

A Framework for Cross-platform Mobile Web Applications Using HTML5

Christos Bouras

*Computer Technology Institute & Press “Diophantus”
University of Patras
Building B, University campus
Rio, 26504, Greece
Email: bouras@cti.gr*

Andreas Papazois, Nikolaos Stasinou

*University of Patras
Building B, University campus
Rio, 26504, Greece
Email: papazois@ceid.upatras.gr, stasinou@ceid.upatras.gr*

Abstract—In recent years the use of smart mobile devices has become an integral part of everyday life leading to the expansion of applications development for the various mobile platforms. Each of these platforms requires separate software development process, which subsequently increases dramatically the corresponding effort. With the emergence of HTML5 these issues can be addressed efficiently since application development is allowed in a cross-platform manner. An additional important benefit is that users can access the application immediately without any need for installation. In this paper, we investigate the potentials of mobile application development with web technologies and we present a development framework that utilizes the most important state-of-art web technologies for the support of mobile devices. This framework can be used for the implementation of mobile web applications and also for the investigation and experimentation on the main features that HTML5 offers for this specific type of devices.

Keywords—mobile app; web technology; HTML5; framework;

I. INTRODUCTION

In recent years the use of smart mobile devices, such as smartphones and tablets, has become an integral part of everyday life leading the development of innovative mobile applications to a great expansion. Today there are several different platforms for mobile application development with the iOS and Android being the dominant among them. Each of these platforms requires separate software development process, based on different tools, SDK and programming languages to implement the same functionality which subsequently increases dramatically the corresponding effort [1]. This fact has a negative effect on the lifecycle of the mobile application development since several issues usually arise, including:

- multiple software implementation,
- inconsistencies between versions for different devices and platforms,
- longer quality control process.

From a user’s point of view, each native mobile application requires downloading and installation prior to its usage which adds an overhead of time and memory resources even for applications that are rarely used [2].

The emergence of HTML5 introduced a new paradigm for the mobile application development and there are several reasons why the popularity of HTML5 for mobile application development is increasing steadily [3]. HTML5 supports interfaces to the various mobile device sensors and operations and thus it can provide nearly the same functionality as the native applications. Additionally, through new features such as application cache and file system, selected data can be stored so that the application can be executed even when the mobile device is not connected to a data network. Given that the mobile application runs on a web browser, application portability is assured in a cross-platform manner saving a large amount of costs for the software development process [4]. The capabilities and the development process with HTML5 are so important that platforms for building native applications using web technologies are also becoming dominant. Such interesting examples are PhoneGap¹ and Apache Cordova² that have been also extensively documented by technical literature [5], [6] and [7]. Last but not least, with mobile web applications users are not obliged to install any software to access the application features [8]. The significance of HTML5 as a technology for mobile application development has increased in light of Adobe’s cease of developments on Flash Player for mobile browsers and its turn to invest on HTML5 instead³.

In this paper, we present the design and implementation of a framework built for experimentation on mobile web application development. This framework is an HTML5 laboratory that makes use of the most important features and provides indicative methodologies of their use. We use it to investigate the potentials of the HTML5 technology for mobile application development and to explore new fields of services and functionalities that can be engineered. Part of this study is also the investigation and proposal of best practices as well as ideas for future work on mobile applications based on HTML5. The main purpose of this framework is

¹PhoneGap. Online at: <http://phonegap.com/>

²Apache Cordova. Online at: <http://cordova.apache.org/>

³.net Magazine. Adobe stops developing Flash for mobile. Available at: <http://netm.ag/tkF0j7>

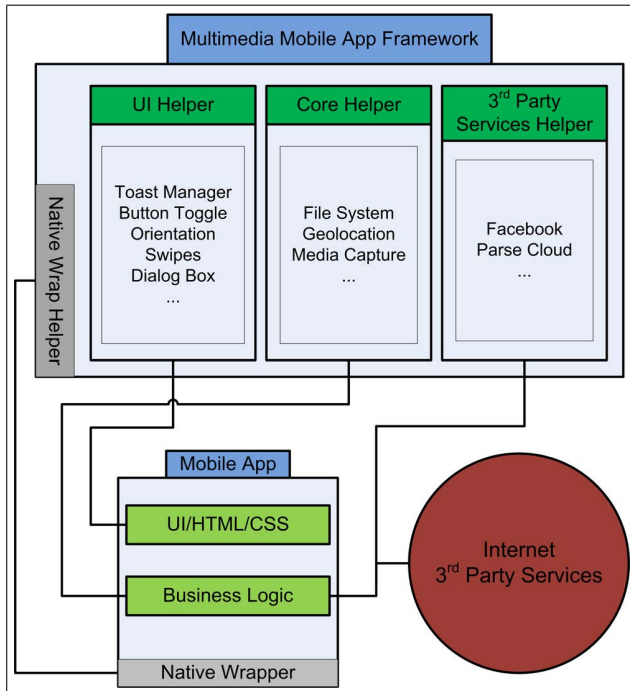


Figure 1: Framework's overall architecture.

to provide the developer with simple and easy to implement features of HTML5 in an cross-platform aspect. Most of the framework's features are essential for a modern mobile application. They make use of the information a device can provide and also features such as social network interaction and data on the cloud are presented. It should be noted that the application that demonstrates the features of the framework is publicly available online⁴ to be used by interested researchers or web engineers. Additionally, the application's features are briefly demonstrated in the demo-video created for this purpose and being available in our website⁵.

This paper is structured as follows: In Section II we present the architectural design of the framework for mobile web application development as well as selected important implementation issues. Section III provides details on the most important features offered by the framework. Finally, in Section IV we summarize this work.

II. FRAMEWORK FOR MOBILE WEB APPLICATIONS

A. Architectural Design

The design of the framework has followed a modular approach with clear interfaces between the discrete system

⁴RU6 Mobile Group: Mobile Apps. Available at: <http://ru6.cti.gr/mobile/apps.php>

⁵RU6 Mobile Group: Mobile apps demo video. Available at: http://ru6.cti.gr/mobile/uploads/demo_html5app.mp4

architectural elements. Figure 1 depicts the system's overall architecture with the short descriptions of the various components provided in the corresponding blocks. The framework's architecture has been divided into 3 main sections. The UI Helper, Core Helper and 3rd Party Services Helper. Each one of them is autonomous and can be used separately in order to prevent unneeded code from loading. The main concept is that every module refers to specific provided services and features.

More specifically, the following components are included in the framework design:

- **UI Helper:** This component refers to the provided features that are relevant to the user interface. Contains basic user interface functions that a modern application has to have in order to provide a good user experience. Consequently this component interacts with the HTML and CSS code of the application. The features provided are: Toast Manager, Orientation Manager, Swipes, Button Toggle and Dialog Box.
 - The Toast Manager provides a quick, small and custom graphic on-screen notification that can be used to provide simple application information to the user.
 - The Orientation Manager provides the appropriate interface to handle the change of the device's orientation. This can be used to adjust the interface according to the hand-held device's position.
 - Button Toggle provides a function to handle the change in the status of a button, clicked to non-clicked and vice versa.
 - Swipes provide functions to integrate and handle swipe events on a specific element. Swipe left and right are supported. Any other kind of swipe can be supported too.
 - Dialog Box provides a custom dialogue interaction interface to confirm or inform the user for various application actions.
- **Core Helper:** This component relates to the business logic of the application and provides services related to the core functionality of it. The main features it offers are File System, Geolocation and Media Capture.
 - File System provides an easy way for basic file functionality like creating file or folder, deleting and more without getting into HTML5.
 - Geolocation provides a simple function to get device's location.
 - Media Capture offers a way to get and handle media files captured directly from device.
- **3rd Party Services Helper:** This component contain functions that are based on 3rd party on-line services. This also relates to the business logic of the application especially with the part that communicates with other services on the Internet. Here, the two main services

are Facebook and Parse integration.

- Facebook feature provides basic Facebook JavaScript SDK functions such as login with Facebook account, upload/download photos and videos, sending requests, wall posts, user details and more. All this in a simple and easy way to integrate and use.
- Parse feature provides easy access and manipulation of cloud data on the Parse Cloud and also a user database based on the Facebook login method. The user has to use his/her Facebook account credentials to login.
- **Native Wrap Helper:** offers the appropriate modifications in order the services of the framework can be integrated without additional code or changes in a native application built with Cordova or PhoneGap.

B. Implementation Issues

The main technology used in our framework is the HTML5 which consists of new JavaScript API's, CSS3 attributes and new HTML elements. Below the individual technologies and services provided are further analyzed. Most of the functions provide a “success” callback with which the developer can handle the output of the function. The features make use of HTML5 File API including directories and system, Geolocation API, HTML Media Capture, CSS3 new features, Facebook and Parse JavaScript SDKs and more. These features has to be supported by the browser. Latest version of modern web browser are recommended and preferably Chrome. Additionally cross-platform implementation is supported by Cordova, preferably version 2.9.0.

- **Core Module:** The core module provides File System, Geolocation and Media Capture functions. Indicative functions that are included in this component are the following:
 - *mfwk_saveFileToFS(filename, file, success)*: this can be used in order to save a file in the initiated file system providing the file name, the data of the file and a success callback to handle the successful run of the function.
 - *mfwk_getFileFromFS(filepath, success)*: providing the file path we can get the data of the file, from the file system, which is included in the success callback function as data URL.
- **UI Module:** The UI module provides functions for features such as Toast Notifications, Button Toggle, Orientation Change Event, Swipe Events and Dialogue Box. All those features are relative to the user interface of the application and are meant to provide a better user experience. Indicative functions that are included in this component are the following:

- *mfwk_drawToast(message, delay)*: this function provides a simple and quick notification graphic that appears for a specific period of time on the screen to inform the user about something.
- *mfwk_orientChange(portrait, landscape)*: a function with callbacks for portrait and landscape orientation management.

- **ParseFB Module:** this module offers functions to provide easy access to features and services from Facebook JavaScript SDK and Parse JavaScript SDK. Its main role is to make it easier for the developer to implement 3rd party services. It can be abstractly divided into two sections the Parse one and the social networks one. The Parse section refers to the implemented Parse services. It offers Parse Cloud data and users management. The social networks one contains Facebook services but others can be implemented too, such as of Twitter and Google+.

Indicative functions that are included in this component are the following:

- *parseFb_meDetails(success)*: this function's callback contains Facebook's user details such as name, Facebook user ID and profile picture URL.
- *parseFb_getAlbumCover(auid, success, error)*: success callback contains the album cover URL.
- *parseFb_getAlbumPhotos(auid, success, error)*: given a specific album ID the success callback contains photos data.

III. FEATURES DEMONSTRATION

Below an application built with the help of framework is presented to demonstrate the framework's features and the possible future ones.

A. User Interface

The User Interface provides access and manipulation of the application's features, i.e., the user can either capture a new photo or record a new video or a new audio file. Each of the above functions calls the appropriate native application of the device's OS and then the file is transferred to the web application. The last file that was created can be shown to the last file “preview” field. Moreover the user has the ability to save the current device's position with the file. After the completion of each function a simple “toast” notification will appear for a short time. We also have the ability to hide the address bar of the browser for a more efficient usage of the available free screen space. Figure 2 depicts the application's main menu, which consists a toolbox providing access to the various supported activities.

In the gallery page the user can preview the photos or play the video or audio files that he created with the application. The application has the ability to handle the device's orientation change so that the photos and the videos can be displayed correctly and fit to the new screen width/height.

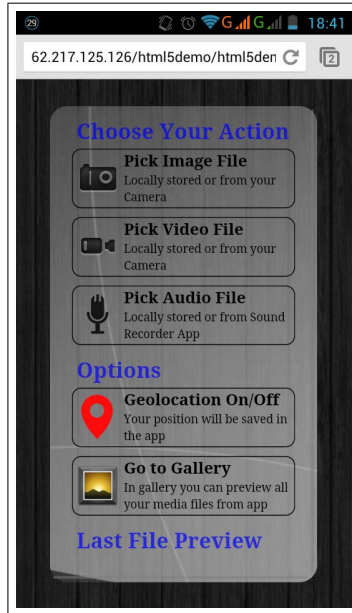


Figure 2: Application's main menu.



Figure 3: Video playback.

Additionally the user can use touch swipe gestures (left or right) to navigate to the next or previous file. For functions such as file upload, Facebook file share or file deletion the application asks for the confirmation of the user with a dialog box. Immediately after the completion of these functions, a simple notification appears. Finally we created a zoom-in/out effect for the navigation in the gallery.

Finally, it is important to mention that we adjust the content of the web page to the device's screen height and width so that we do not have very big or very small display scaling and keep the display of the content usable of every screen size.

B. Offline Use

A very important feature of a web application is the ability to run offline without the need of a data connection such as Wi-Fi, 3G, GPRS, etc. This enables the user to have immediate access to the application from his browser, everywhere he is, without the need of mobile network or wireless network access at all. We also have to consider the benefits we gain of the decrease of the server load as the application run locally on the user's device. Only the requests that cannot be served locally will be sent to the server. The application has full functionality while offline, excluding the functions that must have Internet access. The user can take photos, record video or audio even save the current location or preview or manage his files in gallery and all that while been offline. Figure 3 presents the video playback feature of the application through standard web technologies.

One more important functionality is the ability to store

files in the application. The user now has the ability to write and read big or small files, such as photos, videos, stored in the application while those files been available every time he needs them.

C. Access to Sensors & Devices

A very useful capability is the access to the device's hardware from the web application. We now have the ability to use the device's camera either to directly live stream the input we get from it or to use a native application to gain access to the camera, i.e., photo or video. The native application sends to our application the final file. The same applies to the microphone too.

We have the ability to detect the points that the user touches in the application because the touch events been transferred to it. This functionality is really useful because we can improve the user's experience with the application adding new features such as the ability to drag elements in application or to create swipe gesture events making the User Interface more usable and impressive. Of course such a feature is really important for web games too.

Furthermore, it offers access the device's sensors such as GPS, orientation, gyroscope, acceleration or compass. Therefore information such as the orientation or the motion of the device are available and enables the web engineer to adjust the displayed content of a website or web application properly depending on how the user holds the device, i.e., landscape or portrait, or use these information from the sensors to manipulate web-based games.

D. External Interfaces

The user has the ability to share his content to social networks such as Facebook or to upload his files to a file repository server. To upload his files to Facebook the user has to log in with his account credits and then he can upload a photo or video on his profile page so that his friends can see it too. The files will be saved in a specific user's album with the same name as the application's name we set and also in the feed news and profile page the upload event will be displaying the specific application that was used. Something like that can bring new users to our application. Additionally, the application can be used from the Facebook applications page directly.

Finally, the application has the functionality to upload files on a server, an ability that can be used to build a backend service, i.e., streaming video files, multimedia file hosting, etc.

IV. CONCLUSION

In this paper, we presented a mobile web framework for experimentation purposes that makes use of the most important features and provides indicative methodologies of their use. Additionally, along with the presentation of the architectural design of the implemented framework. HTML5 proved to be an excellent technology for mobile application development. Its main strengths are that it is able to offer the same functionality as native installable applications do, but in a platform-independent manner, and that it creates applications that do not require installation on the mobile device prior to its use.

REFERENCES

- [1] N. P. Huy and D. vanThanh, "Evaluation of mobile app paradigms," in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, ser. MoMM '12. New York, NY, USA: ACM, 2012, pp. 25–30. [Online]. Available: <http://doi.acm.org/10.1145/2428955.2428968>
- [2] P. E. Dickson, "Cabana: a cross-platform mobile development system," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, ser. SIGCSE '12. New York, NY, USA: ACM, 2012, pp. 529–534. [Online]. Available: <http://doi.acm.org/10.1145/2157136.2157290>
- [3] H. Heitkötter, S. Hanschke, and T. Majchrzak, "Evaluating cross-platform development approaches for mobile applications," in *Web Information Systems and Technologies*, ser. Lecture Notes in Business Information Processing, J. Cordeiro and K.-H. Krempels, Eds. Springer Berlin Heidelberg, 2013, vol. 140, pp. 120–138. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36608-6_8
- [4] J. Cho, J. Jeong, and E. Seo, "Twob: a two-tier web browser architecture optimized for mobile network," in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, ser. MoMM '12. New York, NY, USA: ACM, 2012, pp. 267–270. [Online]. Available: <http://doi.acm.org/10.1145/2428955.2429006>
- [5] R. Ghatol and Y. Patel, *Beginning PhoneGap: Mobile Web Framework for JavaScript and HTML5*. Apress, 2012.
- [6] R. Mahesh Babu, M. Kumar, R. Manoharan, M. Somasundaram, and S. Karthikeyan, "Portability of mobile applications using phonegap: A case study," in *Software Engineering and Mobile Application Modelling and Development (ICSEMA 2012), International Conference on*, 2012, pp. 1–6.
- [7] T. Green and M. Clawson, "Edge animate goes mobile," in *Foundation Adobe Edge Animate*. Apress, 2012, pp. 335–362. [Online]. Available: http://dx.doi.org/10.1007/978-1-4302-4351-9_10
- [8] M. Firtman, *Programming the Mobile Web*. O'Reilly Media, Incorporated, 2013. [Online]. Available: <http://www.google.gr/books?id=gswdarRZVUoC>