

# **GOPE: A MVC Driven Mobile Framework using HTML 5 for Pervasive Device Container**

Tong-Ming Lim, Siew Hoong Angela Lee and Yeong Terng Choi

*Mobile devices such as smart phones and tablets have become part of our life these days. Well designed mobile applications for business specific mobile commerce are still very lacking. GOPE is designed to allow group-buy consumers to access to an online group-buy commerce portal with ease, efficiency and flexible functional needs. The GOPE mobile application offers few key features: location selecting option, deals viewing and selecting option, deals sharing, mobile log in, payment on-the -move, and user account management. The proposed mobile framework is designed using Model-View-Controller (MVC) design pattern with 'hoke' to enrich the group buy flexible functional requirements. The GOPE MVC 'hoke' technique has successfully fulfill the objective of the project. The GOPE prototype has been tested with a select group of testers and it has demonstrated to have achieved goals of the project.*

**Keywords:** Model-View-Controller, Pervasive Device Container, Design Patterns, HTML5, Software framework

## **1. Introduction**

Mobile devices are getting popular nowadays and this contributes to the growth of mobile application. Mobile devices have evolved rapidly in terms of performance, processing capacity, storage capacity and user-friendliness in order to meet the expectation of different groups of users. In addition, the price of mobile devices is becoming more affordable to the group of consumers. The enormous number of available mobile applications is one of the main reasons for increasing number of consumers buying smart phones and tablets. The increase demand of mobile applications is due to the different set of functions or services that they provide such as basic telephony and messaging services, gaming and synchronization service for email and database. Trends of buying behavior among the consumers have changed over the years. Figure 1 shows the trends of buying behavior among the consumers over the years and the importance of mobile applications in this respect.

## **2. Problem Statements**

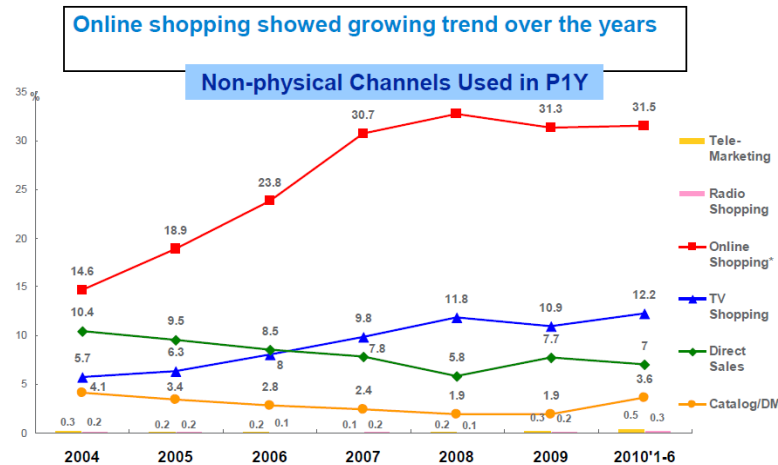
In this research project, several key problems have been identified. These issues are consolidated based on past research works and issues raised by players in the mobile app industry. The fast, dynamic and demanding mobile software development needs as a norm in today's application world; hence, requires an adaptable software design

---

Department of Information Systems, Sunway University, 5 Jalan Universiti, Bandar Sunway, Malaysia.

Email: tongmingl@sunway.edu.my, angelal@sunway.edu.my, jcyt05@hotmail.com

framework. Improper plugin due to differences in technological support for some web browsers on mobile devices and this may cause part of the web contents not viewable by the users. Mobile applications allow on-time purchase of products and services so that they do not miss the deals that interest them.



**Figure 1: Trends of Buying Behavior**

### 3. Scope of Work and Objectives

The scope of works for the GOPE mobile framework includes the investigation and application of the robust GOPE MVC design proposed in this research project and application and development of the functional requirements include Deals selection, Location selection, Deals sharing, Mobile Log in, Deals purchased and My Account management.

A set of objectives has been defined and these include an agile and robust MVC driven mobile application framework, access to latest active deals, identify user location for nearby deals, recommend deals, and digitized purchased ID.

### 4. Literature Review

To ensure that the project carries novel values, some review of existing technologies and past research works were examined. Research reviews the mobile applications, payment gateways, mobile software development technologies and tools, mobile operating systems and Model-View-Controller (MVC) design pattern.

#### A. Group-Buy Mobile Applications

There are a number of mobile applications in the market; for examples, mobile application for Groupon, LivingSocial, MyDeal and Qoo10. Groupon [2] is the mobile version of Groupon. The key features of this mobile application include allowing users to buy and redeem Groupons directly on any Android devices and track the purchased

coupons by location and expiration date. LivingSocial [3] is another mobile application for LivingSocial.com. The key features of this mobile application include helping the users to find best deals closest to them and allow them to purchase and redeem the coupon instantly. MyDeal [4] is a mobile application for MyDeal.com.my. This mobile application allows the users to access to the best deals offered everyday as well as purchase the coupon. Qoo10 [5] is a mobile application for Qoo10.my under Giosis Group and eBay. Qoo10 also allows users to view the available deals and purchase the deals directly.

	<b>Groupon</b>	<b>LivingSocial</b>	<b>MyDeal</b>	<b>Qoo10</b>
<b>Select city option</b>	Available	Available	Available	Not Available
<b>Deal's List</b>	Available	Available	Available	Available
<b>Recent Deal</b>	Not Available	Not Available	Not Available	Available
<b>Deal Details</b>	Price, Time remaining, Number of purchased, Deal Highlights, Deal fine-print, Deal Location, Deal Description	Price, Time remaining, Number of purchased, Deal highlights, Deal highlights, Fine Print, Deal Location, Deal Description	Price, Time remaining, Number of purchased, Deal highlights, Fine Print, Deal Location, Deal Description	Price, Time remaining (only for "Time Sale" category), Number of purchased (not available for "Time Sale") Deal highlights, Deal Description
<b>Deal Discussion</b>	Not Available	Not Available	Not Available	Not Available
<b>Option to share</b>	Email, Message, Facebook, Twitter.	Facebook, Twitter, Email, SMS.	Not Available	Not Available
<b>Gift Voucher and Redemption</b>	Not Available	Not Available	Not Available	Not Available
<b>Sign in or sign up page</b>	Sign in only	Available	Available	Available
<b>My Account</b>	Account info, Subscription, My Groupon (nearby, expiring, recent).	Account info, My Vouchers (unused, nearby, expiring, all)	Account info, My Deals (no category)	Coupon, Wish List, Shopping List, account info.
<b>Purchase from mobile devices</b>	Available. Choose to use credit card or PayPal. Allow to register credit card.	Available, using credit card. Allow to register credit card.	Available. Choose to use credit card, online banking (IPay88) or PayPal.	Available. Choose to use credit card, online banking (IPay88), or Q-account money.
<b>Pass code for additional security</b>	Available	Available	Not Available	Not Available
<b>Shopping cart feature</b>	Not Available	Not Available	Not Available	Available

**Table 1: Comparison of Groups Buy Mobile App**

### **B. Payment Gateways**

Payment gateways [9] help to protect payment details by encrypting sensitive information such as credit card number and bank account login information to ensure that the payment information is transferred securely. PayPal is a global e-commerce business allowing payments and money transfers to be made through the Internet [10]. PayPal provide payment processing service for online vendors, auction sites, and other commercial users, for which it charges a transaction fee. 2Checkout simplifies credit card processing by bundling the merchant account and payment gateway together [11]. 2Checkout provides global credit card processing service in multiple currencies and

languages. iPay88 helps online stores to accept comprehensive of online payment options such as credit cards, local direct bank transfer and e-wallets to complete a sales transaction via Internet channel securely [6]. MOLPay is a multi-currency payment gateway that accepts cash payments for online purchase [7]. MOLPay provides a wide range of e-commerce payment solutions such as online payment acceptance and processing as well as fraud management to payment security.

### **C. Mobile Software Development Technologies and Tools**

Table 2 illustrates open source mobile application development frameworks that are available in the market. The reasons for choosing and considering these two technologies besides Objective-C, mobile Java and Adobe Mobile developer are as follow:

- The chosen technologies are based on HTML5, which is open source and non-proprietary technology.
- No charges are required to develop the mobile application using these technologies.
- Mobile application developed using these technologies can be run on multiple platforms with minor changes or no changes required.

	<b>PhoneGap</b>	<b>Sencha Touch 2</b>
Type	Mobile App Development Framework	Mobile App Development Framework
License	Open Source (Apache Public License v2)	Open Source (GPL v3) / Commercial
Free?	Yes	Yes
Platform Support	iOS, Android, Windows Phone, Blackberry, Symbian, Bada, WebOS	Apple iOS 4 +, Android 2.3 +, and BlackBerry 6+ devices
Native Features Support	See <a href="http://phonegap.com/about/feature">http://phonegap.com/about/feature</a> for a full list	Connection, Notification, Orientation, Camera
Native UI Support	No	No
Technologies Used	HTML + JS + CSS	SenchaTouch_Library + HTML + JS + CSS
Requirements	Platform SDK, PhoneGap	SenchaTouch_Library + PhoneGap
Strengths	<ul style="list-style-type: none"> <li>• All native wrapper source code is provided so it can be customized further.</li> <li>• Simple 'drop-in libraries' concept makes it easier to develop.</li> <li>• Apps built purely in HTML, JavaScript and CSS.</li> <li>• Provide access to more native device features</li> </ul>	<ul style="list-style-type: none"> <li>• Provides UI components to give the exact look-and-feel of mobile native App (e.g. momentum-assisted scrolling, pinch to zoom and swipe gestures to navigate between screens)</li> </ul>
Weaknesses	<ul style="list-style-type: none"> <li>• Assumes that normal capabilities of a web-based application are available.</li> <li>• Recommended for applications which are heavily web dependent.</li> <li>• Lack of support for native UI components, design patterns and development tools.</li> </ul>	<ul style="list-style-type: none"> <li>• A little more complicated as it is almost a purely programmatic model (you can't manage design using html, you need to programmatically add elements to a page)</li> <li>• Provide access to limited device features</li> <li>• Its native packager only support iOS and Android</li> <li>• Steep learning curve – will be working with Sencha's API and JavaScript, with little to no HTML/CSS</li> </ul>

**Table 2: Phone Gap and Sencha Touch 2**

#### **D. Operating System Platforms for Mobile Devices**

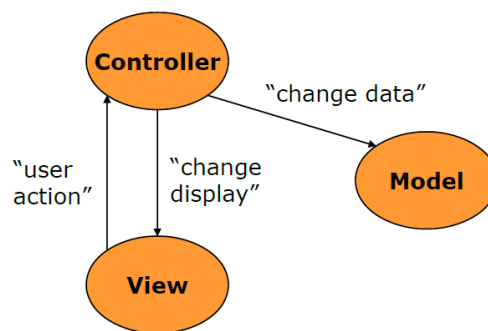
An investigation was carried out on two of the popular open source mobile development OSs: iOS and Android. Comparisons of their features' differences when applications are developed using Apache Cordova have been tabulated in Table 3.

	iOS	Android
Software Development Kit (SDK)	iOS SDK and Apache Cordova	Android SDK and Apache Cordova
User Interface	Navigation bar located on top of the application.	Navigation bar located on bottom of the application.
Language	Objective C	Java
Accessibility	XCode is only available for Mac	Eclipse is available for Mac, Linux and Windows.
Layouts	.xib files	XML

**Table 3: Comparison on Development Differences between iOS and Android**

#### **E. Design Patterns**

Design pattern was coined and popularized by the "Gang of Four" or GoF in 1994 [1]. The purpose of software design pattern is to compile and consolidate set of general reusable 'patterns' to a commonly occurring problem within a given context in software design. Popular design patterns are Façade, Singleton, Factory Method and etc. In this project, MVC is used to manage dynamic change of functional requirements, different interface design requirements for different mobile device's specifications, and deals' promotion and pricing strategies, easily and robustly. Figure 2 is an illustration of how components of the MVC design interacts cooperatively to response to the needs of the three major components of any software design project.



**Figure 2: MVC design pattern**

### **5. GOPE Application Design**

This section includes designs of the overall system architecture using various design notations such as use case models, activity diagrams and Entity-Relationship Diagram (ERD) to guide the design of GOPE mobile application and development process.

### F. GOPE Architecture

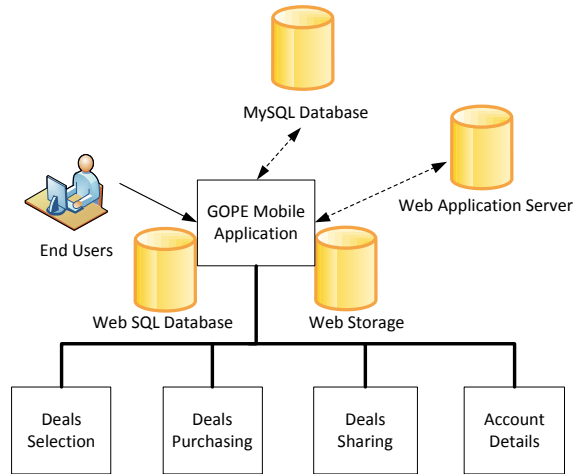


Figure 3: GOPE Architecture

The overall GOPE system architecture (Figure 3) consists of end users, a backend MySQL database, a Web Application Server, a local device Web SQL Database, a Web Storage in the mobile devices, a GOPE Mobile Application, a Deals Selection panel, a Deals Purchasing, a Deals Sharing and an Account Details.

### G. Design of GOPE Functions

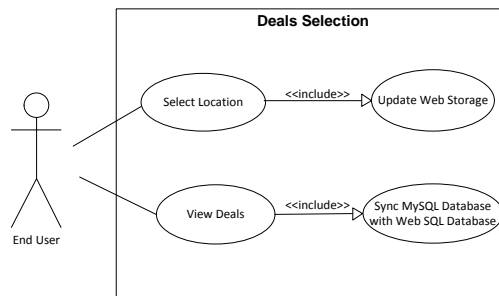
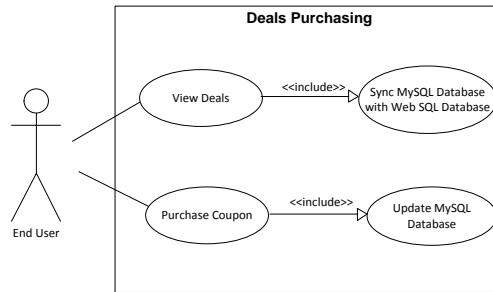


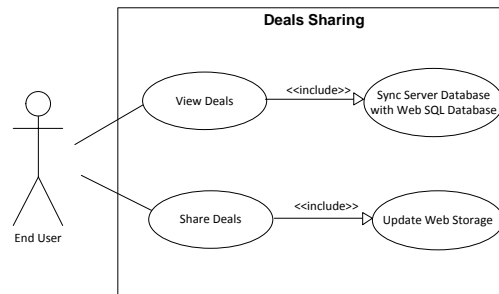
Figure 4: End User Select Deals' Location and View Deals' Information

Figure 4 illustrates “Update Web Storage” where *Select location* allows end users to select the available location that will then update the Web Storage; and “Sync MySQL Database with Web SQL Database” provides *View deals* option where end users can view a list of on-going deals that is stored in Web SQL Database. It synchronizes with MySQL database every time when end users request to view the available list of deals.



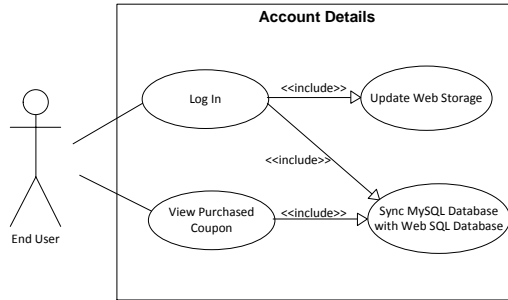
**Figure 5: End User Purchase Coupon**

Figure 5 shows “Sync MySQL Database with Web SQL Database”. The *View deals* allows end users to view a list of on-going deals that is stored in Web SQL Database, which carry out synchronization with MySQL database every time when end users request to view the available list of deals. As for “Update MySQL Database”, *Purchase coupon* permits end users to purchase coupon from desired deals using PayPal account. The MySQL database will be updated once the payment is successfully made. A new coupon with unique id and password will be inserting into the MySQL database under the user id.



**Figure 6: End User Share Deals**

Figure 6 illustrates “Share Deals”. The “Sync MySQL Database with Web SQL Database” has a *View deals* option where end users can view a list of on-going deals that is stored in Web SQL Database, which carry out synchronization with MySQL database every time when end users request to view the available list of deals. The “Update Web Storage” provides *Share deals* where end users can share the deals with the friends and family through various communicating channel such as email, Facebook and Twitter. The Web Storage will be updated with the deals’ Uniform Resource Locator (URL) as each of the deals has their own unique URL. Users can share the deals with the URL retrieved from the Web Storage.

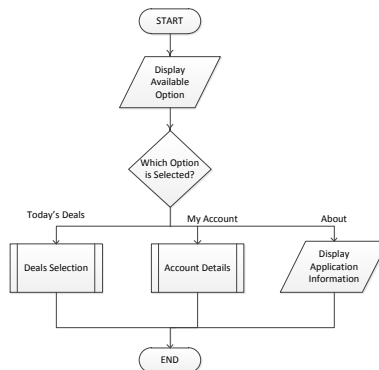


**Figure 7: End User Access to Account Information**

Figure 7 illustrates “Account Information”. The “Log In” has a *Sync MySQL Database with Web SQL Database* where based on the end user’s email. Once the end user’s email and password had been verified by the server, information of the end users will be synchronizing from MySQL database to the Web SQL Database. As for *Update Web Storage* where based on the end user’s email and password. Once the end user had successfully log in, the Web Storage will be updated so that the application will keep the end users log on to their account every time the mobile application is launched. In the “Sync MySQL Database with Web SQL Database”, the *View purchased coupon* where end users can view a list of purchased coupon that is stored in Web SQL Database, which carry out synchronization with MySQL database every time when end users request to view the list of purchased coupon.

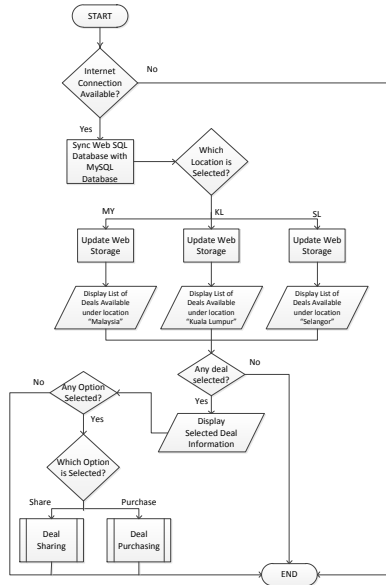
**H. System Logic Flow**

When the mobile application launched (Figure 8), three options will be display to the users. Depending on different options that chose by the users, different actions will be carried out. Three of the available options are *Today’s Deals*, where users can view a list of on-going deals; *My Account*, where users need to login to their account in order to access their account information; and *About*, where information about the application will be displayed.



**Figure 8: Overall System Flow**

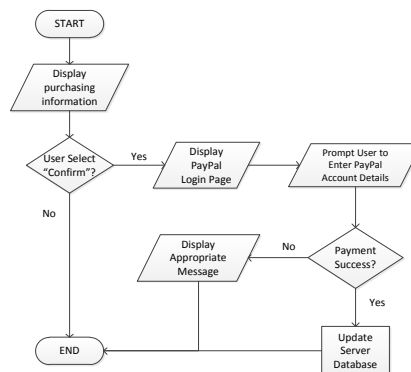




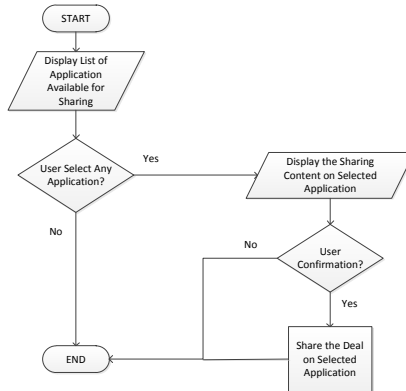
**Figure 9: Deals Selection**

In Figure 9, users are allowed to view the list of deals according to the location. The default value of the location in Web Storage is MY, which represent Malaysia and all the deals available in Malaysia can be view here. After users select their desired location, the value of location in Web Storage will be updated and different list of deals will be display according to the value of location. Users can choose to get more detail information about a deal by selecting on it. After the deal’s information is displayed to the users, they can choose to purchase or share the deal with others.

In Deals Purchasing (Figure 10), the mobile application will first display the purchasing information which includes the name of the deal and the total price to the users. Users can then click on the “Confirm” button to proceed to the next process. The PayPal login page will be shown after users click on the “Confirm” button. The page will prompt users to enter their PayPal account details. If the payment had been successfully made, the mobile application will connect to the server and update the MySQL database.

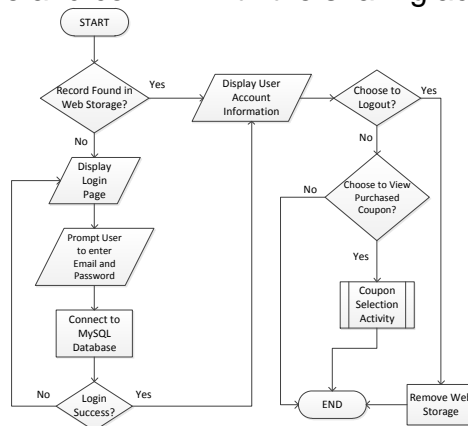


**Figure 10: Deals Purchasing**



**Figure 11: Deals Sharing**

A list of available mobile application will be displayed as users choose Deals Sharing (Figure 11). The users can choose any mobile application that is displayed to share the deals with their friends and family. The deal’s URL will be shared on the selected mobile application once user approve and confirm with the sharing action.



**Figure 12: Account Detail**

In Account Detail (Figure 12), the mobile application will first search for the record in Web Storage. If the record exists, the user’s account information will be displayed. If the record does not exist in Web Storage, a login page will be displayed to the user. The login page will prompt user to enter their account email address and password. The mobile application will then connect to the MySQL database to verify the user account. User account information will be shown if the user login successfully. After login to the account, users can choose to view their purchased coupon. Besides that, users can also choose to logout from their account. The record in Web Storage will be removed if the users choose to logout.

Figure 13 shows a sub process in Account Accessing activity. The mobile application will download the data from MySQL database and store the data in Web SQL Database when there is internet connection. Depends on the category selected by the users,

different category of coupon list will be displayed. More detail information about selected coupon will be displayed when users click on any coupon on the list.

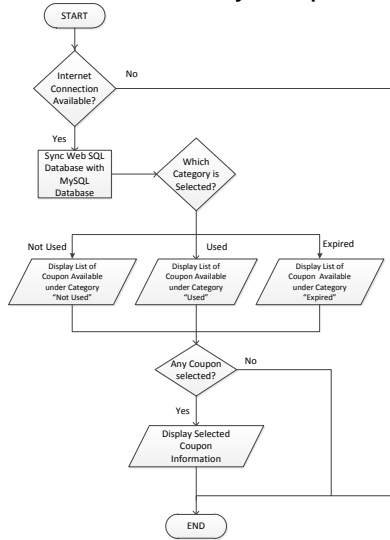


Figure 13: Coupon Selection

### I. Database Architecture

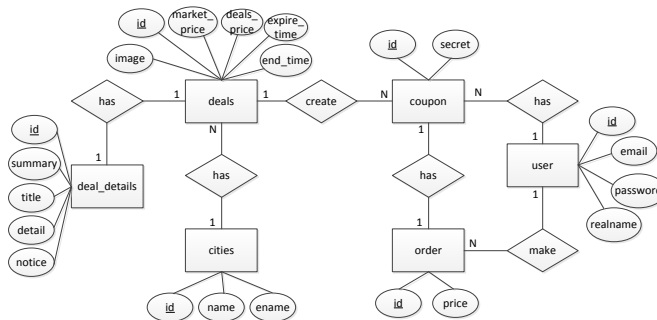


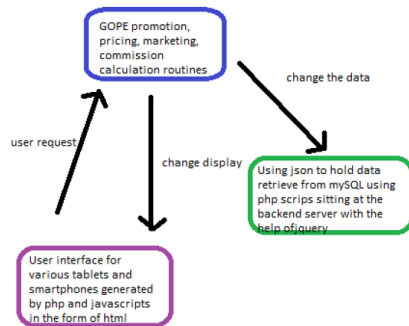
Figure 14: Database Architecture

There are six entities in both of the databases which are deals, deal\_details, cities, coupon, order and user. The *deals* entity stores details such as price, image and expire time of all the deals that are available for users to purchase. The *deal\_details* entity is the extension of *deals* entity. This is because the information will be very complicated if all the data were to store in the *deals* entity. The *deal\_details* entity contains detail information such as fine print, highlights and summary about the available deals. The *deals* entity is linked to *cities* entity because the deals are categorized by their location. The *cities* entity contains information about the deals' location. Other than that, the *deals* entity also linked to the *coupon* entity because one deal can create many coupons for the users. The *coupon* entity consists of the unique ID and secret code of the coupons. The database is designed in the way that one order can only purchase one coupon. The *order* entity consists of the unique order ID and the total price. One user can make more

than one order and that is the reason why the *user* entity is linked with the *order* entity. Information about the user account is stored in the *user* entity.

## 6. Deployment of MVC Design Pattern

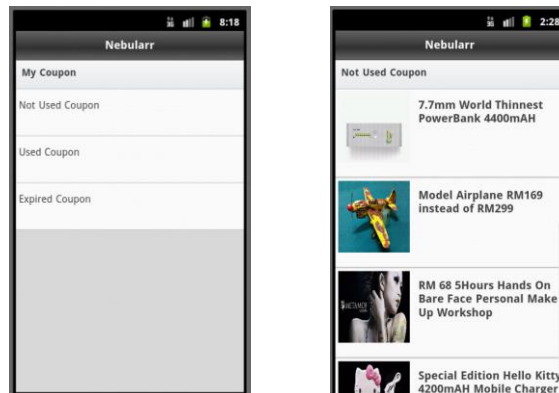
GOPE mobile application is designed using the MVC design pattern. The “View” is a set of hypertext (HTML) and Cascading Style Sheets (CSS) files that prescribes how should “View” looks like on mobile devices. The “Model” is a set of PHP and JavaScript scripts with the help of JSON data containers holding the data of the models required by GOPE. As for “Controller”, they are a set of PHP scripts and Javascript sitting at the front and backend server carry out GOPE promotion and pricing strategies as well as other processes required by the engine. Due to Non-Disclosure Contract signed by the University and the commercial partner, detail of the engine will not be disclosed in greater detail. Figure 15 illustrates the conceptual framework will be presented here for research sharing.

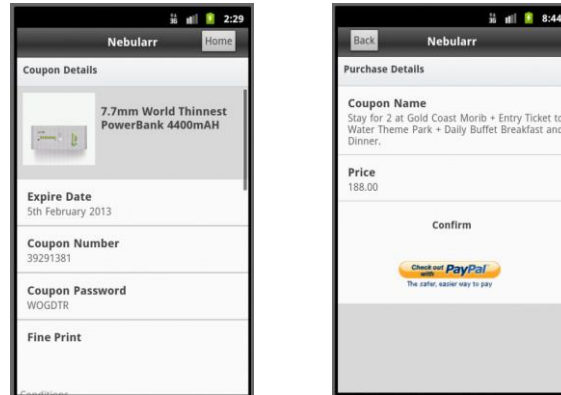


**Figure 15: Deployment of MVE in GOPE**

## 7. GOPE: A MVC Driven Mobile Group Buy Prototype

The following are some sample screen shots (Figure 16) of the GOPE prototype. The project has successfully developed into a working system and the system is now being tested.

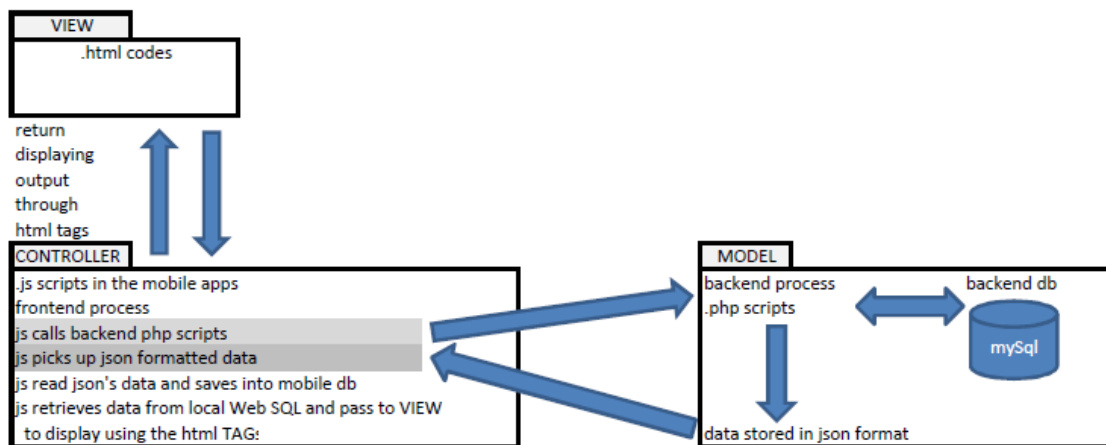




**Figure 16: Sample snapshots of GOPE prototype**

## 8. The MVC Driven GOPE Framework

The MVC driven GOPE framework is presented in Figure 17. The MVC design pattern has enabled the separation of “View”, “Controller” and “Model” in the framework. A software framework is a set of classes that are linked to provide flexibility through “hokes” for dynamic software requirements and ease of GUI adaptation. Figure 18 demonstrates the “Deals” function of GOPE framework utilizing MVC pattern.



**Figure 17: MVC driven GOPE framework**

The “Deals” activity at the “View” triggers the “Controller” to carry out data extraction from the “Model”. The “Model” selects rows of data from the database. The selected records are stored in the “Model” where json is used to hold the content of the “Model”. Contents of the “Model” are returned back to “Controller” where these contents are then displayed in the “View”. Clear separation between “View”, “Model”, and “Controller” provided by MVC design allows ease of dynamic real world requirement.

## **9. Conclusion and Proposed Enhancements**

GOPE has included all the features and it has fulfilled the objectives of the research project. However, there are rooms for future enhancements. The future enhancements identified are *increase Purchase Quantity for each Transaction* where GOPE has to adopt some interactive features that allow users to input any number of units as one purchases a merchandize; *increase the Number of Payment Gateway* where payment gateway such as online banking and credit card that are popular nowadays should be made available to the end users; *improve the Sharing Feature* where GOPE may choose to launch the sharing channel using the mobile web browser if the mobile application of the selected sharing channel is not supported; *increase more design patterns in the GOPE framework* where the GOPE framework will consider other design patterns such as Singleton and Factory in the future design.

## **References**

- Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.
- Google (2012). *Groupon – Daily Deals, Coupons*. [Online] (Updated 21 November 2012) Available at: <https://play.google.com/store/apps/details?id=com.groupon&hl=en> [Accessed 25 November 2012].
- Google (2012). *LivingSocial*. [Online] (Updated 1 November 2012) Available at: <https://play.google.com/store/apps/details?id=com.livingsocial.www> [Accessed 15 November 2012].
- Google (2012). *MyDEAL.com.my*. [Online] (Updated 22 November 2012) Available at: <https://play.google.com/store/apps/details?id=com.rebatenetworks.android.client.my> [Accessed 25 November 2012].
- Google (2012). *Qoo10 MY*. [Online] (Updated 24 October 2012) Available at: <https://play.google.com/store/apps/details?id=net.giosis.shopping.my> [Accessed 15 November 2012].
- iPay88 (2011). *Who Are We*. [Online] Available at: <http://www.ipay88.com/about.asp> [Accessed 5 November 2012].
- MOLPay (2012). *About MOLPay*. [Online] Available at: <http://www.molpay.com/About-MOLPay/q?cid=57&doit=showclass> [Accessed 5 November 2012].
- The Nelson Company (2011). *2011 Nelson Online Shopping Behavior*. [Online] Available at: <http://tw.en.acnielsen.com/site/documents/2011ShopperTrendOnlineshopping-en.pdf> [Accessed 28 August 2012].
- Wikipedia (2012). *Payment Gateway*. [Online] (Updated 23 November 2012) Available at: [http://en.wikipedia.org/wiki/Payment\\_gateway](http://en.wikipedia.org/wiki/Payment_gateway) [Accessed 25 November 2012].
- Wikipedia (2012). *PayPal*. [Online] (Updated 2 November 2012) Available at: <http://en.wikipedia.org/wiki/PayPal> [Accessed 10 November 2012].

2Checkout.com (2011). *Process Payments and Credit Cards Online*. [Online] Available at: <https://www.2checkout.com/online-credit-card-processing/> [Accessed 5 November 2012].