A Solution for a Common Front-End File System for Desktop and Smart Device Platforms

Arman Farahzadeh
2013

Master of Science (120 credits)
Computer Science and Engineering

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
A solution for a common front-end file system for desktop and smart device platforms

Arman Farahzadeh

Supervisor:
Josef Hallberg

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Division of Mobile and Pervasive Computing

January 2013
Abstract

Many businesses have to release cross-device/cross-platform applications to remain in the market. The challenge is that each platform demands its especial programming language making this goal very costly to achieve. Web technologies have improved a lot in recent years by newly introduced standards. Web seems mature enough to address this challenge both for existing desktop native applications and native mobile applications which run on smart devices. HTML5 as the latest web standard introduces many new features and capabilities for application developers. In this thesis I will argue why HTML5 is a right choice to select and how it is beneficial in both desktop and mobile platforms. In defending this claim, I will present and evaluate proof-of-concept prototypes.
Acknowledgment

This is a report describing the master thesis project by Arman Farahzadeh. The project was carried out at Designtech Projektsamverkan AB in Luleå, Sweden. Designtech Projektsamverkan AB provides web based collaboration solutions for other companies and organizations.

I would like to thank Josef Hallberg for being incredibly generous with his time and constantly supporting of my work. I was fortunate to have him as my supervisor. I would also like to thank Patrik Svanerudh in Designtech Projektsamverkan AB for his support through this process. I would like also to thank Andreas Andersson for his technical advises.

Finally, I am especially thankful of the constant support of my dear Parents: Ashraf and Rahman, and my dear sister, Elahe. Special thanks also to all of my great friends in Luleå University of Technology.
# Table of Contents

Chapter 1 Introduction .............................................................................................................................. 1  
1.1 Background ......................................................................................................................................... 1  
1.2 Problem Definition .............................................................................................................................. 2  
1.3 Requirements ...................................................................................................................................... 4  
1.3.1 File Drag and Drop functionality .................................................................................................. 4  
1.3.2 Mobile Application ....................................................................................................................... 4  

Chapter 2 Theoretical Information ........................................................................................................... 6  
2.1 File Drag and Drop Functionality ........................................................................................................ 6  
2.2 Browser plug-ins ................................................................................................................................. 7  
2.3 Introducing HTML5 ............................................................................................................................. 8  
2.4 HTML5 File Drag and Drop ................................................................................................................ 10  
2.5 Mobile Applications .......................................................................................................................... 10  
2.5.1 Mobile Native application versus Mobile Web application ....................................................... 11  
2.5.2 Overview of web application frameworks ................................................................................. 13  
2.5.3 A sample mobile web application: Twitter application ............................................................. 15  

Chapter 3 Experimental Design and Implementation ............................................................................. 17  
3.1 HTML5 File Drag and Drop desktop web application ....................................................................... 17  
3.1.1 Architecture ............................................................................................................................... 17  
3.1.2 Implementation ......................................................................................................................... 18  
3.2 Twitter mobile web application ........................................................................................................ 22  
3.2.1 Development Environment and Requirements ......................................................................... 22  
3.2.2 Application structure ................................................................................................................. 22  

Chapter 4 Evaluation and Discussion ..................................................................................................... 26  
4.1 HTML5 File Drag and Drop web Module ........................................................................................... 26  
4.2 Evaluation of application usability through questionnaire method ................................................. 27  
4.3 Sample mobile web application: Twitter web application ............................................................... 32  

Chapter 5 Conclusion and Future works .................................................................................................. 36  
5.1) Conclusion ........................................................................................................................................ 36  
5.2) Future Works ................................................................................................................................... 37
5.2.1) Drag and Drop module ................................................................. 37
5.2.2) Mobile Web app ................................................................. 38
References ................................................................. 39
Appendix A: Additional Graphs ................................................. 42
Appendix B: Questionnaire ................................................. 43
List of Figures and Tables

Figure 1.1 Businesses tendency to release cross-platform/cross-device applications ........................................ 2
Figure 1.2 Document Management System of ProjectCoordinator® running on Internet Explorer 8 and Windows XP operating system ..................................................................................................................... 3
Figure 2.1 PhoneGap Architecture .................................................................................................................. 14
Figure 2.2 Sencha Touch 2 application architecture. .......................................................................................... 16
Figure 3.1 HTML5 Drag and Drop web application Architecture ........................................................................... 17
Figure 3.2 HTML5 Multiple File Drag and Drop desktop web application, running on Fireforx 15.0 under Ubuntu Linux Operating System ................................................................................................................. 19
Figure 3.3 Project Schema for HTML5 File Drag and Drop web application ........................................................... 20
Figure 3.4 JavaScript Function calls ............................................................................................................... 21
Figure 3.5 screenshot of the HTML5 File Drag and Drop module running inside Safari web browser on Windows 7 .................................................................................................................................................. 21
Figure 3.6 Twitter web application lists the recent tweet posts with a scrollable indicator and allows the user to tap on and select a particular post ............................................................................................................. 23
Figure 3.7 Twitter web application after selecting a particular tweet post ........................................................... 24
Figure 3.8 Twitter web application running on a phone ..................................................................................... 25
Figure 4.1 Distribution of participants who had worked with similar files drag and drop functionality ... 28
Figure 4.2 Twitter Application architecture which follows Sencha Touch 2 recommended project structure...................................................................................................................................................... 33
Figure 4.3 Twitter application running on iPhone and iPad based on scenario 1 .................................................. 34
Figure 4.4 Twitter web application running Galexy Nexus with Android 4.01 (Google chrome and Firefox browsers) and window Phone simulator (Internet Explorer mobile browser)................................................. 35
List of Tables

Table 2.1 Mobile Native application compared with mobile web application .......................................... 11
Table 2.2 Required developer skills sets for different mobile platforms ................................................... 12
Table 4.1 Evaluating HTML5 Drag and Drop module on different web browser and operating systems.. 27
Table 4.2 The Questionnaire results ........................................................................................................... 30
Table 4.3 Participants extra comments on the application ........................................................................ 31
Table 4.4 Evaluating twitter application based on scenario 1 ................................................................. 33
List of Abbreviations

PC – Personal Computer
HTML - Hypertext Markup Language
OS – Operating System
W3C - World Wide Web Consortium
SDK – Software Development Kit
UI – User Interface
JRE – Java Runtime Environment
RIA – Rich Internet Application
AIR – Adobe Integrated Runtime
WHATWG – Web Hypertext Application Working Group
XML - Extensible Markup Language
SGML – Standard Generalized Markup Language
API - Application Programming Interface
CSS - Cascading Style Sheets
AJAX - Asynchronous JavaScript and XML
MIME - Multipurpose Internet Mail Extensions
GPL – GNU General Public License
IE – Internet Explorer web browser
Chapter 1 Introduction

This chapter presents an overview of the thesis report. It will provide the background of the topic as well as the problems and research questions that are elaborated through the whole report. In the Requirements section we talk about demands and preferences by the company, Designtech Projektsamverkan AB, in which this thesis has been carried out. The chapter ends with the organization section for the rest of the chapters.

1.1 Background

As new trends for accessing information have introduced, many businesses try to release their products compatible with at least major platforms in the market. Desktop users still have a big share of Internet users but they are going to be surpassed by users of other devices, such smart-phones and tablets. [1] As a result, both areas should be considered by companies when providing applications for their products.

Enterprise software applications that target businesses and organizations especially the ones that are used in offices solely address Desktop users. In this context, cross platform portability only target different operating systems support. If the final enterprise applications is web based the challenge still exist by how to support different web browsers as well as underlying operating system. IT companies that provide software products should also take large number of smart devices users into consideration. Even Enterprise applications should release applications for mobile and tablet devices.

In Fig 1.1, a schema of most widely used computing devices / platforms based on number of users is illustrated. As device capabilities such as screen size, input types and display features can vary dramatically from vendor to vendor, achieving cross-platform and cross-device consistency becomes more costly.
All of these platforms provide their own Software Development Kit (SDK) and demand different programming languages. Even rich companies may not be able to afford to release their products for all of the platforms in the market.

1.2 Problem Definition
Designtech AB is a company located in Luleå, Sweden which provides web based collaboration solutions for other businesses. The company’s most important product, ProjectCoordinator®, is a web based comprehensive framework that addresses the organization’s challenges such as project Management, document management, web content management and collaboration, income and expense tracking and etc.[2]

ProjectCoordinator® provides a Document Management System for users so they can download, upload documents with view and edit and saving back to the system. Fig 1.2 shows an overall view of the document Management system in ProjectCoordinator®.
Document Management system services are leveraged by offering exclusive Document Management Kit [3] which allows users to drag and drop files and folders from their computer to the document archive and open, edit locally and save automatically Microsoft Office ® documents (word®, Excel®, PowerPoint®, RTF and MS Project®) directly to the document archive after editing.

The main problem is that Document Management Kit features only work with Windows operating system and Internet Explorer web browser. It is based on Microsoft ActiveX controls technology which has some security issue itself [4]. In addition, the ProjectCoordinator® is only available for Desktop computers and not on smart devices. The company wanted to provide a new drag and drop functionality method which supports different operating systems and web browsers. The final module should be easy to use and efficient so the end-users can adapt to it fast and the company will not be bothered to invest new staff resources on customer training and support for the new features.

Furthermore, the company wanted to choose a correct strategy towards the smart devices. In the past, they had decided to export some functionality of their existing product (ProjectCoordinator®), such as File Manager or Calendar for smart device users. One important aspect for the company was to have a solution which is not costly from development prospective so existing developers can adapt to it easily.

Figure 1.2 Document Management System of ProjectCoordinator® running on Internet Explorer 8 and Windows XP operating system
This research work tries to propose a new solution for existing web solution which has Desktop users in mind as well as suggestions for future releases which involves mobile applications for smart phones and tablets. We can summarize these facts into to research questions:

1) **Is there a comprehensive solution to provide desktop multiple files Drag and Drop upload from local file system to a web browser? This solution should provide cross-platform functionality so it is applicable in different operating systems and on different web browsers.**

2) **Is there a mobile application development method to provide cross-platform/cross-device applications that can provide a similar look and feel as native mobile applications and a reduced final development cost for the companies?**

### 1.3 Requirements

The requirements by the company was both based on improvements of their existing product but on different directions which only had cross platform feature in common. We can categorize the requirements in two groups:

#### 1.3.1 File Drag and Drop functionality

The existing file manager needs extra software to be installed on the user system to be able to provide drag and drop functionality. Some studies was demanded to understand more about the technologies that current document and file manager are based on. Are there are any other technologies that can help us to avoid extra software and plugins to install on end-user operating system to provide web based Drag and Drop functionality. Is web new standards are mature enough to achieve this functionality?

It was also important how the end-users perceive the application. The solution should be able to work on any operating systems, and any web browser users prefer to work with. In the case that the browser cannot provide file drag and drop functionality it is important to allow the users to continue working with an old way of uploading files which includes uploading the files manually from local File System. Furthermore, find a solution showed here and It has to provide an acceptable performance for the end-user and is easy to work with and avoids the company to spend resources to train this feature to the customers.

A proof-of-concept demonstrator was asked to present these features. The back-end of the application had to be compatible with Apache web server to store uploaded files.

#### 1.3.2 Mobile Application

Designtech AB also wanted to choose a correct strategy towards the smart device access. They have decided to export some functionality of their existing product (ProjectCoordinator®), such as File Manager or Calendar for smart device users.
As a result, the company required an evaluation on existing mobile application development in the market. They wanted to see if web is similarly a correct strategy for mobile applications. Demanded solution for the mobile application should include these features:

1) Different mobile platforms support

2) Low development costs (low current developers training costs plus ability to integrate with desktop version)

3) Similar user interface to native mobile look and feel

4) Automatic software update by the company

They also wanted a proposal on evaluation of development frameworks environments that is easy to develop and can provide a comprehensive development tools and an environment that developers can adapt to it easily. Similar to Drag and Drop functionality, they also required a small proof-of-concept demonstrator, developed by this framework and a feasibility test on different platforms.

1.4 Thesis Organization

In Chapter two we will talk about different technologies in File Drag and Drop Functionality see if web is mature enough to provide us this feature. We also talk about mobile applications and compare the main trends in the market as well as some research about mobile web applications development frameworks. In chapter three a detailed overview of the techniques and implementation process for developed sample prototypes is presented. Chapter four contains the discussion and evaluation.

The thesis report ends with chapter five with conclusions and some future works. In the Appendix some additional information has presented
Chapter 2 Theoretical Information

This chapter provides some theoretical information about the technologies that we use throughout this report. This information was gained through literature review of different papers, books and online materials. Furthermore, it addresses the requirements discussed in chapter 1.

2.1 File Drag and Drop Functionality

File Drag and Drop is part of Human-Computer Interaction in graphical user designs. Apple and Microsoft included it as part of their User Interface (UI) in late 80s and early 90s along with their software products. Web applications can also support this feature. It enables them to act and interact similar to native applications as well as enabling users to drag a single or multiple files from the local file system to a web element on the web page. There are two important aspects to cover when it comes to file drag and drops:

- File path address on the local File System
- Reading the file content

Web browsers are not able to provide Drag and Drop functionality without having these two features.

Without the browser supporting this directly, the browser cannot expose the full path to a file or open a reader on the file. Traditional HTML 4, supports file upload in a cumbersome process since the user has to click the “Browse” button to go find a file on his/her hard drive and then click “upload” button. The user has to repeat this process for every file since multi file upload is not supported. JavaScript may seem helpful to make this process less lengthy. However, JavaScript alone can not help us in this scenario because JavaScript cannot intercept such drops. Instead, the browser will simply open the dropped file.

There are some options to solve this problem. The first option is to use additional software that enables the web browser to support this functionality. Most of the applications in this category are provided as plug-ins or add-ons that integrate with the web browser. This method makes the primary web application depend on other companies’ platforms such as Adobe Flash. Another method is to take advantage of new revisions in HTML standard itself that improve the language with support for new features such as Multimedia and Files.

We will discuss both of these methods to evaluate which one can address the demanded requirement by the company.
2.2 Browser plug-ins

Browser plug-in is a program which extends web browser by changing it somehow to be able to work with additional types of web content and also to add specific ability to it. As we discussed, traditional HTML 4 specification, only propose `<input>` tag for uploading files to the web browser. Web Browser plug-ins such as Java applets and Adobe Flash can make it possible for the browser to access to local host file system. Some of the plug-ins are available for different platforms and web browsers but as we mentioned another program rather than a web browser itself should be installed on the system which is a big drawback.

- **Java Applets**

Java applet is a special kind of Java program that a web browser enabled with Java technology, can download it from the internet and run it consecutively. An applet is typically embedded inside a web page and runs in the context of a browser [5]. Since Java is cross-platform or platform independent, Java applets can be executed by browsers on many platforms, including Microsoft Windows, Mac OS and Linux.

Java applet is able to access local host file system, so in the same way it has access to file path of a local file. By this ability they can allow file drag and drop from desktop to a browser. For example, [6] represents an open source Java applet for file drag and drop functionality.

Java applets have some serious concerns: First, it requires the java plug-in to add to the browser and Java Runtime Environment (JRE) must be installed on the system. For example some organizations only allow software's that are installed by the network administrators and this could be a big disadvantage. Above that, some applets may need a specific JRE version to be available and most importantly Java applets do not run on some mobile browsers on platforms such as Apple iOS or Android.

- **Adobe Flash**

Adobe Flash Player is software plug-in for viewing multimedia, Rich Internet Applications (RIA), and streaming video and audio on a computer web browser or on supported mobile devices. Flash Player is freely available as a plug-in for recent versions of most web browsers. It can also combine with Adobe Integrated Runtime (AIR) to have a cross-platform environment on desktop and mobile applications.

Flash is also a tool for delivering RIAs. Adobe-based RIAs can run in the browser or directly on the host Operating System (OS). When deployed in the host system, the RIA applications can access the entire local storage and file system, not just the application server, so they can provide interface functionality such as dragging and dropping files between a browser and the desktop [7].
Flash technology as well as Java applets need Adobe Flash player to be installed along side with the web browser to be able to function correctly. Adobe Flash is notorious for constant crashes. It was also challenged by Apple CEO, Steve Jobs, when he published an open letter on apple website criticizing the fact that “Flash is the number one reason Macs crash” and also “Flash was created during the PC era” and “New open standards created in the mobile era, such as HTML5, will win on mobile devices (and PCs too).” [8]

The core technology which was used in ProjectCoordinator® document manager was also plug-in based, ActiveX controls by Microsoft. We can clearly conclude that plugins method is not a comprehensive solution and restrict the application in different ways.

2.3 Introducing HTML5

Before specifically discuss about HTML5 regarding File Drag and Drop we try to briefly introduce HTML5 and new provided features.

HTML5 is currently a working draft at W3C [9]. It defines new functionalities and thus offers interesting possibilities for rich internet applications. These new features can be utilized both on Desktop and mobile platforms. The first version of Hypertext Markup Language(HTML) was created in 1990 and the current table version, HTML 4.01, was standardized by - World Wide Web Consortium(W3C) in 1999 [10]. W3C started to work on the well-formed, Extensible Markup Language (XML) based version of the language and came up with the XHTML in 2000. [11]

The Web Hypertext Application Technology Working Group (WHATWG) began to develop the new Standard Generalized Markup Language (SGML) based version of HTML in 2004. Finally, in 2009 W3C discontinued the work on XHTML 2.0 and joined the WHATWG on the effort of creating the HTML5. In addition to the markup, HTML5 describes a number of Application Programming Interfaces (API) and many of new technologies are to be used with JavaScript. As a result, part of new standard is implemented in JavaScript libraries integrated in the browser. [12]

HTML5 usually comes with Cascading Style Sheets version three (CSS3) and JavaScript as a bundle which all provide these new opportunities for web applications. HTML5 provides different new features [13] that can be divided into advantages and disadvantages categories.
Advantages

- **More Descriptive Markup**: The new elements and tags that directly describe content like footers, sidebars and articles.
- **Multimedia with less Reliance on Plug-ins**: new `<audio>` and `<video>` tags to embed audio and video files into a web page.
- **More Interactive Applications**: new APIs to completely replace old technologies like ActiveX controls and Adobe Flash to make web applications more interactive.
- **Client-Side Storage**: APIs like File API or Web SQL database or Web storage to make web applications work similar to Desktop applications.
- **Cross-Document Messaging**: introducing Cross-Domain Messaging that makes it possible for scripts hosted on different domains to pass messages back and forth.
- **Web Sockets**: support for web socket which instead of using a back end for updates, a web page can subscribe to a socket and receives push notifications to users.
- **Better Forms**: Reduces the need for CSS, JavaScript and JavaScript libraries such as jQuery, providing more interactive forms by introducing elements such as sliders, calendar date pickers and color pickers.

Disadvantages

- **In progress specification for HTML5 and CSS3**: Most APIs and specifications for HTML5 and CSS3 are just working drafts and not final specifications. As a result, changing the specification leads to changes in the browsers which finally leads to more broken web applications.
- **Browsers support challenge**: One major problem is the adoption of HTML5 and CSS3 by big web browser companies. For example Internet Explorer(IE) has a very weak support for HTML5 APIs so many HTML5 web applications don’t work in IE correctly. In addition, Apple, Google and Mozilla foundation are battling for supremacy. For example they argue about video and audio codecs. Safari play MP3 audio with `audio` element but `ogg` files does not work. Firefox in contrast supports `ogg files` instead of MP3.
- **Deprecated Tags and elements**: In addition to many useful tags and elements introduced by HTML5 standard, the specification deprecates some common elements such as: `center, font` and also more famous tags such as: `frame` and `frameset`. 
Summary

HTML5 gives web applications the ability to approximate the look, feel and usability of desktop applications and native mobile applications. It also provides a consistently meaningful user experience across different platforms and help web applications to rival native applications in terms of rich user experience.

2.4 HTML5 File Drag and Drop

HTML5 provides very powerful APIs to interact with binary data and local file system. After some researches about the HTML5 proposed methods to access local storage such as listing local directories and reading files, I came to conclusion that File API is the best one that can help to implement drag and drop functionality.

The File API [14] gives web applications the ability to conduct tasks such as reading files asynchronously, creating arbitrary Blobs, writing files to a temporary location, recursively reading a file directory, performing file drag and drop from the desktop to the browser, and uploading binary data using XMLHttpRequest2. According to specification it provides methods to list the files which are dropped to an element and also reading the content of those files.

The only challenge which remains here is the web browser support. Modern browsers are supporting more of HTML5 standards every day. Fig A.1 in appendix A shows browser support for most important browsers in the market. However, many big IT companies are offering HTML5 file drag and drop as part of their solutions. Gmail is offering Drag and Drop for email attachments [15], Box.net Inc. a leading company for cloud storage and file sharing, also offers it for their document management system instead of third party plug-ins. [16]

The HTML5 native support for file drag and drop seems a better method compared to a third-party plug-in even considering the facts that the specification is still a working draft, browsers may implement it on different ways and the fact that different parts of the specification may violate with each other.

2.5 Mobile Applications

A mobile application is an application targeting mobile devices that performs a specific task such as a game, a calendar, a music player, etc. Global Intelligence Alliance [17] defines native application as an application that is specifically designed to run on a device's operating system and machine firmware and typically needs to be adapted for different devices. For example a native application developed for the iPhone will need to run on its proprietary iOS platform or on Symbian for many Nokia devices, and so forth.
A web application, or browser application on the other hand is an application in which all or some parts of the software are downloaded from the web during each execution. It can usually be accessed from all web-capable mobile devices.

### 2.5.1 Mobile Native application versus Mobile Web application

There are different criteria when considering whether native or web applications should be chosen for development for a special purpose by an institute or a business. Table 2.1 compares native apps and web apps based on different criteria's.

<table>
<thead>
<tr>
<th>Mobile Native application</th>
<th>Mobile Web application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development for one platform at the time</td>
<td>Development for several platforms at once</td>
</tr>
<tr>
<td>Native look and feel by native UIs</td>
<td>No default look and feel, programmer must provide look and feel</td>
</tr>
<tr>
<td>Development cost is high</td>
<td>Development cost is reasonable</td>
</tr>
<tr>
<td>Access device-specific features is high</td>
<td>Access device-specific features is low</td>
</tr>
<tr>
<td>Installation through applications stores (for example App store or Google play)</td>
<td>No popular app store; access through web server</td>
</tr>
<tr>
<td>Low-level code, fast performance</td>
<td>High-level code, performance is less than native app</td>
</tr>
</tbody>
</table>

**Table 2.1 Mobile Native application compared with mobile web application**

Development for native applications demands the developers to use a specific platform SDK that leads to using a special programming language such as objective-c or java. In addition, a device capability such as screen size, input types and display capability is different from vendor to vendor and platform to platform, therefore challenge of writing cross-device content is significant. Web applications on the other hand use open web technologies such as HTML5, CSS3 and JavaScript. As long as the browser meets the requirements level any device is able to run the application.

Native look and feel, is automatically produced when working with specific platform SDK. For example iOS apps have back button on the left corner of each page and web application developers should provide this look and feel when developing for iOS.

Development costs refer to knowledge that must be acquired for developing the application. Native applications development requires developers to be skilled at different programming languages such as Objective-C, C++, Java, etc. Table 2.1 presents the wide set of knowledge needed for developing on different mobile platforms in the market. Some
businesses that try to offer their products cannot afford to hire developers to publish their products on every device available on the market.

In the case of web applications, they likely to have the skilled people available to them in existing web teams. In our case, Designtech AB can dedicate a specific budget on developer’s education to start implementing mobile web applications.

Every business or institution which tries to build for mobile phones has to answer this question whether they have enough staff resources to offer mobile native application for each platform and what they expect form this application to deliver?

<table>
<thead>
<tr>
<th>Mobile OS Type</th>
<th>Skills Required</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple iOS</td>
<td>C, Objective C</td>
<td>Xcode</td>
</tr>
<tr>
<td>Google Android</td>
<td>Java</td>
<td>Android SDK</td>
</tr>
<tr>
<td>RIM BlackBerry</td>
<td>Java</td>
<td>BB Java Eclipse Plug-in</td>
</tr>
<tr>
<td>Symbian</td>
<td>C, C++, Python, HTML/CSS/JS</td>
<td>Nokia Qt SDK</td>
</tr>
<tr>
<td>Windows Mobile</td>
<td>.NET</td>
<td>Windows Phone Dev Tools</td>
</tr>
<tr>
<td>HP Palm WebOS</td>
<td>HTML/CSS/JS</td>
<td>HP webOS SDK</td>
</tr>
<tr>
<td>MeeGo</td>
<td>C, C++, HTML/CSS/JS</td>
<td>MeeGo SDK</td>
</tr>
<tr>
<td>Samsung bada</td>
<td>C++</td>
<td>Bada SDK and IDE</td>
</tr>
</tbody>
</table>

Table 2.2 Required developer skills sets for different mobile platforms according to [18].

Native applications give the full access to device hardware but web applications can access some levels of hardware, but not the whole stack. Web applications run in a sandbox which prevents them from accessing low level API such as camera or sensors. However, the distance between the native layer and the browser is just a compile process. [18]

When developing web applications, we can offer the app through our own web server and not on application stores. As a result we can skip the approval process and added cost (Apple’s app store costs $99 even if developing free apps) [19]. Native applications have better performance than web applications. Native applications are more suited for highly interactive, CPU-intensive and visually rich applications such as mobile games.
In conclusion, we can argue that the mobile web application is likely to be a better choice for some companies strategy toward mobile applications if the company budget for development is limited and if they require applications that don’t need complicated computation. However, we should have this in mind, that mobile web applications are generally have a weaker performance rather than native applications and always need a mobile web browser to run inside.

2.5.2 Overview of web application frameworks

As we mentioned in chapter 1, The Company wanted a study on comparison of different development environments for mobile applications. As I proposed the mobile web application rather than native mobile applications, a discussion on different development frameworks seems necessary.

1. PhoneGap

PhoneGap [20] is an open-source mobile development framework that enables developers to build mobile applications by deploying web technologies. PhoneGap follows a hybrid web model. In the hybrid model, web app execute inside a browser view embedded into a native app. It provides a set of native app wrappers for all major platforms with an embedded browser which renders the UI and supports the interaction between the web app and the device. The core of the framework is a device-independent JavaScript API that provides functionalities such as storage, geo-location, etc.

PhoneGap is the best option when there is an existing web application that needs to be convert/port to a mobile environment. PhoneGap does not impose any structure so developers have the freedom to architecture what best suit their needs. This feature is a very interesting for experienced developers but could be confusing for new developers. It does not provide any development tools as well[21]. PhoneGap supports all of the platforms in the market: iOS, Android, Blackberry, Windows Mobile, Symbian and Palm. Fig 2.1 illustrates PhoneGap Architecture [35].
2. Appcelerator Titanium

Appcelerator Titanium [22] is developed by Appcelerator Inc. and was introduced in December 2008, provides a rich development tools set for mobile and tablet application developers. It also contains a comprehensive documentation and online resources which helps inexperienced developers to produce their first applications.

Titanium uses a platform-independent JavaScript API which is compiled to different target platforms as required. It takes JavaScript as source code and optimizes it and then creates a hierarchy of all Titanium APIs used for that specific application and then generates a platform-specific native code and builds any specific code that is necessary to compile. Afterwards, it compiles to native excitable using platform specific tools and then packages for native application on either real device or simulator. It does not use any browser engine to render the UI but its own API [21].
Appcelerator used to provide not only cross-platform tools for mobiles and tablets, but also for desktop. They decided to stop providing application development for desktop in January 2012[23]. Titanum supports iOS, Android, Blackberry platforms.

3. Sencha Touch

Sencha Touch[24] is a JavaScript Library and framework which enables developers to create HTML5 based web apps that run on different platforms such as iOS and Android. It can be used to create applications which run inside a browser and are very similar to native applications and are fully based on web standards such as HTML5, CSS3 and JavaScript. It has excellent documentation and online resources as well as different tools for developers. The latest release, Sencha Touch 2, was released on March 7, 2012.

In order to access some of device built-in capabilities such as connection, notification, orientation and camera, the application should be wrapped in native shell using Sencha own packager or PhoneGap packager instead. Sencha Touch supports iOS, Android, Blackberry and WebOS.

Summary

In this thesis, Sencha Touch 2 framework was picked for more investigation and also as the platform for mobile web application prototype. The main reason for picking Sencha Touch platform beside technical aspects and rich development suit was that ProjectCoordinator® uses “Ext JS” Library which is another product form Sencha company. Ext JS is a pure JavaScript application framework for building interactive web applications.

Many core classes of Sencha Touch and Ext JS are the same. Therefore it is easier for Designtech AB developers to work with this platform because they have worked with a very similar platform for the desktop web application and development cost for the company is less since they already have skilled people available to them in the existing web development team

2.5.3 A sample mobile web application: Twitter application

The company needed a sample mobile web application prototype that should be developed with the mobile and tablet developing platform that we proposed, Sencha Touch. They wanted a simple web application that has the following features:

- Native Look and Feel
- Supports both mobile and tablet views
- Cross platform and cross browser
- Multiple pages
- Interact with datasets
For this reason a sample “Twitter mobile web application” was developed using Sencha Touch 2 framework. Twitter Inc. provides a very well documented APIs to write third party mobile applications. Each Sencha Touch 2 application should follow a pattern which is defined by the platform itself and is shown in Fig 3.1.

![Sencha Touch 2 application architecture](http://docs.sencha.com/touch/2-0/#!/guide/apps_intro)

Sencha Touch 2 applications consist of Models, Views, Controllers, Stores and Profile. These concepts have been explained very briefly bellow:

- **Model**: Model is a type of object in the app. For example users or products.
- **View**: They are used to display data to the end user.
- **Controller**: A controller interacts with the app, listens for user taps and swipes and responds afterwards.
- **Store**: store is a collection of model instances. They are responsible for loading data to an application.
- **Profile**: profiles provide simple classes to handle different devices in every sencha touch 2 application allowing the developer to customize app UI for phones and tablets.

The twitter application was developed following this pattern carefully. There is some propriety IDE from Sencha Touch that generate code for the developer with drag and drop components to the design area. However, developers can also develop code with only a simple text editor.
Chapter 3 Experimental Design and Implementation

This chapter describes the prototypes, which were developed in this thesis project. It gives detailed overview of all major parts. The Implementation work of the thesis is divided in two different categories: desktop HTML5 web application and mobile HTML5 web application.

I first explain the former which provides a file drag and drop web application using HTML5 APIs and in the second part I will investigate mobile web application. I have tried to show an overall design and architecture of the application and provide some level of detailed information about the functions and APIs.

3.1 HTML5 File Drag and Drop desktop web application

In this section we discuss the architecture and then a detailed description of the modules that are used in the application.

3.1.1 Architecture

Fig 3.1 presents the overall architecture for our HTML5 Drag and Drop web application. In this architecture we divide our system in back-end and front-end sides. In the front-end we have the local file system where our files that should be uploaded reside. The HTML5 Drag and Drop module is an HTML element which runs inside our web browser. In the Back-end side, our web server is running which enables our application to upload files asynchronously to the server. The web server will save those files in data storage.

![Figure 3.1 HTML5 Drag and Drop web application Architecture](image-url)
3.1.2 Implementation
The module features are as follows:

- File Drag and Drop onto an element in the web page
- Ordinary File selection if HTML5 File API is not supported in the web browser
- Load, parse and analyze files in the client side before uploading
- Opening, parsing and analyzing files in the web browser
- A preview for images before uploading them to the server
- A progress bar when uploading the files to the server
- Uploading the dragged files to the server with Asynchronous JavaScript and XML (Ajax) call with XMLHttpRequest 2 object as well as ordinary synchronous file upload if the web browser does not support XMLHttpRequest 2
- Layout using jQuery libraries

Fig 3.2 shows the HTML5 File Drag and Drop web module running on Firefox 15.0 under Ubuntu Linux Operating System. As you see in this picture some files have been uploaded to the server and the green progress bar indicates they are uploaded successfully to server. Furthermore a new photo has been dragged from the local file system and is being dropped to the Droppable area in the webpage.
Figure 3.2 HTML5 Multiple File Drag and Drop desktop web application, running on Fireforx 15.0 under Ubuntu Linux Operating System

Figure 3.3 illustrates the functional schema of the project. As illustrated in this schema, the user can select and drag multiple files simultaneously from the local file system the web browser is running. It is important to mention that file system structure does not affect the file processing procedure in the web browser and the module works on every file system such as FAT, NTFS, EXT and etc.
When the file is dropped to the web element, File API “List” and “Read” Objects are created to read the files. Then a client side file parsing is carried out on the files. In this step, brief information about each file such as name, size and type is displayed. Every dropped file is asynchronously uploads to the server with a progress bar indicating how much of the total size of the file is uploaded to server.

Fig 3.4 shows the JavaScript function calls and API Objects which triggers when the page loads and a File is dragged and dropped to the drop area page element.
First of All, Init() function checks if the File API is available in this browser. Then a set of mouse events are registered on the window. Events handlers for events such as “dragover” and “dragleave” are handled so that event is stopped from further prorogation. The only handler which is interesting here is “drop” event handler. This event calls two other functions. The parseFile(event) function which gets the DataTransfer object from the event. DataTransfer Object returns a FileList object, which lists the files that are dropped. Then it calls FileReader Object to read the file content. At the same time uploaded files are uploading to the server via XMLHttpRequest Object asynchronously.

One important issue about this module is worth to mention. In cases that the HTML5 File API is not supported by the web browser the user is still able to use the system. In this scenario, the user have the option to work with a simple upload method by clicking the button to search for the file and then click on upload button to save them on the server. Fig 3.5 illustrates this scenario in Safari version 5.1.7 on windows 7(64-bits) operating system which 4 files have been uploaded to the server.

Figure 3.4 JavaScript Function calls

Figure 3.5 screenshot of the HTML5 File Drag and Drop module running inside Safari web browser on Windows 7
The project structure consists of an HTML file as an entry to the web module with two CSS files which format the layout and a JavaScript file which provides the core functionality of the system. There is an `upload.php` file which runs on the web server allowing the files to be saved in a directory on Apache server.

### 3.2 Twitter mobile web application

As we discussed in Chapter 2, the prototype which is developed in this chapter is based on Sencha Touch 2 framework. This framework enables us to write HTML5 mobile applications with web user interfaces which look really similar to native UIs.

#### 3.2.1 Development Environment and Requirements

This sample prototype is developed under Sencha Touch 2 Framework 2.0.1.1 SDK version. Sencha touch framework was released under two different licenses. One of them is Open Source version which is available under the GNU General Public License version 3 (GPLv3) as open source usage and also published under Free Commercial Version which is free for use under the sencha touch commercial license agreement. [25]. This prototype is developed under the commercial version of the SDK version 2.0.1.1.

Other requirements are as follows:

- Apache web server version 2.4.2 running locally
- Sencha SDK tools package version 2.0.0 Beta 3 which is packaged for windows operating system (Sencha SDK tools provide different extra services via command line utilities for generating projects, optimizing the code and also native packaging)
- A WebKit based web browser and Developer tools for testing (I tested it both on Safari 5.1 and Google Chrome 22.0.1)

#### 3.2.2 Application structure

This prototype represents a twitter application which lists the last 20 tweets or timeline of a desired twitter account. The user is able to scroll down or up the list and then select any of the tweets by tapping on it. The second page is a preview of the selected tweet with more information about that tweet. The figure 3.6 illustrates these two pages.

For data source a twitter URL query with a JSON response format is used. [26]. This is a Representational State Transfer (REST) based API that allows HTTP calls to retrieve and post content. The URL which is passed in this sample application is:

```plaintext
https://api.twitter.com/1/statuses/user_timeline.json?include_entities=true&include_rts=false&screen_name=CollTools&count=20
```
This URL represents Designtech AB twitter channel with the latest 20 tweets. The JSON response returns some values which we use to list the tweets. The response includes these tages:

- text
- created_at
- profile_img_url
- description
- name

After receiving these values there are some CSS lines which we use to format the list. The list includes the profile image of the account, then text of the tweet and then the date it was posted on the format: day, month and the year (timestamp).

Fig 3.6 and Fig 3.7 illustrates the mockups depicting the Sencha Touch Components we will deploy. As it is shown in the mockups EXT JS different classes such as "Ext.Toolbar" and "Ext.dataView" have been used to make a nice and smooth user interface for the application.

![Mockup Image]

**Figure 3.6** Twitter web application lists the recent tweet posts with a scrollable indicator and allows the user to tap on and select a particular post
When the user taps on one of the tweets, a tap event occurs and a new list opens with more detailed information about the tweet. It includes the previous data with a new white background section under that section which includes the complete tweet description and the date it was created. There is also a "back" button which appears on the toolbar of the second list. When the user tap on this button, the primary refresh list is shown and this button hides again.

As we discussed on section 2.3.3 this application has multiple pages and is connected to an external resource to load and show datasets with a native look and feel style. Twitter web app was tested on different phones and tablets and with different web browsers to investigate the cross-platform feature of the application. The results are presented in chapter 4.
Figure 3.8 Twitter web application running on a phone
Chapter 4 Evaluation and Discussion

In this chapter we discuss about the evaluation of our prototypes regarding the company's requirements. We will elaborate details of every result and try to analyze the results in a more detailed way. Both desktop application and mobile web application are analyzed here.

4.1 HTML5 File Drag and Drop web Module

The HTML5 Drag and Drop web module was tested on different web browsers on different operating systems. The server side settings were unique on the tests to investigate the tests more accurately on front-end side. The back-end side common configuration was: Apache Web Server 2.4.2 and PHP version 5.4.4. To test the module on the front-end we selected operating systems with more users and also web browsers that are more popular according to statistics [27]. The same release version of the web browser was used. (Apple Safari was an exception because it had different release versions for Mac OS X and Microsoft windows). The results are presented in Table 4.1. According to this table, Mozilla Firefox, Google Chrome and Opera support this module fully.

When writing this thesis, The Drag and Drop module was not working fully on previous versions of the Opera browser. Fortunately they added the support on their last release, version 12 [28]. Internet Explorer 9 does not support it at all. One of the strange results of Table 4.1 is that Apple safari support File API on its OS X version but it does not support it on windows version. In conclusion, it seems that the HTML5 File Drag and Drop is able to provide the functionality regardless of underlying operating system and is only dependent on the web browser running on it. It seems that Chrome, Safari, Firefox implement some part of the File API specification in many different ways that in some cases violate each other. Our module works fine on specified browsers however if we use more aspects of the specification we may not get to the same results until all of the web browsers implement the File API specification fully. In this case the businesses should wait for the web browser companies to support HTML5 APIs fully or they can use third party plug-ins inside browsers. A combination of both also seems fine in case of lack of HTML5 API support.
### Table 4.1 Evaluating HTML5 Drag and Drop module on different web browser and operating systems

<table>
<thead>
<tr>
<th>Web Browser</th>
<th>Mozilla Firefox (ver. 16.0)</th>
<th>Google Chrome (ver. 22.0)</th>
<th>Apple Safari (ver 6.0 on OS X) (ver 5.1 on windows)</th>
<th>Internet Explorer (ver. 9)</th>
<th>Opera (ver. 12.02)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 7 32-bits</td>
<td>✔️</td>
<td>✔️</td>
<td>File API is not supported</td>
<td>File API is not supported</td>
<td>✔️</td>
</tr>
<tr>
<td>Windows XP 64-bits</td>
<td>✔️</td>
<td>✔️</td>
<td>File API is not supported</td>
<td>File API is not supported</td>
<td>✔️</td>
</tr>
<tr>
<td>OS X 10.8 (Mountain Lion)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>N/A</td>
<td>✔️</td>
</tr>
<tr>
<td>Linux (Ubuntu Desktop 12.04)</td>
<td>✔️</td>
<td>✔️</td>
<td>N/A</td>
<td>N/A</td>
<td>✔️</td>
</tr>
</tbody>
</table>

### 4.2 Evaluation of application usability through questionnaire method

To evaluate the prototype application, we performed a usability test to observe how the real users interact with the application. The test results will help us to understand if the application actually solves the problem and how the users perceive the application. To address this, I performed a survey with the help of a questionnaire with a scaled structure. The questions were scaled based on “Likert scale” in which every respondent is asked to indicate degree of agreement by checking on five response categories that are between “strongly disagree” and “strongly agree” end points. [34]

Respondent target were people who work or study in institutions and are familiar with language and context of the questions. The questionnaires were distributed remotely to the 10 participants by using Google Docs online forms service. The link to the web applications was also provided. The participants were warned, not to use Internet Explorer web browser and avoid uploading files larger than 100 MB, because the application does not
allow users to upload files larger than that size. The questions were structured from more general ones to more specific and also from easy to difficult. The questionnaire is available in Appendix B.

The questionnaire starts with a question about the previous experience of the participants about file drag and drop between desktop and web browser. 80% of the respondents had worked with similar functionality before.

![Figure 4.1 Distribution of participants who had worked with similar files drag and drop functionality](image)

The rest of questions are scaled between numbers 1 to 5, 1 representing “strongly disagree” and 5 “strongly agree” consecutively.

The first question was designed to find out if the web application displayed well on the user web browser and received “4.8” average grade which is a relatively high score and proves that initial appearance is working fine in respondent different web browsers. The design of the application may need some changes according to the second question which received an average score “4.3”.

Questions number 3 thorough 5 were designed to understand how the users evaluate the application being easy to interact and how the instructions provided may improve their knowledge about using the drag and drop functionality. The question number 3 received the highest point compared to all the other questions. This suggests that the overall layout and the instructions provided were sufficient for the users to figure out how to start working with the application. Although some participants suggested it’s better to warn the users about the limitations of their browser support on drag and drop functionality.

Participants evaluation about the general performance was tested in question number 6 and has received an acceptable “4.6”grade. Although some people stated that they
experienced a lower performance when the upload file size was larger. This also could be a limitation on the server side rather than the application.

Question number 7 asks if the participants think all the functions for file drag and drop functionality was present. This question was marked with average “4.2” which is the second weakest score for a single question. From this result and users comments it can be deduced that some functions are missing. The users pointed out that it should be an option for the end-user to delete a file from the server after it has been uploaded. Also they want to have a button to ask the user after dropping the file into the draggable area if he/she is sure to send it to the server instead of sending them at once. Results from the question number 8 states that the application is quite stable. This question is graded by average grade “4.6”.

The weakest grade is for question number 9 which ask if the users are confident the files have been uploaded to the server. The green progress bar which indicates the files that are uploaded seems to be a bit misleading to the users. Because it does not clearly show the end of uploading process. So it is better to change the text on the progress bar from “upload” + “file name” to something like “filename”+ “successfully uploaded to the server”. The last question ask about the overall impression of the web application prototype which received the average score “4.6”. At the end of scale questions, participants were kindly asked to add any other comments they might have. These comments are summarized in Table 3.
## Questions

| Q.1 | The layout of the application is well displayed on my browser | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | **4.8** |
| Q.2 | The design fits with the content of the application | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 5 | **4.3** |
| Q.3 | I immediately understood how to work with the application | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | **5.0** |
| Q.4 | The instructions on the screen helped me to start working with the application easily | 4 | 5 | 5 | 4 | 5 | 3 | 5 | 5 | 5 | **4.6** |
| Q.5 | It was easy to work with the application | 5 | 5 | 5 | 4 | 4 | 5 | 5 | 5 | 5 | **4.8** |
| Q.6 | The application performs exactly what I expected from file drag and drop functionality | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 3 | 5 | **4.6** |
| Q.7 | All the functions I expected to find in a file drag and drop application were present | 3 | 5 | 3 | 5 | 5 | 5 | 4 | 4 | 3 | **4.2** |
| Q.8 | The application was working without any errors | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 5 | 3 | **4.6** |
| Q.9 | I feel confident that all of the dropped files are uploaded to server | 4 | 5 | 4 | 5 | 4 | 3 | 5 | 5 | 2 | **4.0** |
| Q.10 | My overall impression of the application prototype was positive | 5 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | **4.6** |

**Table 4.2** The Questionnaire results
<table>
<thead>
<tr>
<th>Participant</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>P 1</td>
<td>I was a little bit confused in the beginning whether the files are uploaded to the server or there are just chosen and not uploaded yet. Later I have realized when the uploading is finished the color of the file name is changed to green.</td>
</tr>
<tr>
<td>P 2</td>
<td>It was better to give the choice of deleting the files so if they are wrongfully uploaded.</td>
</tr>
<tr>
<td>P 3</td>
<td>When choosing a file, better to have a button and upload the files after clicking.</td>
</tr>
<tr>
<td>P 4</td>
<td>Dotted area could be larger than this. The instruction could be simple and shorter. Design and graphic should be attractive; it's a little bit boring. Otherwise the structure and logic is working perfect.</td>
</tr>
</tbody>
</table>
| P 5         | 1. When I tested the application with an old and deprecated browser (Firefox 3.6.24), drag and drop did never work, and the dotted area didn’t exist at all. It would be better if you could warn users with old browsers that their browsers may not work well with this application and should be updated.  
2. I retested the application with an updated browser (Firefox 16.0.2) and everything worked well.  
3. When uploading is done, a text with green background will appear i.e. "upload 'file name'" which is misleading. The term "upload" is rather imperative and thus is confusing, since the uploading process is already done. I recommend to use something like "The uploaded file: 'file name'", or "Uploading is done.", etc. |
| P 6         | It is very convenient and user-friendly way to upload files. |
| P 7         | It is easy to work with this application but using more distinguishable colors or highlights for instruction part will make it easy to start working with the application. |
| P 8         | There should be a status bar that shows the progress of uploading. |
| P 9         | I uploaded a big text file (50 MB) but it didn’t perform well and as expected! Maybe you can give the chance for the people to specify which file type they are uploading before they do that. Good luck! |
| P 10        | -Preview on drag didn’t work for me, however I am using a Linux box and drag and drop doesn’t work in Gmail and I was surprised it did work on this website.-It doesn't show be the process of upload, which would have been very helpful for large files.  
-1 uploaded 40 MB file and the preview was very slow, because the web tried to load the entire file not just a snapshot of it.  
-After upload of large files, site became slow.  
-I don't see a reason to see file type.  
-In terms of preview I expect something like "scroll preview", I don't want to see the files which I already know about, if I want to see a preview, I would scroll on them. |

**Table 4.3** Participants extra comments on the application
4.3 Sample mobile web application: Twitter web application

Twitter web application uses a resource URL from twitter API for data source. The default format of this URL is:

```
http://api.twitter.com/1/statuses/user_timeline.format
```

The format can be XML, JSON or RSS. Any of these formats are compatible with this web application. In case of JSON, it should be passed to the application with `include_rts=false` which prevents the re-tweets to appear, remaining only original tweets from the users in a list. If RSS is used, re-tweets are going to appear in the list regardless of `include_rts` flag. For this reason, JSON format was preferred for this app. Twitter application is tested on different phone/tablet devices and also some simulators. In the development phase, Google chrome desktop edition was used to debug the application.

The primary system for mobile web application development and testing was Ubuntu Desktop version 12.04. Sencha Touch 2 SDK has some problems with Ubuntu to run the apps. When running the application, this error was printed on the console indicating it is unable to access the jar file. This problem was reported by several developers and unfortunately none of the proposed solutions were answered on platform development forum [29].

The next development platform was Windows 7. Sencha Touch 2 platform does not provide any iOS simulation for itself and relies on default iOS simulator of Apple. iOS Simulator is part of Apple’s Xcode IDE for creating apps for Mac, iPhone, and iPad [30]. Xcode IDE is only installable on OS X operating system and there is no way to tweak windows or Linux platforms, which I was using for development, to run Xcode IDE. The iPhone and iPad simulation was performed by using a third party product, “iPhone and iPad Web Browser Simulator Suite”, a product by Electric Plum, LLC. A free trial version is available for 7 days. [31]

Twitter application as well as every other sencha touch application should be built according to the application structure which we talked in section 3.2.1. By following this architecture, we have the guarantee that our final application runs correctly and is easy to debug. Fig 4.1 illustrates the twitter application components and events which have created under appropriate classes.
Twitter application was tested considering two scenarios:

- Scenario 1: On the same platform and same web browser but with different views (phone and tablet)
- Scenario 2: On different Devices and different mobile web browsers

We performed two tests according to scenario 1. The test results are presented in Table 4.2. These results indicate that twitter application is cross-device and gives the same look and feel and user experience on both phone and tablet view. Fig 4.1 also illustrates the screen shots of the twitter application based on test scenario 1.

<table>
<thead>
<tr>
<th>Device</th>
<th>Common configuration</th>
<th>Same behavior?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone</td>
<td>Tablet</td>
<td></td>
</tr>
<tr>
<td>iPhone 4s</td>
<td>iPad 2</td>
<td>• iOS 5.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Safari browser</td>
</tr>
<tr>
<td>Galaxy Nexus</td>
<td>Nexus 7</td>
<td>• Android 4.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Google chrome 18.9</td>
</tr>
</tbody>
</table>

Table 4.4 Evaluating twitter application based on scenario 1
Figure 4.3 Twitter application running on iPhone and iPad based on scenario 1

In test scenario 2, different devices and mobile web browsers were tested. Results showed that if the web browser uses a WebKite layout engine, the twitter application provides almost the same user experience and look and feel on target devices.

If the web browser does not support WebKit engine, a message is displayed on the browser:

_The current browser is unsupported. Supported browsers:_

- **Google Chrome**
- **Apple Safari**
- **Mobile Safari (iOS)**
- **Android Browser**
- **BlackBerry Browser**

This message runs as part of index.html file in every application. Some famous mobile browsers such as Mozilla Firefox (Gecko engine) and Opera web browser (Presto engine) do not use WebKit engine. However, recent statistics show that WebKit engine, as of July 2012, has the most market share of any other layout engine by having over 40% of the browser market share according to StatCounter[27].
Figure 4.4 Twitter web application running Galexy Nexus with Android 4.01 (Google chrome and Firefox browsers) and window Phone simulator (Internet Explorer mobile browser)
Chapter 5  Conclusion and Future works

5.1) Conclusion

In this section, we try to answer the research questions which were listed in section 1.2.

1) *Is there a comprehensive solution to provide desktop multiple files Drag and Drop upload from local file system to a web browser? This solution should provide cross-platform functionality so it is applicable in different operating systems and on different web browsers.*

The answer to the following question is to some extend positive. This is possible with the help of web technology and newly introduced HTML standard, HTML5. In this report I presented a prototype that was working correctly on different popular operating systems such as Windows, Mac OS and Linux as well as popular web browsers such as Mozilla Firefox, Google Chrome and Safari. However, as HTML5 is still a working draft some web browsers may have not implemented all of the APIs and in addition some web browsers may have implemented it differently that can violate each other.

This solution does not need any third-party plug-ins or applications other than web browser to function correctly. According to a survey result I conducted, this prototype was easy to use and does not need prior knowledge to start working with.

2) *Is there a mobile application development method to provide cross-platform/cross-device applications that can provide a similar look and feel as native mobile applications and a reduced final development cost for the companies?*

The answer to the following question is likely to be positive. Mobile web applications with the help of HTML5 are able to address cross-platform/cross-device applications issues for smart devices such as mobiles and tablets. There are different web application frameworks in the market that claim they provide this ability for mobile application developers to write once, run everywhere. However, they may not support all operating systems and mobile web browsers such as mobile versions of Firefox and Internet Explorer. These platforms tend to provide a similar native look and Feel as native mobile applications, leveraging user experience.

However, these platforms are usually not able to provide access to all low level capabilities of mobile phones such as built-in sensors. This usually leads to mobile web applications that can not provide all kind of applications such as games. It is worth to mention that some of these platforms, such as Sencha Touch, that we used to create our prototype, provides some native packaging for Android and iOS operating systems to fill up this gap and makes it possible for mobile web applications to access some of native device features such as camera and GPS.
Regarding development cost, it is likely that many companies can cut off development budget by providing only one web application that can be run on different devices. This reduced cost are notable if they have developers with web development background in house. The performance of mobile web applications compared to native mobile applications was beyond the scope of this report.

5.2) Future Works

5.2.1) Drag and Drop module

• HTML5 Desktop Notifications

This HTML5 feature will really make the prototype to work similar to a Desktop application and improves user experience dramatically which leads to a better adoption of the application.

This feature will use Notification API, part of Web Notification standard by W3C (4). This API allows web applications to display notifications for instance on right low corner of user screen, to alert users outside the context of a web page. In our module we can use Desktop Notifications to alert users about uploading a large file to the server or uploading multi files to the server. The sample code snippet in Fig 5.1 presents a simple function that create a plain text notification instance.

```javascript
function createNotificationInstance(options) {
    if (options.notificationType == 'simple') {
        return window.webkitNotifications.createNotification('icon.png', 'Notification Title', 'Notification content...');
    }
}
```

Fig 5.1 Sample Desktop Notification code snippet

This feature is currently available for Google popular mail service, Gmail, which pushes Desktop notification for new email and chat messages. Google use an older version of the standard and the feature is only available on Google Chrome browser[32]. The code for this feature can be reused for other parts of the ProjectCoordinator© to have better overall user experience.
• **Folder Content Upload**

This prototype only supports uploading multiple files onto the server via Drag and Drop method. There are new APIs and hacks which enable users to upload Folders to the server which automatically uploads subfolders and files as well.

Google recently have added Support for HTML5 Folder Upload in its famous web-based office suite and data storage service Google docs. This feature works only in Google Chrome browser but for other browsers beside Chrome, users should install a java applet into their browser. [33] This feature can be a very potential forward step for future works for this thesis.

### 5.2.2) Mobile Web app

Current Version of Twitter application allows us to view and navigate through a predefined twitter account. The future version can provide a search feature to select a desired twitter time line. Another handy feature could be having a indicator of tweet locations on a Google Map on another tab via a tab panel which is a layout view on Sencha Touch framework.

Another improvement is to package the web application via Sencha Touch native packaging capability for iOS and Android. Native Packaging allows the web applications to act like a native application on a phone and have access to device built-in capabilities such as camera and native notifications. There could be two versions: a version as a web application with generic functionality on all phones and the other version with more specific functionalities such as checking if the phone is online or offline through the application as well as native notifications for Android and iOS phones. In this way, users are able to install them similar to native applications directly on the phone via app store or Google Play rather than accessing them through web browser.
References


[17] Native or Web Application: How best to Deliver Content and Services to your audience over the mobile phone?, Global Intelligence Alliance, April 2010.


Appendix A : Additional Graphs

Fig A.1 File API and Browser Support: http://caniuse.com
Appendix B: Questionnaire

Survey on File Drag and Drop web application

This a study for understanding the end-user views about a prototype which has been developed for a master thesis. Your data will be kept confidential and will only be used for the purpose of the master thesis.

Have you ever worked with websites that allow you to Drag and drop Files and Folders to the browser?

Yes ☐ No ☐

For the next several questions, please choose a number from 1 to 5 to indicate how much you agree with that statement.

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>The layout of the application is well displayed on my browser</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q2</td>
<td>The design fits with the content of the application</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q3</td>
<td>I immediately understood how to work with the application</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q4</td>
<td>The instructions on the screen helped me to start working with the application easily</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q5</td>
<td>It was easy to work with the application</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q6</td>
<td>The application performs exactly what I expected from file drag and drop functionality</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q7</td>
<td>All the functions I expected to find in a file drag and drop application were present</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q8</td>
<td>The application was working without any errors</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q9</td>
<td>I feel confident that all of the dropped files are uploaded to server</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Q10</td>
<td>My overall impression of the application prototype was positive</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
If you have any comments about the application or tips to improve it, please add it in the box below: