change status of the product such as waiting, shipping status. When the member buys the product, they can check status of the product.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. When the product is bought. The member click checks the status of the product.	2. The system shows the status of the product.

- **Expanded description of the “Comment and Like” use case**

**Use case:** Comment and Like

**Actors:** Member

**Goal:** To comment and like of the product.

**Overview:** When the member views the product and would like to comment and like the product.

**Typical course of events:**

<b>Actor action</b>	<b>System response</b>
1. The member clicks like button.	2. The system updates the number of likes.
3. The member types the comment and click sends comment.	4. The system shows the comment on the application.

# Chapter 5

## Software Design

### 5.1 System Architecture

Figure 5.1 shows the architecture of our online marketplace system. There are two main parts in this system, the front end and the back end. In the front end, the user can interact with our application via a web browser and an iOS device such as an iPhone. In the back end, we implemented the server-side application using the Django framework. Django communicates with our web server, which is NginX, using the software called WSGI. The Django framework automatically manages the database, which is stored in the PostgreSQL server.

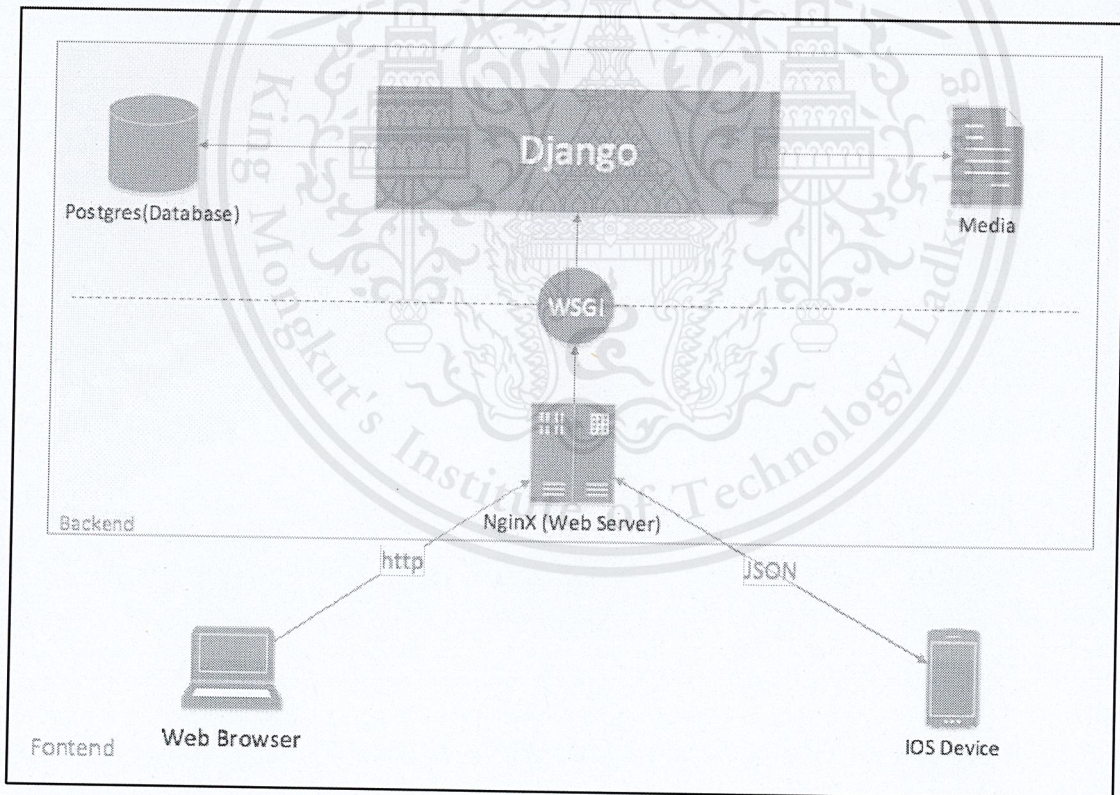


Figure 5.1: System architecture

## 5.2 ER diagram

The ER diagram of our system is shown in Figure 5.2. The ER Diagram of this system is very large, so it is separated into two sections. The first section contains the entities related to the users, including the user's profile [shopping\_member], user's activities on products [shopping\_member\_product], like [shopping\_like\_product]. The second section contains the entities related to the E-commerce system, product information [shopping\_product], store information [shopping\_store], etc.

There are also relationships between products and tags [shopping\_product\_tag] and the relationship between tags and tags [shopping\_interesting\_tag].



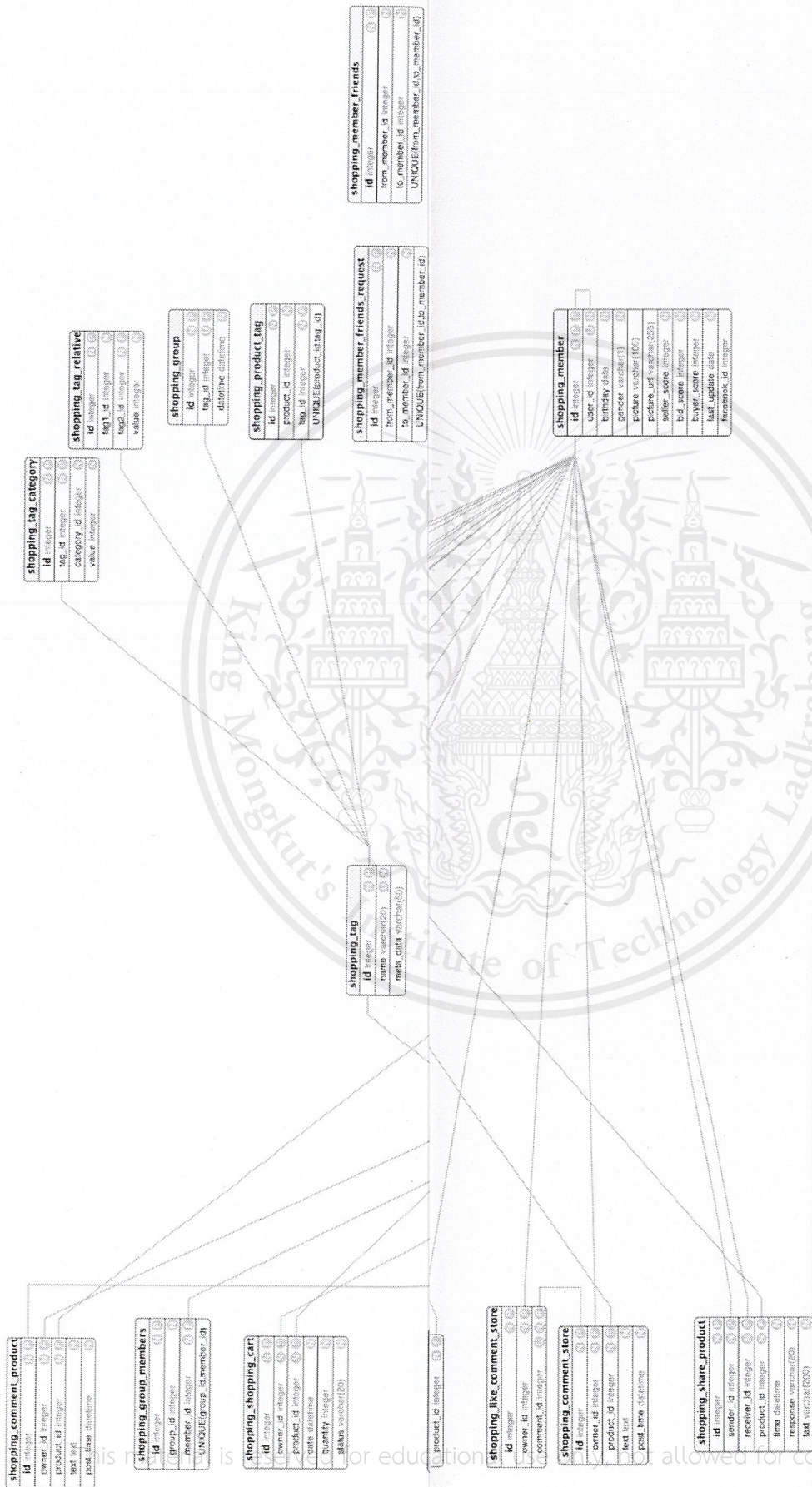
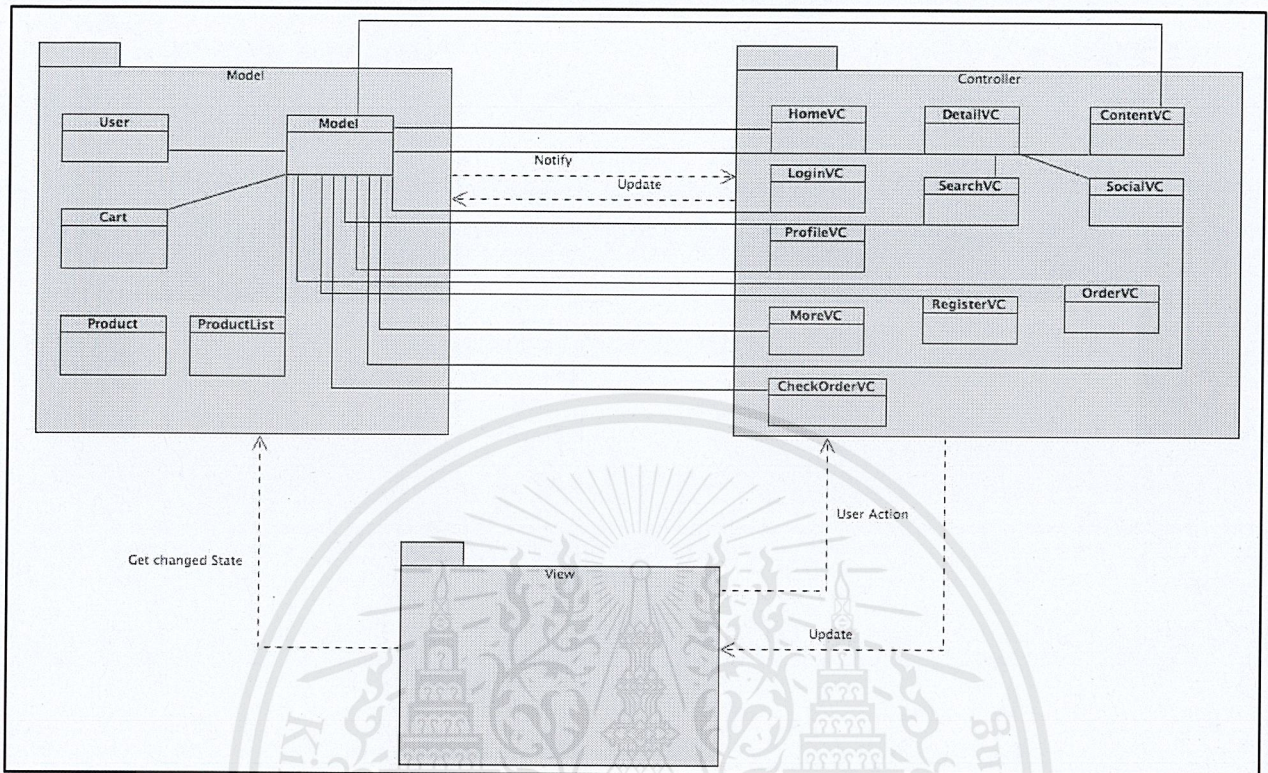


Figure 5.2: ER diagram

### 5.3 Class Diagram for the iOS application



**Figure 5.2:** Class diagram for iOS application

In mobile application, the software design is based on the MVC architecture which consists of three components: model, view, and controller.

The model contains classes which encapsulate data storage in our application. Classes in the model may contain a method for loading the data from database or server. For example in our application the model including the class of user, product, product list and cart.

The controller is the mediator that coordinates all the work. It accesses the data from the model, listens to events and manipulates the data then displays in the view. For example in our application the controller including the class of check order, details and search.

The model will notify the controller of any change in the data, after which the controller will update the view.

# Chapter 6

## Development

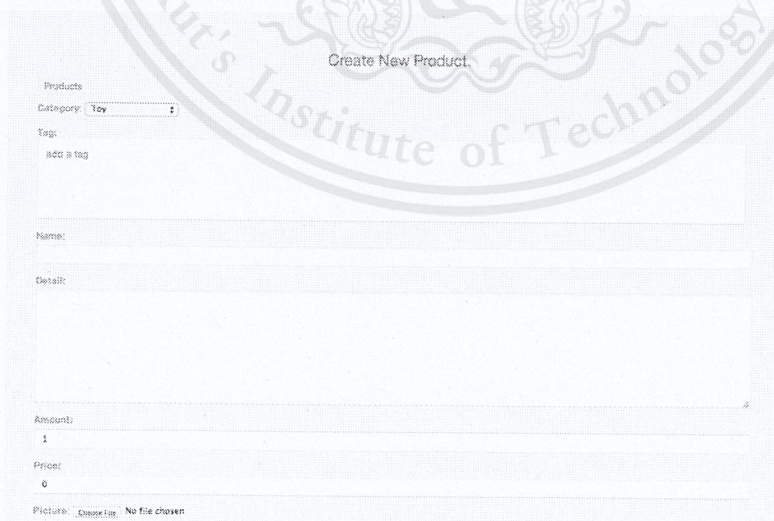
### 6.1 Techniques

#### 6.1.1 Information in online social marketplace system

To effectively provide recommendations for the user, our online social marketplace system needs to store various information about the products and the users, as well as other additional information in the system's database. In this section, we summarize the key entities of our system.

- **Products**

As our system acts as an online marketplace, the system needs to store the data of all the products that were or are being sold. We record many attributes for each product, including category, tag, name, detail, amount, price, picture, payment, and identification number. Therefore, when the user would like to post a product for sale, they need to provide the data for the mentioned attributes of the product. Figure 6.1 shows the entry form for adding a new product to the system with the product's attributed listed



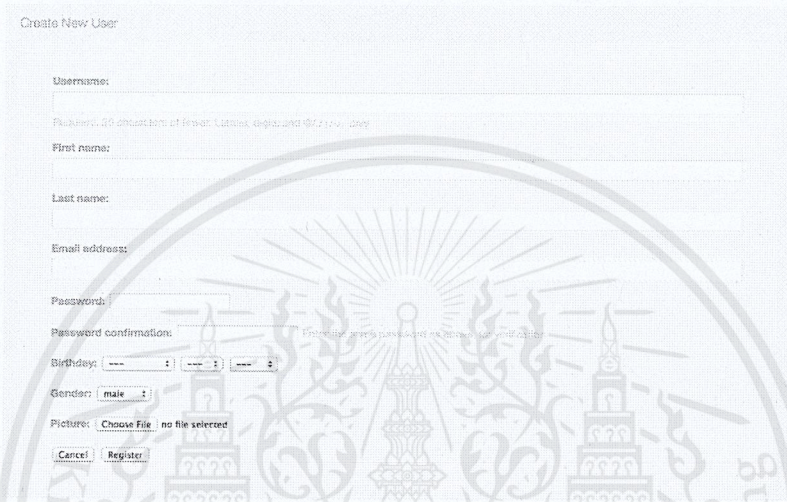
The image shows a web form titled "Create New Product" overlaid on a large, faint watermark of a university seal. The form contains the following fields:

- Products**: A dropdown menu with "Toy" selected.
- Tag:** A text input field with the placeholder "add a tag".
- Name:** A text input field.
- Detail:** A large text area for product description.
- Amount:** A text input field with the value "1".
- Price:** A text input field with the value "0".
- Picture:** A file upload field with the text "Choose file" and "No file chosen".

**Figure 6.1 :** The product's attributes to be provided when creating a new product

- **Users**

To sell or buy a product on our system, each buyer or seller needs to register as a new user. Note that there are no separate accounts for buyers and sellers. Each user can be both a buyer and a seller in our system. When registering a new user, the following information must be provided username, first name, last name, e-mail address, password, birthday, gender and photo. Figure 6.2 shows a page for creating a new user.



The image shows a web form titled "Create New User". The form contains the following fields and controls:

- Username:** A text input field with a small icon to its left.
- First name:** A text input field.
- Last name:** A text input field.
- Email address:** A text input field.
- Password:** A text input field.
- Password confirmation:** A text input field.
- Birthday:** A date selection field with three dropdown menus for day, month, and year.
- Gender:** A radio button labeled "male".
- Picture:** A file upload button labeled "Choose File" with the text "no file selected" below it.
- Buttons:** "Cancel" and "Register" buttons at the bottom.

**Figure 6.2:** New user creation

- **Tags**

A tag is a word or a short phrase used to describe a product. Typically, a tag is named after the type, the brand, the model, the color, or the main feature of a product. For example, a pair of sport shoes could be given the tags such as "shoes", "Nike", "trainer", "running", "sport", etc., as can be see in Figure 6.3. Tags are very important in our system as we use them extensively in recommending product, and also in searching.

Specification of tags for a product is the responsibility of the seller of that product. To prevent an excessive number of tags, up to only five tags may be added to each product. There are many advantages in using these user-generated tags. Namely, tags are dynamic, flexible, and more refined than human-generated product categorization. However, there are also some inherent problems in the use of user-generated tags. The number of tags generated likely grows very fast and the tags have to be managed properly. Some two or more tags may be essentially the same but just different in spelling. Also, the seller might add a misleading tag to his/her products.

- **Categories**

Categories are resulted from grouping the products, which share similar characteristics. Categories may be nested hierarchically, i.e. there could be sub-categories within a category. In our system, categorization of products is done manually in advance. Figure 6.5 illustrates the hierarchy of the categories we designed for our system. Adding category can fill name, description, image, color and property as Figure 6.3. In addition, the website provides adding relative category which it is hierarchy of a category as Figure 6.4.



Add category

Name:

Description:

Image:  No file chosen

Color:

Properties:

Active

**Figure 6.3: Adding categories**



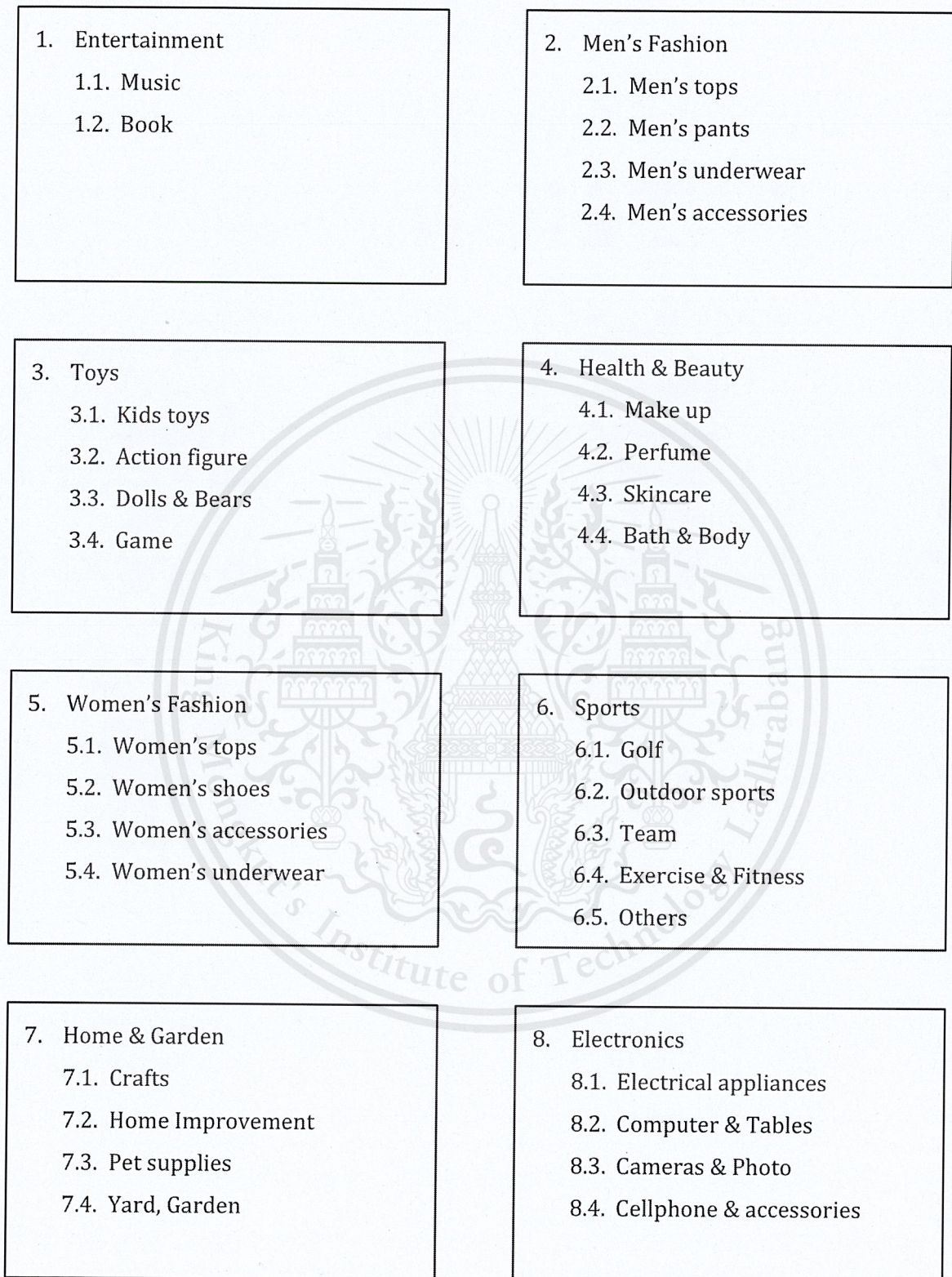
Add relative\_category

Parent category:

Child category:

**Figure 6.4: Adding relative categories**

## Products



**Figure 6.5:** Categories in our website

## 6.1.2 Graphs

Graphs are data structures which consist of nodes and edges. Nodes are typically used to denote some entities and edges are used to represent relationships among the entities. Since one node can be connected to many other nodes, graphs can thus be used to represent many to many relationships.

Our social marketplace system utilizes various kinds of information, such as users, products, and tags. Each of these can have relationships with the others. Such relationships can be modeled as graphs.

Below we describe some key relationships among the entities in our system.

- **Product - Tag**

A product is related to a tag when the product is assigned a tag. For example an iPhone can be given tags like “phone”, “Apple”, “iPhone”, “Touch ID”, “64 bit”.

The relationship between of products and tags induces a relationship among products. For example, consider the following products and their tags:

iPhone 5 = {phone, Apple, iPhone, Touch ID, 64 bit }

Nokia 3310 = {phone, Nokia}

iMac = {Apple, Mac, computer}

Then we can infer that iPhone 5 and Nokia 3310 are related as they are both phones and iPhone 5S and iMac are also related as they are both made by Apple.

- **Tag - Tag**

We can also record how one tag is related to another tag. For example, if whenever a product is assigned Tag 1 it is also Tag 2, then we can infer that Tag 1 and Tag 2 are related.

- **User - Category**

We can also record the relationship between users and categories. When a user buys or clicks “like” on a product, there will be a relationship between the category in which that product belongs and the user.

### 6.1.3 Recommendation

- **Collaborative Recommendation**

In collaborative recommendation, we consider a situation where a customer is looking at some product and we would like to suggest other products that the customer may also like.

A simple algorithm for collaborative recommendation is as follows. Let A be the people who bought product X. Let S be the set of products that a person in A bought. For each product i in S, we define

Rank(i) = the number of people in A who bought product i

We sort the products in S based on its ranks in descending order and output the top 20 products in the list.

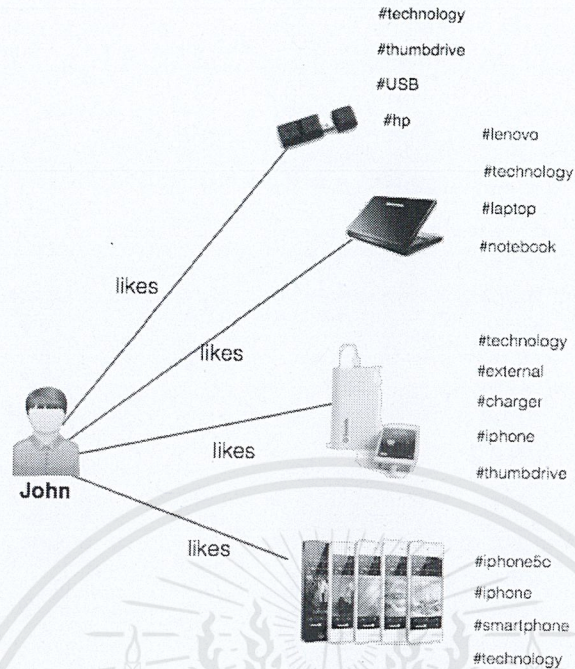
```
1: def CollaborativeBase(self, _product):
2:     _result = []
3:     own_this_product = Order.objects.filter(product=_product)
4:     for order_i in own_this_product:
5:         for order_j in Order.objects.filter(owner=order_i.owner):
6:             result.append(order_j.product)
7:     return most_common(_result,20)
```

**Figure 6.6:** Sample code of collaborative recommendation

- **Content-based recommendation**

In content-based recommendation, we recommend products for a customer by looking at the kinds of the products that the customer likes. We use tags to specify the kinds of products.

Suppose John is a customer. First, we find the tags of the products that John likes. Such products are typically assigned many tags, as shown in Figure 6.7



**Figure 6.7 :** Graph showing the products that a customer likes

We rank each tag by counting the number of products that is liked by John and contains such tag. Precisely, we order the tags by  $\text{TagRank}(\text{John}, t)$  where,

$\text{TagRank}(c, t)$  = the number of the products which  $c$  likes and contains tag  $t$ .

Table 6.1 shows the tag ranks for each tag of the products that John likes.

Next, we then find the products that contain one or more of the top ten tags. Since there might be many products, we sort those products based on the number of the top-ten tags that each product contains. Precisely, let  $T$  denote the top-ten tags. Let  $S$  be the set of products, which contain one or more of the tags in  $T$ . For each product  $i$  in  $S$ , we define

$\text{Rank}(i)$  = the number of the tags in  $T$  that are contained in product  $i$

Then we sort the products based on its ranks in descending order and return the top 20 products. Figure 6.8 contains a simplified implementation of content-based recommendation.

User	Tag	TagRank
John	iphone	2
John	technology	4
John	smartphone	1
John	external	1
John	charger	1
John	notebook	1
John	technology	1
John	laptop	1
John	iphone5c	1
John	laptopHP	1
John	USB	1
John	thumbdrive	2
John	hp	1

**Table 6.1:** List of tags from Figure 6.7 and their tag ranks

Product	Tags	Rank
iPhone5s	<b><u>iPhone</u>,<u>smartphone</u>,<u>technology</u></b>	3
iPhone5	apple, <b><u>technology</u></b> , <b><u>iphone</u></b>	2
Sony USB	<b><u>thumbdrive</u></b> ,computer, <b><u>tecnology</u></b>	2
Segate external	<b><u>thumbdrive</u></b> ,segate, <b><u>tecnology</u></b>	2
Hp notebook	<b><u>hp</u></b> ,computer,cheap	1

**Table 6.2:** List of the products containing the highest numbers of top-ten tags

```

1: def contentBase(self, _member):
2:     _result = []
3:     _tags = Interesting_Tag.objects.filter(member=_member).order_by('value')[:10]
4:     _search = SearchController()
5:     for _item in _tags:
6:         print _item.tag
7:         _result.extend(_search.searchByTag(_item.tag))
8:     return most_common(_result,20)

```

**Figure 6.8:** Sample code of content-based recommendation

- **Social Recommendation**

In social recommendation, we will recommend products for a customer by looking at the products that the customer's friends bought.

First, we find the products that one or more of the customer's friends bought. Since there might be many products, we sort those products based on the number of the friends who bought each product. Precisely, let  $F$  be the set of the customer's friends. Let  $S$  be the set of products, which a person in  $F$  bought. For each product  $i$  in  $S$ , we define

$\text{Rank}(i)$  = the number of people in  $F$  who bought product  $i$

Then we sort the products in  $S$  based on its ranks in descending order and return the top 20 products. Figure 6.9 shows a sample implementation of content-based recommendation.

```

1: def socialBase(self, _member):
2:     _result = []
3:     _products = []
4:     _friends = _member.friends.all()
5:     for friend in _friends:
6:         _orders = Order.objects.filter(owner=friend)
7:         for order in _orders:
8:             _products.append(order.product)
9:     return most_common(_products,20)

```

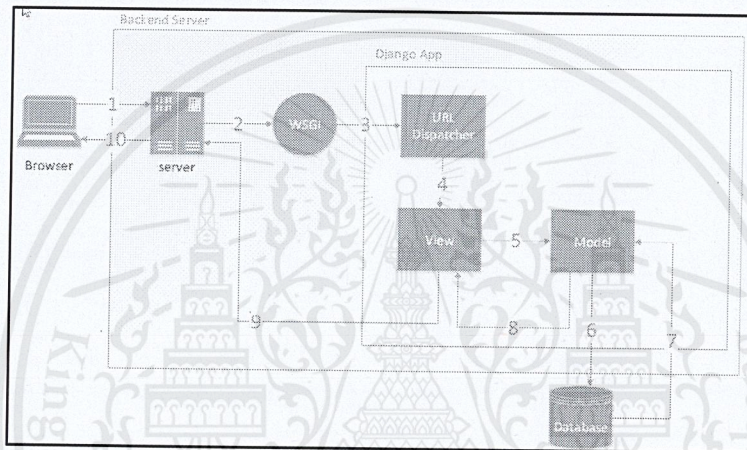
**Figure 6.9:** Sample code of social recommendation

## 6.2 Implementation

### 6.2.1 Back end

This section is concerned with the server and the main core of project. It manages for calculating and processing. When the user enters to website or searches the product then the system will pull the product from server to show on the user screen and prepares the web service for iOS application. All of these are processing of back end.

The main core of the website we have been using Django for implementation because Django is a high level Python web framework that encourages rapid development and pragmatic design.



**Figure 6.10:** Back-end system

For Database, we have been using PostgreSQL because it is a powerful, open source object-relation database system, data integrity and correctness. Besides it has native programming interface for C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC. [2]

For Web service API framework, we have been using Tastypie because Tastypie is a web service API framework for Django. Moreover Tastypie can build REST-style interfaces for forwarding data through Internet by using JSON format. [3]

For REST-style, we have been using REST-style URL because it is often used to indicate a URL that composes its parameters within the URL file path, rather than the query string.[4]

For JSON, we have been using JSON because JSON is a lightweight data interchange format and easy for read and write. Furthermore, it is easy for machines to parse and generate. [5]

## 6.2.2 Web-based front end

This section is used for communication with users. We select web to display the interface because it is easier than using embedded application. The advantages of web front end are easy to use and can run in every platform/OS.

We created front end by using HTML, CSS, Java script, jQuery and AJAX.

We used Html for creating web pages and other information that can be displayed in a web browser. For decoration, CSS was used in our system. Moreover, jQuery was used which was a library that increased the performance for the website and easy to implement.

Then Ajax is used technologies for building dynamic webpage on the client side. Dynamic webpage is an important in our project because in this project has a notification and tracker. It is necessary to some parts of web updates so we need to use AJAX in our website. For designing interface, we use the techniques of responsive web design.

## 6.2.3 iOS application

Nowadays, Most of people use smartphone rapidly. Especially iPhone thus we would like to develop iOS application into our project.

Objective-C language is used for Developing iOS application. It is the main programming language for the OS X and iOS operating systems and their respective APIs, Cocoa and Cocoa Touch. So in our application we have been using Objective-C language for implement.

For API to connect with the back end we have been using Restkit, *“Restkit is a modern Objective-C language framework for implementing Restful web services clients on iOS and Mac OS X. It provides a powerful object mapping engine that seamlessly integrates with Core Data and a simple set of networking primitives for mapping HTTP requests and responses built on top of AFNetworking.”* [6]

# Chapter 7

## Results

### 7.1 Website

As Figure 7.1 displays the home page of website. There are composed with many functions such as login, login with Facebook, recommendation function, register, search and circle.

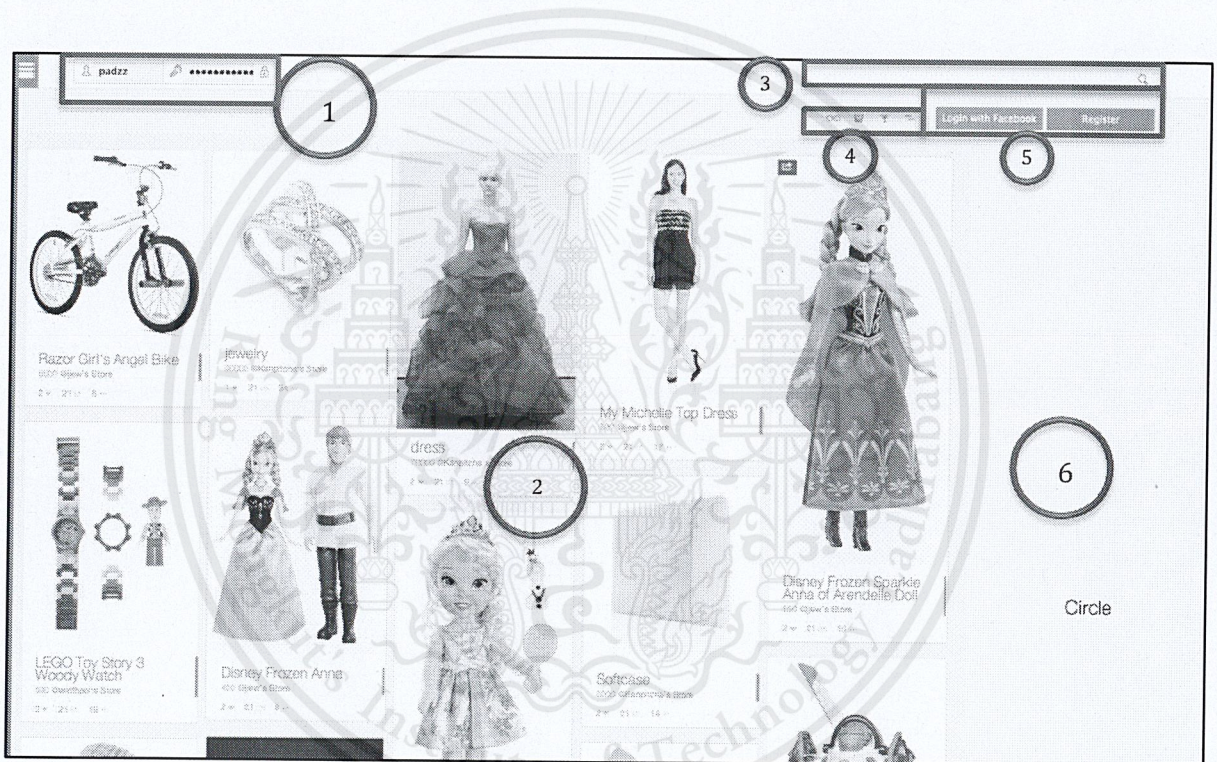


Figure 7.1: Homepage of website

There are six components in the homepage of our website which are log in function (1), product list area (2), search function (3), recommendation function (4), log in with Facebook and register (5) and circle function as shown in Figure 7.1. We will describe those components in detail shortly.

### (1) Log in

When the user enter to our website the user must fill the user name and password then click on the log in button for log in into website. The system will verify the username and password.

### (2) Product list area

The area that displays the product in our website. It will show the product list in type of picture, which you can see the name of product, name of store, price, like and comment.

### (3) Search function

The user can type a key word to search for product. When the user click on search button the screen will show the search result.

### (4) Recommendation function

There are three recommendation functions as shown in Figure 7.2, which are collaborative recommendation, content-based recommendation and social recommendation.



**Figure 7.2: Recommendation function**

### (5) Log in with Facebook and Register

The user can log in to the website by using Facebook account and if the user are not the member in our website the user should click on the register button for registration.

### (6) Circle function

The system suggests the circle to the member and tries to group the same kind of member together.

As Figure 7.3 displays the detail of product page in our website. There are composed with many components such as notification, friend request, menu bar, social features, details of product, recommendation.



Figure 7.3: Detail of product page

There are five components in the detail of product page in our website which are notification and friend request (1), menu bar (2), social feature and buy function (3), recommendation function (4), comment of product (5) as shown in Figure 7.3. We will describe those components in detail shortly.

#### (1) Notification and friend request

As shown in Figure 7.4, the system can notify interesting events such as someone just bought or commented your product and can be friend with other users.

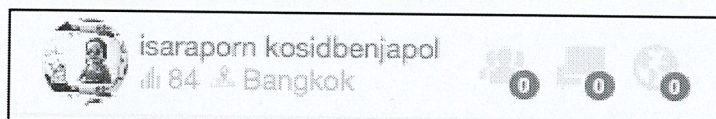


Figure 7.4: Notification and friend request

## (2) Menu bar

There are seven menu as shown in Figure 7.5, which are homepage, setting, profile, shopping cart, order product, check status of product and log out.



**Figure 7.5:** Menu bar

## (3) Social feature and buy function

There are five functions as shown in Figure 7.6, which are like, share, wish list, message and buy function.



**Figure 7.6:** Social feature and but function

## (4) Recommendation function

This system is able to recommend the products for the users. There are recommend the products that the user might purchase as shown in Figure 7.7.

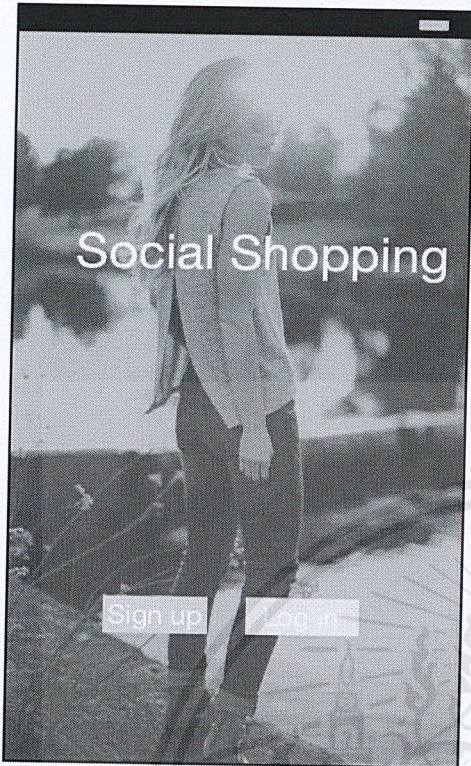


**Figure 7.7:** Recommendation function

## (5) Comment of product

When the user views the product and would like to comment the product.

## 7.2 Mobile application



**Figure 7.8:** Home page

Figure 7.8 shows the first page of iOS application. This page encourages to sign up and login.

1. When you click "Sign up" button, the application will go to the sign up page for new user to register.

2. When you click "Create Account" button, the application will go to log in page for enter user to the application.

A screenshot of the Social Shopping app's register page. The status bar at the top shows "iPod", signal strength, Wi-Fi, time "06:12", and battery level. A "Back" button is in the top left. The title "Social Shopping!!" is centered. Below it, the text "Please confirm your information below:" is followed by several input fields: "Username", "First name", "Last name", "Gender", "Birthday", "Your email", and "Password". At the bottom, there is a "Create Account" button.

**Figure 7.9:** Register page

Figure 7.9 shows register page. This page allows the user to fill in the username, first name, last name, gender, birthday, email and password.

Then click on "Create Account" button for create new account.

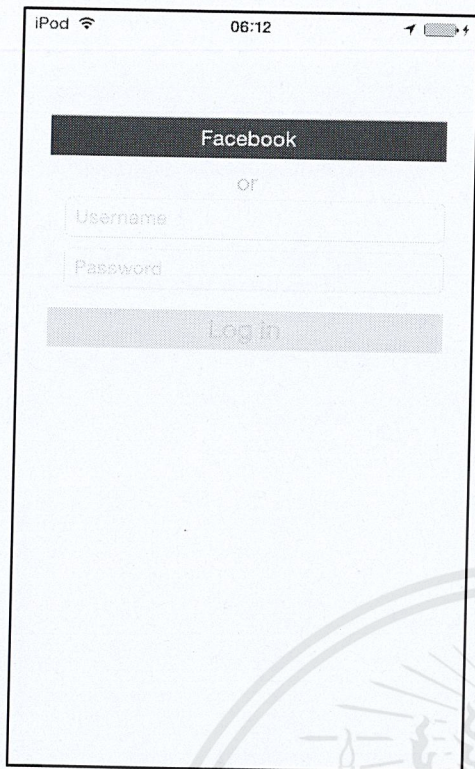


Figure 7.10 displays the login page in iOS application, which allows the user to fill a username and a password and login by using Facebook.

Fill username and password then tap "Log in" button for log in to the application.

Tap "Facebook" for log in to the application by using Facebook account.

**Figure 7.10:** Log in page

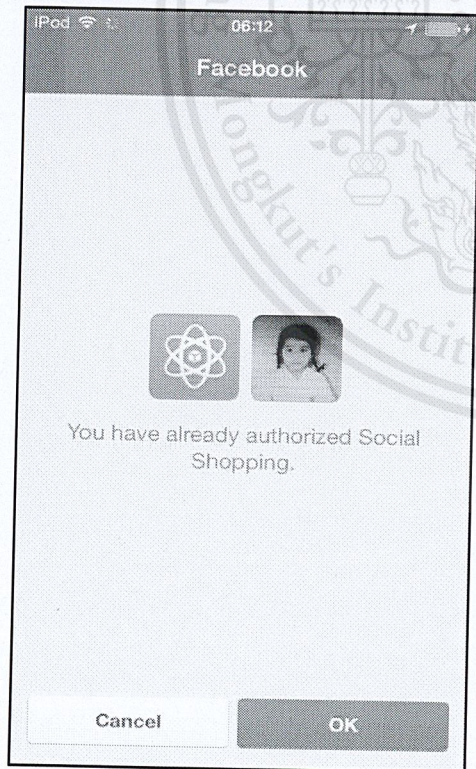


Figure 7.11 displays an access to login by using Facebook.

**Figure 7.11:** Log in by using Facebook



Figure 7.12: Product page

Figure 7.12 displays the product page of iOS application. There are five tabs in the bottom as shown in the Figure 7.12 which are market, content-based recommendation, social recommendation, shopping cart and more.

### 1. Market

When the user clicks on 'market' tab(1), the application will display the list of products in each store.



Figure 7.13: List of products

### 2. Content-based

When the user clicks on 'Content-Based' tab (2), the application will display the list of products in each store by using content-based recommendation as shown in Figure 7.13.

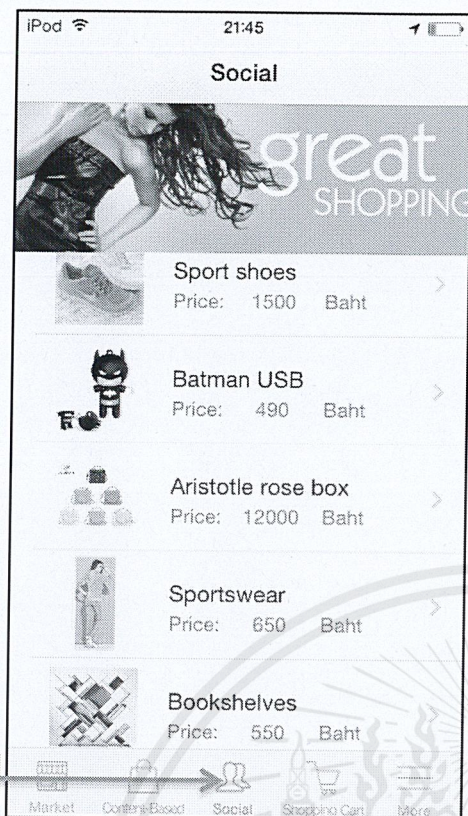


Figure 7.14: List of products

### 3. Social

When the user clicks on 'Social' tab (3), the application will display the list of products in each store by using social recommendation as shown in Figure 7.14.

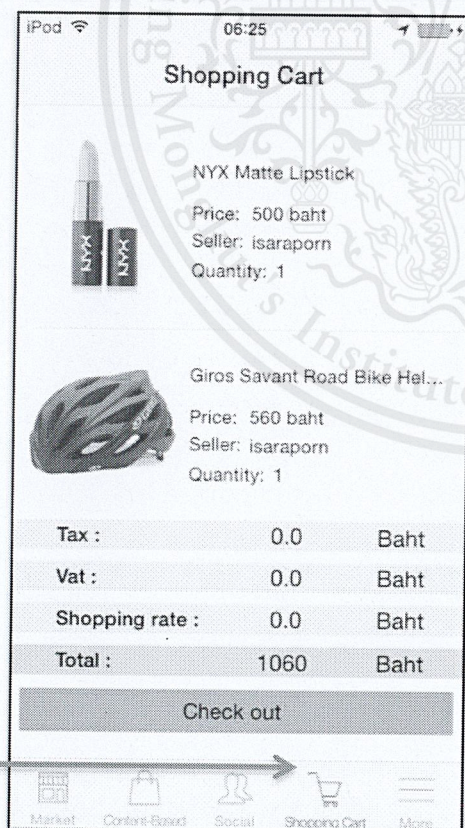
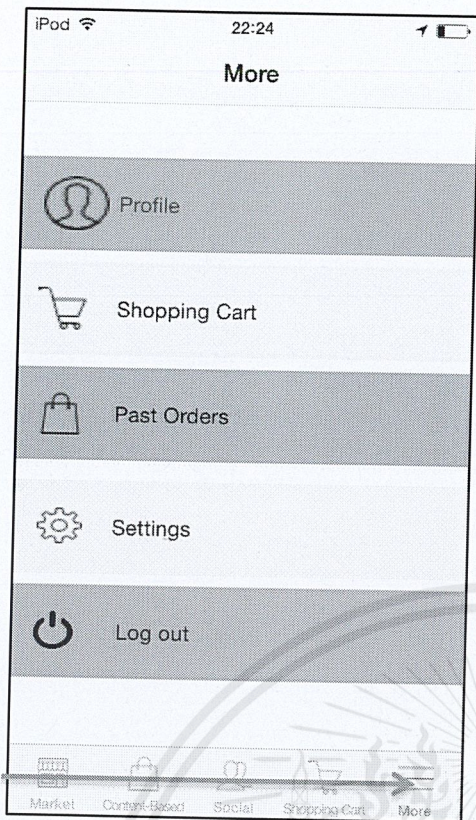


Figure 7.15: List of products

### 4. Shopping cart

When the user clicks on 'Shopping cart' tab (4), the application will display the list of products that you add to your cart and click on 'Check out' button for purchased product as shown in Figure 7.15.




## 5. More pages

When the user clicks on 'More' tab (5), There are five menu as shown in Figure 7.16.

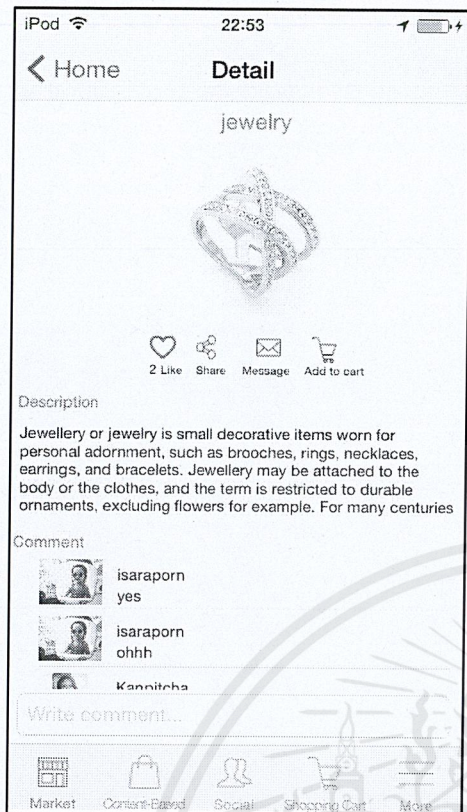
1. Profile
2. Shopping Cart
3. Past Orders
4. Setting
5. Logout

**Figure 7.16:** More pages



When the user want to search the product, clicks on  button in Figure 7.12 then type the keyword of product. Figure 7.17 shows the result of search.

**Figure 7.17:** Search pages



**Figure 7.18:** Detail of product page

When the user want to look the product, clicks on the product then the application will display the detail of the product as shown in Figure 7.18.

The user can also click like, comment and add the product to cart.

# Chapter 8

## Conclusion

### 8.1 Summary

In this project, we have developed a web site and a mobile application for an online marketplace which incorporates a number of social features. , we studied various techniques for developing a recommender system commonly used on e-commerce websites, including collaborative recommendation techniques and content-based recommendation techniques. We adapted and extended such techniques by utilizing relationships among the users and the information obtained from the users' profiles on Facebook to create an enhanced recommender system which not only provides recommendations on products to the user but also suggests "friends", the other users who are expected to share similar interest with the user. We implemented the server side of the software using the Django framework. The web-based front end was implemented in using HTML 5. The mobile application was developed for the iOS platform.

### 8.2 Problems and obstacles

- Lack of customer and product data to evaluate the recommendation techniques implemented in the system.
- There are many different techniques on recommender systems, each of which requires knowledge in some particular field. Consequently, we spent a lot of time on the background study.
- Lack of a large number of testers to evaluate the effectiveness of the developed system and measure its performance.

### 8.3 Further work

#### 8.3.1 Web site

- The home page of the website should be made to load faster.
- Test the system with a large number of testers.
- Utilize more information available on the user's social networks.
- Simplify the interface of the website.

- Study and apply other recommendation techniques.

### 8.3.2 iOS application

- Add more functions to the mobile application, such as selling an item.
- Improve the interface.
- Make the application to work on all iPad and older iPhones with smaller screens.
- Add push notifications.
- Make the application load the data more quickly.
- Utilize the location data from the iOS device.



## Reference

- [1] Dietmat Jannach, Markus Zanker, Alexandar Felfernig, Gerhard Friedrich, **"Recommender Systems: An Introduction"**, Cambridge University Press, 2010.
- [2] PostgreSQL. **"What is PostgreSQL?"** Available: <http://www.postgresql.org/about/>.
- [3] Tastyapi. **"What is tastyapi?"** Available: <http://tastyapi.org>.
- [4] Stackoverflow. **"What does actually mean Rest-Style?"** Available: <http://stackoverflow.com/questions/11437598/what-does-actually-mean-rest-style-url>.
- [5] Json. **"What is JSON?"** Available: <http://www.json.org>.
- [6] Restkit. **"What is Restkit?"** Available: <http://restkit.org>.

