

MOBILE PRICE SHARING AND COMPARISON SYSTEM

By

TAN JIA MING

Supervisor: DR. OOI BOON YAIK

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

Jan 2013

MOBILE PRICE SHARING AND COMPARISON SYSTEM

By

TAN JIA MING

Supervisor: DR. OOI BOON YAIK

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

Jan 2013

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Mobile Price Sharing and Comparison System**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : Tan Jia Ming

Date : 8-4-2013

Acknowledgement

I would like to express my gratitude towards all people that have been supporting me, giving me guidance during the project. Without them supporting me, the project might not be possibly done.

First of all, I would like to express my appreciation to supervisor, Dr. Ooi Boon Yaik, who gave me idea, comments, and guidance during the period of completing this Project 2.

I would also say thanks to Faculty of Information Communication and Technology, UTAR (Perak Campus) that allows me to pursue my degree, and for providing facilities which facilitated the work, such as computer laboratories.

Besides, I would also express my deepest appreciation towards my parent, and my family members, that have been supportive during the time I pursue my degree, either financially or mentally.

Finally, I would like to say thanks to all other friends that have been helping me throughout the project, especially Chong Jia Zheng, and Lee Yew Chyuan, which shared their knowledge and experience.

Abstract

Inflation has brought great impact to everyone, and people would spend their money wisely, they will compare price of similar products, and choose the better offer. And so, many price comparison websites are built, to help them gather information more quickly.

However, the information was decentralized across those different websites. And some websites requires extra effort to reveal the information of a particular product. Besides, some websites contains out-dated information, because they have not been updated for a while. This project, has aimed to solve the above problems.

During this project, several solutions had been analysed. These solutions are either websites, or applications for mobile device. Different criteria have been determined to identify the strengths and weaknesses of each solution. And then, several strengths of those solutions have been combined, to cover up the shorts of other solutions.

The new solution is a mobile application that allows both user and seller to update pricing information. Besides, it also allows users to post reviews for the products. It has a search function, to allow users to find a particular product more easily. It also features a bar code scanner, which users can use to precisely find a particular product in a quick manner. Users can also use the application to compare the prices of the product within an area, and find better prices in that area easier. The project will be focused on retail products that come with a bar code scanner.

Further work can be conducted to allow more features, such as photo taking, or includes products that do not have a bar code.

Table of Contents

Title	i
DECLARATION OF ORIGINALITY	ii
Acknowledgements	iii
Abstract	iv
Table of Contents	v
List of Figures	vii
List of Tables	ix
List of Abbreviations	x
Chapter 1 Introduction	
1.1 Project background	1
1.2 Problem statements	2
1.3 Project scope	3
1.4 Objectives	4
1.5 Contributions	5
Chapter 2 Literature Review	
2.1 Existing solutions	6
2.2 Critical remarks of existing solutions	
2.2.1 Price Check by Amazon.com	8
2.2.2 WhatGas Petrol Prices	8
2.2.3 1Pengguna.com	8
2.2.4 Lowyat.net	9
2.2.5 Mudah.my	9
2.2.6 Lelong.com.my	9
2.2.7 Mobile88.com	10
2.2.8 Shopp paradiso.com	10
Chapter 3 Methodology And Tools	
3.1 Methodologies	
3.1.1 Use Case Diagram	11
3.1.2 Class Diagram	13
3.1.3 Entity-Relationship Diagram	15
3.1.4 Activity Diagram	16
3.1.5 Sequence Diagram	18
3.1.6 Collaboration Diagram	24
3.1.7 State Diagram	26
3.2 Technology involved	28

MOBILE PRICE SHARING AND COMPARISON SYSTEM

3.3	Specifications and requirements	
3.3.1	Functional requirement	30
3.3.2	Hardware and software requirement	31
3.4	Gantt chart	
3.4.1	Project Planning phase	32
3.4.2	Project Analysis phase	33
3.4.3	Project Design phase	34
3.4.4	Project Development and Implementation phase	34
3.4.5	Testing phase	35
3.4.6	Second Project Analysis phase	35
3.4.7	Second Project Design phase	36
3.4.8	Second Project Development and Implementation phase	36
3.4.9	Second Testing phase	37
3.4.10	Documentation phase	37
Chapter 4 Implementation		
4.1	Overall architecture	38
4.2	MPSCS	39
4.2.1	Reporting	39
4.2.2	Scanner	40
4.2.3	Location-based searching	40
4.3	Cloud Service	41
4.3.1	Server	42
4.3.1	Database	42
4.4	Screenshots	43
Chapter 5 Testing		
5.1	Functionality Testing	52
5.2	Usability Testing	56
Chapter 6 Conclusion		
6.1	Conclusion and contributions	59
6.2	Software limitations	59
6.3	Future work and possible enhancements	61
References		62
Appendix		A-1

List of Figures

Figure 3-1-1-1	Use Case Diagram for MPSCS	11
Figure 3-1-2-1	Class Diagram for MPSCS	13
Figure 3-1-3-1	Entity-Relationship Diagram for MPSCS	15
Figure 3-1-4-1	Activity Diagram for MPSCS	16
Figure 3-1-5-1	Sequence Diagram for add location in MPSCS	19
Figure 3-1-5-2	Sequence Diagram for add product in MPSCS	20
Figure 3-1-5-3	Sequence Diagram for view reporting in MPSCS	21
Figure 3-1-5-4	Sequence Diagram for compare price in MPSCS	22
Figure 3-1-5-5	Sequence Diagram for update price in MPSCS	23
Figure 3-1-6-1	Collaboration Diagram for MPSCS	24
Figure 3-1-7-1	State Diagram for MPSCS	26
Figure 3-2-1	High level Architecture view for MPSCS	28
Figure 3-4-1-1	Gantt chart for Project Planning phase	32
Figure 3-4-2-1	Gantt chart for Project Analysis phase	33
Figure 3-4-3-1	Gantt chart for Project Design phase	34
Figure 3-4-4-1	Gantt chart for Project Development and Implementation phase	34
Figure 3-4-5-1	Gantt chart for Testing phase	35
Figure 3-4-6-1	Gantt chart for Second Project Analysis phase	35
Figure 3-4-7-1	Gantt chart for Second Project Design phase	36
Figure 3-4-8-1	Gantt chart for Second Project Development and Implementation phase	36
Figure 3-4-9-1	Gantt chart for Second Testing phase	37
Figure 3-4-10-1	Gantt chart for Documentation phase	37
Figure 4-1-1	Overall architecture for MPSCS	38
Figure 4-4-1	Screenshot of MPSCS main interface	43
Figure 4-4-2	Screenshot of MPSCS login tab	44
Figure 4-4-3	Screenshot of MPSCS showing username after logged in	45

MOBILE PRICE SHARING AND COMPARISON SYSTEM

Figure 4-4-4	Screenshot of MPSCS searching by using product name	46
Figure 4-4-5	Screenshot of MPSCS searching by using Barcode Scanner	47
Figure 4-4-6	Screenshot of MPSCS adding a product name	48
Figure 4-4-7	Screenshot of MPSCS adding a product's price	49
Figure 4-4-8	Screenshot of MPSCS adding a shop name	50
Figure 4-4-9	Screenshot of MPSCS reporting	51

List of Tables

Table 2-1-1	List of existing solutions	5
Table 5-1-1	Functionality testing for MPSCS	52
Table 5-2-1	Usability testing for MPSCS	56

List of Abbreviations

CPI	Consumer Price Index	1
QR code	Quick Response Code	3
MPSCS	Mobile Price Sharing and Comparison System	7
B2C	Business to Consumer	9
IDE	Integrated Development Environment	28
ADT	Android Development Tools	28
APK	Android Package	28
SDK	Software Development Kit	28
PHP	PHP: Hypertext Preprocessor	29
3G	Third Generation	31
ICS	Ice Cream Sandwich	38
HTTP	Hypertext Transfer Protocol	38
URL	Uniform Resource Locator	39
ZXing	Zebra Crossing	40
1D	1 Dimension	40
2D	2 Dimension	40
W3C	World Wide Web Consortium	40
API	Application Programming Interface	40
V3	Version 3	40

Chapter 1: Introduction

1.1 Project background

According to report of Consumer Price Index Malaysia June 2012, Consumer Price Index (CPI) for Food & Non-Alcoholic Beverages had been increased for 2.9%, during period June 2011 till June 2012. The reflected changes had even contributed 54.9% for the total increased CPI (Department of Statistics, Malaysia, 2012).

The above figure conveyed a message, which life expenses would be increased as time goes by. And since we consumed food and beverages every day, the pressure caused by inflation would certainly hit us.

There were shops, or stores, that raised their prices periodically, especially during the time price of petrol increased. However, there would be some shops that remained prices unchanged, which was a relief, for the time like this.

However, finding best prices among shops, stores, or cities requires tremendous time and also effort. This was due to source of information may vary. The pricing information might come from sellers, through catalogues, or from their store. The information could also be shared from buyers, or potential customers.

A system was needed to organise those information, which enabled people to determine the best prices around easily. In addition, with the help of this system, we hoped to fight against inflation. Since better prices could be located easily, the shops that have higher price bracket would face the pressure, and thus reducing their price to a more reasonable level.

There were some websites tried to solve the problem, but their solution was not complete. For instances, some website brought sellers and users together, but incapable to show the best price in a specific area, for example: Lelong.com. Some website was capable to show pricing information of selected area, however, the prices could not be updated by users, where the price might be out-dated. Some website enable users to update prices, and capable to show best price in selected area, but their product were specified into some category, such as petrol.

1.2 Problem statements

Although there were many existing solutions as stated above, there are still problems to be solved. First problem is the parties who update the pricing information. Prices could be updated by users, or sellers. When prices are updated by users, prices may not be accurate, and the pricing information would be stay out-dated when lack of user inputs. When prices are updated by sellers, the prices would be more accurate, but the update might be slow, so the pricing information may be out-dated as well, if the seller seldom updates prices.

Second problem would be time needed to search for particular products. Users may face problems when searching for particular products, due the different name of similar product among the websites. This is because many websites did not take barcode into account, which could be used to distinguish products easily, even if they have a slightly different name.

Third, would be gathering or centralising pricing information. Although there were many price listing websites, but prices were seldom compared between sellers or shops. The pricing information was scattered, and users, which were potential customers, required doing extra work to find the best prices among the sites.

Fourth, would be the distance between users and shops. For example, if a product was the cheapest, but the shop selling it located a few hundred kilometres away from users' location, which would be meaningless for users. So shops that used to compare prices, should within users' vicinity.

1.3 Project Scope

This project was designed to provide a price sharing application in Android, which allowed users to update prices immediately using their mobile devices. Retail products would be the only concern for this project, since they usually consumed by large quantity, and would be easier to monitor because they have barcode to identify them. The application also had the following features:

- **Scanner/reader.** The price sharing application would be having bar code scanner/reader, which users could use to quickly identify products, thus making the price sharing process more efficient. It could also be used to scan QR codes, if the product code was not in barcode format. The price sharing application would be integrating with rear camera of the device to make use of this feature.
- **Price comparison of specific product in selected area.** The price sharing application would have a price comparison function that compares the price of similar products, in selected area according to location of the user. For example, users wanted to search for price of Coca-Cola (325ml) in Kampar area, the price application would show those stores having the product, along with the prices they are selling. Users would be able to quickly identify the best prices nearby, with minimum distractions.
- **Graph showing price changes of specific product of selected shop.** The price sharing application would have reporting, which would show users a graph. The

graph would show price changes of specific product, of selected shop, according to months. So that users would have a better idea if a shop had more stable price level, and sees the changes of price level for the last few months.

1.4 Objectives

The system would be developed in order to accomplish the objectives stated as below.

- **Allow users and store owners to share pricing information.** The system would allow users and store owners to share pricing of products. Both users and store owners could update the prices if they found that there is a mismatch of pricing information.
- **Allow quick product search,** since there are many products available, a quick way to search for particular product is needed. By using bar code scanner or reader, products could be found easily, thus making the price sharing process more efficient.
- **Allow users to search for pricing of particular product on specific area,** where users could easily compare pricing of particular products within a specific area. This would be useful for users, as users would be able to quickly identify the best price, without the distraction of too much information.
- **Allow users to have better understanding in product pricing,** where users could easily view the timeline of price changes, thus they were able to predict the pricing of specific product. They were able to learn if a shop increased its price frequently, or at unreasonable amount.

1.5 Contributions

The project could help users to find better price within their vicinity easily, which may helped to reduce living expenses during the days of inflation. Besides, it could help to create a healthy competition environment, where the prices of product would be more transparent, thus sellers would have to adjust their prices accordingly.

The project could bring a win-win situation for both users and sellers, because users now could find better prices easily, which could reduce their living expenses. And sellers, on the other hand, could have more customers, if their prices were reasonable compared to other competitors.

Chapter 2: Literature review

2.1 Existing solutions

Table 2-1-1 List of existing solutions

	Allow users to post comments/reviews on products	Range of products listed	Physical location or address of the store	Information updated by?	Search for prices in selected area	Barcode scanner or reader	Require registration
Price Check by Amazon	Yes	General	No	Sellers	No	Yes	No
WhatGas Petrol Prices	No	Specialised in petrol	Yes	Users	Yes	No	No
IPengguna.com	No	General	Yes	Sellers	Yes	No	Yes
Lowyat.net	Yes	Specialised in electronic devices	May include address in sellers' posts	Sellers	No	No	Yes
Mudah.my	No	General	No	Users and sellers	Yes	No	No
Lelong.com.my	No, only can comment on sellers	General	May include address in sellers' catalogue	Sellers	No	No	Yes
Mobile88.com	Yes	Specialised in mobile devices	No	Users and sellers	No	No	No
Shopparadiso.com	Yes	General	Yes	Sellers	No	No	No
Mobile Price Sharing and Comparison System (My solution)	Yes	General	Yes	Users and sellers	Yes	Yes	Yes

Table 2-1-1 briefly showed that the strengths and weakness of each existing solutions. 1Pengguna.com was the closest solution, because it could allow users to search for prices of particular in selected area, and it had addresses of those physical stores that were selling the product (Portal 1Malaysia Pengguna Bijak, 2012). The information was updated by Ministry of Domestic Trade, Co-Operatives & Consumerism, which information was actually from sellers themselves (Ministry of Domestic Trade, Co-Operatives & Consumerism, 2012). Thus, the price might be out-dated if it was not monitored periodically. Besides, it does not have bar code scanner/reader, which would help users in identifying products more easily. It did not allow users to post comments or reviews on a particular product.

The only existing solution that has bar code scanner/reader was Price Check by Amazon (Google Play, 2012). It allowed users to post comment or reviews on a particular product, for buyers' further consideration. However, it lacked of the strengths of 1Pengguna. Mobile Price Sharing and Comparison System (MPSCS) would be combining strengths of the various existing solutions from different criteria, and covering the weaknesses.

2.2 Critical remarks of existing solutions

2.2.1 Price Check by Amazon.com

Price Check was an application made by Amazon, which allow users to search for product prices that were available on their website. It had general products, and allowed users to upload photos of products. It also allowed users to post comments/review on the product, where other users may make further consideration for other users. Besides, it also had bar code scanner, which allowed users to search products more efficiently. However, it did not allow users to compare prices in selected area, and also the physical location of the store. Pricing information was update by store owners, which may not be able to reflect price changes in a timely manner.

2.2.2 WhatGas Petrol Prices

It was an Android application by Chris Bailey. It allowed users to update prices, which prices could be update in a timelier manner. It also allowed users to compare price in a selected area, which made them spent lesser time in finding best price in a selected area. Users could also easily locate the stores, because the physical locations were shown on map. However, it was only specialised in petrol, which was not scalable into other products.

2.2.3 1Pengguna.com

Pengguna.com was a price sharing portal by Ministry of Domestic Trade, Co-Operatives & Consumerism. It allowed users to search for best price in selected area for a particular product, and had address of the store, where users could go to the store and purchase. It showed pricing information for daily food products, and the information was updated by the Ministry, which in this case, were not inputs from users. Users could neither update prices, nor upload photos of the products. It also

lack of bar code scanner/reader to allow quick search for product. This was closest solution to the project, but there were still improvements.

2.2.4 Lowyat.net

Lowyat.net was actually a forum, which had a section for trading. However, the products were more focused on electronic devices. Prices were updated by sellers, and photos could be uploaded by both sellers and users. It also allowed users to post comments/reviews; however, registration was required before they could do so. The main problem was sellers may not have a physical store, and thus it could not compare prices in a selected area easily.

2.2.5 Mudah.my

Mudah.my was a website which allows users to post advertisement for limited period without charge, for buying or selling a products or services. The products range listed was very general, which includes vehicle, job and services, properties, and home and personal item. Since advertisements could be posted by sellers or users, we could say that the price was updated by both sellers and users. It allows users or sellers to upload photos of the products, but they could not leave comments/reviews on the advertisement. It also allows users to search for best price in selected state, which may filter some unwanted information, but the context was still too wide.

2.2.6 Lelong.com.my

Lelong.com.my was a Business to Consumer (B2C) e-commerce website. Sellers would list their products, and state their prices, which means the prices was updated by sellers. Photo upload was available only for sellers. But it had the same problem with Lowyat.net, which sellers may not have a physical store, and thus it could not compare prices in a selected area easily. Users were not allowed to comment on a

particular product, but they could comment on sellers, however, this also requires registration.

2.2.7 Mobile88.com

Mobile88.com was a website specialised for mobile device. Buyers and sellers could state the price they willing to pay / offer, so we could say that the price was updated by both sellers and buyers. It also allowed users to post comments and reviews on particular mobile device, unlike Lowyat.net and Lelong.com.my, registration was not required. However, it could not compare best price on selected area, and there were no information on physical location of the seller.

2.2.8 Shopparadiso.com

Shopparadiso.com was website that Online Shopping Search Engine that allowed users to search for products, and compare prices. It contained price from online and local stores, and had address of physical store, so that users could go to the store and have a touch on real product. The price was updated by seller, and had general products. Users could post comments or reviews, on the website, without registration. It, however, was incapable of showing best prices in a selected area. Besides, it did not allowed users or sellers to upload photos. There were also some errors, when users were being redirected to online store's website.

Chapter 3: Methodology and Tools

3.1 Methodologies

3.1.1 Use Case Diagram

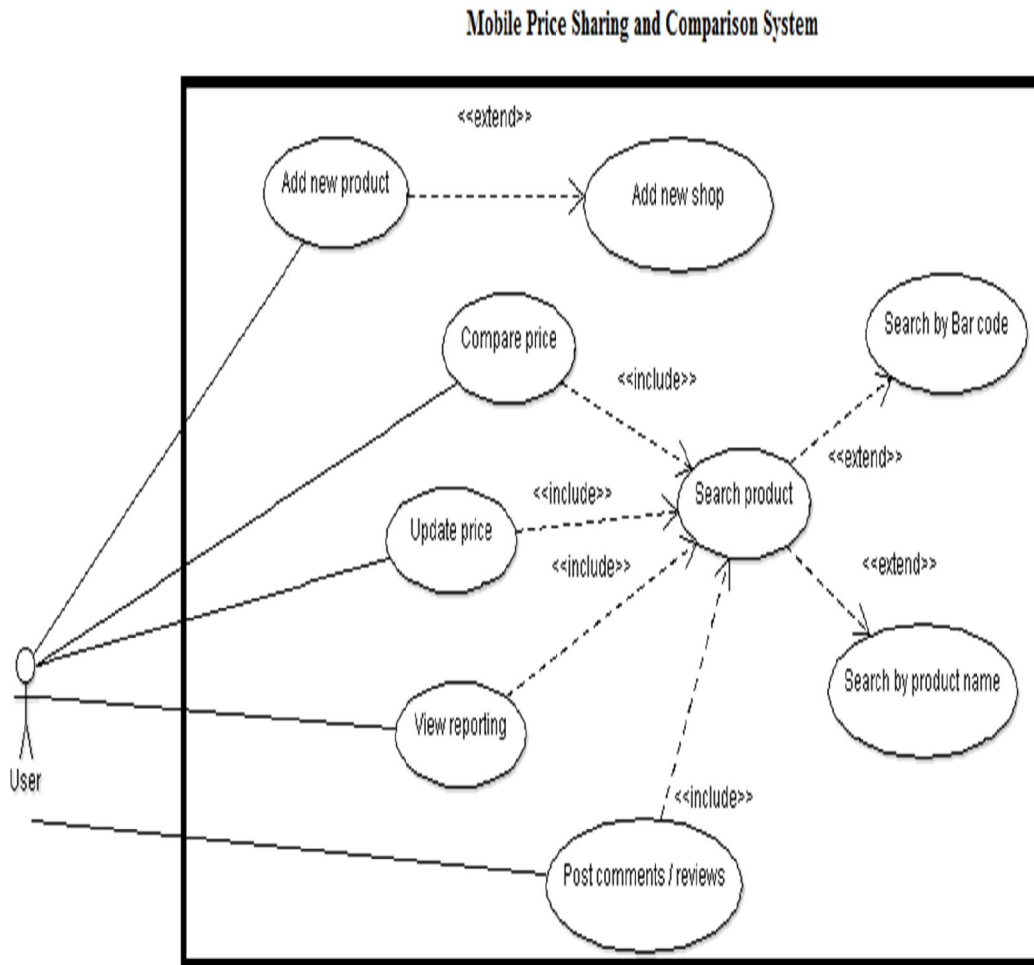


Figure 3-1-1-1 Use Case Diagram for MPSCS

The use case diagram had been constructed according to the key actions that users could perform. Users could update product prices, search for products, compare price and view reporting. Users and sellers could also upload photos, post reviews, check reviews, but we would not focus on that particular use case, and would leave it for future work.

3.1.2 Class Diagram

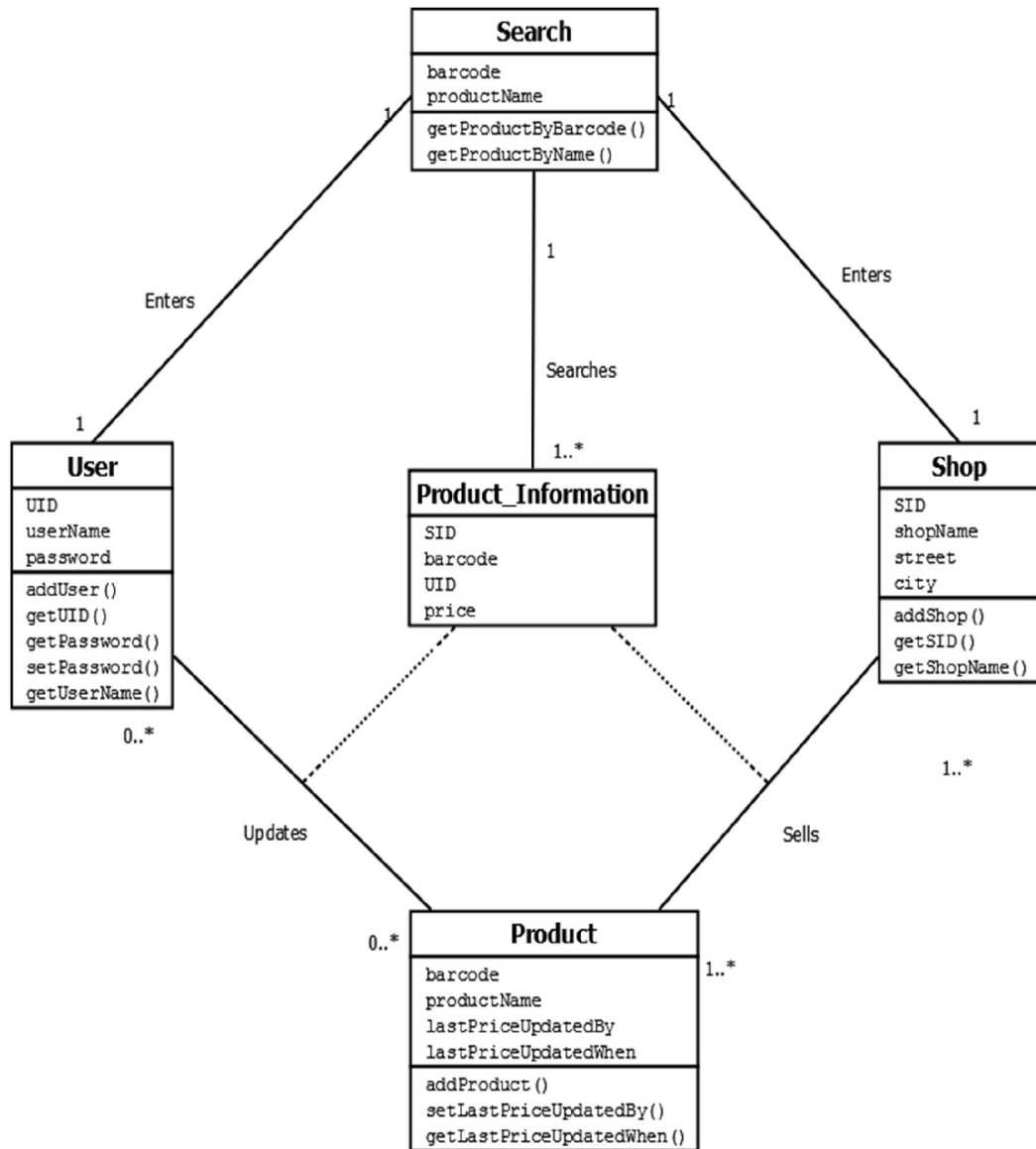


Figure 3-1-2-1 Class Diagram for MPSCS

MPSCS would be having five main classes, which were User, Seller, Product, Search, and one associative class, which is Product_Information. Product_Information would be responsible in holding record for different information of a particular product, such as price, review, and who submitted the price and/or review.

3.1.3 Entity-Relationship Diagram

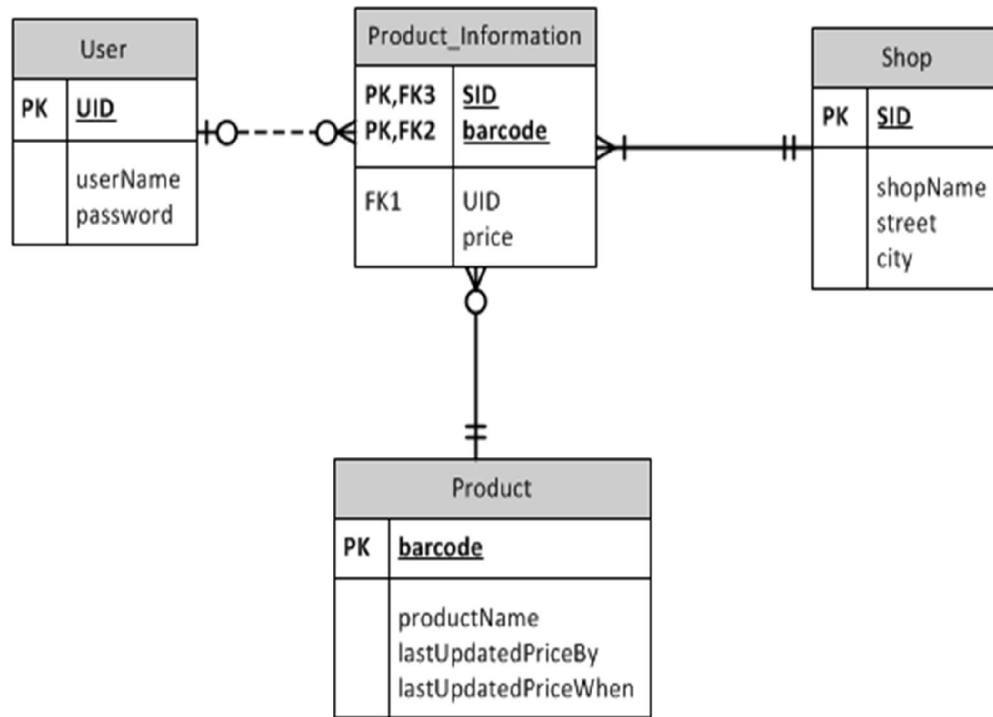


Figure 3-1-3-1 Entity-Relationship Diagram for MPSCS

In this Entity-Relationship Diagram, Search class did not appear, because Search class does not store any values. It would still receive values, but rather than storing them, Search class would perform actions on them. For example, if user entered product name “cola”, it would not store it, but would perform a search, that searches for products that contained the word “cola”.

3.1.4 Activity Diagram

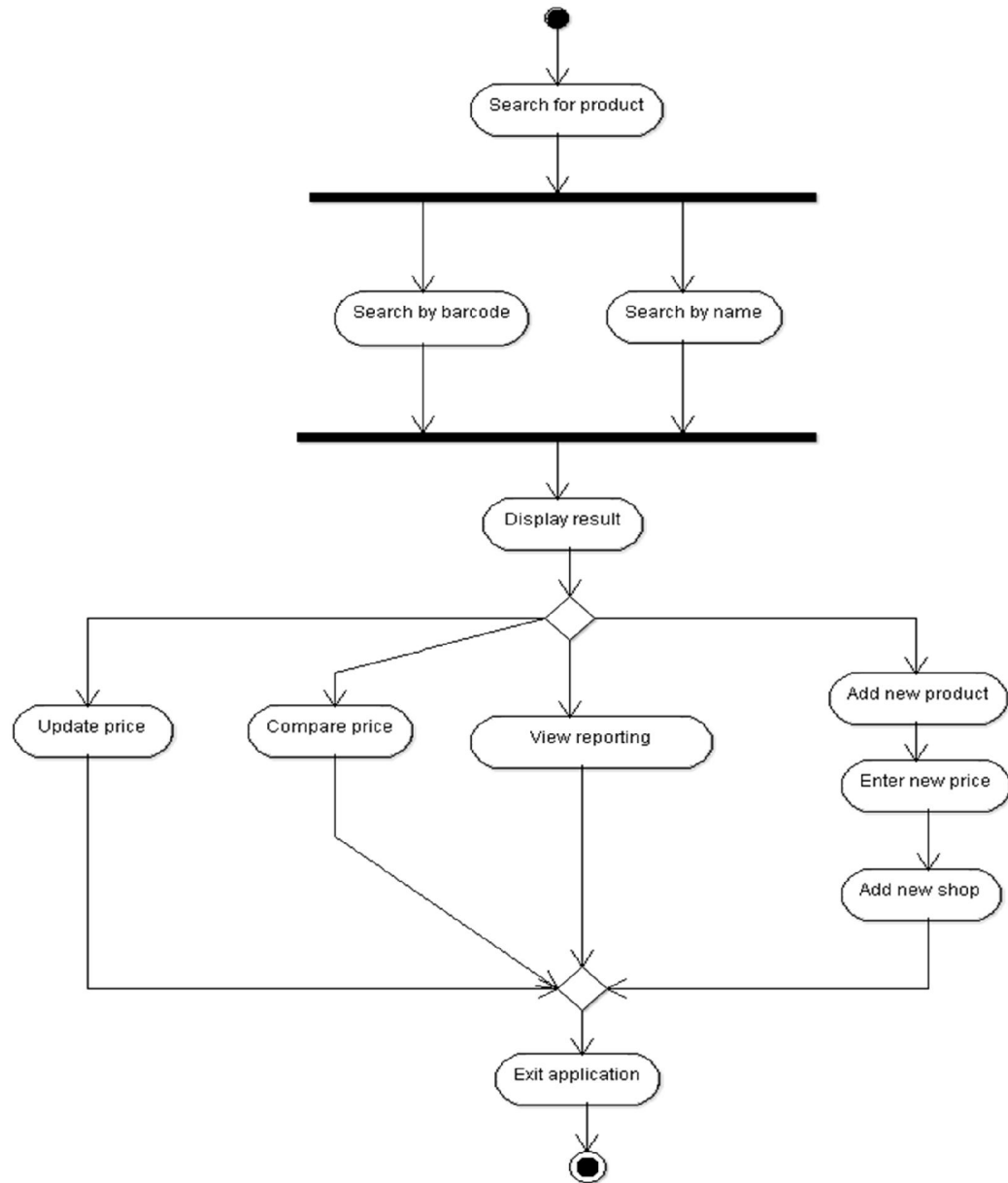


Figure 3-1-4-1 Activity Diagram for MPSCS

As shown in Figure 3-1-4-1, firstly, users and sellers would be searching for products. After that, they would be deciding the next action, where they would choose to update price of the product, or post a review, or to add a new product, or other actions as stated in the figure.

3.1.5 Sequence Diagram

The sequence diagram of MPSCS had been broken down into smaller sequence diagrams, according to use case of MPSCS. This was to avoid confusion, as at a point users would have a set of choices. Basically, almost all sequence diagrams would start with a search, as pointed at section 3.4, except for add product, where users could add products before or without searching. The sequence diagrams were constructed based on six use cases, which was add location, add product, compare price, update price, and view reporting.

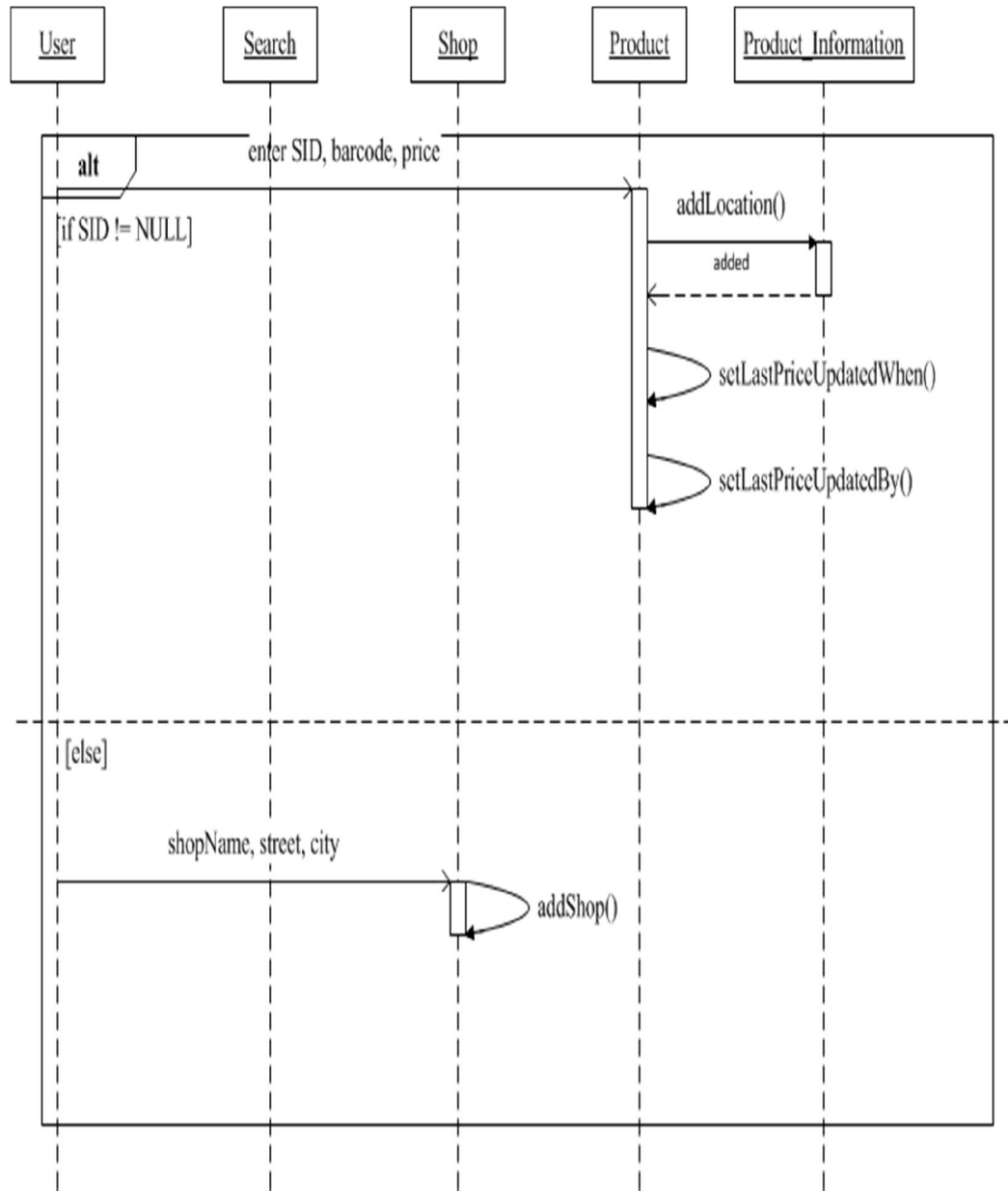


Figure 3-1-5-1 Sequence Diagram for add location in MPSCS

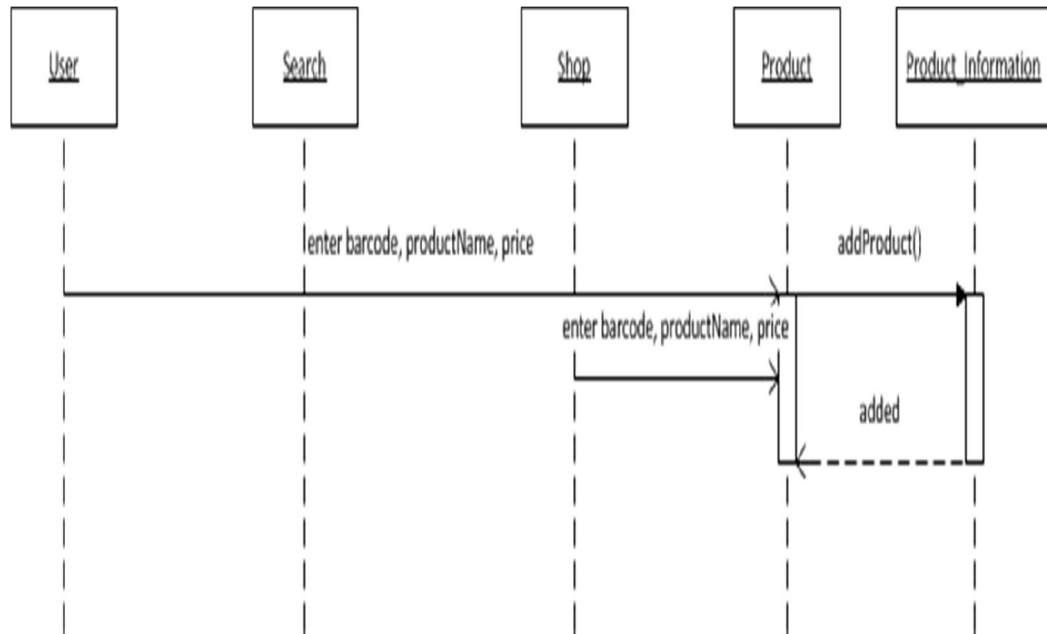


Figure 3-1-5-2 Sequence Diagram for add product in MPSCS

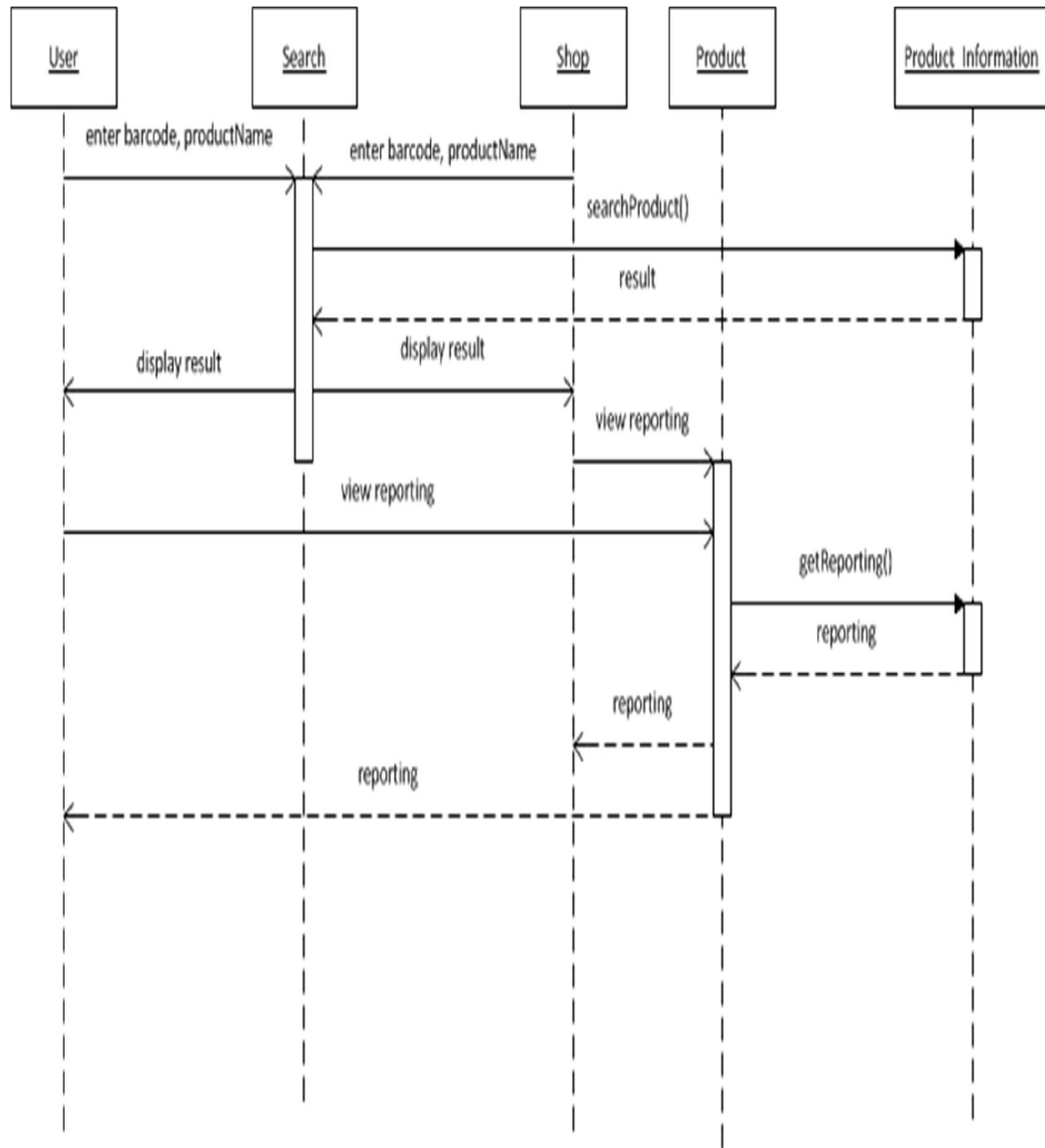


Figure 3-1-5-3 Sequence Diagram for view reporting in MPSCS

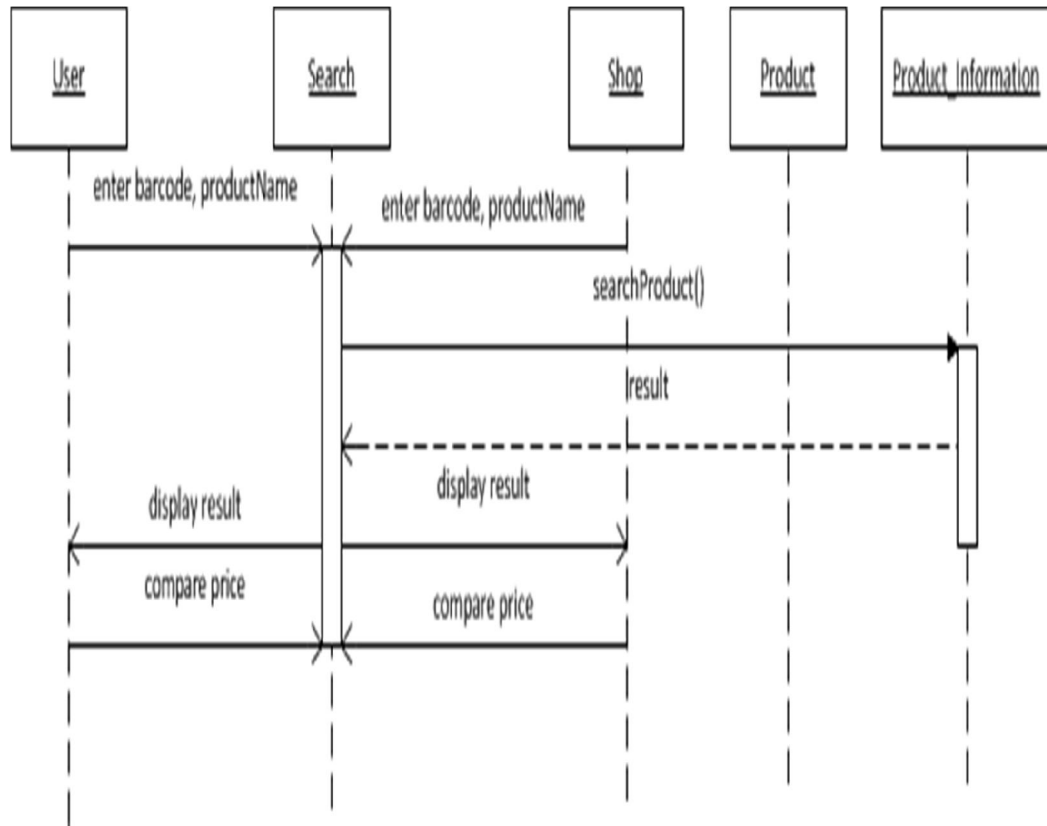


Figure 3-1-5-4 Sequence Diagram for compare price in MPSCS

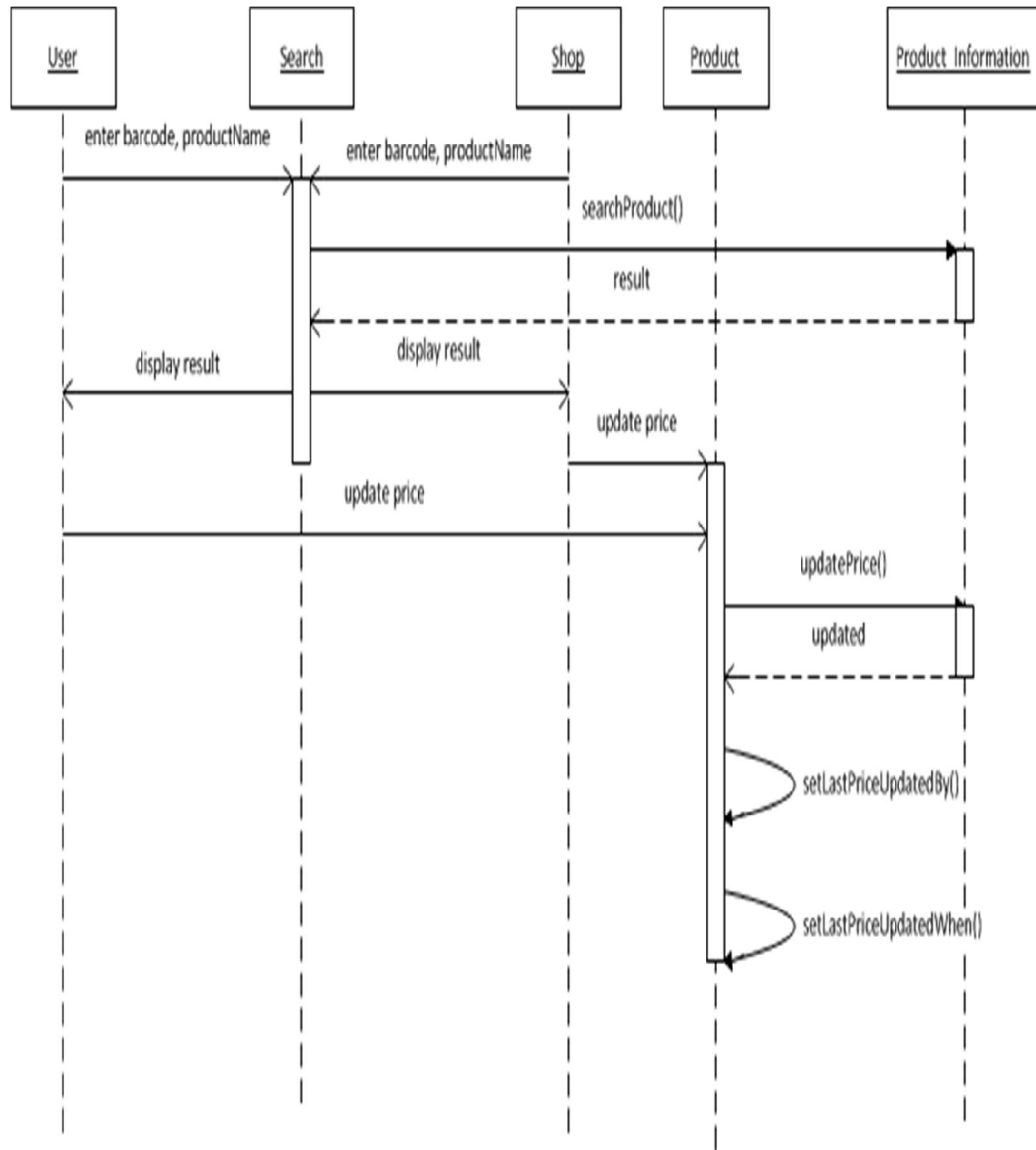


Figure 3-1-5-5 Sequence Diagram for update price in MPSCS

3.1.6 Collaboration Diagram

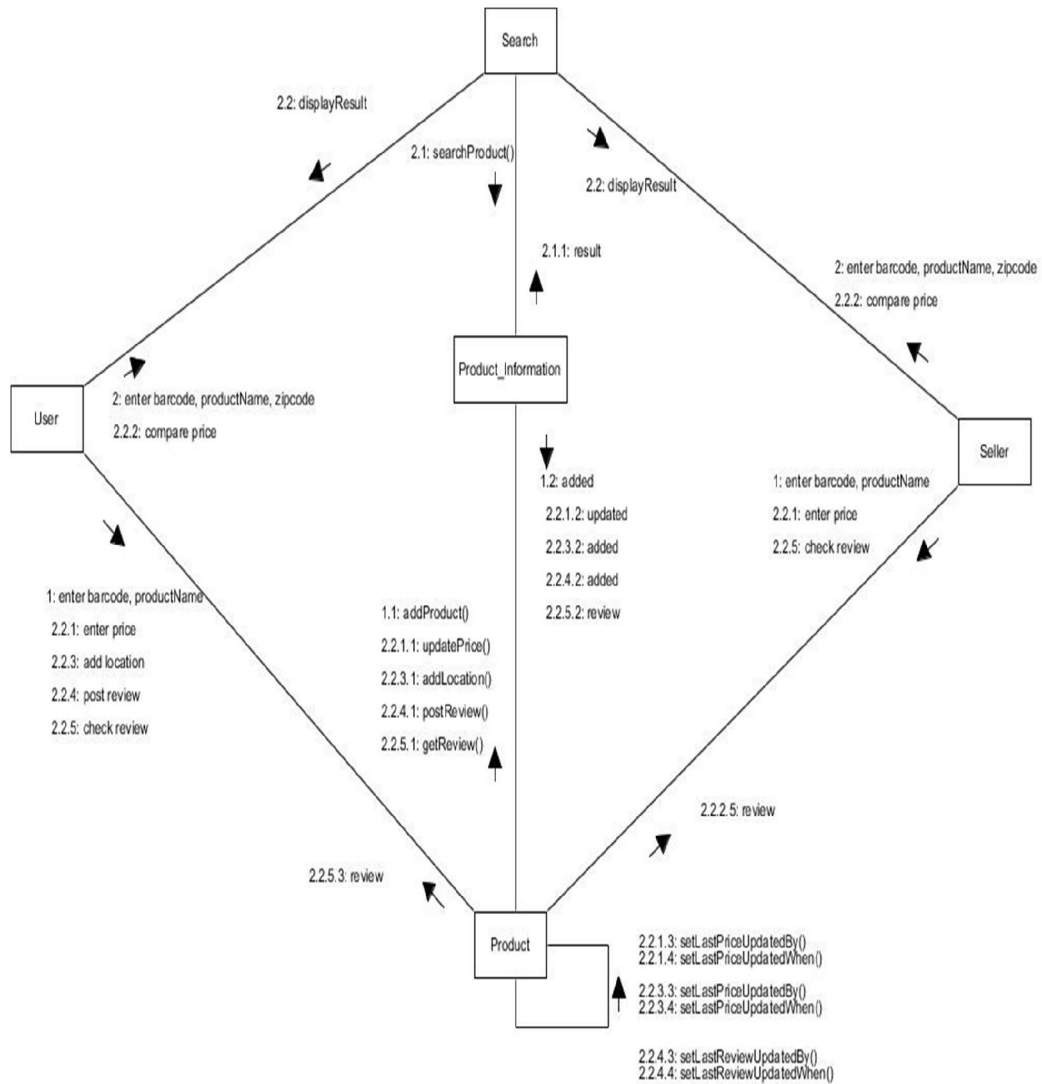


Figure 3-1-6-1 Collaboration Diagram for MPSCS

The collaboration diagram showed the interactions of the classes, some actions were performed. Search class would be communicating with Product_Information for search results, and Product class was responsible to create information about products, and store them in Product_Information.

3.1.7 State Diagram

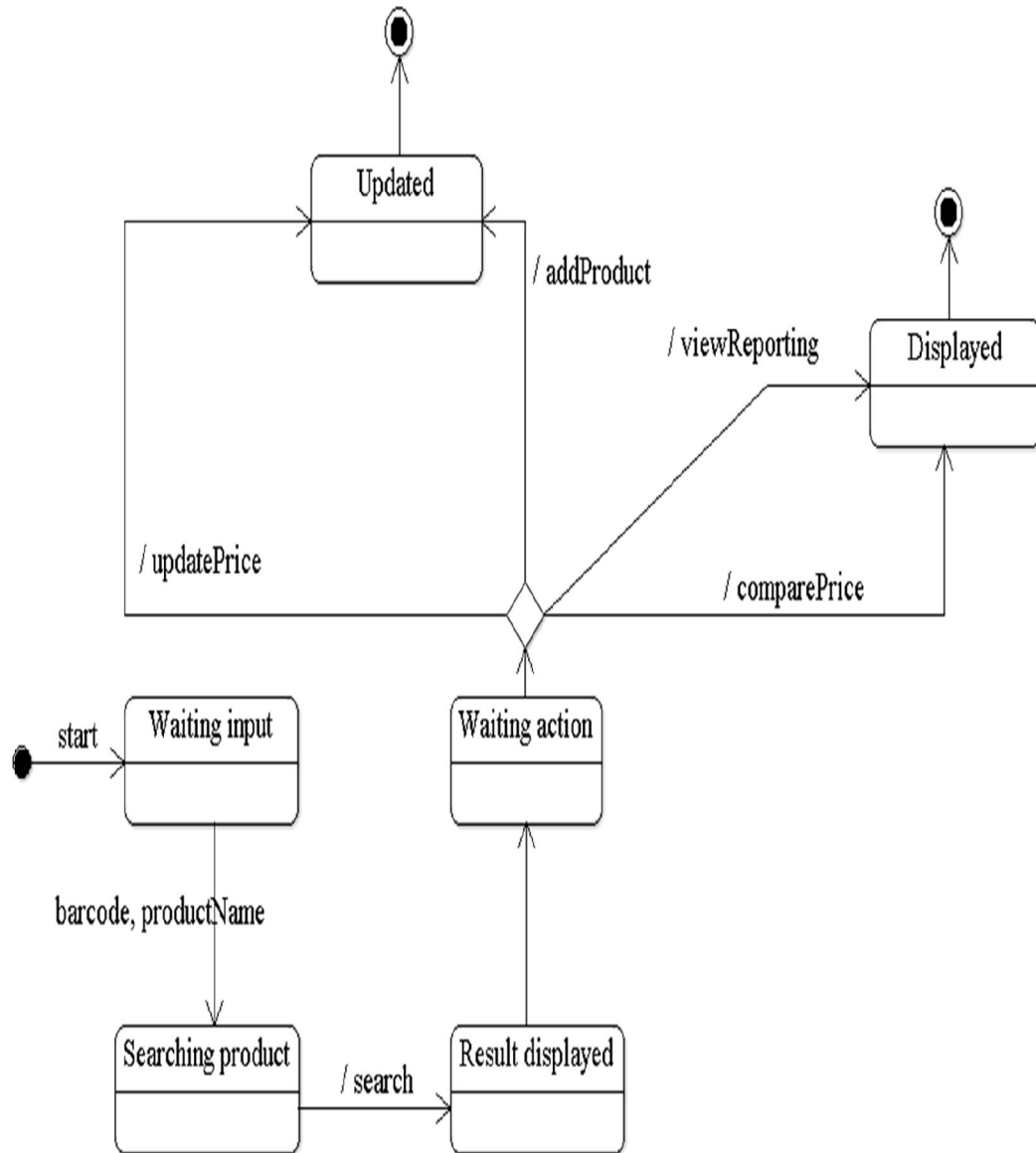


Figure 3-1-7-1 State Diagram for MPSCS

The state diagram showed how MPSCS changed from one state to another during its execution, and during it performs actions. It would be in “Waiting input” state while waiting users or sellers to input searching criteria, such as barcode and product name. And it turns to “Searching product” when it starts to perform searching. Next would turn to “Displaying result” state when display search results, and afterwards “Waiting action” to wait for users selection on which action to perform. And then, it would turn to “Updated” state or “Displayed” state, depending on users’ choice of action.

3.2 Technology involved

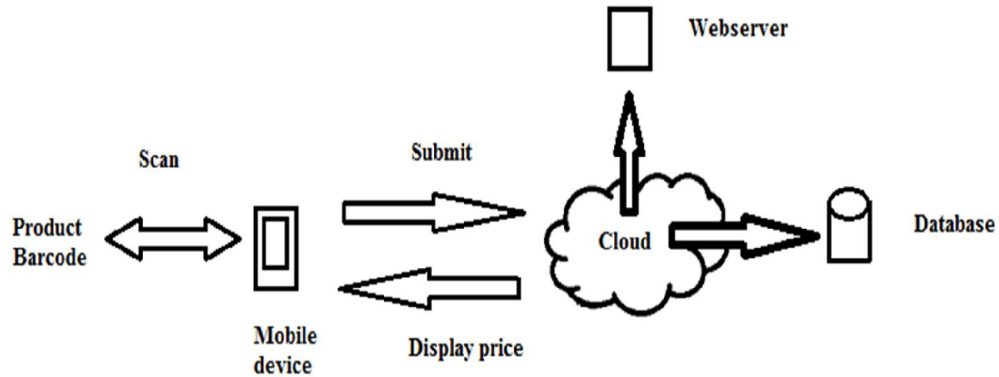


Figure 3-2-1 High level Architecture view for MPSCS

- Eclipse Integrated Development Environment (IDE)

Eclipse IDE would be used for us to develop the application for Android devices. There is an plugin named Android Development Tools (ADT) that can helps us set us Android projects, and even export it to APK files, which is Android application package file (Developer.android.com, 2012).

- Android Software Development Kit (SDK)

The APK file created by Eclipse IDE can be tested using Android SDK. Using Android SDK, application that was built can be tested in an emulated environment, before deploying it to real Android devices. This enables us to identify any possible errors, and also prevent us from harming the real devices, before the system is ready.

- PHP: Hypertext Preprocessor (PHP) Programming Language

Server is needed to serve the request of users, such as pricing information of particular product. PHP is a server side language that used to interact with the database. It would retrieve information from the database, and fetch it to users.

- MySQL database

To store information, a database is needed. MySQL is selected as the database, because it is open source. Besides, it is free, which would lower our cost of development.

3.3 Specifications and requirements

3.3.1 Functional requirement

- Login/Logout

Users were able to log in to the system, or log out from the system. System was able to identify the logged in user.

- Barcode Scanner

The Barcode Scanner could be invoked by pressing a button of the webpage, and pass the scanned result to MPSCS.

- Geolocation

MPSCS would be able to identify the current location of users, and do searching based on the location.

- Reporting

Users would be able to search product, using both barcode and also product name.

- Search product

Users can view a graph, showing the timeline of price changes of a product.

- Add price

Users could add a price when the price was not existed in the database.

- Add product

Users could add a product when the product was not existed in the database.

- Add shop

Users could add a shop when the shop was not existed in the database.

- Update price

Users could update the price of a product.

3.3.2 Hardware and software requirement

- Mobile device with rear camera

Rear camera was needed in order to work with Barcode Scanner.

- Android Operating System

Android version 2.3 or higher, Android version 4.0 and above was recommended.

- Internet connection

Internet connection was required for MPSCS to connect to server; users can use over 3G or Wi-Fi.

- Barcode Scanner

While MPSCS can still be used without Barcode Scanner, users were recommended to install it to get the full use of MPSCS.

3.4 Gantt chart

3.4.1 Project Planning phase

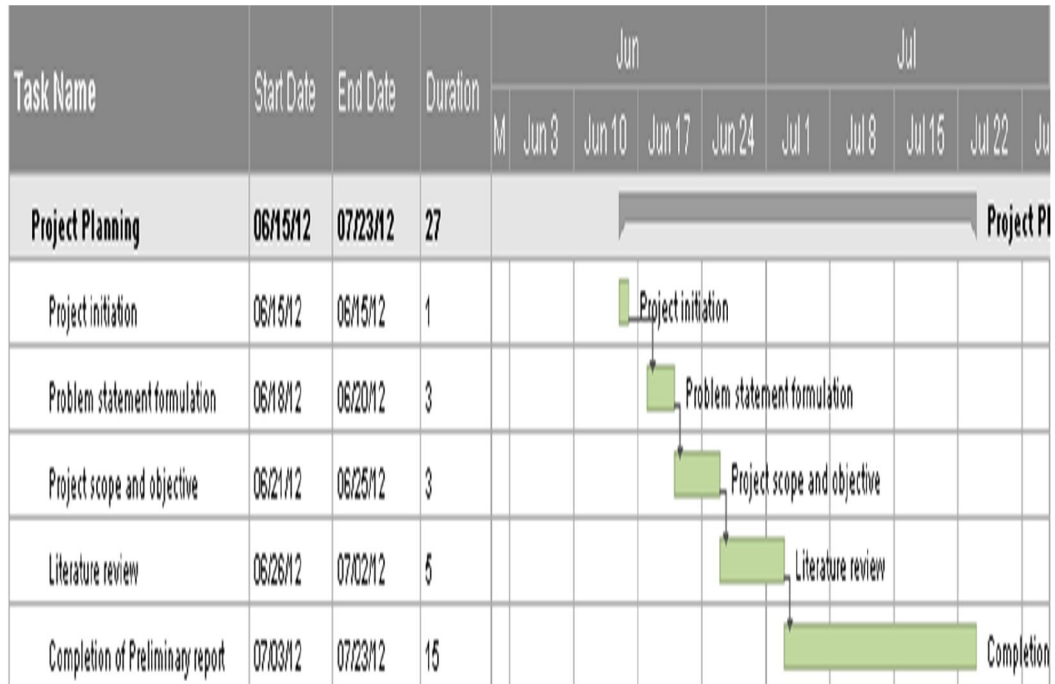


Figure 3-4-1-1 Gantt chart for Project Planning phase

3.4.2 Project Analysis phase

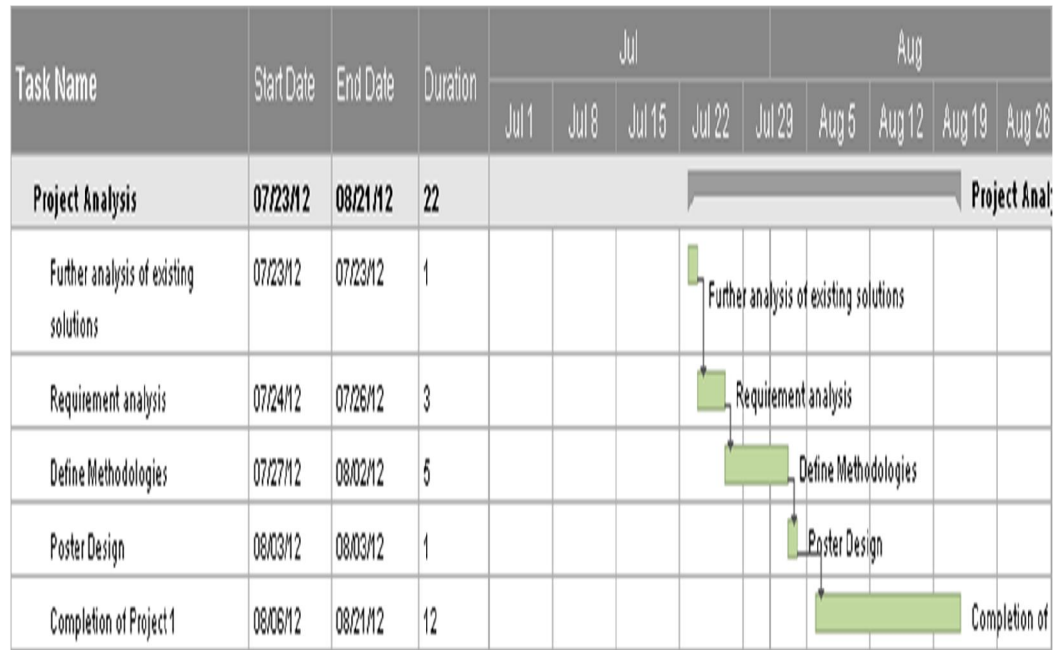


Figure 3-4-2-1 Gantt chart for Project Analysis phase

3.4.3 Project Design phase

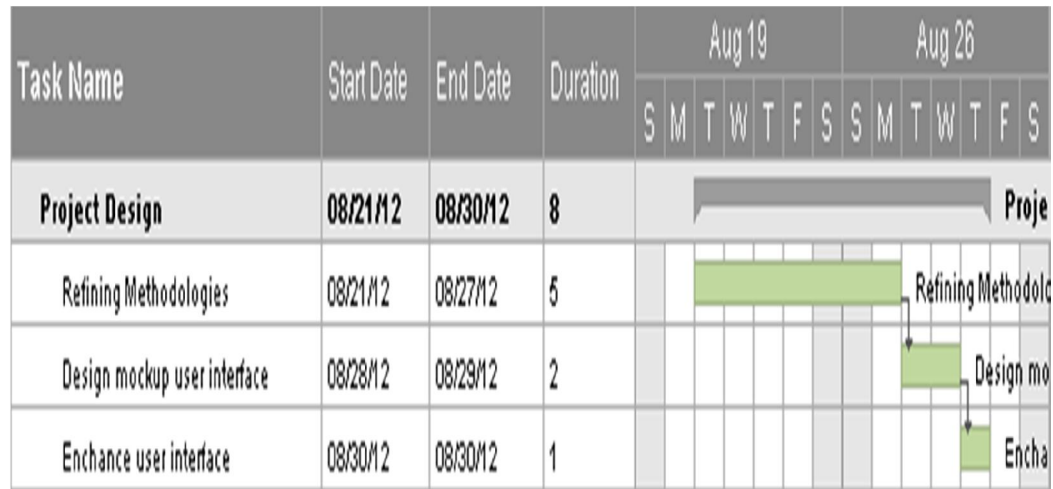


Figure 3-4-3-1 Gantt chart for Project Design phase

3.4.4 Project Development and Implementation phase

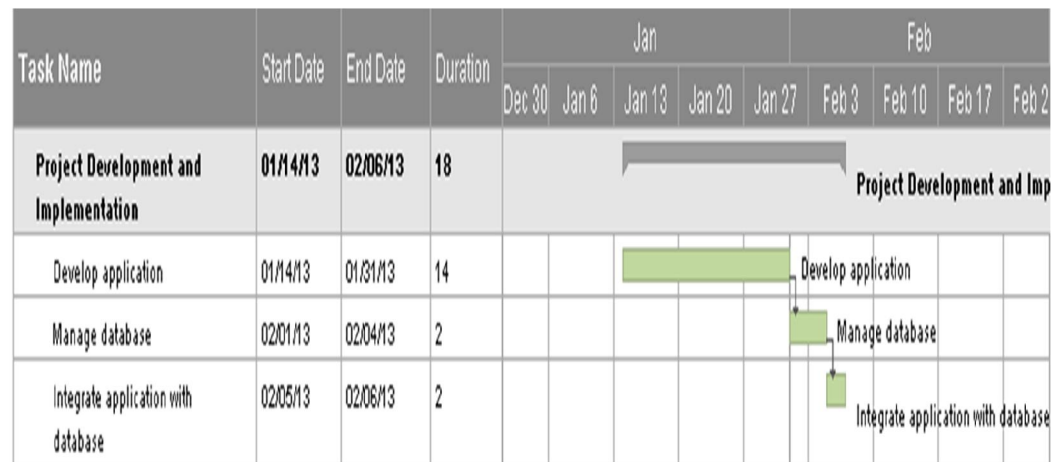


Figure 3-4-4-1 Gantt chart for Project Development and Implementation phase

3.4.5 Testing phase


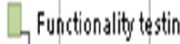
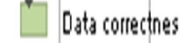
Task Name	Start Date	End Date	Duration	Feb				
				Ja	Feb 3	Feb 10	Feb 17	Feb 2
Testing	02/06/13	02/08/13	3	 Testing				
Functionality testing	02/06/13	02/06/13	1	 Functionality testing				
Data correctness	02/07/13	02/08/13	2	 Data correctness				

Figure 3-4-5-1 Gantt chart for Testing phase

3.4.6 Second Project Analysis phase

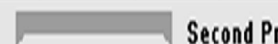

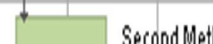
Task Name	Start Date	End Date	Duration	Feb				
				Ja	Feb 3	Feb 10	Feb 17	Feb 2
Second Project Analysis	02/08/13	02/19/13	8	 Second Pro				
Second Requirement analysis	02/08/13	02/12/13	3	 Second Requirement ana				
Second Methodologies Defining	02/13/13	02/19/13	5	 Second Meth				

Figure 3-4-6-1 Gantt chart for Second Project Analysis phase

3.4.7 Second Project Design phase

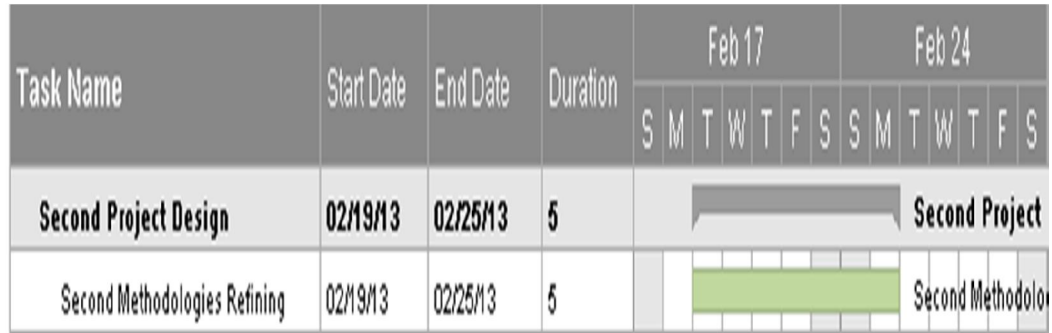


Figure 3-4-7-1 Gantt chart for Second Project Design phase

3.4.8 Second Project Development and Implementation phase

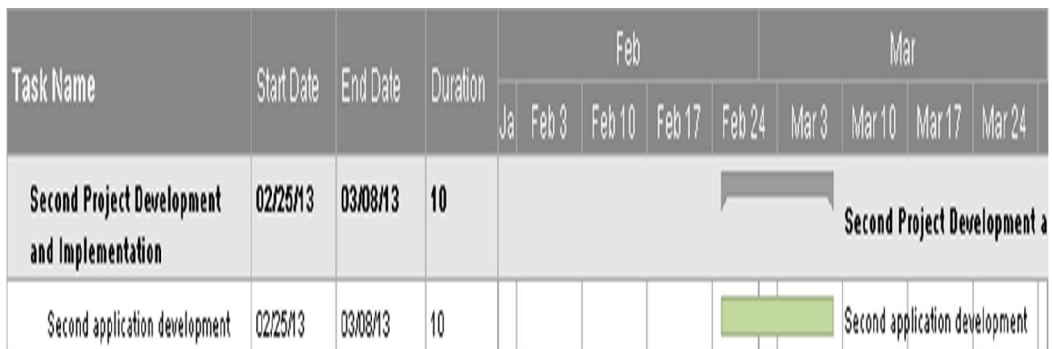


Figure 3-4-8-1 Gantt chart for Second Project Development and Implementation phase

3.4.9 Second Testing phase

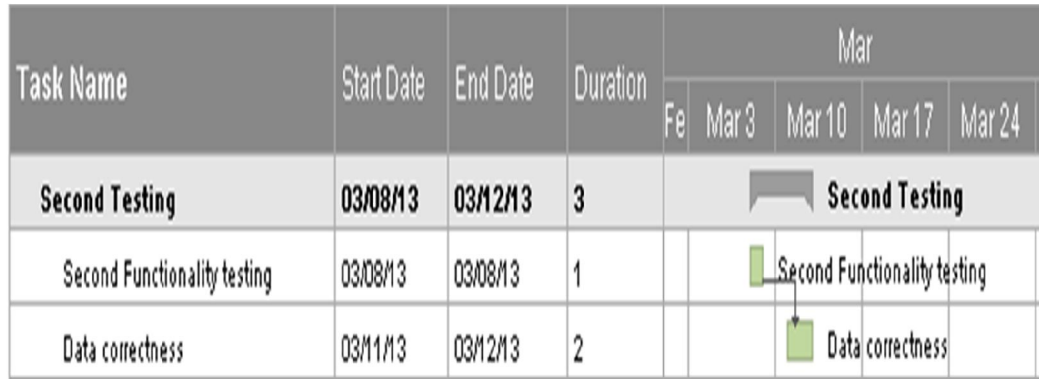


Figure 3-4-9-1 Gantt chart for Second Testing phase

3.4.10 Documentation phase

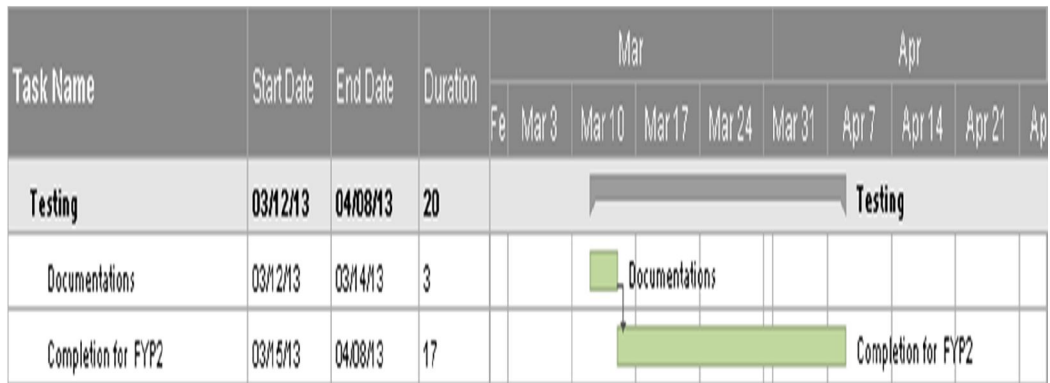


Figure 3-4-10-1 Gantt chart for Documentation phase

Chapter 4: Implementation

4.1 Overall architecture

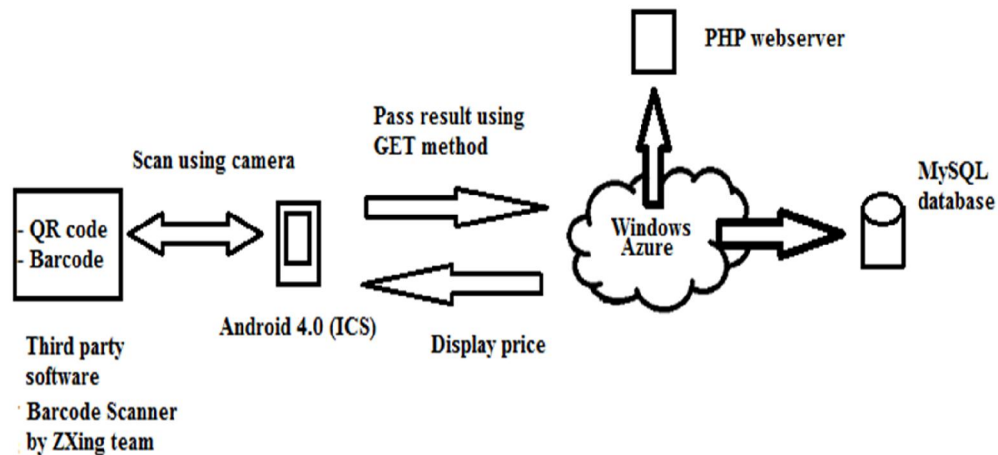


Figure 4-1-1 Overall architecture for MPSCS

Figure 4-1-1 shows the overall architecture that we implemented. We have chosen Android as our development platform, specifically Android 4.0 (ICS), together with Eclipse as our development tool. Eclipse IDE had a plugin named Android Development Tools (ADT) that helped us set up Android projects, and even compiled it to APK files, which would be an Android application package file. We built an application, MPSCS in Android that shows webpages from the PHP webserver, which resided in the cloud service, Windows Azure (Windows Azure, n.d.). MPSCS would communicate with the PHP webserver through HTTP requests and responses. If users needed to use Barcode Scanner, there would be a button in the webpage, and Barcode Scanner would be invoked if users clicked it. MPSCS would submit the scanned result to the PHP webserver. The webserver would then serve webpages that would display

appropriate data, which were retrieved from MySQL database, which resided in Windows Azure too.

4.2 MPSCS

MPSCS was built as an Android application. It made use of Android's WebView (WebView | Android Developers, n.d), which was used to load webpages. In our implementation, we used it to load webpages, served by the PHP webserver resided in Windows Azure. We needed to enable JavaScript and also Geolocation for the WebView, to ensure the functionality of MPSCS. Upon launched, MPSCS, which used WebView, would request page from the URL that we specified, which in this case, would be PHP webserver inside Windows Azure. Most of the functionalities of MPSCS were on the webpage itself, so JavaScript would need to enable.

4.2.1 Reporting

Reporting is used in MPSCS, to show the average price of a product sold by the shop, by particular month. When the average price of different months combined, there would be a timeline of price changes of a product, sold by particular shop. Users could use the reporting to learn and predict the pricing trend of the product, by the shop. For the reporting used in MPSCS, we had used a JavaScript plugin, named Highcharts (Highcharts 2013). To show the reporting, firstly, we fetched appropriate records from MySQL database using PHP. Then, we put the data into the plugin, along with product name. Average price level would be the x-axis, while y-axis would be months. We would also need to do some configurations to the charting that we need. When users requested it, the graph would be generated, by Highcharts. Users were able to click on the graph. As users clicked on the dots, the exact amount of average price level of the month would be shown.

4.2.2 Scanner

For the scanner, we used a third party application, which was Barcode Scanner developed by ZXing. Barcode Scanner was able to scan 1D/2D bar codes, which proved useful when product code were printed as QR codes. Users must have Barcode Scanner installed as standalone application before they can use it, because it was not integrated within MPSCS itself. When user clicked on a button on the webpage to invoke Barcode Scanner, Intent is used to invoke the Barcode Scanner application. The rear camera of the mobile device would be used for scanning. After a scanned result was returned, MPSCS would submit the result through GET value. The PHP webserver would then perform a search for the submitted GET value.

4.2.3 Location-based searching

MPSCS featured shop searching based on users' location. We used W3C Geolocation API (Geolocation API Specification, 2012), coupled with Google Maps JavaScript API V3 to achieve our location-based searching. First, we used W3C Geolocation API, to get coordinates of our devices. In order for this to be worked, the mobile device must enable Google's location service, and Geolocation must be enabled for the WebView, alongside with permission to get mobile device's position. Second, we used the Google Maps JavaScript API V3, which reverse geocode into address of our current location, after we provided the acquired coordinates using W3C Geolocation API. We would search for the long name of the city, and that will be used as a criterion when searching prices or shops.

4.3 Cloud service

We used cloud service rather than shared hosting to host our webserver and database; because cloud service could be have more flexibility and scalability (Janakiram, 2012) (Dialogic Corporation, 2010). The flexibility and scalability of cloud service was we were able to choose hardware or software that was used, that fit our business nature. For example, we could choose to run our PHP webserver in a Linux machine, or Windows machine, we could also choose how many cores for the processor, how many processor are needed, even the version of PHP. We could scale up the resources when in a growth, such as increasing the number of processors, or scale down when resources are underutilized.

Another interesting part of cloud service was on-demand self-service scaling, which traditional shared hosting services could not offer. For example, if there were peak-load season, we could scale up our resources to meet the demand, and scale it down after the peak-load season, so we could still offer the same user experience, without external technical assistance from service provider. Another example would be the total limit of bandwidth. For shared hosting, the limit of bandwidth was fixed as we subscribed. So, once the bandwidth limit was reached, subsequent users would not be able to reach our webserver. But for cloud service, we could increase the limit of bandwidth within a few clicks.

We used Windows Azure as our cloud service, because it was well documented, so we could set up our account easily by following the documentation. Unlike shared hosting, we did not wait for days for our account to be set, but rather several minutes.

4.3.1 Server

We used PHP for our server language, because it can work very well with MySQL. The installation process was skipped, because we were using cloud service. We would need to specify the version of PHP to be used. We would need to connect to MySQL database, which resided in Windows Azure. The server would be our main website for MPSCS, which most of the functionalities lied within.

4.3.2 Database

We used MySQL as our database, because it worked very well with PHP. As we were using Window Azure, installation process was skipped. We were needed to specify which database to use, and linked it to our website. We would need to configure the connection string to the database, so that our website would be able to connect to the database. For database storage, we could scale it, so that increased amount of data could also be stored.

4.4 Screenshots

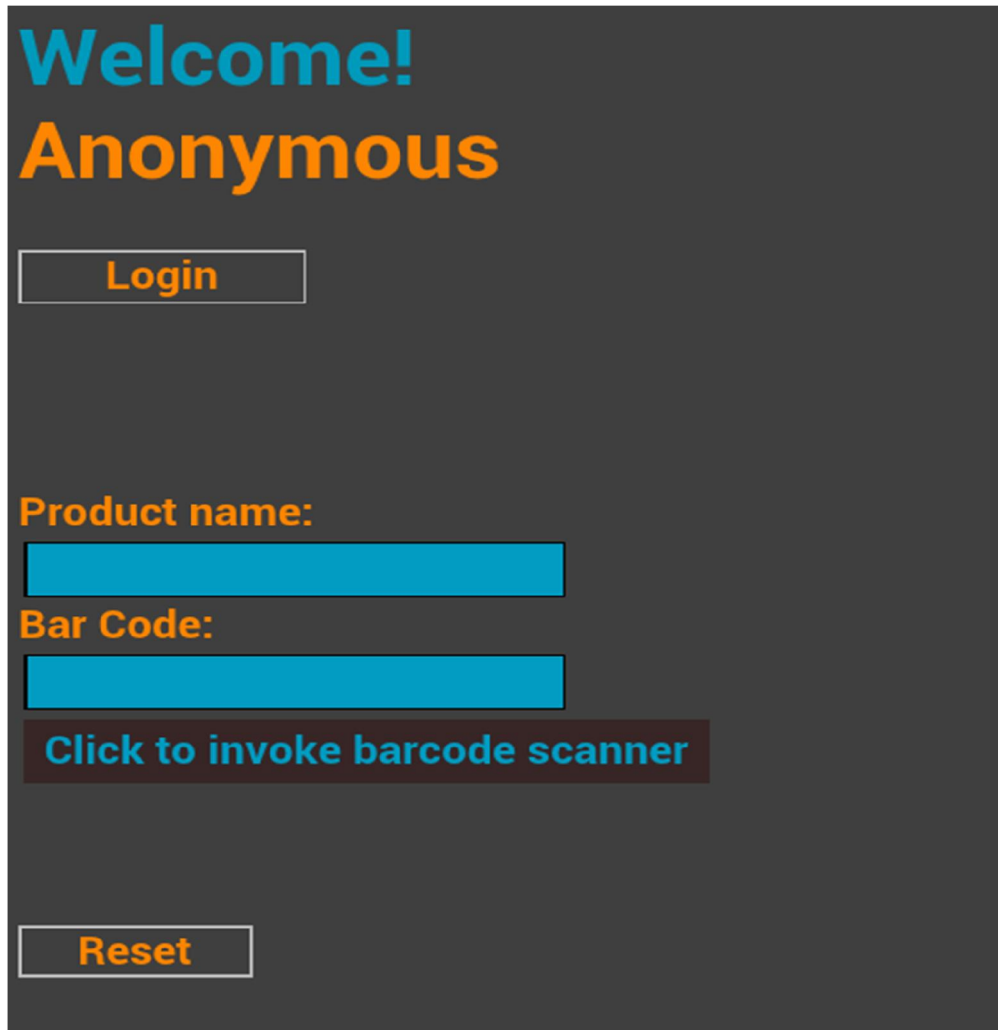


Figure 4-4-1 Screenshot of MPSCS main interface



Figure 4-4-2 Screenshot of MPSCS login tab

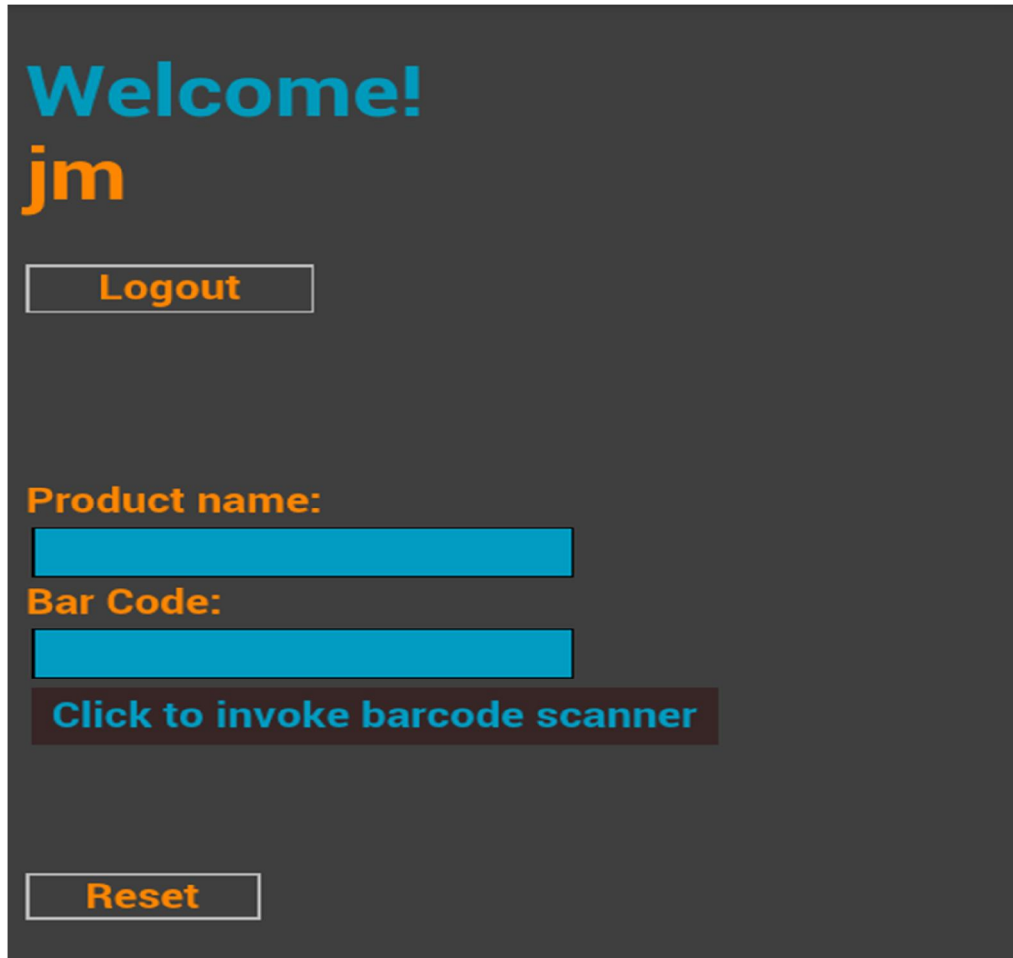


Figure 4-4-3 Screenshot of MPSCS showing username after logged in

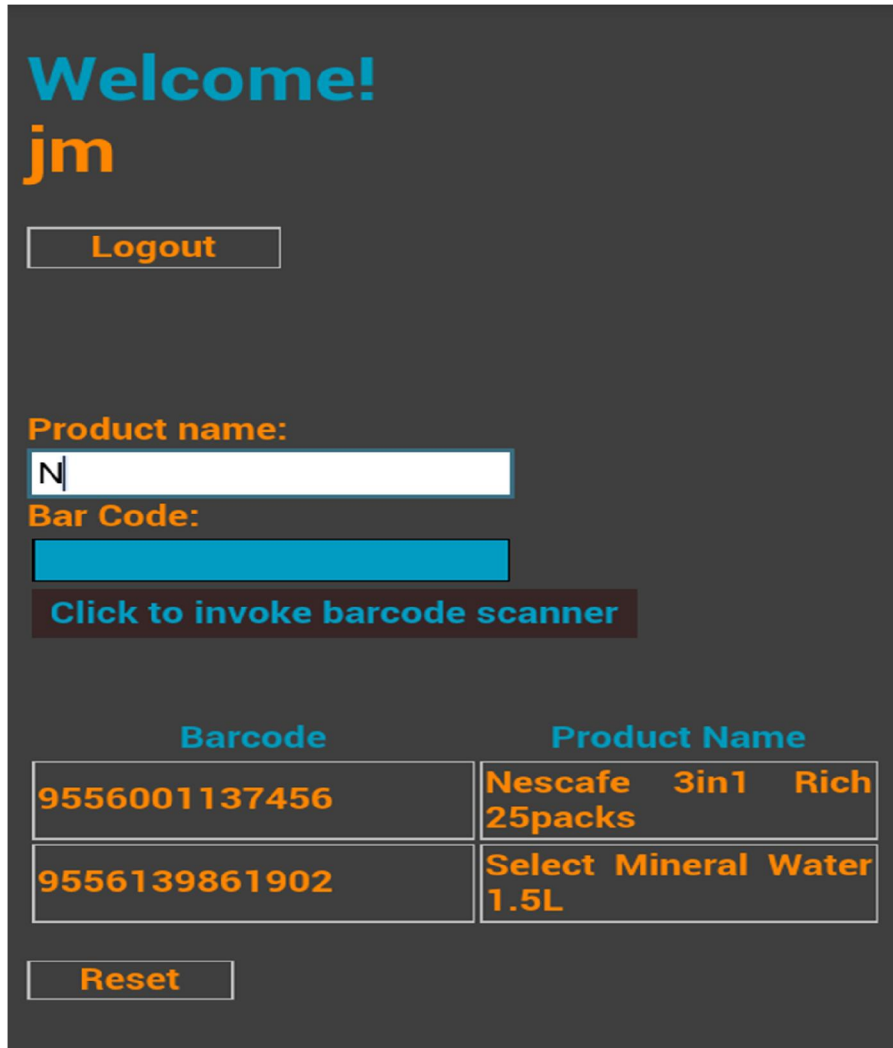


Figure 4-4-4 Screenshot of MPSCS searching by using product name

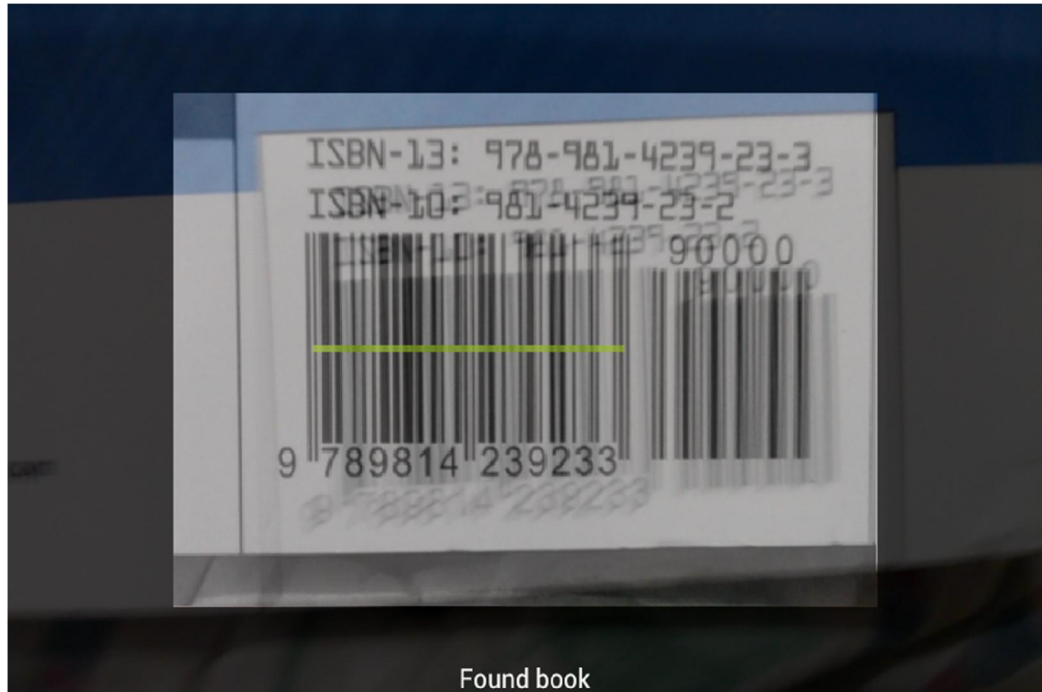


Figure 4-4-5 Screenshot of MPSCS searching by using Barcode Scanner

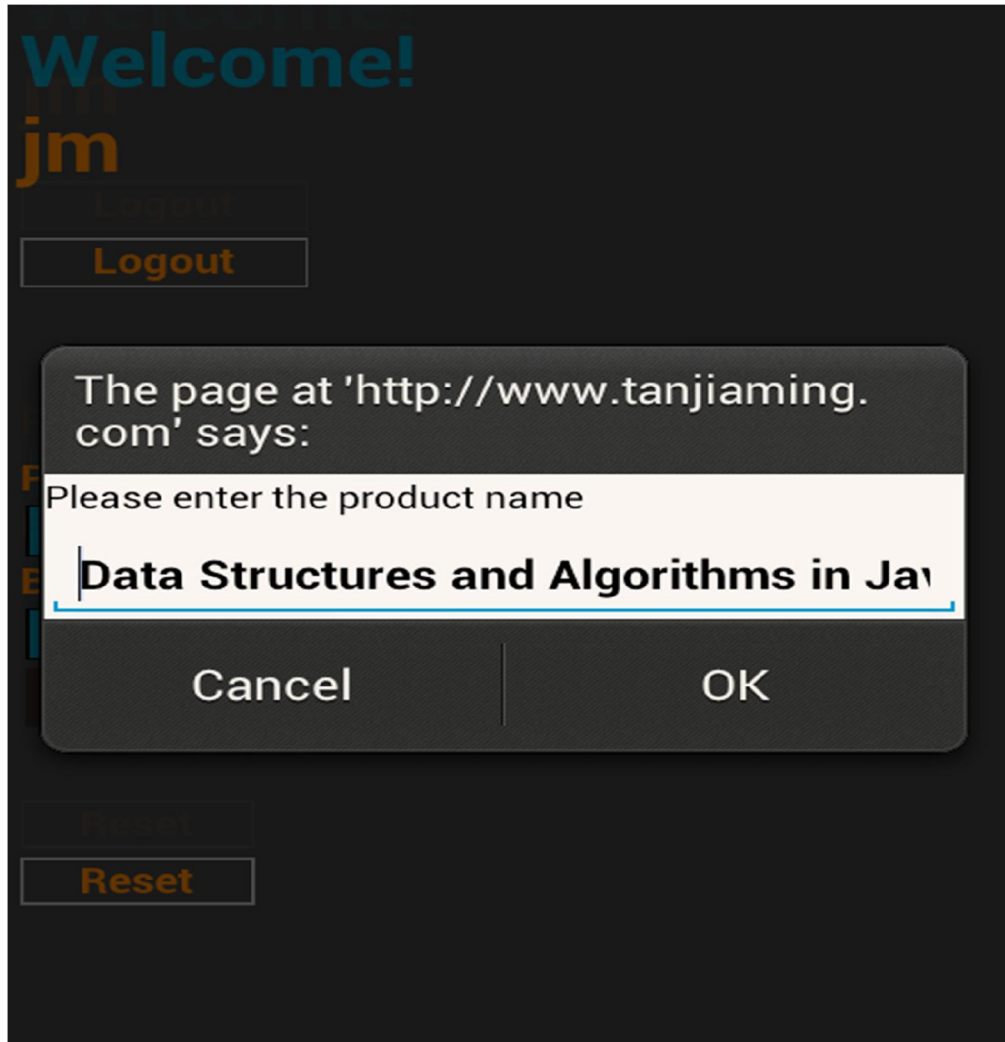


Figure 4-4-6 Screenshot of MPSCS adding a product name

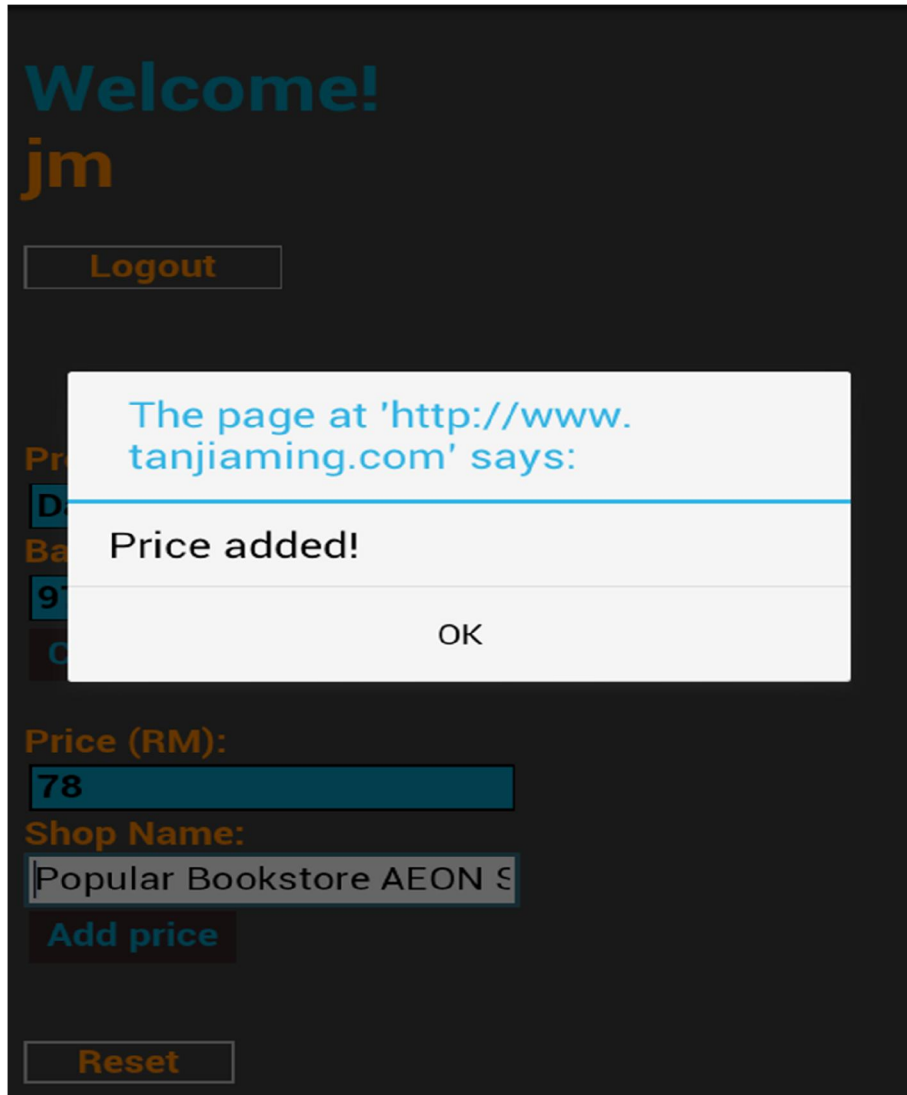


Figure 4-4-7 Screenshot of MPSCS adding a product's price

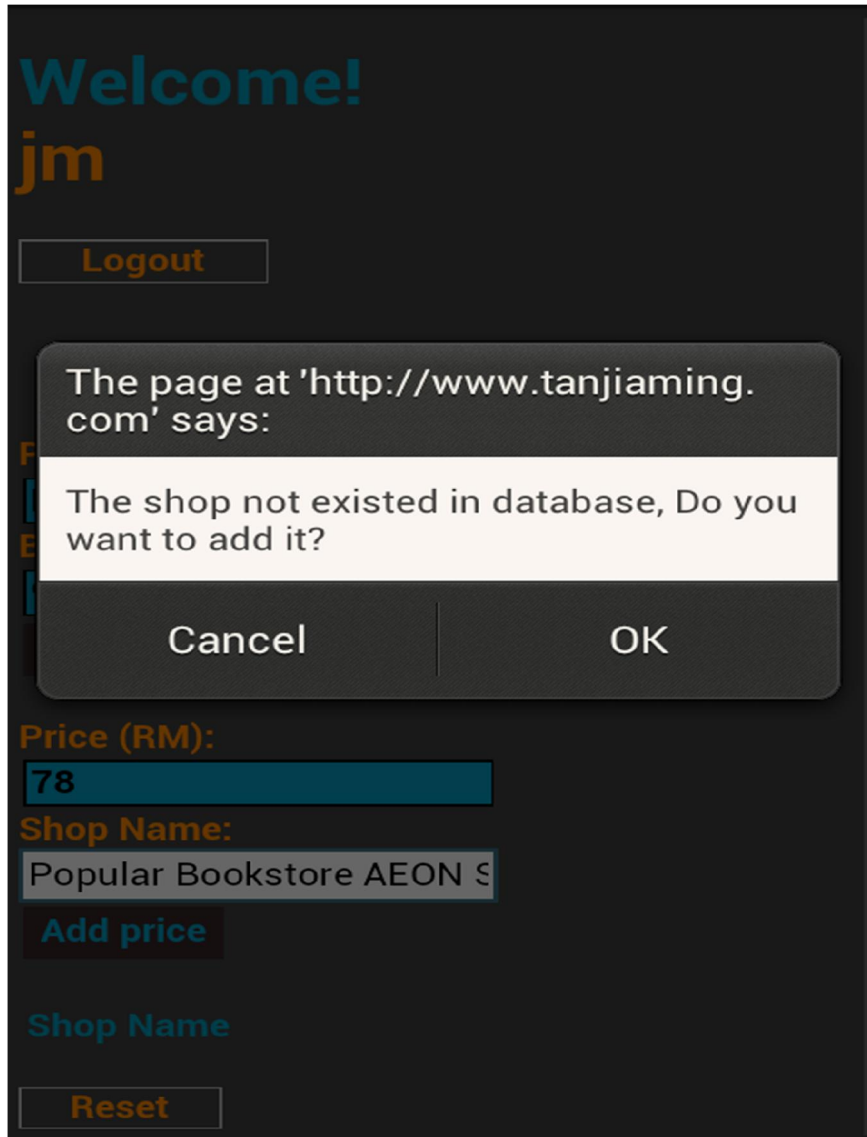


Figure 4-4-8 Screenshot of MPSCS adding a shop name

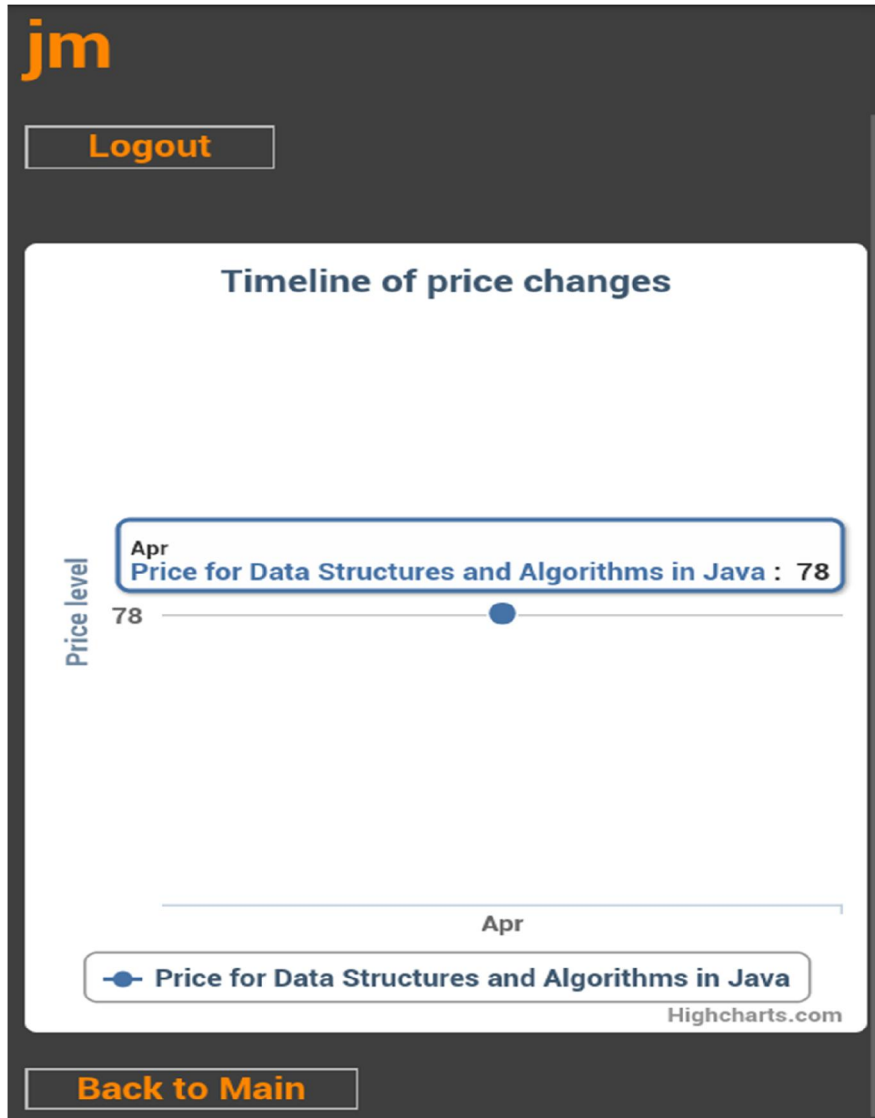


Figure 4-4-9 Screenshot of MPSCS reporting

Chapter 5: Testing

5.1 Functionality Testing

Module	Goal	Reached the goal?	Additional Comment
Login/Logout	Check against empty field inputs	Yes	
	Check if username and password were matched with database record	Yes	
	Log in to the system	Yes	
	Show user's name while logged in	Yes	User name was used to show who updated the price
	Log out from the system	Yes	
Barcode Scanner	Click button to invoke Barcode Scanner	Yes	Users must have Barcode Scanner installed, else MPSCS would crash
	Scan codes	Yes	
	Pass scanned result to server	Yes	

Geolocation	Get coordinates	Yes	
	Reverse geocode into address	Yes	
	Get city name from address	Yes	Sometimes, city name was incorrect, this was due to limitation of Google Maps JavaScript API
Reporting	Correctness of the graph	Yes	
	Show actual amount when the dot was clicked	Yes	
Search product	Suggest product as product name was being entered	Yes	
	Search for product price after clicked on suggested product	Yes	
	Search for product price after scanned 1D/2D barcodes	Yes	
	Prompt to add price when product price was not found	Yes	

	Prompt to add product when product price was not found	Yes	
	List prices located in the same city	Yes	
	List product prices located in other city, if prices in the same city are not available	Yes	
	Correctness of the shown list	Yes	
Add price	Check against empty field inputs	Yes	
	Validate against price input	Yes	
	Suggest shop as shop name was being entered	Yes	
	Prompt to add shop when shop was not found	Yes	
	Price was successfully added	Yes	

Add product	Check against empty field inputs	Yes	
	Check if product name existed	No	This validation has not been implemented
	Product was successfully added	Yes	
Add shop	Check against empty field inputs	Yes	
	Check if shop name existed	No	This validation has not been implemented
	Shop was successfully added	Yes	
Update price	Shop name were entered when user clicked on the price list	Yes	
	Check against empty field inputs	Yes	
	Validate against price input	Yes	
	Price was successfully added	Yes	

Table 5-1-1 Functionality testing for MPSCS

5.2 Usability Testing

Task/Usability goals	Expectations	Result	Additional Comment
Price search using Barcode Scanner	Task completed in ONE (1) step	Task completed within expected number of steps	
Price search using product name	Task completed in TWO (2) steps	Task completed within expected number of steps	
Add a price	Task completed between FOUR (4) and FOURTEEN (14) steps	Task completed within expected number of steps	Steps required was varied, which depended of the existence of product or shop
Add a product	Task completed in FOUR (4) steps	Task completed within expected number of steps	
Add a shop	Task completed in SEVEN (7) or EIGHT (8) steps	Task completed within expected number of steps	Although task completed within expectation, some users complained steps taken to complete task was not straightforward

Update a price	Task completed in FOUR (4) or FIVE (5) steps	Task completed within expected number of steps	Steps required varied depended the search method of users, and some users complained steps taken to complete task was not straightforward
View reporting	Task completed in TWO (2) or THREE (3) steps	Task completed within expected number of steps	Steps required varied depended the search method of users, users complained about difficulty in finding the button to view reporting
Login	Task completed in TWO (2) steps	Task completed within expected number of steps	
Logout	Task completed in ONE (1) step	Task completed within expected number of steps	
Easy navigation between pages	Users able to navigate between pages easily	Users had no trouble in navigation pages after they get used to the HTML button	Users unable to back to previous page using back button, so a HTML button was used to navigate to previous page

Page Presentation	The page layout should show all elements in a logical way	All elements were placed in a logical sequence	While all elements were placed logically, user complained about the limited view of suggestion lists, and also price list. This was due to the limited screen size. A scrollbar was displayed to remind users.
-------------------	---	--	--

Table 5-2-1 Usability testing for MPSCS

Chapter 6: Conclusion

6.1 Conclusion and contributions

The system to be developed was hoped to fight against inflation. The system enables users to share pricing information, which would help them in having updated, comprehensive price lists of products within their area.

The reporting, which was the graph showing the timeline of price changes would help users to have better understanding about the pricing trend of a product, so that they could be prepared when there were signs indicating inflation.

The system would have scanner for 1D/2D barcodes, as well as QR codes, which can be used to quickly identify a particular product, and would made the process of price searching and sharing more efficient.

The system would have location-based searching, so it could search prices of product, or shops that were within vicinity of users, which would make the results more meaningful to users.

At last, the system was hoped to ease the effort on organising pricing information conventionally. With centralised pricing information, along with inputs from other users, users could save time from adding, or updating prices of each product by themselves.

6.2 Software limitations

- **Incorrect city returned by Google Maps JavaScript API V3.** Although we used coordinates geocode our location, but there were scenarios where returned address had the wrong city name, which was the city next to users' located city. For example, coordinates from Kampar, might have Kampong Malim Nawar returned as city name, rather than Kampar.

- **State or data lost when screen orientation of WebView changed.** If users enabled auto-rotation, and they changed screen orientation during the usage of MPSCS, the WebView would lose its state and data. This was because when the screen orientation changed, Android would destroy the Activity first, and then recreate it again. So the WebView would be reloading the same page, after the Activity has been recreated.

6.3 Future work and possible enhancements

The project could still be enhanced, despite all of the limitations mentioned in Section 6.2. Some limitations could even be overcome, by further refining and enhancing the system. But due to the size of scope and limited resources, it could be our future work. There were some possible enhancements for the future, such as:

- **Add review module for products.** The system would have a review module in future, so that users can share their thoughts about the products, and their pricing. This could serve as a guide for other users, and may result in smarter buying decision.
- **Preserve WebView's state or data when screen orientation has changed.** The system could be enhanced to address the problem of lost state of data, when the screen orientation changed during using MPSCS. We could be disabling auto-rotate screen orientation
- **Add register/sign up module.** Users were unable to register for an account upon the completion of this project. The register module, along with login module,

could provide more details on which users updated the price, and could use for the rating module below.

- **Add rating module for preventing unrealistic price input to sabotage shops.** Upon the completion of this project, MPSCS were unable to determine the accuracy of the price input. Users with malicious intention could sabotage a shop, by simply update a price of a product unrealistically, such as RM 1000.00 for a canned drink. By using rating module, other users could rate if the price were helpful or not. Price updated that received many negative votes could be hidden from other users, or even delete.
- **Calculate and show driving distance to the shops listed.** The system was able to find shops that were in the same city as users. However, the distance of the shops was not determined. By showing driving distance between the shops and users, users could find a balance between time spent to travel and money saved on buying the product with cheapest price.
- **Price comparison by graph between two shops.** Upon completion of this project, users were not able to do price comparison between two shops by graph. Using a graph to compare price between two shops could help users in identifying the better price in a glance, which could facilitate the price comparison process.

References

- Bernneth, S. et al. (2006) *Object-Oriented Systems Analysis and Design Using UML*. 3rd ed. McGraw-Hill Education.
- Careerride.com (2009) *Phases in the incremental model - Project management (PMP)*. [online] Available at: <http://www.careerride.com/pmp-phases-in-incremental-model.aspx> [Accessed: 19 Jul 2012].
- Code.google.com (n.d.) *zxing - Multi-format 1D/2D barcode image processing library with clients for Android, Java - Google Project Hosting*. [online] Available at: <https://code.google.com/p/zxing/> [Accessed: 1 Feb 2013].
- Department of Statistics, Malaysia (2012) *Consumer Price Index Malaysia June 2012 (Updated: 18/7/2012)*. [online] Available at: http://www.statistics.gov.my/portal/index.php?option=com_content&view=article&id=1655%3Aconsumer-price-index-malaysia-june-2012-updated-1872012&catid=71%3Aconsumer-price-index-malaysia-&Itemid=153&lang=en [Accessed: 9 Jul 2012].
- Dev.w3.org (2012) *Geolocation API Specification*. [online] Available at: <http://dev.w3.org/geo/api/spec-source.html> [Accessed: 3 Feb 2013].
- Developer.android.com (2012) *ADT Plugin | Android Developers*. [online] Available at: <http://developer.android.com/tools/sdk/eclipse-adt.html> [Accessed: 19 Jul 2012].
- Developer.android.com (n.d.) *WebView | Android Developers*. [online] Available at: <http://developer.android.com/reference/android/webkit/WebView.html>

[Accessed: 2 Feb 2013].

Developers.google.com (2013) *Google Maps Javascript API V3 Reference - Google Maps JavaScript API v3 — Google Developers*. [online] Available at: <https://developers.google.com/maps/documentation/javascript/reference?hl=en> [Accessed: 2 Feb 2013].

Dialogic Corporation. (2010) *Introduction to cloud computing*. [e-book] p.7. Available through: <http://www.dialogic.com/http://www.dialogic.com/~media/products/docs/whitepapers/12023-cloud-computing-wp.pdf> [Accessed: 12 Mar 2013].

Google Play (2012) *Price Check by Amazon - Android Apps on Google Play*. [online] Available at: <https://play.google.com/store/apps/details?id=com.amazon.pricecheck&hl=en> [Accessed: 10 Jul 2012].

Groups.google.com (n.d.) *Google Groups*. [online] Available at: https://groups.google.com/forum/#!msg/zxing/PYY934MJeHI/r2wmQ_uJi3IJ [Accessed: 1 Apr 2013].

HK Tsang, C. et al. (2005) *Object-oriented Technology: From Diagram to Code with Visual Paradigm for UML*. McGraw-Hill Education.

Highcharts.com (2013) *Highcharts - Interactive JavaScript charts for your webpage*. [online] Available at: <http://www.highcharts.com/> [Accessed: 6 Mar 2013].

Informit.com (2006) *Software Development Strategies and Life-Cycle Models | Software Development Methodology Today | InformIT*. [online] Available at: <http://www.informit.com/articles/article.aspx?p=605374&seqNum=2> [Accessed: 17 Jul 2012].

Lelong.com.my (2004) *Malaysia Online Shopping & Auction - Lelong*. [online] Available at: <http://www.lelong.com.my/> [Accessed: 11 Jul 2012].

Lowyat.net (2012) *Lowyat.NET : Malaysia's Largest Online Community*. [online] Available at: <http://www.lowyat.net/v2/> [Accessed: 10 Jul 2012].

Ministry of Domestic Trade,Co-Operatives & Consumerism (2010) *Official Portal Ministry of Domestic Trade,Co-Operatives & Consumerism*. [online] Available at: <http://www.kpdnkk.gov.my/en/> [Accessed: 10 Jul 2012].

Mobile88.com (2012) *Mobile88 : All About Mobile Phone / Handphone / Gadget*. [online] Available at: <http://www.mobile88.com/> [Accessed: 11 Jul 2012].

MSV, J. (2012) *Top 10 Reasons why Startups should Consider Cloud*. [online] Available at: <http://cloudstory.in/2012/07/top-10-reasons-why-startups-should-consider-cloud/> [Accessed: 15 Feb 2013].

Mudah.my (2012) [online] Available at: <http://www.mudah.my/> [Accessed: 10 Jul 2012].

Portal 1Malaysia Pengguna Bijak (2012) [online] Available at:
<http://www.1pengguna.com/11pengguna/index.php> [Accessed: 10 Jul 2012].

Qtp.blogspot.com (2006) *QTP Tutorials & Interview Questions: Explain Incremental Model.* [online] Available at: <http://qtp.blogspot.com/2009/04/incremental-model-software-development.html> [Accessed: 18 Jul 2012].

Stackoverflow.com (2013) *How can I access Android's Barcode Scanner from WebView? - Stack Overflow.* [online] Available at:
<http://stackoverflow.com/questions/2562629/how-can-i-access-androids-barcode-scanner-from-webview> [Accessed: 30 Jan 2013].

Shopparadiso.com (2010) *Online Shopping, Online Price Comparison, Product Reviews, and Find Deals from Local Stores for Singapore @ Shopparadiso.* [online] Available at: <http://www.shopparadiso.com/> [Accessed: 11 Jul 2012].

Whatgas.com (2012) *Whatgas.com | "The price sharing community".* [online] Available at: <http://www.whatgas.com/> [Accessed: 10 Jul 2012].

Appendix

Appendix A : Poster

Mobile Price Sharing and Comparison System

Objectives

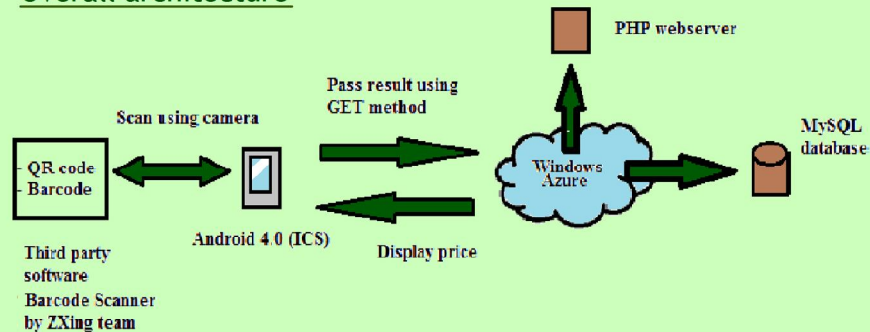
Allow users and store owners to share pricing information

Allow quick product search, since there are many products available

Allow users to search for pricing of particular product on specific area

Allow users to have better understanding in product pricing

Overall architecture



Contributions

Hoped to ease the effort on organising pricing information conventionally with centralised pricing information

Could help users to easily find better price within their vicinity, which may helped to reduce living expenses during the days of inflation

Could help to create a healthy competition environment, where the prices of product would be more transparent