響應式網頁技術嵌入於行動應用之設計
A Study of Embedded Responsive Web Design in Mobile Application Development

吳哲安, 陳英一
國立台北科技大學資訊工程系
Che-An Wu, Ing-Yi Chen
E-mail: {t103598033, ichen}@ntut.edu.tw

摘　要

本論文以軟體工程中的軟體再利用(software reuse)為研究方法，探討行動應用程式在開發的過程中，若欲開發的服務已經存在於網頁平台上，開發人員應該如何克服異質平台的困難，整合這個已開發的服務，並使其能夠達到軟體再利用。藉此，降低在行動應用程式中重覆開發相同服務的成本。

為了達到上述所提之目標，本研究使用的核心技術包含，以響應式網頁設計(Responsive web design)技術，使網頁能夠隨著各種不同解析度的螢幕，自動調整頁面的排版及呈現方式，以便頁面能夠跨平台的再利用。除此之外，藉由行動裝置系統所提供的網頁容器(WebView Container)將網頁嵌入至行動應用程式，以達到網頁資源的再利用。最後，透過網頁容器(WebView Container)所提供的介面(JavaScript Interface)，讓載入的網頁，能夠以JavaScript 的語法，直接調用行動裝置的功能及原生 APIs，以達到網頁與行動應用程式的整合。

利用本研究所提出之方案，以降低開發成本、提高開發效率為目標，並且遵循軟體工程中的軟體再利用。藉由上述所提及之技術，讓網頁服務能夠整合至行動應用程式，使行動應用程式在開發的過程中，不需要根據特定平台，而重覆開發相同服務或是相同畫面，因而有效降低開發成本，同時提高行動應用程式的開發效率。

關鍵字：Software reuse、Responsive web design、Mobile application、WebView

一、研究背景

近年來隨著智慧型行動裝置逐漸地普遍，以及無線網路技術的進步，越來越多的傳統產業亦或是企業，慢慢開始將傳統建置在網頁應用程式上的服務，移轉至行動裝置上，不管是建置行動版的網頁，或是開發行動應用程式，這些現象都足以證明行動化世代的來臨。

隨著行動化世代的來臨，使用者的操作習慣也同時有所改變，從過去沒有智慧型行動裝置時，以桌上型電腦或筆記型電腦為主的定點式瀏覽，改變成現在的行動式瀏覽，只要使用者攜帶著行動裝置，便可隨時隨地的接收資訊、瀏覽網際網路、以及使用網頁服務。「低頭族」這個名詞恰好可以表現出在行動化世代下的使用者行為。

智慧型行動裝置普及的情況下，使用者開始頻繁的在智慧型手機、平板電腦、桌上型電腦、或是筆記型電腦之間交互使用，顯示了現在是多螢幕的世代，也因為使用者透過各種不同的媒體載具瀏覽網頁，使得以傳統的方式設計出的網頁，因為固定的版面配置、固定的畫面元素，無法同時滿足多螢幕瀏覽的問題。舉例來說，傳統網頁設計模式設計出的頁面，以桌上型電腦或是筆記型電腦瀏覽，可以正常舒適地瀏覽其內容(圖 1)，但此時，將瀏覽的載具取代為智慧型手機或是平板電腦(圖 2)，此時可以發現，若網頁沒有做進一步地優化其網頁顯示的內容，整張網頁就會被縮的非常小，甚至完全看不到字體。因此，為了因應多螢幕世代的需求，網頁開發人員不管是使用響應式網頁設計[1]，又或是重新開發行動版網頁，都必須重新思考多螢幕的條件下，應該要如何設計網頁呈現的內容，才能夠提供使用者在不同解析度的裝置下，有著相同舒適的使用者體驗。

圖 1 傳統網頁於桌上型電腦瀏覽

圖 2 傳統網頁於行動裝置瀏覽

二、研究動機與目的

本論文之研究動機來自於行動化的驅動下，越來越多企業或是傳統產業，已經逐漸投入行動應用程式的開發，將其產業背景中所提供的服務行動化，以便滿足目前使用者對於行動應用的需求。

然而，開發行動應用程式的成本，因需求及功能而異，若行動應用程式中欲提供的服務，已存在於網頁平台上，那麼開發人員是否能夠透過現有行動應用的技術，將網頁平台上的服務直接嵌入行動應用程式達到再利用的目標，藉由此目標，降低開發行動應用程式的成本。

基於混合式行動應用程式的開發模式，使得開發人員能夠以網頁的技術開發行動應用程式，這也剛好能滿足前一段所提出「將現有網頁服務再利用」之需求，因此，本研究將以混合式行動應用程式的開發模式，進行後續的系統開發，以達到本論文提出之研究目標。

本研究最主要之研究目的為重覆利用現有網頁服務，藉此降低 APP 重覆開發相同服務之成本、並且提高 APP 開發效率以及品質。除此之外，希望能透過軟體工程中的軟體再利用方法，將現有已存在於網頁平台上的服務，套用響應式網頁設計技術，使該網頁能夠相容於各個不同螢幕解析度的裝置。最後，以行動應用程式中的 WebView 元件載入該網頁，藉由這樣的方法，讓網頁中的服務能夠於行動應用程式中再利用，而不需要再重新將同樣功能的服務，依據各個不同平台的原生語言各自實作。

三、相關技術分析

響應式網頁設計（Responsive Web Design）

響應式網頁設計為一種網頁的設計技術，能夠使網頁隨著不同解析度的螢幕，自動調整頁面的排版或位置，使網頁只需要同一套程式碼，便能夠讓使用者在桌上型電腦、筆記型電腦、平板電腦、或是智慧型手機等不同的平台，都能夠看到一樣的網頁內容，且不會影響閱讀的便利性。

圖 3 響應式網頁設計示意圖

CSS Media Query

CSS Media Query 是響應式網頁設計的主要核心技術，網頁開發人員可以透過 Media Query 偵測目前瀏覽網頁的裝置解析度或方向，並且自動套用相對應的 CSS 檔，進而使網頁隨著螢幕大小自動改變排版方式或顯示內容， 使用 Media Query 時主要分為三個部份，第一個部分，是偵測媒體種類，也就是 Media Type，其中種類包括一般螢幕(Screen)、投影機(Projection)、列表機(Print)…等，第二部分為 Query，條件查詢例如 And、Only、Not，第三部分為 Media Feature，媒體特性，例如: 最大寬度、最小寬度、旋轉等…，Media Query 基本語法(圖 4)所示：

```
/*螢幕小於480px套用*/
@media screen and (max-width:480px) {
  .demoDiv { color: red }
}
```

圖 4 CSS Media Query 基本語法

Ex1.單一條件
範例為偵測螢幕寬度大於 480px 才會套用此 CSS
@media screen and (min-width:480px) {
}

Ex2.同時符合兩種條件
範例為偵測螢幕寬度在 480px 以及 960px 的範圍內，則套用此 CSS
@media screen and (min-width: 480px) and (max-width: 960px){
}

行動網頁（Mobile Web）

廣義的來說，行動網頁就是一般網頁執行於行動裝置上，無須安裝程式在行動裝置，但必須透過行動裝置中的瀏覽器，輸入網址或搜尋的方式才能夠執行。

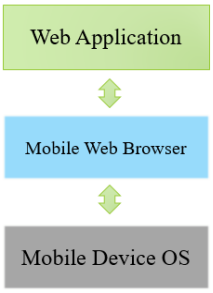

圖 5 網頁行動版

原生行動應用程式（Native App）

原生行動應用程式指的是，利用行動裝置作業系統廠商，所提供的 SDK，或建議的程式語言進行開發，舉例來說 Android 以 Java 開發、iOS 則是使用 Objective C，因此，需要針對平台進行不同的開發，無法跨平台。

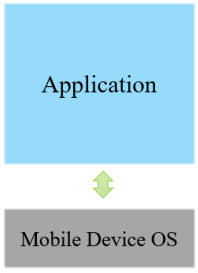

圖 6 原生行動應用程式

混合式行動應用程式（Hybrid App）

混合行動應用程式，開發的過程中，部分程式碼會以網頁的技術編寫，例如：HTML、JavaScript、CSS，而這些程式會被包裹在原生行動應用程式中，透過行動裝置的瀏覽器引擎，將畫面渲染出來，使得行動應用程式能夠以網頁當作使用者介面，而不需要耗費額外成本及時間，針對特定平台，撰寫多套使用者介面。

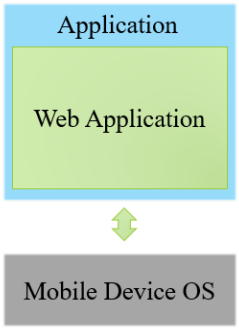

圖 7 混合式行動應用程式

PhoneGap/Cordova Framework

PhoneGap/Cordova 為一套開源的行動應用程式開發框架，目的在於能夠讓開發人員，以 Html、JavaScript、CSS 等 Web API 開發跨平台的行動應用程式，PhoneGap/Cordova Framework 提供一個介面，讓開發人員能夠以 JavaScript 的方式調用行動裝置的 API，例如相機、羅盤、加速器等硬體系統資源，再搭配上 Html5、CSS3 等技術，使得開發人員可以很快地開發跨平台 APP，而不需要去寫任何的原生程式碼。

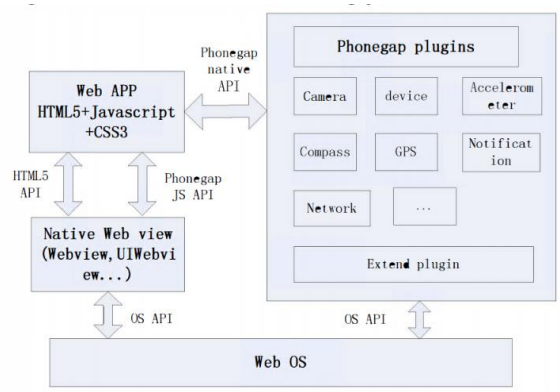

圖 8 PhoneGap 系統架構圖[22]

WebView

WebView 元件為行動應用程式中用以顯示網頁內容的元件，它的運作原理為與瀏覽器相似，透過 Http Get 或是 Http Post 的方式，向後台發出請求以取得網頁內容，並且利用 Web Kit 這個渲染引擎來將網頁顯示出來。基於這個元件，開發人員能夠客製化自己的網頁瀏覽器，或是將這個 WebView 元件嵌入在行動應用程式中，使得應用程式不需要額外開啟外部瀏覽器視窗，而可以直接於應用程式中顯示網頁內容。

行動應用程式的開發模式分為三種，其中混合應用程式(Hybrid APP)之所以能夠同時以原生語言和網頁的技術開發，但最後封裝出來的卻又有著原生應用程式(Native APP)的特性，其中最為關鍵的元件就是 WebView。其架構(圖 9)所示，而它背後運作的原理為利用原生語言建構出原生應用程式的外殼，接著在這個外殼內 New 一個由原生平台所提供的 WebView 元件，利用這個 WebView 元件，將網頁資源包含 Html、JavaScript 和 CSS 載入至 WebView 元件中，再透過原生語言設定 WebView 元件的相關參數，使得在混合應用程式中，能夠同時以原生語言的方式以及網頁的技術來開發行動應用程式。

圖 9 混合應用程式元件架構圖

WebView 的用法以 Android 為例：

```
WebView webView = new WebView(this);
webView.loadUrl("http://www.demo.com");
```

透過 WebView 的 load URL，以 Http Get 或是 Post 的方式，取得網頁並且嵌入在 WebView 元件中，開發人員只要設定這個元件的 Visibility 即可顯示或隱藏這個網頁，另外，可以透過取得 WebView 物件的 WebSettings 來設定一些網頁屬性，例如，是否允許嵌入的網頁使用 JavaScript、或是否保留網頁暫存機制等功能。

除此之外，以 Android 為例，有兩個與 WebView 有關的重要類別：

1.WebViewClient

此類別主要幫助 WebView 處理各種網頁的事件，像是當 WebView 發出 Request 時，就會觸發 WebViewClient 類別中的 onPageStart 事件，當網頁載入成功，會觸發 onPageFinish 事件，載入失敗，則會觸發 onReceiveError，開發人員可以透過這個類別，加以實作客製化自己預期的執行邏輯。

2.WebChromeClient

此類別主要幫助 WebView 處理網頁中 JavaScript 的動態事件、或是其他動態效果，例如 onJsAlert 事件、onJsPrompt 事件、onJsConfirm 事件等，開發人員可以透過此類別，客製化相關元件。

JavaScript Interface

JavaScript Interface 又可以稱作為 JavaScript Bridge，這個介面是由原生程式碼所撰寫而成，其目的顧名思義為 JavaScript Code 與原生程式碼 (Native Code)之間建立出一個介面以及橋樑，它的實作原理為透過 JavaScript Runtime 與 WebView 元件的底層機制溝通，因此，必須搭配上 WebView 元件才能夠使用，(圖 10)所示，在混合應用程式中，JavaScript 可透過 JavaScript Interface 直接呼叫原生語言所提供的 APIs，藉此達到調用行動裝置底層硬體功能的目的。

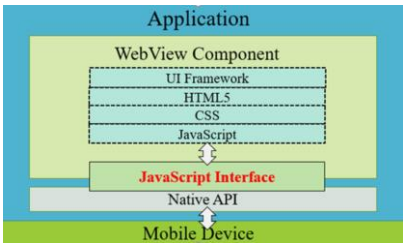

圖 10 混合應用程式元件架構圖

當開發人員以混合的方式同時以網頁的技術以及原生語言開發行動應用程式時，在某種情境之下，可能會需要以 JavaScript 的程式碼呼叫及調用原生程式的方法或是 APIs。舉例來說，JavaScript 以及 Native Code 兩者都有提供顯示訊息視窗的 APIs，JavaScript 使用的是 Alert function，Native Code 則使用 Dialog Builder，但 JavaScript 的 Alert 視窗往往沒有 Native Code 的訊息視窗來的好看，此時 JavaScript Interface 則可以扮演著中繼的角色，以原生語言在介面中實作一個原生的訊息提示視窗，使得 JavaScript 的程式碼得以透過 JavaScript Interface 呼叫這個由原生程式撰寫而成的訊息視窗，取代原本 JavaScript 的 alert 方法，藉由這樣的方式，使得用網頁技術開發的開發人員，得以讓應用程式中的提示訊息視窗更像行動應用程式。

另一方面，以網頁的技術開發行動應用程式，免不了還是會遇到需要使用行動裝置所提供的硬體功能，像是相機、藍芽、檔案系統、加速器、陀螺儀…等硬體功能，單純的使用 JavaScript 語法是無法使用這些功能的，唯有透過作業系統建議使用的原生語言，才能呼叫這些硬體功能相對應的 APIs，在這種情況下，就必須透過 JavaScript Interface 這個介面，先在介面中實作各種方法後，再由 JavaScript 透過 JavaScript Interface 去轉呼叫相對應的方法，藉此達到以 JavaScript 的語法調用行動裝置底層硬體功能。

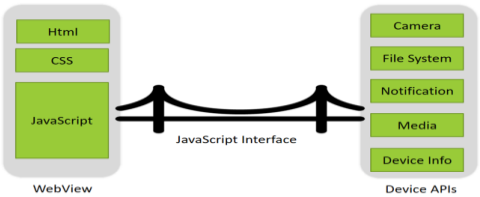

圖 11 混合應用程式元件架構圖

JavaScript Interface 實作範例：

Step1 建立介面類別並實作方法

```
public class JSInterface {
    @JavascriptInterface
    public void showToast(String text) {
        Toast.makeText(this, text, Toast.LENGTH_SHORT).show();
    }
}
```

Step2 於 WebView 建立橋樑

```
webView.addJavascriptInterface(new JSInterface(this), "Android");
```

Step3 透過 JavaScript 呼叫介面中的方法

```
function triggerJavaCode(){
    Android.showToast('Hello');
}
```

Multi Page Application (MPA)

MPA 是一種傳統網頁應用程式的設計模式，每當應用程式顯示資料或是將資料送回伺服器時，都必須向伺服器請求新的頁面，並且利用網頁瀏覽器引擎將這個頁面重新渲染，也就是說使用者在網頁上做的每一個動作，都需要重新更新網頁畫面。

圖 12 MPA 架構示意圖

Single Page Application (SPA)

相較於傳統網頁設計模式(MPA)，Single Page Application，為單一網頁應用程式，所謂的單一網頁應用程式，指的是應用程式只有在初始化時，會向伺服器請求一個網頁，而這個單一網頁在載入的同時，會將所有網頁必要的程式碼，如 HTML、JavaScript、CSS 通通都載入到這一個單一頁面上，而在使用者在使用網頁的過程當中，所有的網頁資源都不需要重新載入，每次只需要透過資料的更新，使用者體驗較好，也是近幾年，網頁設計的趨勢。

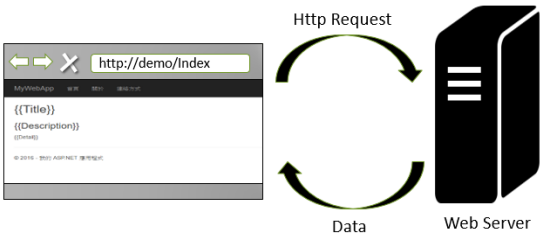

圖 13 SPA 架構示意圖

四、系統分析

需求分析

本論文提出之主要研究目的為以軟體再利用之方法，降低開發人員重複開發相同服務的成本，以及提高產品開發效率以及其品質，因此為了能夠檢視本文提出方法之可行性，本研究將因應行動化及行動應用的需求，並且以金融證券業為例，藉由重覆利用網頁上的服務，快速地建置出一個股市行動應用程式。

功能分析

股市服務行動應用程式，依據其產業特性及其提供之服務，可以歸類出兩個主要的功能，第一為商品報價功能，第二則是商品下單功能，第三個功能為商品庫存功能，以下將針對這兩大功能進行個別的分析。

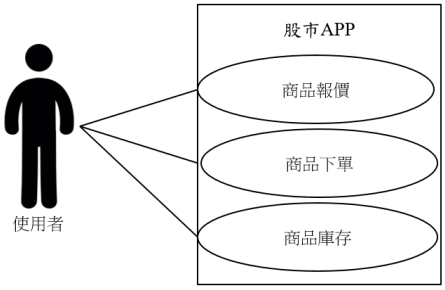

圖 14 使用者案例

1.報價功能

商品報價主要功能為呈現商品相關資訊以及市場上的最新價格，以幫助使用者選擇適當的商品。股市中的報價功能其中包含，當日行情報價、自選股報價、證券報價、期貨報價、選擇權報價…等功能，而每一種商品的報價資訊，不外乎有成交價、均價、漲跌價、檔跌幅、最高與最低價…等資訊需要提供給使用者參考。

2.下單功能

商品下單功能主要為提供使用者進行商品的買賣行為，根據股市交易行為，下單的模式分為以下幾種，一般現股買賣、零股買賣、融資融券、現股當沖、資券當沖等交易模式。

3.庫存功能

商品庫存之主要功能為顯示使用者目前所持有的商品，以及該商品買進時的價位、數量、及其他商品相關資訊，透過此功能讓使用者能快速的瞭解目前持有的庫存資訊。

流程分析

根據目前線上的股市服務系統，藉由進行使用者操作流程分析，進而歸納出使用者在使用股市系統時，一般較為常見的操作流程。首先，使用者透過股市服務系統使用報價功能的服務，針對當日行情、或是其關注的商品進行報價查看，瞭解完該商品的市場行情後。其次，則會使用下單功能，對於他有興趣的商品進行交易，交易完成後，透過商品庫存之功能，使用者可以隨時追蹤該商品的最新狀況，或是利用商品資訊這項功能，來更進一步的瞭解該商品的詳細資訊。

五、系統設計

報價模組設計

報價模組依據商品的種類，將分為三個部分，依序為證券報價、期貨報價、選擇權報價。

由於本系統之報價功能，要直接使用已經建置在網頁平台上報價服務，因此，報價模組中的證券報價、期貨報價、選擇權報價，三個報價模組都會擁有一個 WebView 元件，分別載入網頁平台上相對應的服務(圖 15)所示。

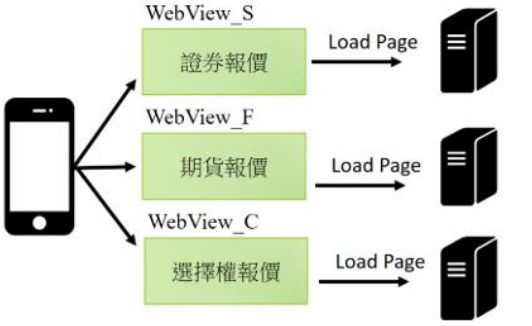

圖 15 使用者案例

下單模組設計

下單模組依據商品的種類，將分為三個部分，依序為證券下單、期貨下單、選擇權下單。

表 1 證券物件設計

prodInfo 參數	說明
1.stock_id	商品代碼
2.tradeKind	0:現, 3:資, 4:券
3.buysell	B:買進, S:賣出
4.quantity	數量
5.price	價錢
6.priceLH	0:限價單, L:跌停, H:漲停
7.time	西元年/月/日/時/分/秒/毫秒

庫存模組設計

本服務透過後台 API 提供即時庫存及與市價差距的功能，以「商品種類」、「商品代碼」、「履約月份」、「是否為海外商品」、「顯示類別」、「單式/複式」、「買權/賣權」做查詢。

庫存模組依據商品的種類，將分為三個部分，依序為證券庫存、期貨庫存、選擇權庫存。

表 2 查詢庫存電文格式

查詢庫存電文參數	說明
1.marketType	S:證券, F:期貨, C:選擇權
2.idNo	使用者身分證號

表 3 證券庫存物件格式

庫存回傳參數	說明
1. prodId	商品編號
2. prodName	商品名稱
3. numberOfShareHeld	持有數量(單位：股)
4. costPrice	均價
5. costValue	成本
6. marketPrice	現價
7. marketValue	現值
8. gainOrLoss	損益
9. returnOnInventment	報酬率

流程設計

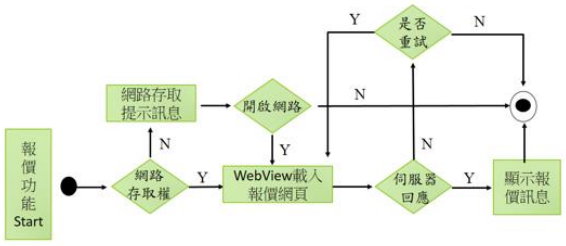

圖 16 報價功能流程圖

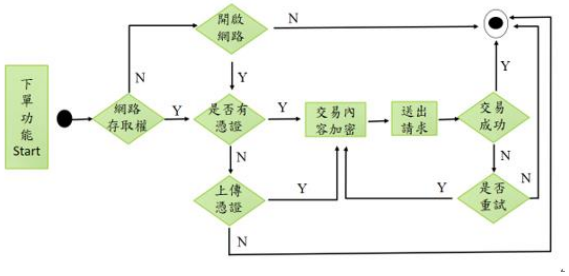

圖 17 下單功能流程圖

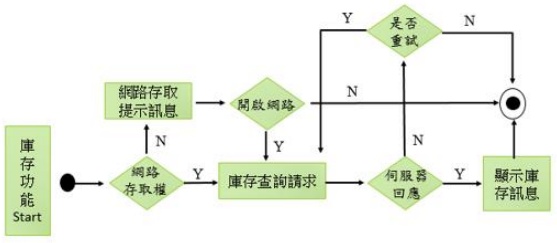

圖 18 庫存功能流程圖

系統架構設計

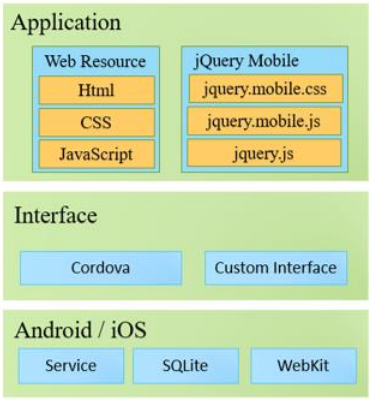

圖 19 系統架構圖

六、系統實作與成果

```
<!-- 報價模組 -->
<application>
    <id>FUNCT001200</id>
    <name>證券報價</name>
    <uri>apps/FUNCT001200/index.html</uri>
    <imgUri>images/icon_1200.png</imgUri>
</application>

<application>
    <id>FUNCT001205</id>
    <name>期貨報價</name>
    <uri>apps/FUNCT001205/index.html</uri>
    <imgUri>images/icon_1205.png</imgUri>
</application>

<application>
    <id>FUNCT001210</id>
    <name>選擇權報價</name>
    <uri>apps/FUNCT001210/index.html</uri>
    <imgUri>images/icon_1210.png</imgUri>
</application>
```

圖 20 報價模組程式碼

```
<!-- 下單模組 -->
<application>
    <id>FUNCT001015</id>
    <name>證券下單</name>
    <uri>apps/FUNCT001015/index.html</uri>
    <imgUri>images/icon_1001.png</imgUri>
</application>

<application>
    <id>FUNCT001016</id>
    <name>期貨下單</name>
    <uri>apps/FUNCT001016/index.html</uri>
    <imgUri>images/icon_1004.png</imgUri>
</application>

<application>
    <id>FUNCT001010</id>
    <name>選擇權下單</name>
    <uri>apps/FUNCT001010/index.html</uri>
    <imgUri>images/icon_1006.png</imgUri>
</application>
```

圖 21 下單模組程式碼

```
<!-- 庫存模組 -->
<application>
    <id>FUNCT001050</id>
    <name>證券庫存</name>
    <uri>apps/FUNCT001050/index.html</uri>
    <imgUri>images/icon_1050.png</imgUri>
</application>

<application>
    <id>FUNCT001060</id>
    <name>期貨庫存</name>
    <uri>apps/FUNCT001060/index.html</uri>
    <imgUri>images/icon_1060.png</imgUri>
</application>

<application>
    <id>FUNCT001070</id>
    <name>選擇權庫存</name>
    <uri>apps/FUNCT001060/index.html</uri>
    <imgUri>images/icon_1070.png</imgUri>
</application>
```

圖 22 庫存模組程式碼

圖 23 應用程式選單畫面

圖 24 報價系統畫面

圖 25 下單系統畫面

圖 26 庫存系統畫面

七、結論

在行動應用尚未普及前，各行各業都透過建置網頁應用程式的方法，將其所提供的服務電子化，例如：線上訂票、線上購物、線上報價…等等服務，使得民眾能夠透過電腦便利地使用這些服務。隨著現今智慧型行動裝置的普及，各行各業為了因應這股行動化的趨勢，紛紛投入開發行動應用程式。然而，開發行動應用程式的成本，因需求及功能而異。因此，本研究希望提出一個方法，將原有的網頁套用響應式網頁設計技術並嵌入至行動應用程式中，藉此重複利用網頁服務，以降低開發行動應用所耗費的成本。

由第四章的系統建置以及系統建置成果可以發現，使用本研究所提出之方法「響應式網頁嵌入於行動應用程式」，透過行動裝置作業系統中所提供的 WebView 元件載入網頁中的報價服務，不僅能夠直接在欲開發的行動應用程式中，直接使用已開發完成的網頁報價服務模組，同時還能夠藉由此方法，有效地降低行動應用程式重複開發該模組的成本。

總結以上，未來有關行動化的應用會持續發展，若網頁技術能夠與行動技術進行整合，取網頁的跨平台優點以及行動能存取裝置底層功能的優點，相信在網頁與行動相輔相成的發展之下，兩者之間的鴻溝將會越來越小，而在未來，這樣子的混合式應用也將會越來越廣泛及多元。

八、未來展望

在未來，行動應用的發展會越來越多元，而網頁技術與行動技術的整合應用也會越來越廣泛，基於這個原因，行動應用程式中的 WebView 元件就顯得更為重要。舉例來說，現在網頁中隨處都可以看見有投放廣告，這些廣告也都是用網頁的技術開發而成，且隨著行動化的趨勢下，使用行動 APP 的人也越來越多。若將原本的廣告內容套用響應式網頁技術，並使用 WebView 元件將此廣告網頁載入，透過響應式網頁技術的特性，自動偵測解析度的大小調整其畫面的內容，如此一來，便能更以最小的開發成本，重覆的再利用既有的廣告網頁，達到廣告投放於行動應用程式的目的。除此之外，透過這種以 WebView 載入網頁的方式，還有一個優點，也就是能夠快速因應內容異動，由於網頁放在伺服器上，因此，只要將新的網頁內容佈署至伺服器上，在行動應用程式中可以很快地看到變更後的結果，而不需要透過相關機制送至官方審核，這對於廣告這種內容快速變化的產業相當有幫助。

圖 27 廣告投放示意圖

參考文獻

[1] Wikipedia, Responsive web design, https://en.wikipedia.org/wiki/Responsive_web_design

[2] Wenhui Peng, Yaling Zhou. "The Design and Research of Responsive Web Supporting Mobile Learning Devices". 2015 International Symposium on Educational Technology (ISET). IEEE, 2015

[3] S. Mohorovičić. "Implementing responsive web design for enhanced web presence". Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on. IEEE. 2013

[4] 陳雪鉼，具使用者行為資訊之 B2B2C 響應式網頁平台設計，碩士論文，世新大學資訊管理學研究所，台北，2015

[5] 陳顥憲，響應式網頁設計對線上購物意願之影響-以單速腳踏車網站 Vesolo.com 為例，碩士論文，靜宜大學資訊管理學研究所，宜蘭，2016

[6] Pinku Hazarika. "Recommendations for Webview Based Mobile Applications on Android". Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on. IEEE, 2014

[7] 張崴，混合式 Android 應用程式安全機制之研究，碩士論文，臺灣大學電機工程學研究所，台北，2015

[8] Hermann Kaindl. "Software Reuse Based on Business Processes and Requirements". 2013 20th Asia-Pacific Software Engineering Conference (APSEC) (Volume:2). IEEE, 2013

[9] M. D. Lubars. "Reusing designs for rapid application development". Communications, 1991. ICC '91, Conference Record. IEEE International Conference on. IEEE, 1991

[10] Meena Jha. "A comparison of software reuse in software development communities". Software Engineering (MySEC), 2011 5th Malaysian Conference in. IEEE, 2011

[11] Wikipedia, Code Reuse, https://en.wikipedia.org/wiki/Code_reuse

[12] Wikipedia, Use case, https://en.wikipedia.org/wiki/Use_case

[13] Wikipedia, Sequence diagram, https://en.wikipedia.org/wiki/Sequence_diagram

[14] Wikipedia, Code reuse, https://en.wikipedia.org/wiki/Code_reuse

[15] 蓋索林(gasoline), Google!Android 手機應用程式設計入門(第五版), 松崗, 2013

[16] Wikipedia, Mobile app, https://en.wikipedia.org/wiki/ Mobile app

[17] Wikipedia, Android KitKat, https://en.wikipedia.org/wiki/Android_KitKat

[18] Wikipedia, iOS, https://en.wikipedia.org/wiki/IOS

[19] Heena Ahuja. "Optimization of the issues in the migration from Android Native to Hybrid Application: Case study of Student's Portal application". Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on. IEEE. 2014.

[20] David Jaramillo. "A secure extensible container for hybrid mobile applications". Southeastcon, 2013 Proceedings of IEEE. IEEE. 2013

[21] Apoorva Jindal, Christopher Crutchfield, Samir Goel, Ravi Kolluri, Ravi Jain. "The mobile web is structurally different". INFOCOM Workshops 2008, IEEE, 2008

[22] LiTian, HuaichangDu, LongTang, YeXu. "The discussion of cross-platform mobile application based on Phonegap". Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on, IEEE, 2013

[23] Ali Mesbah, Arie van Deursen. "Migrating Multi-page Web Applications to Single-page AJAX Interfaces". 11th European Conference on Software Maintenance and Reengineering (CSMR'07). IEEE, 2007

[24] Wang Ning, Li Liming, Wang Yanzhang, Wang Yi-bing, Wang Jing. "Research on the Web Information System Development Platform Based on MVC Design Pattern". Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on (Volume:3). IEEE, 2008