

PROTOTYPING OF AN E-COMMERCE MOBILE MARKETING SYSTEM

Criswanto D. Nugroho¹ and Veronica S. Moertini²

^{1,2}Informatics Department
Parahyangan Catholic University, Bandung, Indonesia
¹arthur2112@gmail.com, ²moertini@unpar.ac.id

ABSTRACT

Electronic marketing is an essential part of e-commerce. With mobile device usage trends that grow exponentially in the last several years, providing electronic marketing services for mobile users has been becoming business opportunities. Despite these opportunities, mobile marketing system for e-commerce has not been widely studied. As mobile devices are personal tools, mobile marketing systems must be properly designed such that the systems will be accepted by users. This paper presents the design and prototype of an e-commerce mobile marketing system resolving user acceptance criteria, such as information personalization, privacy protection, up-to-date content and social network. The proposed system integrates a marketing provider website and mobile applications used by shoppers. The design and prototype are provided for the website, mobile application and the communication between both sub-systems that employs web services and a cloud service. Prototype testing and evaluation have been performed to ensure that the proposed system works properly.

KEYWORDS

Electronic marketing system, mobile marketing system design, e-commerce marketing, publish/subscribe system

1. INTRODUCTION

Electronic marketing (e-marketing) has been viewed as an essential part of e-commerce [9][14]. With the trend of mobile devices usage that grows exponentially in the last several years [26], providing e-marketing services for mobile users has been becoming business opportunities [1]. As mobile devices are personal tools, mobile marketing systems must be designed properly such that the systems will be accepted by users.

Contribution: Despite the opportunities, so far the authors have only found very limited results related to mobile marketing system design for e-commerce. Wium in [23] presents the design of mobile applications supporting tourists with travel information that use information of user's location, time and personal preferences to provide personalized recommendations. Balan *et al.* in [2] reports design and evaluation of a near-field communication-based mobile P2P payment application that is designed to replace cash-based transactions. Therefore, this paper is intended to contribute in the designs of mobile e-marketing systems for e-commerce.

In our previous research presented in [15], we have proposed a conceptual model of a mobile marketing system resolving user acceptance criteria (customer permission, usefulness, relevant content, up-to-date content, social network, privacy protection and so on). In the proposed model,
DOI:10.5121/ijitcs.2013.3101

the components of the system with their required services have been defined. In this paper we present the results of our next work, which is the design and prototype of the system by highlighting the techniques to resolve information personalization, privacy protection, up-to-date content, and market targeting.

Research methods: Based on the conceptual model in our previous work [15], we studied related technical literatures, developed the detailed design of the system, coded the designs, conducted some testing and evaluated the prototype.

This paper is organized as follows: The next section presents the related literatures, followed by the excerpts of the proposed conceptual model, the techniques for personalizing information and protecting privacy, website design, mobile application design, design of communication between the website and the mobile application and some part of its implementation, prototype testing and evaluation, and conclusion.

2. RELATED LITERATURES

2.1. e-Commerce and e-Marketing

In [9], a many-to-many communication model used in e-commerce marketing is proposed. With this model, consumers can communicate with the medium (such as website) and with each other, firms can provide content to the medium and interact with each other, firms can also interact with consumers, and so on. Consumers may collaborate in the marketing communication effort itself. It is stated that the web as both medium and market is more likely to be successful if it frees consumers from their traditionally passive role as receivers of marketing communications, gives them much greater control over the search and acquisition of information relevant for consumer decision making, and allows them to become active participants in the marketing process.

On the other hand, [14] defines that e-commerce e-marketing refers to a new comprehensive marketing model which is based on e-commerce and uses every kind of e-marketing methods and means to achieve online business activities. So, the content of e-commerce e-marketing should insist e-marketing, make e-marketing as the core business activities of e-commerce and combine e-commerce and e-marketing that will give e-commerce new contents as well as create integrated and comprehensive modern new marketing model which is e-commerce e-marketing.

E-marketers can select among four targeting strategies, which are mass marketing, multi-segment marketing, niche marketing (when a firm selects one segment and develops one or more marketing mixes to meet the needs of that segment), micromarketing/individualized targeting (when a firm tailors all or part of the marketing mix to a very small number of people) [22]. Market segments can be designed based on the geographic, language, psychographic and behavior. Consumer psychographics comprise personality, values, lifestyle, activities, interests, and opinions. There are few e-marketing tactics that can be adopted, such as permission, viral, email, search engine, partnering marketing, banner advertising, online promotion campaigns, and so on.

2.2. Publish/Subscribe System for Mobile Network

One option in designing an event-based communication system on mobile ad hoc networks is publish/subscribe system, a loosely-coupled paradigm for communication between entities [11]. In this architecture, two new roles are defined: Subscriber (or Consumer) on one side and Publisher (or Producer) on the other side. Subscribers are nodes who express their interest in an event or group of events with specific pattern and submit these interests to event service (or Broker node). On the other hand, the publishers are nodes who generate the events and dispatch them to the broker node. Then, the broker node notifies the subscribers of the published events in an asynchronous communication model (Figure 1).

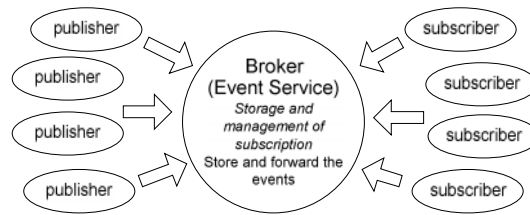


Figure 1. The Publish/Subscribe system components and basic diagram [11].

3. MOBILE USERS ACCEPTANCE CRITERIA AND PROPOSED SYSTEM MODEL

In our previous research result [15], we have identified 13 mobile users acceptance criteria searched from literatures, which are: Cr-1 customer permission [1]; Cr-2 ease of use ([20], [19], [12] and [24]); Cr-3 usefulness ([20], [19] and [25]); Cr-4 relevant content ([1] and [13]); Cr-5 compelling content ([1], [7], [20], [19]); Cr-6 timely and up-to-date content [19]; Cr-7 personalized content [12]; Cr-8 location-aware advertising [3]; Cr-9 increasing gains in shopping performance [24]; Cr-10 incentive, rewards ([1], [3], [7], [19]); Cr-11 enhancing customers image ([20] and [19]); Cr-12 services providing interpersonal and external influence, social network ([20], [25] and [24]); Cr-13 security and privacy protection ([20], [7] and [13]).

We have also proposed a model of an integrated information system that provides services/functions resolving all of the criteria, which is similar to the Publish/Subscribe system [11], adopted the concept depicted in [9] and [14]. The system consists of a website and mobile application where both communicate directly via http as well as via cloud services (see Figure 2.(a) and (b)). The website acts as both Publisher and Broker while the mobile application functions as Subscriber system component.

Website: This will be used by the mobile marketing provider administrator and merchant members (with free, silver, gold and platinum membership). The services provided are: Managing product catalogs; managing ads, deals and market segmentation variables used in computing the targeted market segment for the corresponding ads and deals; updating/adding product categories, catalogs, ads, deals (customers will be updated promptly each time new data is posted); managing stores location.

Mobile application: It is used by shoppers (as free members) by first downloading the application from the website and then installed in their mobile devices. This application provides 16 services, such as: Sign up/registration to become a free member; viewing personalized product catalogs,

receiving up to date (real time) personalized ads and deals/coupons based on customer profile where the deals can be redeemed in the merchant store; displaying the location or map of each merchant; comparing products; marking favorite products and receive notifications if the products are on sale; asking opinions from peers regarding the products of interest; sharing information of products, advertising and promotions/deals to peers; filling shopper profile that will be stored in the local mobile database and are not accessible by other parties (the profile will be used to filter the incoming ads and deals); storing favorite products and deals taken in the local mobile database.

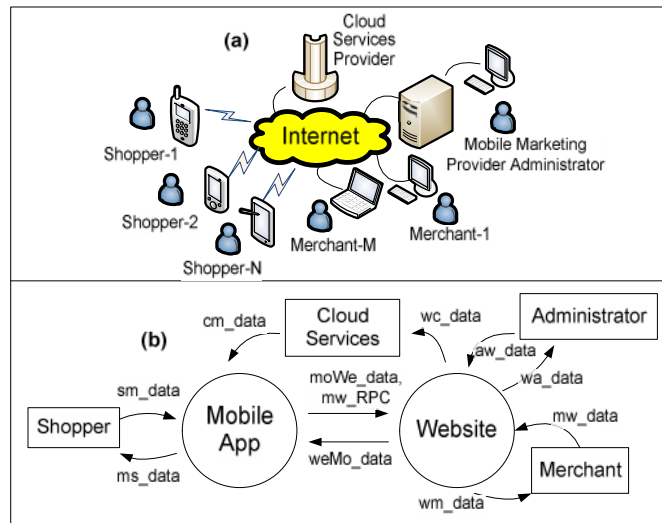


Figure 2. (a) Network topology of the system [15]; (b) Diagram context of the system.

The selected cloud computing service: In this design and prototype, we select a PaaS (platform as services) provided by Google, namely Android Cloud to Device Messaging (C2DM) that allows third-party application servers to send lightweight messages to their Android applications [27]. This service is not designed for sending a lot of content via the messages. Rather, it is used to tell the application that there is new data on the server, so that the client application can fetch it. In our case, this service is adopted to provide push notification services or broadcast events.

4. PERSONALIZING INFORMATION, TARGETING MARKET AND PROTECTING PRIVACY

In order to facilitate the services that resolve the criteria associated with security and customer privacy (Cr-1 and Cr-13) and content (Cr-3 usefulness, Cr-4 relevant content, Cr-7 personalized content, incentive and Cr-10 rewards) that must also be timely and up-to-date (Cr-6) and to provide niche marketing (see Section 2.1) for merchants we propose the following techniques.

First, we design two separate databases: One is the database in the provider server that is accessed by the website and the other is the local database in the mobile devices accessed by the mobile application. The database in the server stores a variety of large data needed by the mobile marketing system while the local database stores small private data belonging to the shoppers. Among other things, each of the database stores the data as shown in Figure 3. The product categories stored in the mobile database will be used to “filter” the product catalogs that can be viewed as well as ads and deals received by the shoppers (mobile users). The user profile in the

mobile database also contains data of the market segmentation criteria and their values used to personalize ads and deals received.

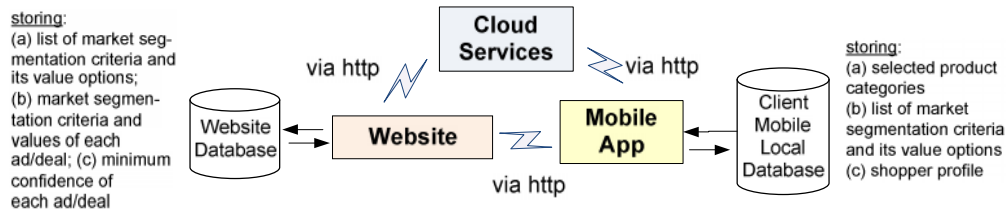


Figure 3. The two separate databases.

Secondly, we design the following communication scheme to maintain data synchronization between the server and mobile database, and send ads/deals in real time:

The scheme of data synchronization (Figure 4.(a)): (1) The product categories and market segmentation data in the server database are updated/added; (2) The data is stored in the server database; (3) The website sends notification to mobile devices via C2DM service; (4) C2DM service broadcasts the notification message to mobile devices; (5) Upon receiving the notification, the mobile application, requests the complete data to the website by executing certain web service functions; (6) The website sends the requested data; (7) the mobile application stores the received data into the mobile local database;

The scheme of broadcasting new ads/deals (Figure 4. (b)): (1) Merchant creates an ad/deal and define its market segmentation criteria, its values and the minimum confidence level; (2, 3, 4) The steps are the same with the data synchronization steps; (5) the mobile application requests the complete data of market segmentation criteria, its values and the minimum confidence level to the website by invoking web service functions; (6) The website sends the requested data; (7) By using the market segmentation criteria and its values, the mobile application computes the confidence of the shopper (that determine whether the ad/deal is suitable for the shopper), then, if this value if greater than or equal to the minimum confidence (defined by the merchant), the ad/deal notification will be stored in the mobile database, otherwise the ad/deal will be ignored.

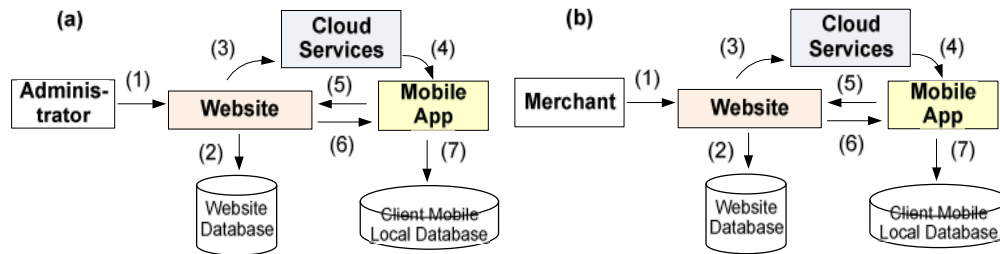


Figure 4. The sequences of: (a) updating and synchronizing market segmentation criteria and its option values between the server and mobile client; (b) broadcasting new ads/deals.

Ads/deals confidence level computation: While [10] proposes a formula for computing user interest of an item using common taste of the population and individual taste of a set of items, here, we proposes a formula for computing user confidence of an item using shopper profile (defined by the shopper) and market segmentation criteria (defined by the merchant in creating an ad or deal of a product). We assume that based on merchants' knowledge, merchants will be able

to define the targeted segmentation market of their ads/deals. Our proposed formula is: $confidence = \sum_{i=1}^{i=n} s_i \times w_i$, provided that $s_i = p_j$. Here, s_i is the value of i^{th} market segment, w_i is the weight of s_i and p_j is the value of j^{th} shopper profile. With this formula, if a market segment does not equal with any of the shopper profiles, it will not be used in the confidence computation. Hence, the more the market segment equal to the shopper profile, the higher confidence will be. In the proposed system, if the confidence is equal or higher than minimum confidence defined by the merchant, then the ad/deal will be passed through the shopper.

5. WEBSITE DESIGN

The provider website is further divided into two modules, namely *Phost A* and *Phost M* (see Figure 5), where each module consists of several sub-modules that are used by certain users and each sub-module has functions materializing the services designed (see [15]). In the context of the Publish/Subscribe system, *Phost M* is the Publisher while *Phost A* is the Broker, which in this case, utilizes C2DM service to dispatch messages.

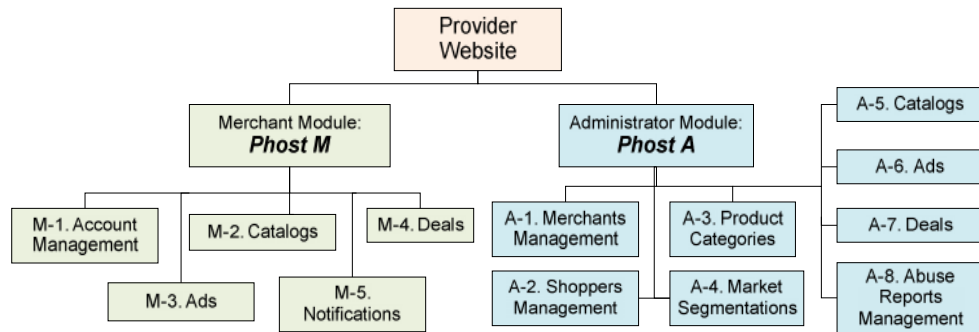


Figure 5. The architecture of the website owned and run by the provider [15].

Some examples of user functions provided by the modules are listed below:

Phost M module with merchants as the users:

- M-3. Ads: View, insert, update and delete product advertisings including their market segments, minimum confidence and period of the advertisings;
- M-4. Deals: View, insert, update and delete product deals offered to shoppers/customers including their market segments, minimum confidence and period of the advertisings; Process deals taken by shoppers.

Phost A module with provider Administrator as the users:

- A-4. Market Segmentations: View, insert, update and delete market segmentations (market variables and values). Examples of variables: age (values: <20, 21-30, 31-40), gender (values: male and female);
- A-7. Deals: View and verify product deals posted by merchants. After the deals are verified by Administrator, notifications are promptly broadcasted to shoppers' mobile devices via a cloud service.

Database Design All of *Phost A* and *M* functions access a single database where its relational design is depicted in Figure 6. There are 20 relations and 4 views (shown without attributes). As can be seen, the database stores system configuration, merchants, shoppers, category of products, products, market segmentation variables and their value options, also data and

transactions/notifications related to ad and deal. Relation *ad_criteria* and *deal_criteria* are used to store the criteria that define the targeted market segmentation for the corresponding ad and deal. Relation *ad_deal_notification* stores the notification of ad and deal sent to shoppers, while *taken_deal* stores the transactions of shoppers of taking/booking deals in responding to deal notifications received (in shoppers mobile devices).

User Interface Design The user interfaces designed for *Phost A* and *M* are mainly forms and reports needed by the administrator and merchants to conduct their tasks. The forms and dialogs are designed by providing self-explanatory menus and pages having buttons and list of choices, simple navigation between pages and using few button colors such that the function of each button is easily recognized. Some of *Phost M* user interfaces designed is shown in Appendix (Figure A1). It is shown in the figure that for providing personalized product ads and deals, the administrator manages product category and market segmentation variables and their value options, while each merchant uses these variables with their values in creating a product ad or deal intended for a certain market segment.

6. MOBILE APPLICATION DESIGN

In this section, we present the design of the subscriber system component in Publish/Subscribe system (see Section 2.2), which is a mobile application, namely *Phost C*, in the form of use-case model, packages and classes, mobile database and user interfaces. The use-case, packages and classes are presented using UML [5] that is commonly used in designing object-oriented application such as application in Android platform.

6.1. Use Case Model

The functions of *Phost C* are designed and depicted in UML use-case diagram. The top level of use-case showing the main functions of *Phost C* is presented in Figure 7.(a). Each of the use in Figure 7.(a) is then broken down into a detailed use-case which represents the use-case of a sub-package (see Section 6.2 for the package design). The detailed functions are extension of the main functions. For instance, Figure 7.(b) and (c) show the extensions of manage catalog and deal. The extensions of other main functions are: (a) manage account: sign-in/out, set profile, set preferred product categories (product filter); (b) manage ads: view ad content, delete ad, share ad to shopper peers, view product information having the ad; (c) manage inbox and shopper: view list of notifications in inbox, view notification, delete notification, view list of shopper peers, add shopper peer, delete shopper peer.

The scenario of the use-case is constructed for each detailed function. For example, the scenario of *take deal* use is as follows: With the pre-condition that User is already log in and view a deal, User press "Take Deal" button, the application updates the status of the deal in the database table as "taken" and sends a message to the provider website stating that the deal is taken by this user, the website then notifies the merchant who posted the deal. Exception: (a) If the products offered are out of stock, the provider website will notify User that the deal cannot be taken; (b) If the deal is expired, the application will reject the request of taking the deal.

6.2. Package Design

It is known that MVC (Model, View and Controller) pattern is suitable to separate data, engine and user interfaces for better organization and maintainability of the applications. In MVC, Model

contains the data that users work with, View is used to render some part of the model as a UI and Controllers processes incoming requests, performs operations on the model, and selects views to render to the user [6]. In designing the package of *Phost C*, we first define two packages, namely controller and model. The model package has classes used to access and update database tables. The controller package, which combines the Controller and View component of MVC, is further broken down into several sub-packages (see Figure 8) to better organize the classes into modules.

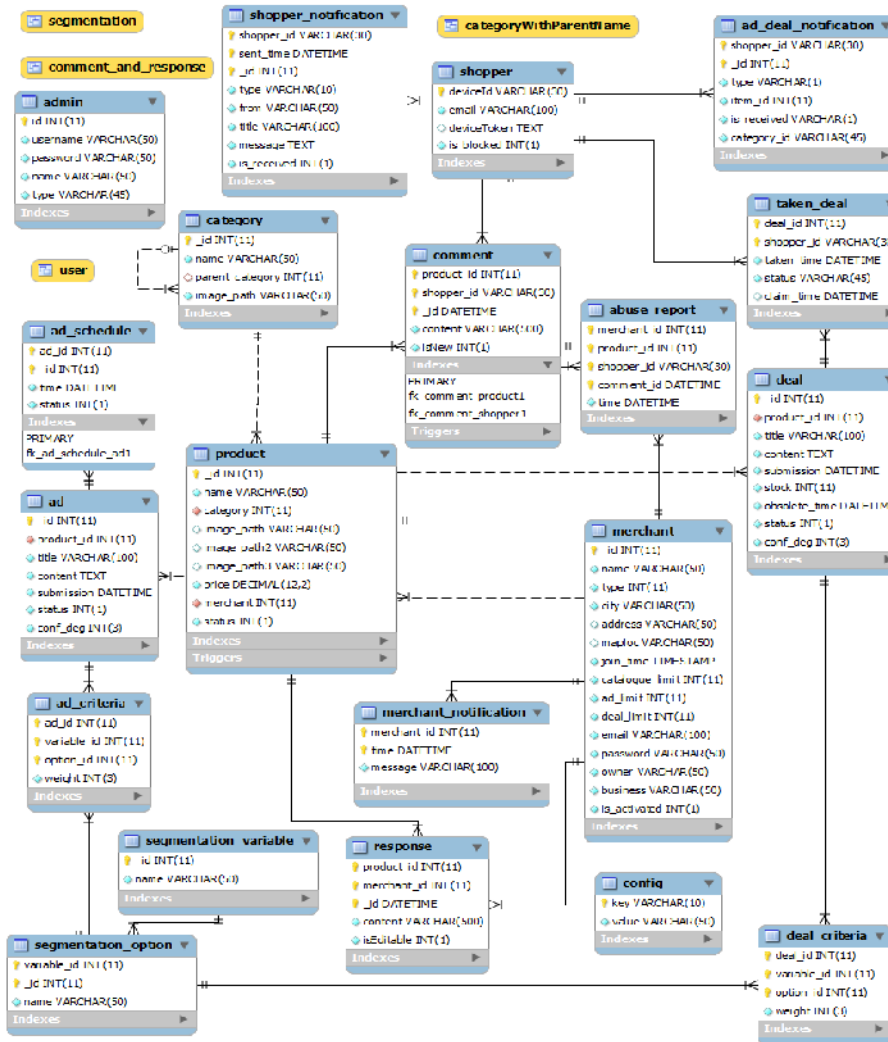


Figure 6. Relational model of the website database.

6.3. Classes Designs

Basically, Android applications consist of components of activity (a single screen of user interfaces), intent (a message sent between components), service (a process run in the background and does not have any user interface components), content provider (an interface for sharing data between applications), broadcast receiver (an Observer pattern, which is dormant codes, that gets activated once an event to which it is subscribed happens) and application context (the application

environment and the process within which all its components are running, hence, an application context gets created whenever the first component of this application is started up) [8]. Android provides libraries for the components such that the designed application classes can simply be the children of them. Currently, only the content provider library that is not used in *Phost C* as it does not share resources with any other application yet.

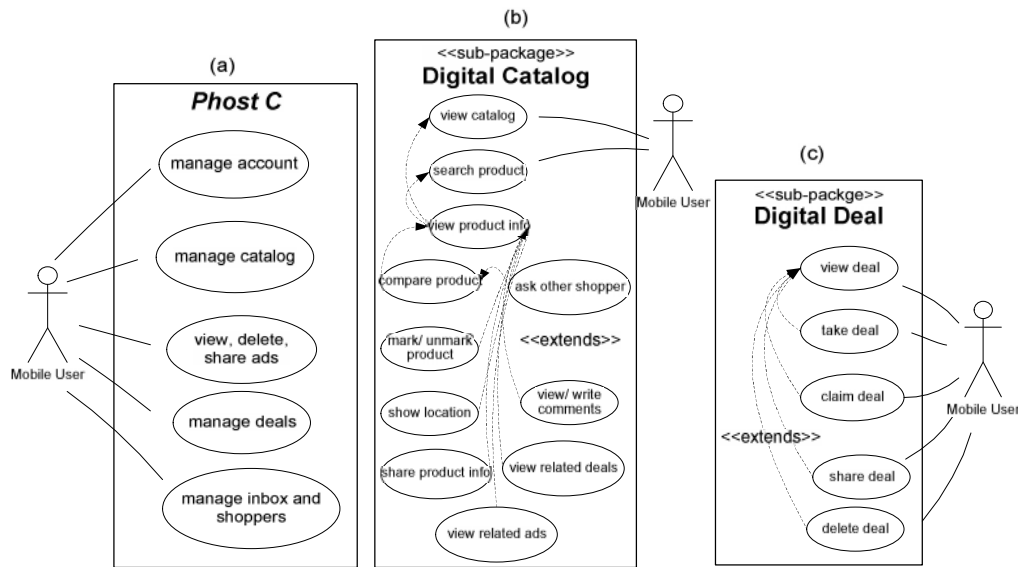


Figure 7. Use case diagram of: (a) *Phost C* that shows its main functions; (b) Detailed use-case of manage catalog in Digital Catalog sub-package; (c) Detailed use-case of manage deal in Digital Deal sub-package.

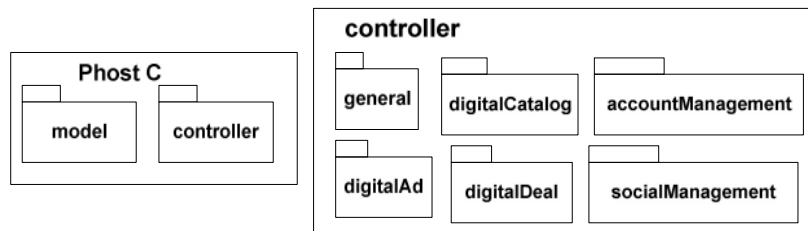


Figure 8. Packages designed of *Phost C* mobile application.

The design of classes and their relationships of *Phost C* is constructed for model package and each sub-package of controller package. However, due to space limitation, this section can only present brief description of model package, general sub-package and digitalDeal sub-package.

Classes in model package As the classes in model package are used/instantiated to access the local mobile database and its tables, most classes contain attributes and constructors declarations only. There are 13 classes designed in model package, which are DBManager, Profile, ProfileItem, Ad, AdDealNotification, Category, Comment, Criteria, Deal, Notification, Product, ProductDetail (sub-class of Product) and Segmentation. Some highlights of DBManager class: Has attributes and methods used to create, access and update database tables; has methods used to synchronize data (such as product categories and market segmentation variables) stored in mobile database and provider’s server database; defines static variables and their values used commonly across class objects.

Classes in controller package Controller package contains purely engine classes as well as classes which are sub-classes of Android classes, such as Activity, Intent, Context, BroadcastReceiver and IntentService. In designing classes, the name of each class in sub-packages is defined such that it is semantically meaningful. For example, C2DMCommandHandler class tasks are handling incoming push messages sent by C2DM service. In sub-packages, there are a number of classes which are derived class of Android Activity class to provide many screens in the application that support easy navigation between screens and ease of use of the application.

Classes in general sub-package There are 11 classes defined in this sub-package. It contains “base” classes that will be instantiated by other classes in other sub-packages. While most classes are sub-class of Android Activity class, this sub-package also contains classes to handle incoming push messages coming from C2DM service (see Figure 9). Some highlight of the classes and their tasks/functions is as follows:

- (a) MainMenuActivity (child of Android Activity class): This class is instantiated when *Phost C* application is first launched and provides “home page” of *Phost C*, then allows users to navigate among its pages.
- (b) ApplicationContext (child of Android Application class): Defines the application environment and the process within which all its components are running. This is instantiated whenever the first component of *Phost C* is started up, regardless of whether that component is an activity, service, or something else. The object/instantiation lives as long as *Phost C* is alive.
- (c) C2DMReceiver (child of BroadcastReceiver class): Processes incoming messages from the cloud service, C2DM.
- (d) C2DMCommandHandler (child of Android IntentService class): Handles various incoming commands passed via intent and passes a message to the appropriate activity or activities to perform further actions (i.e. if the command is equal to “0112”, then “create active notification” message will be sent to object of Notifier and “update product category” message will be sent to object of DBManager).
- (e) Notifier: Handles notifications for users, such as show and hide notifications.

Classes in digitalDeal sub-package This sub-package, among other classes, contains DealEngine class having the tasks to handle deals as well as activity classes (children of Android Activity class) providing user interfaces related with viewing, taking and claiming deals offered by merchants. Classes of model package that are instantiated and used in accessing and manipulating database records are DBManager and Deal (see Figure 10).

Classes design in other sub-packages, namely digitalCatalog, digitalAd, accountManagement, socialManagement, generally are analogous to the classes in digitalDeal sub-package. The sub-packages contain classes having tasks to handle incoming messages, provide user interfaces as well as access the local database.

6.4. Mobile Database Design

In designing the database for a mobile application, it is suggested that the database only manage small size of data most needed by the application [18]. If the data stored must be up to date from time to time, the data must then be synchronized with the source database (that manage the current data). In this context, the mobile database must be synchronized with the provider website database.

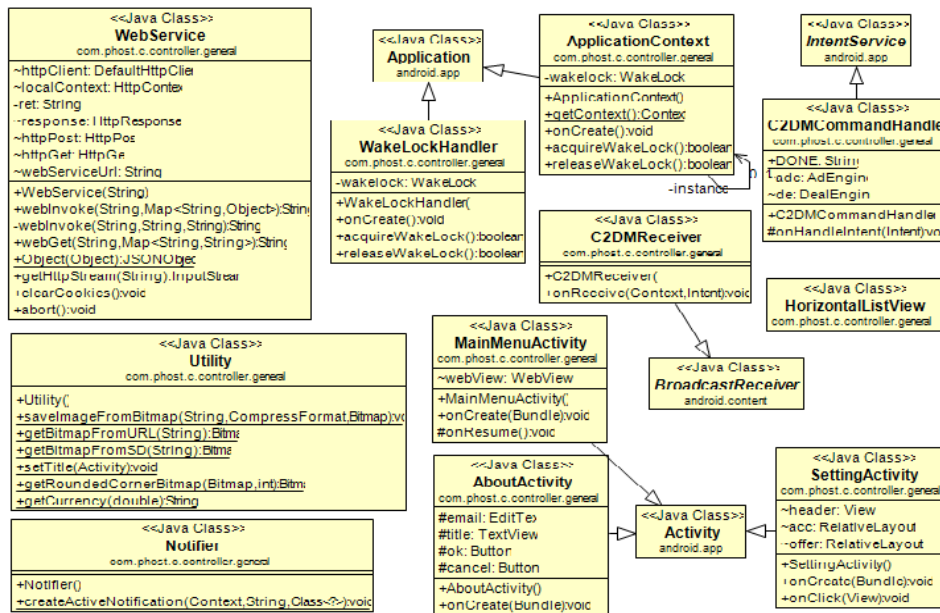


Figure 9. Class diagram of general sub-package.

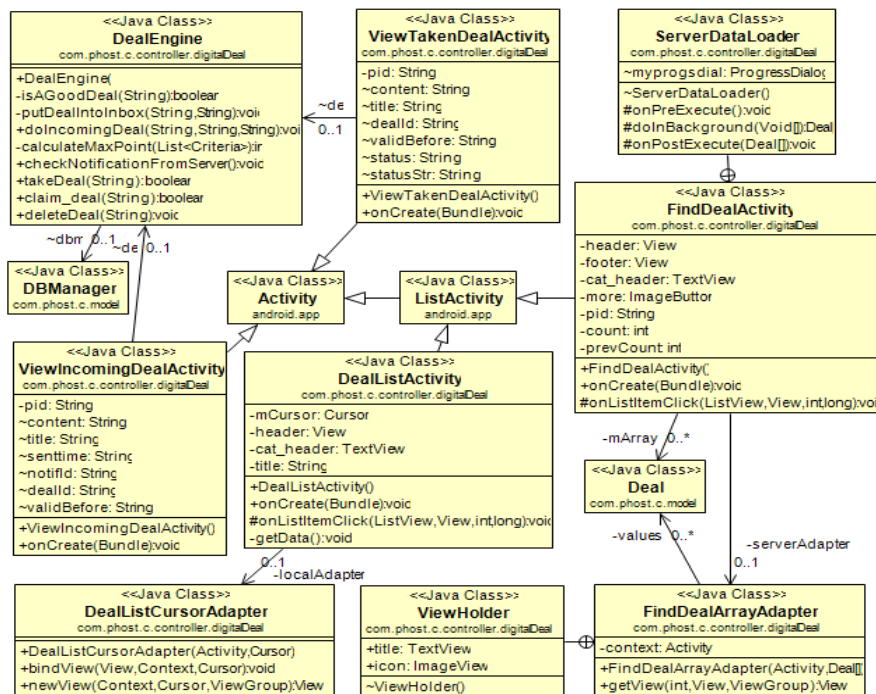


Figure 10. Class diagram of digitalDeal sub-package.

Some highlights of the relations are: (a) shopper: Storing peers email address and mobile device identification; (b) notification: Functioning as the user “inbox” storage; (c) segmentation: Storing

variables and their options used in the confidence level computation of every incoming ad and deal (the confidence level determines whether the ad/deal is matched with user profile); (d) filter: Storing product categories that used to filter product information, ads and deals received by the user; (e) config: Storing the application configuration; (f) compare_list: Storing product IDs being compared; (g) profile: Storing user profile used to personalize ads and deals received.

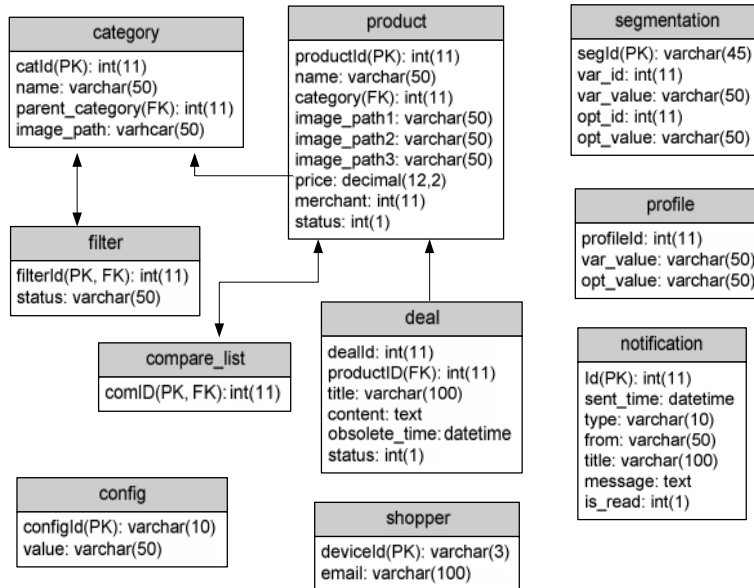


Figure 11. The relational schema of the mobile database.

6.5. User Interfaces

The user interfaces are designed in such a way to resolve ease of use criteria (Cr-2) as well as to support all of other criteria. The ease of use is resolved by providing self-explanatory menus and pages, simple navigation between pages, using semantically meaningful icons, providing list of choices (using drop down list and toggle button) if users must fill forms and using various elegant colors in pages such that information and menus are easily recognized.

Some of *Phost C* pages are shown in Figure 12, where the descriptions are presented in the figure caption. It can be seen in the figure that all of user functions shown in Figure 7 (the use-case diagram) are materialized by the designed menus.

7. SYSTEMS COMMUNICATION DESIGNS AND IMPLEMENTATION

In materializing real time communication between the website (*Phost A* and *M*) and mobile application (*Phost C*) via C2DM service, some designs must be constructed thoroughly and implemented properly. This section presents the design results and some example of communication source codes.

The designs or implementation include: (1) Systems interaction scenarios that are presented in sequence diagrams; (2) Messages structure sent from *Phost C* to the website; (3) Messages structure sent from the website (*Phost A* and *M*) to the cloud service, C2DM; (4) Web services in

the website (*Phost A* and *M*); (5) Classes/methods in *Phost C* having tasks to remotely call functions in the web services; (6) Functions in *Phost A* and *M* having tasks to send messages to C2DM; (7) Classes/methods in *Phost C* having tasks to receive and process push messages sent by C2DM.



Figure 12. Some user interfaces of *Phost C*: (a) The home page showing the menus that can be selected; (b) The form of setting user profile which will be used to personalized incoming ads and deals; (c) Product sub-categories that can be set on-off as desired by the user such that she/he will only receive ads and deals in this sub-categories; (d) Product detailed information and the related menus (the user then can find related deals, more information, view comments for this product, compare with other products, ask peer opinions, mark it as product favorite, share with peers and show the merchant location); (e) Product deal page where the user then can view the related product and “take” it; (f) Form for adding a shopper peer; (g) Inbox page where the user can view and reply the messages.

(1) Systems interaction scenarios

The systems interaction design is presented in the diagram sequence as depicted in Figure 13.

(2) Design of messages sent from Phost C to the website (Phost A and M)

Each of the messages sent by the mobile application has the structure as a collection of strings, {“c”, MESSAGE_TYPE, variable1*, value1*, variable 2*, value2*, . . . , variable n*, value_n*}, where * denotes optional variable/value. There are 27 message types, which are grouped in five categories as follows:

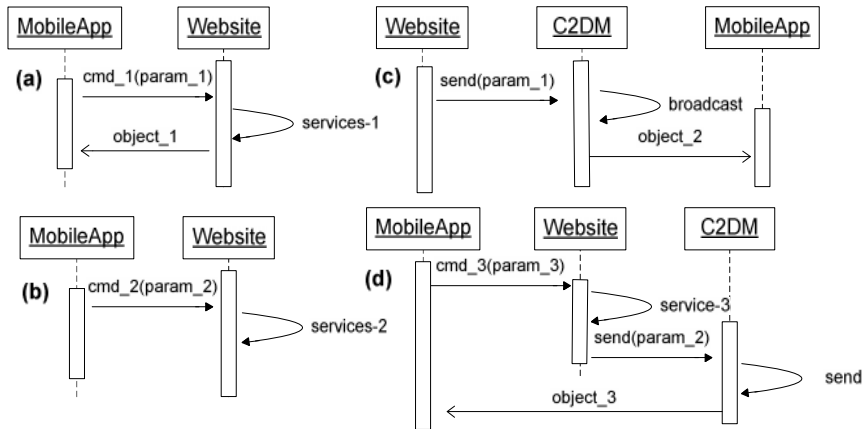


Figure 13. Diagram sequence of: (a) Mobile App call a web service with return value; (b) Mobile App call a web service with no return value; (c) Website broadcasts short messages to Mobile App via C2DM; (d) Mobile App call a web service triggering Website to send message to the Mobile App that makes the call.

- (a) Shopper messages (3 message types: 1121, 1221, 1321), for example: “1121” means adding a shopper peer in the server database and the message format is {“c”, “1121”, “f”, *MOBILE DEVICE_ID*, “t”, *EMAIL_ADDRESS*}; “1321” means requesting to remove other shopper request (to be a peer) with variables and and the message format is {“c”, “1121”, “f”, *MOBILE DEVICE_ID*, “t”, *EMAIL_ADDRESS*};
- (b) Inbox messages (4 message types: 2221, 2321...2421), for example: “2221” means synchronizing notifications data stored in the server database and client (local mobile) database and the format is {“c”, “2221”, “i”, *MOBILE DEVICE_ID*}; “2321” means requesting newest notifications data and the format is {“c”, “2321”};
- (c) Ad messages (6 message types: 3121, 3231,...,3621), for example: “3121” means requesting market segmentation criteria of an ad and the message is {“c”, “3121”, “i”, *AD_ID*}; “3621” means sharing an ad to one or more shopper peer(s) and the message is {“c”, “3621”, “f”, *MOBILE DEVICE_ID*, “t”, *EMAIL_ADDRESSES*, “i”, *AD_ID*};
- (d) Deal messages (9 message types: 4121, 4221,...,4921), for example: “4621” means taking/booking a deal and the message is {“c”, “4621”, “d”, *MOBILE DEVICE_ID*, “f”, *DEAL_ID*}; “4821” means claiming a deal and the message is {“c”, “4821”, “d”, *MOBILE DEVICE_ID*, “f”, *DEAL_ID*};
- (e) Catalog messages (5 message types: 5121, 5221,...,5521), for example: “5221” means asking opinion about a product to one or more shopper peer(s) and the message {“c”, “5221”, “f”, *MOBILE DEVICE_ID*, “t”, *SHOPPER_IDS*, “i”, *PRODUCT_ID*}; “5421” means notifying a merchant that one of its product is being marked, f, i, product ID {“c”, “5421”, “f”, *MOBILE DEVICE_ID*, “i”, *PRODUCT_ID*};

(3) *Designs of messages sent from the website (Phost A and M) to the cloud service, C2DM*

There are two groups of messages designed as the following:

- (a) System configuration messages (message type: 1121...1321): “0112” means notifying a mobile application to update their product category; “0212” means notifying a mobile application to update their market segmentation variables and the values.
- (b) Inbox message: “2112” means notifying mobile applications that there is a new notification (there are 14 notification types, i.e. 001: new request of adding a shopper; 006: a new deal offer; 009: opinion request).

(4) Designs and implementation of web services in the website (Phost A and M)

The main web service function, which is called by the client (mobile application) with or without parameter (depending on the service invoked), is `cserver.php`. The libraries included in `cserver.php` are as follows:

- (a) `catalog.php` (5 functions): `share_product`, `ask_product`, `reply_ask_product`, `mark` and `unmark` handle incoming messages sent by the mobile application with type of 5121, 5221, 5321, 5421 and 5521.
- (b) `ad.php` (6 functions): `return_ad_min_point`, `return_ad_criteria`, `add_ad_notification`, `return_unread_ad_notification`, `set_ad_notification_received` and `share_ad` handle incoming messages with the type of 3121, 3221, 3321, 3421, 3521 and 3621.
- (c) `deal.php` (9 functions): `return_deal_min_point`, `return_deal_criteria`, `add_deal_notification`,
- (d) `return_unread_deal_notification`, `set_deal_notification_received`, `get_deal`, `take_deal`, `claim_deal`, and `share_deal` handle incoming messages with the type of 4121, 4221, 4321, 4421, 4521, 4621, 4721, 4821 and 4921.
- (e) `shopper.php` (3 functions): `add_shopper_request`, `accept_shopper_request` and `delete_shopper` handle incoming messages with the type of 1121, 1221 and 1321.
- (f) `inbox.php` (3 functions): `return_last_notification`, `return_all_notifications`, and `delete_notification` handle incoming messages with the type of 2221, 2321 and 2421.

(5) Implementation of classes/methods in Phost C having tasks to remotely call functions in the web services

`WebService` class (see Figure 9 for its design) provides functions to invoke services in the website. The `WebService.webGet` method code, which is called whenever the mobile application invoke a service, is depicted in Figure 14.

```

package com.phost.c.controller.general;
import java.io.*; import java.net.*; import org.apache.http.*; import org.json.*; import
android.util.Log; import com.google.gson.Gson;
public class WebService{
    DefaultHttpClient httpClient; HttpContext localContext; private String ret;
    HttpResponse response = null; HttpPost httpPost = null; HttpGet httpGet = null;
    String webServiceUrl;
    //Constructor
    public WebService(String serviceName){ . . . }
    //Method for invoking a service from the website
    public String webGet(String methodName, Map<String, String> params) {
        String getUrl = webServiceUrl + methodName; int i = 0;
        for (Map.Entry<String, String> param : params.entrySet()) {
            if(i == 0)getUrl += "?"; else getUrl += "&";
            try{getUrl += param.getKey() + "=" + URLEncoder.encode(param.getValue(),"UTF-
8");
                }catch (UnsupportedEncodingException e) {e.printStackTrace(); i++; }
            httpGet = new HttpGet(getUrl); Log.e("WebGetURL: ",getUrl);
            try{ response = httpClient.execute(httpGet);} catch(Exception e){. . .}
            . . . }
    }
}

```

Figure 14. Some part of `WebService.webGet` method implementation.

Example of invoking a service from the website which is some part of the codes in `AdEngine.isAGoodAd(String)` method is as follows:

```

//get market segmentation criteria from the website server
WebService webService = new WebService(DBManager.getInstance().
    getConfigValue(DBManager.SERVER_ADDR)+"/cserver.php");
Map<String, String> param = new HashMap<String, String>();
param.put("c", "3121");

```

```
param.put("i",ad_id); String response = webservice.webGet("", param);
```

The variable SERVER_ADDR is declared in the method initApp() in DBManager class, where it is executed only once (when the mobile application is launched):

```
public synchronized void initApp(){
    String isconfigured = getConfigValue(CONFIGURED)+"";
    updateConfig(SERVER_ADDR, "http://xwantodn.com/phost/C"); . . . }
```

(6) Implementation of functions in Phost A and M having tasks to broadcast messages via C2DM

There are two functions declared in C2DMMan.php having tasks to broadcast messages via C2DM: sendMessageToOnePhone sends a message of a certain type to a mobile device, while sendMessageToPhone sends a message to active mobile devices where their IDs are stored in the server database (see Figure 15).

```
function sendMessageToPhone($msgType, $messageText){
    $conn = connect();
    $devices = rquery("SELECT deviceToken FROM shopper where is_blocked='0'");
    if($devices){ //--- send notification to every active registered mobile device
        foreach($devices as $p)
            sendMessageToOnePhone($p['deviceToken'], $msgType, $messageText);}
    disconnect($conn);
}

function sendMessageToOnePhone($deviceRegistrationId, $msgType, $messageText) {
    //--- construct the message being sent
    $headers = array('Authorization: GoogleLogin auth=' . $GLOBALS['authCode']);
    $data = array( 'registration_id' => $deviceRegistrationId, 'collapse_key' => $msgType,
        'data.message' => $messageText);

    $ch = curl_init();
    //--- use C2DM service
    curl_setopt($ch, CURLOPT_URL, "https://android.apis.google.com/c2dm/send");
    //--- configure cURL session and send data to a mobile device
    if ($headers) curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false); curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); curl_setopt($ch,CURLOPT_POSTFIELDS,
    $data);
    $response = curl_exec($ch); curl_close($ch); return $response;
}
```

Figure 15. PHP codes of sendMessageToPhone and sendMessageToOnePhone function.

Example code for calling the function with the intention to broadcast a deal notifications is as follows: sendMessageToPhone("payload".time(), "2112#006<sepa>".\$_POST['id']. "<sepa>".\$id."<sepa>".\$category."<sepa>".\$pid);

(7) Implementation of classes/methods in Phost C having tasks to receive and process push messages sent by C2DM

Whenever there is an incoming push message detected in the mobile device, C2DMReceiver object processes the message by creating an intent object associating with C2DMCommandHandler then start the service (see Figure 16). Method of C2DMCommandHandler.onHandleIntent then handles the message accordingly (see Figure 17). It is shown in the figure, how a message having type of “2112” is handled.

```
package com.phost.c.controller.general;
//import Android classes
import android.app.PendingIntent; import android.content.BroadcastReceiver; import
android.content.Context; import android.content.Intent; import android.util.Log;
//import classes of model package and accountManagement sub-package . . .
public class C2DMReceiver extends BroadcastReceiver {
    //override onReceive method accordingly
    @Override
```



```

public void onReceive(Context context, Intent intent) {
    String action = intent.getAction();
    if ("com.google.android.c2dm.intent.REGISTRATION".equals(action)) {
        . . . . .
    }else if("com.google.android.c2dm.intent.RECEIVE".equals(action)){
//new push message, resolve command
ApplicationContext app = (ApplicationContext) context.getApplicationContext();
if(app.acquireWakeLock()){
    String[] msg = intent.getStringExtra("message").split("#");
    Log.v("c2dm","received "+msg[0]+" command message");
    Intent i = new Intent(context,C2DMCommandHandler.class);
        i.putExtra("command", msg[0]);
    if(msg.length>1){ i.putExtra("data", msg[1]); }
    context.startService(i); //start the service
    . . . . .
}
}

```

Figure 16. Some part of C2DMReceiver.java implementation

```

package com.phost.c.controller.general;
//import Android classes . . . . //import other classes . . . . .
public class C2DMCommandHandler extends IntentService{
    AdEngine ade; DealEngine de;
    //override onHandleIntent method accordingly
    @Override
    protected void onHandleIntent(Intent intent) {
    try{
        String command = intent.getStringExtra("command"); Log.v("c2dm message",command);
        //process the command
        if(command.equals("0112")){ . . . . .
        }else if(command.equals("2112")){
            DBManager.getInstance().updateNotificationFromServer();
        if(intent.getStringExtra("data") != null){
            String[] buff = intent.getStringExtra("data").split("<sepa>");
            if(buff[0].equals("004")){
                ade = new AdEngine(); ade.doIncomingAd(buff[1],buff[2],buff[3],buff[4]);
            }else if(buff[0].equals("006")){
                de = new DealEngine(); de.doIncomingDeal(buff[1],buff[2],buff[3],buff[4]);}
        }
    }
}
}

```

Figure 17. Some part of C2DMCommandHandler.java implementation

8. TEST AND EVALUATION

8.1. Prototype Testing

During the prototype development, two types of test were performed, which are white and black box test. The white box was conducted mainly in the design phase to ensure that the databases, algorithms, functions and user interfaces were designed properly. Then, after the prototype was fully developed, the black box was performed to test the functionalities of the prototype. A brief description of the black box test is as follows:

The website prototype was installed and launched in a host available in the Internet while the mobile application was installed and launched in Samsung Android mobile devices. *Phost A* URL was <http://xwantodn.com/phost/A> and *Phost M* was <http://xwantodn.com/phost/M>. *Phost C* was installed in a few mobile devices. In testing every function, positive (accepted) and negative (unaccepted) cases were used (the steps in performing the tests were in accordance with the system procedures depicted in [15]). Ten groups of functional test were performed, which were: Managing merchant and shopper registration, product categories, products, market segmentation variables; creating ads and deals; updating products, deals and ads; taking and claiming deals; managing shopper peers and sharing ads/deals with peers. All of the tests produced results as

expected. For instance, after merchants created ads/deals with certain market segment defined and the administrator approved them, the website broadcasted the ads (via C2DM) and the ads/deals notifications showed up in the *Phost C* inboxes in the mobile devices (detailed test result is presented in [4]).

8.2. Prototype Evaluation

Among the services designed and depicted in [15], the service that has not been materialized is location-aware advertising as collaboration with telecommunication companies is needed to implement it. Another service that has not been fully supported is providing fun/funny/entertaining content of products, ads and deals. Collaborations with web content designers are required to design such content.

During the system test, we did not perform stressed test yet. Therefore, it is not known how C2DM service would deliver/broadcast messages if the number of shopper reaches thousands or even millions.

9. CONCLUSION

A mobile marketing system for e-commerce implementing information personalization, privacy protection, up-to-date content, market targeting as well as social network has been designed and prototyped. The system involves two sub-systems, which are the provider website used by the provider administrator and merchant members, and the mobile application for Android mobile devices used by shoppers. C2DM service is employed to broadcast messages.

In designing and prototyping the system, the works involve: Personalizing information, targeting market and protecting privacy; designing and prototyping the provider website, the mobile application in Android environment, and the systems communication between the website and mobile application.

For the website, the functions, database and user interfaces must be designed in such a way to provide required functions for the provider administrator and merchants. As for the mobile application, packages, classes, local database as well as user interfaces must be designed and coded carefully to meet the mobile services criteria. Then, in designing the communication between the website and the mobile application, scenarios and messages format must be constructed. Web services that can be invoked by the mobile application are provided in the website.

In the future, further works are needed to provide location-aware advertisings, fun/entertaining content as well as to perform stressed test to ensure that the cloud service can handle high volume of traffic. It is also suggested to enhance the market segmentation algorithm (to personalize ads/deals), possibly by employing data mining techniques (some techniques are described in [16] and [17]). In e-commerce systems, the proposed mobile marketing system can be implemented as a standalone system as well as a sub-system of an integrated e-commerce system as suggested in [14].

REFERENCES

- [1] Ashraf, M. F., Kamal, Y. (2010), "Acceptance of mobile marketing among university students", *Mustang Journal of Business & Ethics*.
- [2] Balan, R.K., Ramasubbu, N., Prakobphol, K., Christin, N., Hong, J. (2009), "mFerio: The Design and Evaluation of a Peer-to-Peer Mobile Payment System", *Proc. of MobiSys'09*, June 22–25, Kraków, Poland.
- [3] Broeckelmann, P. (2010), "Exploring consumers' reactions towards innovative mobile services", *Qualitative Market Research: an International Journal*, Vol. 13 No. 4, pp. 414-429.
- [4] Criswanto D. Nugroho (2012), *Development of Android-Based Mobile Commerce System Providing Mobile Marketing Services*, Final Project, Informatics Department, Parahyangan Catholic University.
- [5] Eriksson, H.E., Penker, M., Lyons, B., Fado, D. (2004), *UML™ 2 Toolkit*, Wiley Publishing, Inc., USA.
- [6] Freeman, A. and Sanderson, S. (2011), *Pro ASP.NET MVC 3 Framework 3rd Ed.*, Apress Inc., USA.
- [7] Gao, T., Sultan, F. and Rohm, A. J. (2010), "Factors influencing Chinese youth consumers' acceptance of mobile marketing". *Journal of Consumer Marketing*, Vol. 27 No. 7, pp. 574–583.
- [8] Gargenta, M. (2011), *Learning Android*, O'Reilly Media, Inc., USA.
- [9] Hoffman, D.L. and Novak, T.P. (1996), *A New Marketing Paradigm for Electronic Commerce*, Vandebilt Univeristy, USA.
- [10] Jiang, W. and Sadaoui, S. (2012), *Evaluating and Ranking Semantic Offers According to Users' Interests*, *Journal of Electronic Commerce Research*, Vol. 13, No. 1, pp.1-22.
- [11] Khakpour, A.R., Demeure, I. (2008), *Designing and Prototyping an Event-Based Communication System on Mobile Ad Hoc Networks*, Institut TELECOM - TELECOM - ParisTech Département Informatique et Réseaux, France, July.
- [12] Liu, Y. and Li, H. (2009), "Mobile internet diffusion in China: an empirical study", *Industrial Management & Data Systems*, Vol. 110 No. 3, pp. 309-324.
- [13] Liu, Z., Bonazzi, R., Fritscher, B. and Pigneur, Y. (2011), "Privacy-friendly business models for location-based mobile services", *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 6 Issue 2, pp. 90-107
- [14] Meng, X. (2009), "Developing Model of E-commerce E-marketing", *Proceedings of the 2009 International Symposium on Information Processing (ISIP'09)*, Huangshan, P. R. China, August 21-23, pp. 225-228.
- [15] Moertini, V.S. and Criswanto, D.N. (2012), "e-Commerce Mobile Marketing Model Resolving Users Acceptance Criteria", *International Journal of Managing Information Technology (IJMIT)*, Vol. 4, No. 4, pp 23-40.
- [16] Mohammadnezhad, M. and Mahdavi, M. (2012), "An Effective Model for Improving The Quality of Recommender Systems In Mobile E-Tourism", *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol 4, No 1, Feb 2012, pp. 83-92.
- [17] Mohammadnezhad, M. and Mahdavi, M. (2012), "Providing a Model for Predicting Tour Sale in Mobile E-Tourism Recommender Systems", *International Journal of Information Technology Convergence and Services (IJITCS)*, Vol.2, No.1, pp. 1-8.
- [18] Nori, A. (2007), "Mobile and Embedded Databases", *Proc. of SIGMOD'07*, June 12–14, Beijing, China.
- [19] Palka, W., Pousttchi, K. and Wiedemann, D.G. (2009), "Mobile word-of-mouth – a grounded theory of mobile viral marketing", *Journal of Information Technology*, Vol. 24, pp. 172–185.
- [20] Rao, S. and Troshani, I. (2007), "A Conceptual framework and propositions for the acceptance of mobile services", *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 2 Issue 2, pp. 61 – 73.
- [21] Silberschatz, A., Korth, H.F., and Sudarshan, S. (2006), *Database System Concept 5th ed.*, McGraw Hill, USA.
- [22] Svedic, Z. (2004), *E-Marketing Strategies for e-Business*, Research Project Report for the Degree of Master of Business Administration, Management of Technology Program, Faculty of Business Administration, Simon Fraser University, Burnaby, BC, Canada.

- [23] Wium, M. (2010), Design and Evaluation of a Personalized Mobile Tourist Application, Thesis of Master of Science in Computer Science, Norwegian University of Science and Technology, June.
- [24] Yang, K. (2010), “Determinants of US consumer mobile shopping services adoption: implications for designing mobile shopping services”, Journal of Consumer Marketing, Vol. 27 No. 3, pp. 262–270.
- [25] Zarpou, T., Saprikis, V., Markos, A. and Vlachopoulou, M. (2012), “Modeling users’ acceptance of mobile services”, Electronic Commerce Research, Vol. 12, pp. 225–248.
- [26] (ITU, 2012) International Telecommunication Union, Mobile-cellular Subscriptions, <http://www.itu.int/ITU-D/ict/statistics/> [accessed, 22 Sept. 2012].
- [27] Google Developers (2012), “Android cloud to device messaging framework” available at <https://developers.google.com/android/c2dm/> (accessed 26 June 2012).

APPENDIX

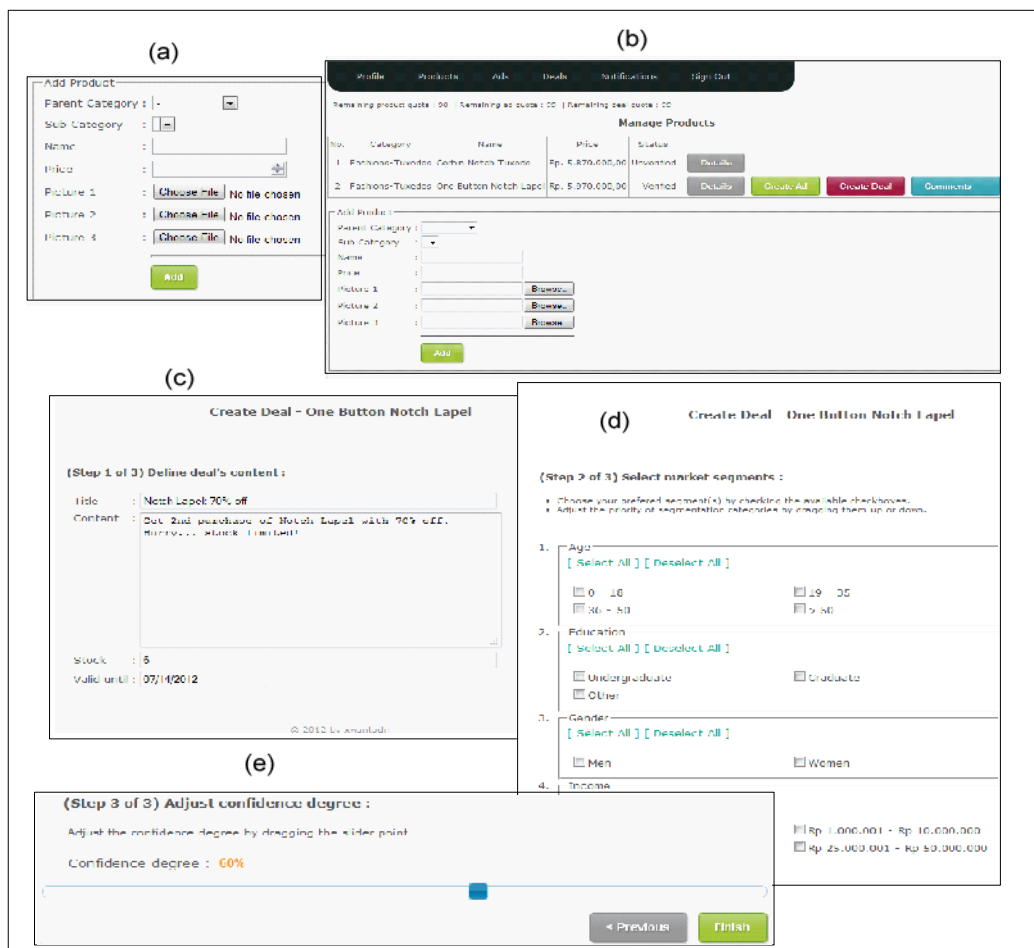


Figure A.1. Some user interface designs of *Phost M*: (a) Adding a product; (b) Managing products where the administrator can add an ad, deal or answer to incoming comments for a particular product; (c) Step 1 of adding a deal; (d) Step 2 of adding a deal: Administrator determines the market segmentation of the deal; (e) Step 3 of adding a deal: Administrator sets the desired minimum confidence level which will be used in determining target shoppers receiving the deal based on the confidence computed by the website.