

## The Research and Application of Modular Mobile Phone Web Front-end Based on MVCS Mode

Zhang Wen-sheng<sup>1, a</sup> and Chen Hong<sup>1, b</sup>

<sup>1</sup>*Xi'an Eurasia University, No.1 Eurasia Road, Xi'an, Shaanxi, China*  
<sup>a</sup>*496690080@qq.com, <sup>b</sup>7646524@qq.com*

### Abstract

*With the popularity of mobile devices in recent years, the application of mobile phone Web front-end has become a research hotspot. This paper has constructed the MVCS (Model-View-Controller-Store) model by taking the MVC model as the framework, the ExtJS modular as the basis, and data storage (Store) as the data container, explained by taking the book information system the methods and steps of adopting such model to complete the design of modular mobile phone Web front-end, and displayed the real-time data chart of mobile phone web front-end. Finally, it also has discussed the research trend.*

**Key words:** *Modular; MVCS Model; MVC Model; WEB UI*

### 1. Introduction

Recently, with the development of the HTML5 and CSS3, the demand for mobile devices to be developed by Website or Web application program is expanding gradually. The viewpoint that the Website or the Web application program based on HTML 5, CSS 3 and JavaScript is the trend of the future" is widely accepted by more and more IT sessions that start to look for a framework that can use HTML5 and CSS3 to develop the Website or the Web application program [1].

The MVC mode (Model-View-Controller) is a type of software architecture pattern in the software engineering; it divides the software system into three basic parts: Model, View and Controller. The MVC model is designed to achieve a dynamic programming, so as to make it simple in the subsequent modifications and extensions, make it possible to put the repeated procedures into use. In addition, the model can make the program structure more intuitive by simplifying the order of complexity. The software system gives each basic part its proper function through the simultaneous separation of its basic part. Professionals can achieve its specific functions by grouping its expertise; [2]

Such as:

Controller---- Responsible for forwarding a request and processing the request;

View---- Interface designers carry out the graphical interface design;

Model-Programmers write program functions (to implement algorithm), database experts carry out the data management and database design (to achieve specific functions).

Since the past five or six years, with the various sizes of the intelligent mobile phone and tablet computer emerging in an endless stream, more and more people use mobile devices to access the Internet, which gives the mobile front-end code platform higher requirement on the cross-platform, reusability and maintainability. If there is a framework that has a good reused cross platform and is easy to be maintained, it can greatly improve the development efficiency [3]. Therefore, the mobile front-end developers are also exploring constantly, and MVC model is widely used in the

background of Web development, which has achieved the separation of the display module and the data logic processing module. Therefore, many companies has began to learn the application of the MVC in the back-end, and established the front-end MVC architecture, with the representative framework of ExtJS, Backbone *etc.*, [4].

However, the more the mobile phone front-end development pays attention to the user experience, the more complexity there is in the development, which makes the development of mobile front-end become more difficult, increases the possibility of occurring bugs, and increases the complexity of software testing, and then those complexity will extend the software development process, developers have to pay extra development costs to organize and manage their own framework.

The design of mobile phone front-end based on MVCS model has proposed the methods to solve the above problems, and made it possible for the front end reconstruction. It can guarantee the code clean, easy to test and easy to maintain by using the MVCS software design pattern.

In addition, this model also supports modular programming. The basic of modular is object oriented, on the bases of object oriented programming, one or more widgets are packed together to form a module, and make it form a new component (components that meet our needs), and each module should have the ability to be operated singly. Component based programming is a further development of the object oriented technique extension, the concept of class is still a fundamental concept of component technology, but the core concept of component technology is the interface.

This paper has taken the book information system as an example, explained the principle and the implementation steps of modular mobile phone front end based on the MVC mode, and made it standardized, which allows the developers to focus more on the development of core JS code, and reduce the time in doing repetitive work such as the preparation of interface and organization code etc. At the same time, this paper verifies the validity of MVCS by comparing with MVCS model and MVC model.

## 2. Design of the Modular Mobile Front-End

Mobile Web front-end application is the best form of mobile Internet applications, but with the escalating of the system version and difference of the devices increasing constantly, the mobile client application development face the problem of the compatibility efficiency development and maintenance as the same as the Web front-end development. The modular Web front-end architecture design using the Model-View-Controller-Store (or MVCS, for short) mode is based on the *Sencha Touch* (*ExtJS JQTouch* integration, Raphael Library) framework, and the script file classified storage is named by the directory naming conventions. It not only can make the code organization more easily and clearly, but also can reduce the amount of code compilation and make the maintenance more convenient [4] by configuring and managing of the required software function in the way of "Modular".

### 2.1. MVCS Model

MVCS mode is aimed to achieve a dynamic programming which will make it simple in the sequent process of program modification and expansion, and make it possible to reuse certain portion of the program. In addition, this model can make the program structure more intuitive by simplifying the complexity of the structure. Software system gives each basic part of its function while separating their basic parts [6].

MVCS model is different from traditional MVC Web application, in the traditional Web application, the user browser client does low levels of interaction, and the domain logic mainly focuses on the server side, therefore, the control layer, view layer and model

layer of the traditional Web application are distributed on the server side, which is also known as the server MVC mode [7].

As the front end of the mobile phone, MVCS mode learns from the classic server side MVC models, but there are some changes, which are defined as follows [8]:

Model is a collection of domain and its data type. Model knows how to retain them through data package, and can be linked to other models through relevance. Generally, it is used together with the Stores to provide data for grids or other components. [9]

(1) View folder defines all the widgets, and each widget is corresponding to a class file. View code is static code, which can be generated automatically.

(2) The Controller in the Controller folder is divided into different class files according to the management region, and control logic includes initialization, component event listening, and references, *etc.*

(3) Store is an abstract of the process of loading data through the network, and Store relies on Model and knows how to retrieve the objectification of data through the relevant Model objects for benefit of displaying View, so the View is dependent on Store object; Store is similar to the (DAO) and used as DAO (Data Access Object) mode in the server end [10].

## 2.2. Managing Dependencies with MVCS

There are two main places that application program used to define dependencies, i.e. application itself and the application internal classes. When creating an MVC application program, *Ext.application* will provide an intuitive way to set the data model, views, controllers, data storage, and configuration files that will be used in application. As shown in Figure 1. The 5 configuration items are shortcuts for loading the common application files (data model, view, controller, memory, configuration files) [11].

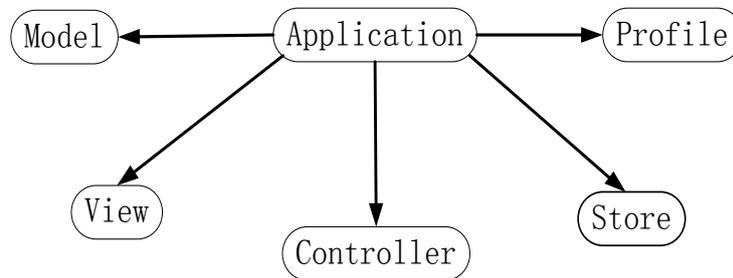


Figure 1. Managing Dependencies with MVCS

Thus, after using MVCS model, JS code is no longer written directly in HTML, JS is classified and stored in different directories and corresponded to different files according to logic and responsibility. Decoupling is needed in each module so as to reduce dependence between each other; bind the interface (View) and data storage model (Store), so that when one side changes, the other side will also automatically change.

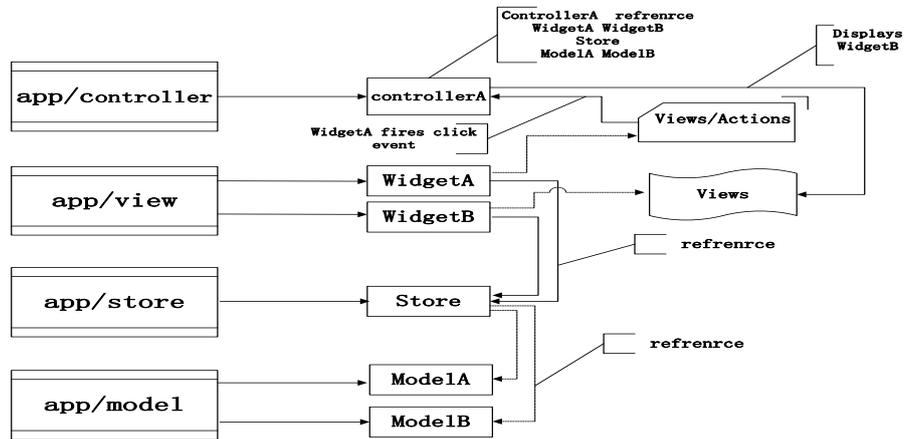
## 3 Implementation of Modular Mobile Phone Front-End Based on MVCS Mode

### 3.1. Metadata Description Models of Book Information Management

First of all, to design a Widget A, this is a tab panel used to display basic information record of books in the Viewport A. Model A is used to define the recorded data types. Store A is responsible for retrieving basic book data from the server.

So, a window will open when users click on a record from Widget A, and the data type of information displayed is from Model A. Of course, this Store will be responsible for retrieving the appropriate data from the server. Open the Widget B, the home window opens [12].

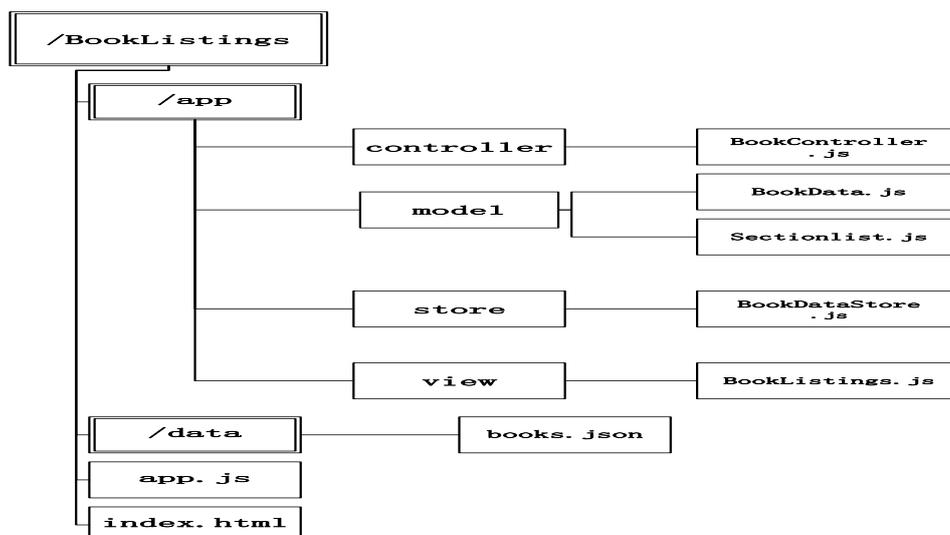
Meanwhile, Controller A captures the Click event from Widget A and loads data information required by Widget A, and finally shows the corresponding book list. The description model is shown in Figure 2.



**Figure 2. Metadata Description Model**

### 3.2. Construction of Application Program Structure

Application architecture defines a unified convention, especially a unified directory structure when application architecture establishes applications. This structure requires all the class files to be put in the directory of the app, in such directory; it can build 4 sub directories according to the namespace, such as models, views, controllers and stores. It is required to strictly abide by the naming convention of the four folders, i.e. controller, data model, data storage and view, and separate the visual interface, interaction of logic and data processing clearly, which can ensure that you can use the SDK Tools beta to optimize the application [6]. The directory structure is shown in Figure 3.



**Figure 3. The Directory Structure of Application Program**

### 3.3. Achievement of Each MVCS Module

View module needs to build a tab Panel component to set the entire interface to use Panel and set the property of full screen as true. When the application starts, it shows Ext.Container at the beginning and the location of *TabPane's TabBar* is at the bottom of the application view, while clicking, it will switch to the corresponding container interface.

Model module is divided into *BookData* and *Sectionlist* – *BookData* defines the basic data of the book information and *Sectionlist* defines the book list data. Each model definition includes *id*, *img*, *name*, *items*, *products* and other fields, we can write validation rules according to the definition such as the length of name field, the format of products field, etc.

Store module is also divided into *BookDataStore* which can provide data for the data list component. After obtaining the data, it needs to analyze the data by the data parse, and then becomes the data which conforms to certain format and are accepted by the clients.

Proxy is a way for the framework to load and save data from Store and Model. The proxies we can choose are AJAX, JSON-P, HTML5 *localStorage* and other agents. Here we use AJAX proxy to tell the program to load data from the data '*book.json*' which we called *url*.<sup>[13]</sup>

The purpose of Controller is to monitor events and take appropriate actions to respond, which are usually operated by View users. The *BookController* of the Controller module can capture the click event from table Panel and load data information that *BookListing* needs, and finally display the corresponding book lists and pictures.

Data module places resource files which are the data that we use application program to exchange with serve-side in the AJAX way. The returned data format is lightweight text data interchange format *json*.

Each *ExtJS* application must start by creating an instance of Ext.app.Application, app.js creates an Application instance and completes the initialization.

The purpose that MVCS module increases Store is that the Proxy can build bridges between Model and data. Proxy will establish a relationship between them, and then we can do some related GRUD(Create, Read, Update, Delete) operation by Proxy.

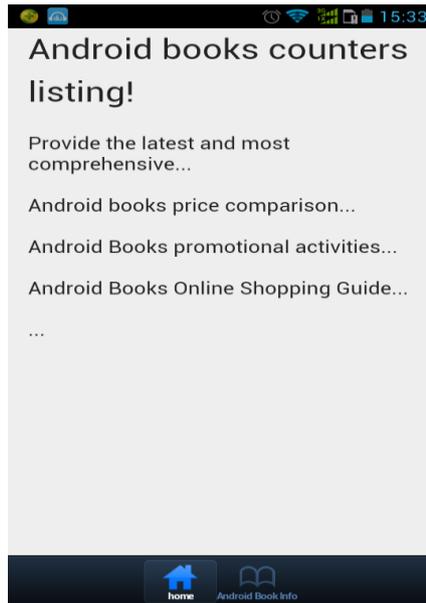
Store data set which is cache of client-side data model object Model can provide the data input for the model components(such as *GridPanel*, *ComboBox*, *DataView* and so on). Store can load data by Proxy and also can load by using *loadData* manually. The parsed data object can cache in Store data set and access via accessor functions.

Besides, this pattern also supports access of the history(applications will receive full back button functionality, and each part can be linked), deep linking(deep linking can open any screen in application program), device profiles(it can be easy to customize application user interface for mobile phones, tablet and other devices while sharing common code,) *etc.*

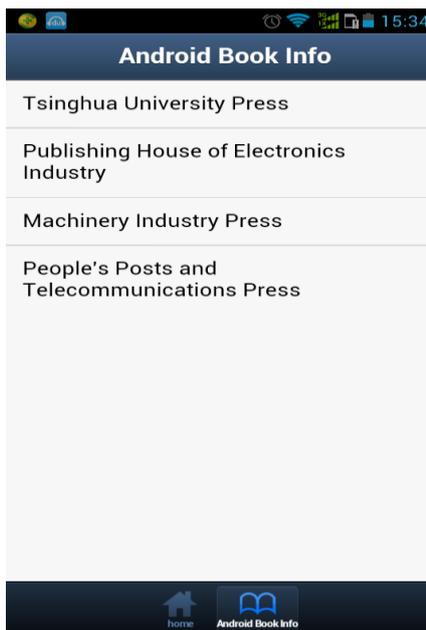
## 4. Operational Results and Analysis

This paper takes the book information management system as an example for explanation. The MVCS model implements mobile phone Web front-end display of book information, as shown in Figure 4 to Figure 7.

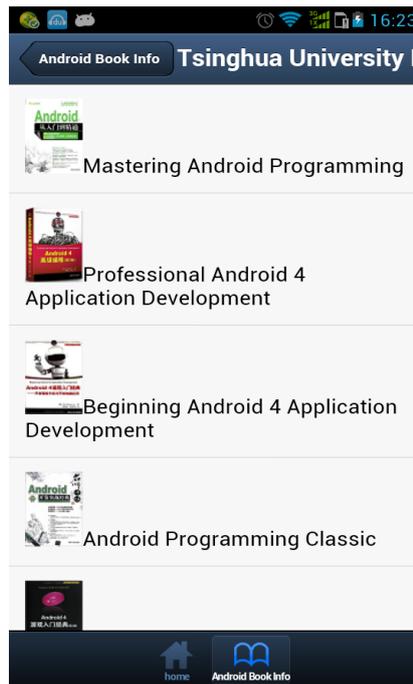
Two navigation buttons on the *TabBar* can be used to switch the main page and query page respectively. See Figure 4 and Figure 5.



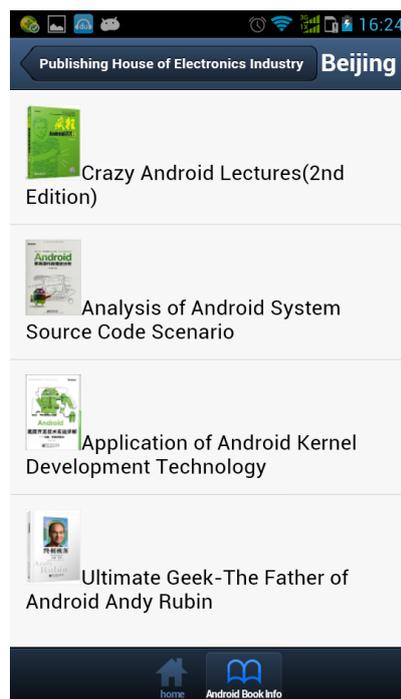
**Figure 4. The Home Page of the Application**



**Figure 5. The Directory Structure of Publishing House**



**Figure 6. Book List of the Publishing House**



**Figure 7. Book List of the Area Agency**

In the application, the dynamically generated list can list the basic book information and automatically list according to different publishers and regions. After the handled events are correctly bound to the click event of each publisher and region row, it can show the relevant book title and cover picture through the relevant click event. See Figure 6 and Figure 7.

MVCS mode displays the data in the standard way which reduces the difficulty of development, so that the planning of the whole phone front-end code becomes more standard and the presentation of charts and graphs of front end real time data becomes easier to be expanded and maintained, which is more suitable for people to write complex interface together.

The system takes the cross-platform development framework PhoneGap/Cordova to deploy the mobile phone application. It is a professional mobile application development framework and a framework for development of a comprehensive Web App that provides access to the Web in the form of the API functions of terminal equipment. Taking Web App to exploit can avoid some function of native development. Cordova is a native shell and App kernel is a complete Web App, native functions which are needed to call will be realized in the original form of raw plug-in and invoked by exposing JS interface.

## 5. Conclusions and Prospect

On the basis of MVC, it forms MVCS of component-based extension and comes up with the implementation through the practical application.

Comparing with MVC, MVCS split the previous business model to data model Model and data storage model Store, in the view of database, the Store encapsulates a client's record cache object, which provides data entry for UI components. This is the inevitable result on account of finer particles of the division of responsibilities by Proxy, Reader/Writer and bringing in the Model. The advantage is that the Store can avoid searching data in UI components as a data access layer and it can regard the browser as the presentation layer and view the server as the a data access layer. So, the server just provides the necessary data by XML or JSON format and then all performance can be achieved in the client schema by MVCS, which can avoid mixing the server-side code with client code and realize to decouple the client and server side.

In order to facilitate the deployment on different mobile devices, profile is a startup-configuration file which can configure each platform. If the application program needs to be run on different platforms and different devices, it needs to configure exclusive configuration file for each device platform, such as Android, iPhone or iPad platforms and devices.

With the development of mobile Web front-end, the characteristics of engineering become more obvious. MVCS does some researches on the standardization, maintainability and scalability of Web front-end technology, and we will take further steps to optimize and improve the MVCS pattern to enhance the reusability of modules and fault tolerance and flexibility of the entire system in the future.

Special thanks Shaanxi Provincial Department of Education funded this project (Number is 2013JK1175).

## References

- [1] H. Fang, I. Chen and B. Xu, "The Interaction Mechanism based on J SON for Android Database Application", *Information Technology Journal*, vol. 12, no. 1, (2013), pp. 224-228.
- [2] L. Ling-niu, "Sencha Touch: The Definitive Guide", CHNIA MACHINE PRESS, (2012), 9: preface.
- [3] Y. Chun-na, W. Chen-sheng, Y. Guang, G. Shi-long and L. Feng, "Significance of Web in front of MVC framework", *Estate and Science Tribune*, vol. 13, no. 1, (2014), pp. 52-53.
- [4] P. Octavian, "Enabling telecare assessment with pervasive sensing and Android OS smartphone", *Medical Measurements and Applications Proceedings (MeMeA)*, 2011 IEEE International Workshop on IEEE, (2011). pp. 288-293.
- [5] K. Sokolova, M. Lemercier and L. Garcia, "Android Passive MVC: A Novel Architecture Model for the Android Application Development", *PATTERNS 2013, the Fifth International Conferences on Pervasive Patterns and Applications*, (2013), pp. 7-12.

- [6] Y. Yu-xi, "Open Front Frame talk about", Programmer, vol. 3, (2014), pp. 26-29.
- [7] Z. Wen and L. Ren-yi, "Design and Application of RIA WebGIS Framework based Client-side MVC", Computer Applications and Software, vol. 5, (2011), pp. 75-77.
- [8] A. Kumar, "Sencha MVC Architecture", Packt Publishing Limited, (2012).
- [9] L. Groner, "Sench Architect App Development", Packt Publishing Limited, (2013).
- [10] J. Hyeyoung, R. SungYul, "Mobile Software Architecture Technique and Application for Using Business Logic and View", Convergence and Hybrid Information Technology. Springer Berlin Heidelberg, (2012). pp. 458-464.
- [11] M. Britta, K. Johannes and K. Harald, "A Mobile Player for Interactive Non-Linear Video", Proceedings of the 19th ACM international conference on Multimedia, ACM, (2011), p. 779-780.
- [12] K. Sokolova, M. Lemerrier and L. Garcia, "Towards High Quality Mobile Applications: Android Passive MVC Architecture", International Journal on Advances in Software, vol. 7.1 and 2, (2014), pp. 123-138.
- [13] W. Zhang and C. Hong, "The Research and Application of Configuration Software in the Web Mobile Front-End Design Method", Electronic Design Engineering, vol. 22, no. 17, (2014), pp. 162-164.

## Authors



### **Zhang Wen-sheng**

Born in 1967, M.S., lecturer/ Engineer. His research interests include Web front-end architecture, software architecture design, software engineering and so on.



### **Chen Hong**

Born in 1977, M.S., lecturer. Her interests research include Graphic Image algorithm, Web front end design, software engineering and so on.

