Design and Construction of Manuscript Acquisition, Editing and Distribution Platform Based on iOS System

Yansong Cui¹, Mingli Guo² and Weizheng Ren³
¹,²,³Beijing University of Posts and Telecommunications, Beijing, China
Email: cuiys@bupt.edu.cn; guomingli1992@163.com

Abstract. With the popularization and application of mobile Internet, the paper or word document office used by the traditional news publishing group becomes more and more inefficient and cumbersome. In this context, a complete set of manuscript editing and distribution platforms needs to be established to meet the modernization and transformation of the news media industry. The design of manuscript acquisition, editing and distribution system in this article is the core part of News Publishing Group manuscript acquisition, editing and distribution platform. This paper uses Objective-C language, AVFoundation framework, base64 codec technology, and SQLite3 database storage data. Through the division of user rights, the manuscript's life cycle and processing flow are elaborated. The system adopts the MVVM design mode. Performance analysis and optimization of the system through the Instruments tool. The system function is complete, performance good, reliable and stable.

1. Introduction
Social environmental pollution and waste of resources are serious in the current international environment, and Sustainable development is a lasting issue in human society. So every business and citizen should be held responsible. Premised on protecting the environment, improving productivity and economic efficiency are also unavoidable topics for every company. China's news and publishing industry is in the midst of an important strategic opportunity for development, so the news publishing industry has a lot to offer[1]. Based on this background, there is an urgent need to build a complete system of manuscript acquisition, editing and distribution. This design could improve the efficiency of the news media industry and achieve resource conservation goals. The design and implementation of manuscript acquisition, editing and distribution system is a working platform for collection, editing and distribution of manuscripts within the group. The platform clearly sorts the context of the manuscript, improves the efficiency of the group, and facilitates the work of users with different rights. It plays an important role on promoting the development of China's news media publishing industry and improving the quality of news publications.

The iOS client is a proprietary mobile operating system developed by Apple, which mainly used for iPhone, iPad and iPod touch[2]. Apple first announced the streaming iOS system at the 2007 Macworld conference. Like Mac OS based systems, iOS is based on Darwin. iOS is divided into four levels: the Core OS layer, the Core Services layer, the Media layer and the Cocoa Touch layer. The development language of the iOS system is the Objective-C language. Objective-C is an object oriented language which add some new object-oriented syntax to C language. Objective-C can be compiled on GCC and Clang operating systems.[3]

2. Summarize
This document is composed of two modules: the manuscript distribution module and the display
module. The manuscript distribution module includes shoulder title, main title, subtitle, keyword, author, content, attachment, manuscript depository, and source. The main functions of the attachment section are video, audio, and file which upload via base64. The manuscript depository part uses the multiplexed packaged model in the cell to reduce the system coupling degree. The display module includes html parsing, download files, local storage, gesture interaction, and manuscript processing.

3. Manuscript distribution module

3.1 Video compression
Video compression is realized by the multimedia carrier class AVAsset under the AVFoundation framework. It converts source video path to AVAsset multimedia carrier object and determines whether the compression format supported by the source media carrier object contains the compression format we selected, then set the compression format through the source media carrier object and transcode AVAsset content through AVAssetExportSession. At this point, create a folder to store compressed video. Finally, set the output path and output format of the compressed video so that the output path of the compressed video can be returned. The overall compression process is shown in Figure 1.

![Video compression flow chart](image)

**Figure 1.** Video compression flow chart

3.2 Base64 encoding and uploading
The internal pictures of the mobile manuscript are all taken by the mobile phone, and the size is relatively small, so compared to multipart/form-data form uploads, you can reduce HTTP requests. The use of base64 encoding is not only faster in processing small files, but also more compatible in the transmission of different languages on different platforms and not susceptible to other coding. Finally, the system chooses base64 upload mode. The method of upload uses asynchronous upload of dispatch_async() and join the serial queue serial queue, in order to observe the asynchronous execution result of the serial queue. The method of dispatch_async() will not execute on the current thread, it will open a new thread and will not block the current process.

4. Display module

4.1 Html parsing
Push the html data obtained in the background into the local richTextEditorGGD.html file, and then realize the interaction between js and OC. This article uses the stringByEvaluatingJavaScriptFromString method, so that the html language can be displayed in the
webView and this method cannot be placed in a thread other than the main thread. We should try to avoid executing the stringByEvaluatingJavaScriptFromString method for a long time because of executing the main thread for a long time will cause the UI update to be blocked.

This article selects "Chinese phone brands dial up efforts to go global" as the test sample. The backend scriptNowId of this manuscript is 2244, the smlId is 984, and the creatorId is 92. Firstly, the scriptContent is extracted from the backend, and the richTextEditorGGD.html path is obtained by [NSBoundle boundle]. Secondly, print to get richTextEditorGGD.html path to "file:///var/containers/Bundle/Application/86476969-B7B6-4843-ADC8-E6122D8EED3D/XWCB.app/

richTextEditorGGD.html", The html data obtained by the backend is shown in Figure 2.

![Figure 2](http://cbf.chinaxwcb.com/upload/scriptcontentphoto/105152987961_dkw0.jpg)

**Figure 2.** Manuscript html data map

Use the stringByEvaluatingJavaScriptFromString method to parse data into NSString data that Objective-C can recognize. Dynamically get the height of the webView to dynamically refresh the height of the webView.

Call method as "webViewDidFinishLoad", "webView:didFailLoadWithError: "and "webView:shouldStartLoadWithRequest:navigationType: " can display the content of the manuscript on your iOS device. As shown in Figure 3.

![Figure 3](http://cbf.chinaxwcb.com/upload/scriptcontentphoto/105152987961_dkw0.jpg)

**Figure 3.** Manuscript details page

4.2 Manuscript process

The manuscript life cycle and process are as follows: The draft manuscript sent by the reporter, after editing, after the first trial, second trial, three trials, finally accomplish. Among them, the first trial, second trial, third trial has the right to return to any of its previous auditors, manuscript status updated to return. The first trial, second trial, third trial has the right to use the one-click pending permission to make the manuscript skip the follow-up review, manuscript status is updated directly to be released. The first trial, the second trial, the third trial, the manuscript has the operation function of the conversion section, the manuscript after the conversion section is changed to be pending review, and the manuscript re-enters the review process. The manuscript life cycle is shown in Figure 4.
4.3 Download and storage

Download files can use the NSURLSessionDownloadTask and NSURLSessionDataTask methods. Both methods belong to the NSURLSessionTask property. Compared to NSURLSessionDataTask, NSURLSessionDownloadTask not only implements breakpoint downloads, but also solved the problem of memory soaring when downloading large files. NSURLSessionDownloadTask internally completes the operation of writing data to the sandbox while accepting.

The NSURLSessionDownloadTask flow Figure5 is as follows.

Figure 4. Manuscript life cycle flow chart

Download file s can use the NSURLSessionDownloadTask and NSURLSessionDataTask method s, Both methods belong to the NSURLSessionTask property. Compared to NSURLSessionDataTask, NSURLSessionDownloadTask not only implements breakpoint downloads, but also solved the problem of memory soaring when downloading large files. NSURLSessionDownloadTask internally completes the operation of writing data to the sandbox while accepting.
5. Testing and optimization

Performance testing and optimization includes system responsiveness, startup time, and energy consumption, corresponding to the use of the instruments test tool in Xcode to analyze the performance\(^5\). By adding Xcode project under environment Variables DYLD_PRINT_STATISTICS in scheme – Run – Arguments. The system can print output response time when the program is running.

This system uses Timer profiler under Instruments, profiler project, estimate the time between the two peaks by starting the CPU Usage. The test time of this system is shown in Figure 6. It takes about 983ms for the app to start running to render the first interface.

5.1 Compare the impact of different number of frameworks on program startup time

Under cocoapod, write empty items, 6, 9, and 13 frames respectively. The comparison of the impact of the test framework on the initializer time is shown in Figure 7. The comparison of the influence of framework on pre-main time is shown in Figure 8. In the four different cases, the system running time is shown in Figure 9.
Figure 9. Running time of the manuscript publishing platform in four

After the test, the number of frames in the cocoapod will be seriously affected by the horizontal comparison. Optimize system uptime by managing third-party libraries manually instead of dynamically loading frames through cocoapod.

5.2 Function test
Analysis of test results: The manuscript is compiled and tested by the configured account, and the user can edit the manuscript and publish it to the platform, and the user can find a manuscript on the platform and track the status of the manuscript in real time.

6. Constitution
The manuscript release and manuscript display described in this article are the core part of the manuscript acquisition, editing and distribution system. This article makes a detailed explanation of html parsing, gesture interaction, file upload, file download storage, and video compression. It build a complete collection editing and publishing system. The completion of this system will promote the development of China's news media publishing industry.

7. References
[1] Wurenqun, "The analysis of the income distribution in digital publishing industry chain," 2016 13th International Conference on Service Systems and Service Management (ICSSSM), Kunming, 2016, pp. 1-4.
[2] K. W. Tracy, "Mobile Application Development Experiences on Apple's iOS and Android OS," in IEEE Potentials, vol. 31, no. 4, pp. 30-34, July-Aug. 2012.
[3] O. C. Novac, M. Novac, C. Gordan, T. Berczes and G. Bujdosó, "Comparative study of Google Android, Apple iOS and Microsoft Windows Phone mobile operating systems," 2017 14th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, 2017, pp. 154-159.
[4] C. Xu, Y. Chen and K. Chiew, "An Approach to Image Spam Filtering Based on Base64 Encoding and N-Gram Feature Extraction," 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, Arras, 2010, pp. 171-177.
[5] B. Gao, S. Zhang and N. Yao, "A Multidimensional Pivot Table Model Based on MVVM Pattern for Rich Internet Application," 2012 International Symposium on Computer, Consumer and Control, Taichung, 2012, pp. 24-27.