Codile

A Mobile Web Development Environment

California State University,

Monterey Bay

Capstone Proposal

Jeremy Laskar

Advisors

Andrew Coile & Bude Su

Table of Contents

Executive Summary

Codile aims to be a mobile web development environment, for the scripter who is always moving. Imagine the following scenario: a web developer is at lunch with a client. The client mentions a few bugs and tweaks which are on the developer's site. Currently, the developer would either have to jot notes down about the problems, to fix later; or, he would have to pull out a laptop at lunch and make changes to the code from there. With Codile, web development and site management fits right into the developer's pocket. With Codile, the developer can connect to their remote server, make changes to their site, preview the changes, and publish them to their live server - all from their iPhone.

The Codile project will be a web application for the Apple iPhone, and will consist of a framework for loading "plugins" to facilitate the different functions that the program will include at release. The framework will allow for the development of new plugins, to add new functionality to the application after its release. Initially, the four included plugins will allow for complete web development from the iPhone, and are as follows:

- File View - an FTP client, to view files on a remote server
- Code View - a color-coded editor, with support for standard website file formats and languages (PHP, HTML, CSS, Javascript)
- Web View - a live web preview of the site you are editing, to see how the changes you have made to code look, before they are published live
- Reference Guide - a web languages reference, formatted for the iPhone, and containing information about common tags in HTML, CSS, and PHP.

INTRODUCTION/PROBLEM DESCRIPTION

BACKGROUND ON PROJECT

The internet is arguably the most important invention of the last century – a worldwide network of knowledge, freely accessible, and relatively unregulated. With all the conflict in the world today, the internet is one of the few resources that countries around the world have come together to participate in.

In what is today known as "Web 1.0", the internet was created mostly by web developers. If a company or person wanted to publish their idea to the world, they hired a web designer, who would create a site for them to publish their work in a format friendly to the viewer. Keep in mind, at this point, broadband had not become ubiquitous yet, so the sites were made with the intention of loading fast and looking good on your standard computer resolution of 1024x768 pixels.

However, in the past few years, a new generation of internet technologies has risen; called Web 2.0 by many, this buzzword refers to an internet that anyone can easily contribute to, not just a web designer. Wikipedia, Youtube, Facebook – these are all web 2.0 sites which are run on user input, not a designer sitting at a desk, being paid to publish others information. The internet has become a place that encourages personal creativity, and makes YOU the singer, the actor, the philosopher – and behind this movement. Helping this movement are two new technologies that help expand the internets capabilities and reach: ubiquitous broadband, and mobile browsing. Ubiquitous broadband means that sites can be made with less worry about size and load times; it means that streaming audio and video, and interactive sites can be created, that only have the designers imagination as their limit. Mobile browsing brings the internet to the user, wherever they are – and in a society where web 2.0 contributors are always posting their pictures on an online gallery, or managing their online social network – it provides access to the resources of the internet wherever you are.

But where does this put the web developers of the 1.0 era? Now that anyone can create content for the web, 2.0, what is left for the creators of all this? Web developers have taken on a new role – they are the ambassadors of the internet, creating the sites and technologies that allow people to share and create whatever they please. The internet has turned from a source of information, to a new development platform, and web developers have had to change their ways to become application developers in the process.

WHAT'S THE PROBLEM?

Web developers need to be on-call most of the day and night – the internet is a resource available around the world, it does not go to sleep. And for large websites, close to 100% uptime is a must – if there is a database error on YouTube, rest assured that the problem is usually fixed quickly. However, developers are not always near a computer, and even if they are, its often hard to work from a station without the right tools to view and edit code.

THE SOLUTION

This capstone aims to create a mobile development environment for web developers, so that when one needs to make a quick change to their site code, they can do so from their mobile device, with similar functionality

to a full-fledged desktop editing program. The application will provide a coding environment, remote file manager, a preview mode, and a web coding reference. It aims not to replace a desktop editing environment, but to supplement it, allowing for quick changes to a site from any location.

TARGET AUDIENCE

The target audience for this capstone is web developers – it will provide an interface which is easy enough for aspiring designers to work with, but also provide powerful functions for more advanced designers and developers. The target audience are also users of the Apple iPhone, as that is the platform which the capstone will be developed for.

SIMILAR PROJECTS

There are other students and faculty who are working on mobile related projects this semester, and in past semesters. However, few of these projects entailed advanced PHP and JavaScript programming as mine will, instead relying on more simple coding for viewing on mobile devices. Because I will be working with a specific platform, the iPhone, I can program my application to take advantage of the unique technologies offered in this device.

For the desktop computer, there are several programs and sites which provide some or all of the functionality I would like to include in my capstone. Panic Software's Coda editor gives a good idea of the type of interface and features I would like my capstone to have, but is a full-fledged desktop software package, not a web application.

As far as web applications are concerned, I have found products which contain some of the different types of functionality I would like my project to have; however, I have not found any similar projects being developed for the iPhone or mobile devices. PHPFTP features online ftp access, and 9ne is a web-based code editor which provides the highlighting features I would like to incorporate into my project.

PROJECT METHODOLOGY

PROJECT DEVELOPMENT PLAN

To properly code this capstone, it will be separated into several "plugins", which will be completed individually, and linked in the application at the end of coding. The capstone is actually several smaller web applications, and to try and focus on all of them at once would be foolish and unproductive. The Codile application will provide a framework to load "plugins", the initial four of which will be presented with the capstone. In the future, more plugins can be made, and they will integrate with the Codile framework automatically. The four initial plugins are as follows:

- File View - an online FTP client, to allow the user access to their remote site for viewing and opening files
- Edit View - a code editor, which will color code PHP, HTML, and CSS for easy viewing and editing
- Web View - a preview frame for your edited code, showing you the changes on a live page before you save them to your site
- Reference View - a reference guide for common tags in HTML, PHP, and CSS

I will begin the capstone by coding the overlying Codile framework - this will link all of the plugins, and declare the variables which will be used for the different plugins to relay data to each other. It will also be expandable, meaning that I (or another developer) can code a custom plugin (a CSS editor would be cool) and integrate it into the Codile framework.

After the framework is completed, the first plugin to code will be the FTP client - the edit and web plugins rely on the FTP client to provide information to them, so it must be done initially. There are a few web-based FTP clients available, many of which are open source, but none of which are formatted properly for viewing on the iPhone. An FTP client will need to be coded to fit in with the iPhone UI's look and feel.

After the FTP client is completed, the next step will be to code the editor. This will probably be the most complex part of the capstone - the limitations on the mobile browser of the iPhone, along with the complexity of a dynamically generated, color-coded layout, will make this take up most of the time spent on the capstone.

As well as being able to edit code, the editor plugin should be able to create temporary preview files, which can then be relayed to the Web View plugin, so that the developer can see what their changes look like on a live server. That said, the Web View plugin will probably be relatively easy to code.

The final plugin will be the reference guide, and while it will not have to have to relay data between the other plugins, it will be time consuming because of the amount of information which will go into it. The reference guide will provide most, if not all, of common web development tags and short summaries of their functions and parameters.

REQUIRED SKILLS

This capstone will draw knowledge and skills from many different courses and content taught under the TMAC Web Development track; specifically, they will rely heavily on knowledge of Web Design and Web Scripting,

as well as incorporate content from Interface Design and Digital Art. The design of the application will need to reflect the "look and feel" of the UI used in the iPhone, while providing a clear, easy to understand interface with a minimal amount of screen real estate used.

PROJECT RESOURCES

The resources used for this project are minimal - a computer, and an iPhone to develop on are all that are needed to code this. Access to both of these devices are already available, so there should not be any problem acquiring access to these resources for the capstone.

RISK ASSESSMENT

The biggest risk associated with this project, is that I will not be able to complete the coding on time, or that certain features which I would like to include in the application will not be codeable, because of software restrictions. An example of this is color-coding the editor; while I am sure that I can color-code text on a page, editboxes cannot be formatted without special coding; I will need to find a way to incorporate color-coded tags, while still providing an intuitive interface for the user. As long as I keep on track with the milestones I have setup for myself, time spent coding the capstone should not be an issue, and I should have the entire capstone completed on time.

PROJECT DELIVERABLES

INFORMATIONAL WEBSITE

The first deliverable will be the website which will advertize and provide the Codile application - http://www.codile.com. This site will include links to a version of Codile which a user can run directly from the website. It will also provide the source code, released under an open source license to users. With the source will be an installer file which a developer can use to install Codile on their own web server.

WEB SERVER INSTALLER

The web server installer will be a set of files included with the source code of Codile. A user can upload Codile to a server of their choice, and configure their own personalized copy of Codile, specifically for their domain and site. Alternatively, a demo version of Codile will be available from the website, which will allow the user to connect to a server of their choice. The package will be programmed in PHP, and will provide the setup for a MySQL database, to manage records for the user, and provide AJAX functionality.

MOBILE DEVICE APPLICATION

The mobile application is the main part of the Codile capstone package. It will provide the framework for the mobile device, as well as all of the plugins provided with the initial Codile package. Codile will be programmed in PHP, and will use extensive CSS, for layout and formatting. The application will be designed to be viewed in both portrait (360x480) and landscape (480x360) resolutions on the device, and to maintain the look and feel of the iPhone UI in it's design. In addition to CSS, the <canvas> tag will be used extensively for layout, which provides a means of inserting design elements without images. The Codile application will also feature AJAX technologies, for smooth, live updates to user code, without reloading or changing pages. The goal is to make a product which works as smoothly as the rest of the iPhone interface, and to make the link to the internet transparent.

TESTING AND EVALUATION

FUNCTIONALITY TESTING

This project is being developed with the intent that it will be used on an iPhone or iPod Touch; as such the iPhone will be used for testing the program, to make sure that the coding and layout appears correctly on the device.

While the coding will be done on a desktop computer, attention must be paid to the standards and languages used in the programming; only certain types of code work on the iPhone, and there are exceptions to many languages as to how complex their coding can become. In addition, the layout of the program should adapt depending on whether the user is viewing the application at a landscape or portrait orientation.

The initial functionality testing will be done by the clients who this capstone is being built for, and I. Cuban Council, a San Franciscan web development firm, has asked that I build this mobile development interface, in order to make quick changes to their site code from anywhere. As such, they will be using the development versions of the software as it is being built, and giving me professional advice on the functionality of the application. I will make sure to look for input on the four essential tasks Codile aims to accomplish: file viewing, color-coded editing, web previews, and a streamlined web reference.

USABILITY TESTING

Usability is important, especially when designing an interface to be used on such a small device. The user has to be able to have all the options within a few screen touches, but they must be laid out logically and clearly. To this extent, I will have a hyperlink built into the application for user input, so that those using initial builds of Codile will be able to provide their input on the application design, directly from the application.

As far as refining and perfecting the code and functionality of the application, I will be using a multi-tiered testing system, to get as broad of an analysis of the program as possible. I will have a test routine document, which will ensure that a tester does a standard set of operations with the application, to test its' usability. These tasks will include the four essential tasks mentioned above (file management, editing, previewing, and referencing code). Initially, I will be asking colleagues to go through the testing operations, and I will observe them using the program to see how they use or ignore parts of the UI. Following the initial testing, I will release beta test code to my clients, and to a Google Code site. The site will provide builds of the program, notes on my progress, and allow for input on the design and features of the program. Using input from these three sources – peer testers, my client's input, and bug testing through Google Code, I will be able to further refine the usability of Codile to be useable to as wide of an audience as possible.

CAPSTONE EVALUATION PLAN

After completion of the capstone, I will continue to publish the Codile source code and revision information on it's website, as well as provide for a community to develop plugins for the Codile framework.

The success of the application will be directly measurable through the Google Code site. Using this resource, along with other Google services, such as Google Analytics, I will be able to see how much publicity the application is getting, who is downloading the code, and what people's overall thoughts and opinions are about

Codile.

Approximately 2-3 months after publishing the initial release of Codile, I will be able to see if the capstone turned out to be an effective product. I will ask Cuban Council, my original clients, what their reflections are about using the program, and if they continue to use it after it's initial release. I will also study the Google Code site to see how many times the source code is getting downloaded; if there are more than a dozen people who have used the product, I would consider it successful. The ultimate proof of this capstone's success will be if the experience shown in my application development ultimately helps me forward my career as a web developer.

PLANNED BUDGET

Product	Qty.	Cost	Units	Actual Cost	Theoretical
Personnel (source: http://www.salary.com)					
Web/DB Developer	50	$50	/hr.		$2500
Interface Designer	30	$50	/hr.		$1500
PR Representative	20	$40	/hr.		$800
Subtotal:					$4800
Equipment/Hardware (Source: http://www.apple.com/store)					
iPhone	1	$400	/unit		$400
MacBook	1	$1500	/unit		$1500
Subtotal:					$1900
Software (Sources: http://www.panic.com/coda; http://www.newegg.com)					
Mac OS 10.5	1	$100	/copy		$100
Panic Coda	1	$70	/copy		$70
Microsoft Office	1	$500	/copy		$500
Adobe Photoshop CS3	1	$300	/copy		$300
MySQL/Apache/PHP	1	$0	/lic.		$0
Subtotal:					$970
Services/Other (Sources: http://www.godaddy.com, http://www.amazon.com)					
Domain Name	1	$10	/yr.	$10	$10
Web Hosting	12	$12	/mo.	$144	$144
Google Code account		$0	/user		$0
Reference Guide (PHP/MySQL)	1	$40	/unit		$40
Apple Human Interface Guidelines	1	$0	/unit		$0
Subtotal:				$154	$194
Total Costs:				$154	$7864

CAPSTONE MILESTONES				
#	Task	Start	Finish	Months
Conceptual Phase				
1	Write Capstone Proposal	Aug. 07	Dec. 07	4mo.
2	Concept Meetings w/ Capstone Advisor	Nov. 07	Dec. 07	2mo.
3	Brainstorming Meetings w/ Client	Nov. 07	Jan. 08	3mo.
4	Planned Feature List	Nov. 07	Nov. 07	2wks.
5	Conceptual Sketches	Dec. 07	Dec. 07	2wks.
6	Open Source Research	Dec. 07	Jan. 07	2mo.
Coding Phase				
7	Base Framework Completed	Jan. 07	Jan. 07	2wks.
8	Codile Plugin Framework	Jan. 07	Jan. 07	1wk.
9	FTP plugin coded	Jan. 07	Feb. 07	2wks.
10	Code Editor plugin coded	Feb. 07	Feb. 07	3wks.
11	Web Preview plugin coded	Feb. 07	Mar. 07	1wk.
12	Web reference content building	Mar. 07	Mar. 07	1wk.
13	Web reference content coded	Mar. 07	Mar. 07	1wk.
Design Phase				
14	Base Interface Coded	Apr. 07	Apr. 07	2wks.
15	Advanced Design Elements Built	Apr. 07	Apr. 07	1wk.
16	Advanced scripting, dynamic reload complete	Apr. 07	May 07	1wk
Production Phase				
17	Informational/Promotional Website completed	Feb. 07	Apr. 07	3mo.
18	Client input integrated into project	Apr.07	May 07	3wks.
19	Production meetings with capstone advisor	Feb. 07	May 07	4mo.
20	Functionality Testing	Feb. 07	Apr. 07	3mo.
21	Usability Testing	Mar. 07	Apr. 07	2mo.
21	**Obtain capstone advisor approval**	**May 07**	**May 07**	**1 day**
22	**Obtain capstone professor approval**	**May 07**	**May 07**	**1 day**
23	Integrate post-release bug testing into final project	May 07	May 07	2wks.