

Cite as: Huynh, M., & Ghimire, P. (2015). Learning by doing: How to develop a cross-platform web app. *Journal of Information Technology Education: Innovations in Practice*, 14, 145-169. Retrieved from <http://www.jite.org/documents/Vol14/JITEv14IIPp145-169Huynh1842.pdf>

Learning by Doing: How to Develop a Cross-Platform Web App

Minh Huynh and Prashant Ghimire
Marketing and Supply Chain Management,
Southeastern Louisiana University, Hammond, LA, USA

Minh.Huynh@selu.edu; Prashant.Ghimire@selu.edu

Abstract

As mobile devices become prevalent, there is always a need for apps. How hard is it to develop an app, especially a cross-platform app? The paper shares an experience in a project that involved the development of a student services web app that can be run on cross-platform mobile devices. The paper first describes the background of the project, the clients, and the proposed solution. Then, it focuses on the step-by-step development processes and provides the illustration of written codes and techniques used. The goal is for readers to gain an understanding on how to develop a mobile-friendly web app. The paper concludes with teaching implications and offers thoughts for further development.

Keywords: Web-app, mobile-friendly web app, open-source software, web database, cross platform app, HTML5, PHP, CSS, JavaScript.

Introduction

As users increasingly move to mobile and tablet devices instead of desktop computers, there is a growing demand for good apps to run on mobile devices. According to recent study by Pew Research Center, Millennials (ages 18-34) are by far the most technologically advanced user group, with a 95% cell phone ownership. Educators are inspired not only to use mobile devices to communicate learning content in new formats to these Millennials (Caverly, Ward, & Cavarly, 2009; Zickuhr, 2012) but to come up with apps to meet the needs of these tech-savvy students. Being able to deliver a great user app, especially educational apps, is always a challenge. There is no easy answer to this challenge. For many involved in teaching or developing mobile apps, it is difficult to teach something when the only constant is change and the pace of change is fast as pointed out by Lutes (2004, 2012), a veteran expert in teaching cross-platform mobile app software development. Another challenge is that there is no established platform for mobile devices. Each mobile device platform provides its own development tools. Unfortunately, these tools are

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

quite different from one to the other. For instance, tools of Apple iOS cannot be used on Google Android and vice versa. Hence, an app that runs on multiple mobile devices has to be written for each of the different platforms. For educators who often have little time to become skilled on any one platform, trying to master multiple tools is almost impractical. Furthermore, the learning

Editor: Lynn Jeffrey

Submitted: March 24, 2015; Revised: March 14, May 17, 2015; Accepted May 19, 2015

curve to master just one tool is already quite demanding, let alone learning multiple tools. Therefore, learning and mastering a development tool is another tough barrier to climb.

In this paper, we want to share our experience in the development of a cross-platform app. It is a simple app, but the approach in developing this app may be useful for teaching purpose. Moreover, the knowledge presented here may also be used for developing more sophisticated apps based on the common open-source technologies and built upon the codes already written. Therefore, the primary purpose of this paper is to describe our development processes for the simple app, to illustrate our codes, and to share our experiences in this project as well as our vision for further work. We hope that what we share might help to lower the barriers in the mobile app development process as well as to reduce the learning curve for those interested in teaching mobile app development. The focus of this paper centers on our project—that is to design and develop a student services app. Its main function is to display information on student services at our campus. Specifically, our paper is structured as follows. First, it describes the background of the project, the clients, and the proposed solution. Then, it describes the step-by-step development processes and shows the illustration of written codes and techniques used. Next is the presentation of how the student services web app works. Finally, the paper concludes with practical implications and future development work.

Background of the Project

Identifying the Need for an App

Initially, our interest was to explore and learn different ways to develop a mobile app for students to use. We wanted an app to be useful and relevant to the students. We came up with several ideas for potential apps, but in the end we decided to focus on student services. The main purpose of this app is to provide students with information on services available on our campus. Such an app, we conjecture, would be useful not only to current students but also to new students as well as visitors to our campus.

Based on this need, we set out to collect information on all the services relevant and available to students. We identified over 20 different services that are typical at most university campus. They included financial, academic, as well as technical supports offered to students. With this information, we intended to design and create an app that would allow students to search for a specific student service and retrieve basic information on that particular service from their mobile devices. More importantly, we wanted the app to be able to run on cross-platforms including iPhone, Android, and Window devices.

Understanding the Three Major Options for Mobile Apps Development

Based on a discussion with industry professionals, Agrawal and Gill (2013) suggested that there are basically three major options available for the development of mobile apps. The first option is to use platform-specific developer tools to develop platform-specific applications. The second option is to create web applications using HTML/JQuery and CSS. The last option is to use RAD tools such as Sencha and RhoMobile to create platform-neutral web-based applications (Agrawal and Gill, 2013).

With the first option, using platform-specific developer tools, the result is the delivery of native apps. Native apps are normally apps that are written using tools and an application program interface (API) provided by the hardware/device providers such as Google and Apple and are specifically tied to a particular platform such as Google Android, Apple iOS, Microsoft Windows (Lutes, 2012). Native apps are typically coded in compiled programming languages such as Java,

Objective C, or C#. Native apps are often robust because they have access to device specific features (Lutes, 2012). All native apps are obtained via app stores such as Apple App store, Google Play store. A native app is installed directly onto a device itself, is available locally and hence can run without a connection to the network.

With the second option using HTML/JQuery and CSS, the result is the delivery of mobile web app. Mobile web apps function essentially like websites designed with a user interface that is optimized for mobile devices. It is not device-dependent because it is typically coded using standards-based web browser technologies such as HTML, CSS, JavaScript, PHP (Lutes, 2012) rather than compiled program languages. Although it works like a native app, a web app does not need to be downloaded or installed on a device. The web browser on the mobile device invokes it through a designated URL. Hence internet connection is required to run it.

The last option using platform-neutral web-based applications creates hybrid apps. This kind of app draws upon the characteristics of both a native app and a mobile web app. Like a mobile web app, it is written using HTML, CSS, JavaScript, or PHP, but it is compiled into a native app for distribution via the vendor app stores. From a user's point of view, there is little difference between a native app and a hybrid app. As long as the device has a web browser, a hybrid app can run just like a native app. From a developer's point of view, a hybrid app is truly cross-platform. The same code of HTML can run on different platforms. This is possible because all the major smartphone OSs support mobile browsers that can run HTML5, CSS, and JavaScript code, and use local device storage. Open-source libraries and tools provide a shell project with the main user interface (UI) being a single web browser widget. It is this widget that processes the HTML5, CSS, and JavaScript code and packages them into a native app (Lutes, 2012).

Choosing between Native Apps vs. Web Apps/Hybrid Apps

Over the years, there has been a great deal of discussion on native apps vs. web apps/hybrid apps (Charland & Leroux, 2011; Heitkötter, Hanschke, & Majchrzak, 2013; Hinchly, 2014). For instance, in his keynote blog, Hinchly (2014) described how the Financial Times weighed in on Web vs. native app debate. The blog pointed out the advantages and disadvantages in both ways of developing an app. With the rapid pace of change, it is hard to know which of the two would prevail. In the near future, the lines between the two may become even more blurred as there is no longer a distinction between them. Since the focus of this paper is not to explore which of the development options is a better choice, we are more interested in sharing an approach that was suitable to our project. Therefore in this paper, we discuss in more details how we actually used the chosen approach to design, develop, and deliver a functional app with our available resources and knowledge.

At the initial stage of the project, we did consider the native app approach because of its many inherent advantages. From a user's perspective, everyone is familiar with this type of app. It is also easy to distribute a native app to users via established app stores. From a developer's perspective, its performance tends to be better because the application is installed once and the installed app is run every time the user wants to use the app. Furthermore, there are lower-level APIs that allow native code to access device storage, sensors, and data (Charland & Leroux, 2011). However, to develop a native app for different platforms is quite challenging for a number of reasons. One, it would require knowledge of different developing environments. For instance, iOS requires skills in Object C or Swift; Google Android in Java; Window mobile in .NET that uses C#. Two, it would incur higher costs because each platform requires different development kits and tools. Three, the distribution is quite restrictive because a native app would have to get approved by the respective third party and is distributed only through them. For iOS, it goes through Apple store, and for Android, it is Google Play store. Some of the app stores like Apple's App Store require the developers to pay a charge for having the application on their app store.

Given these challenges, we did not choose the native app approach. With the two choices left, the web app option vs. the hybrid option, we decided to choose the web app option. We will pursue the hybrid approach in a future development. We like the use of RAD tools such as PhoneGap to create platform-neutral web-based applications. However, it would take some time to learn. Therefore, at this point in time, the web app approach is considered to be more feasible and suitable for two reasons.

1. We want to develop one app that can run on cross-platform mobile devices. Web app allows us to write codes once and run them from any mobile device including a personal desktop or laptop computer as long as it has a web browser. This approach also fits well with our limited resources, time constraints, and tools on hand because we do not have the necessary knowledge or the funds to undertake the development of a native app and hybrid app.
2. We have knowledge and access to the tools needed for the development of a web app. These tools include PHP, JQuery, JavaScript, as well as a server running Ubuntu OS and MySQL as our database. The best part about these tools is that they are open-source. Hence, they can be obtained without cost. Furthermore, they are widely available and supported by a large community of users.

Step-by-Step Development Processes

Step 1: Conceptual Design, Basic Requirements, and Development Tool Selection

One of the first things that we did was to conceptualize our web app. We determined the basic requirements and then selected tools needed for the development. After identifying the need and understanding different development options, we had a good idea about what we wanted for our student services app and how we approached the task of building the app. Essentially, we wanted to develop a web app for students in a university environment. The content for this app consists of the basic information of available services on our campus. The app allows students to search for a particular service by name and type from cross-platform mobile devices. The result displays the following information: A Click for Map link, name of the service, location, phone, email, web, and a brief description. Figure 1 is the use case diagram for our student services web app. In the diagram, “user” refers to students who use the web app and “administrator” refers to us whose main task is to design, develop, and maintain the web app. “User” can run the app to search for the information with a search term. If the search term matches the term in the database, the app would display the information. Otherwise, “administrator” has to update the information or include the missing information.

Although the use case diagram shows who the users are and what actions the users might take, it lacks the sequence of activities. In Figure 2, we created an activity diagram to capture the sequence of activities as well as the logic involved. This simple diagram helped guide the development of the key functions in the student services web app.

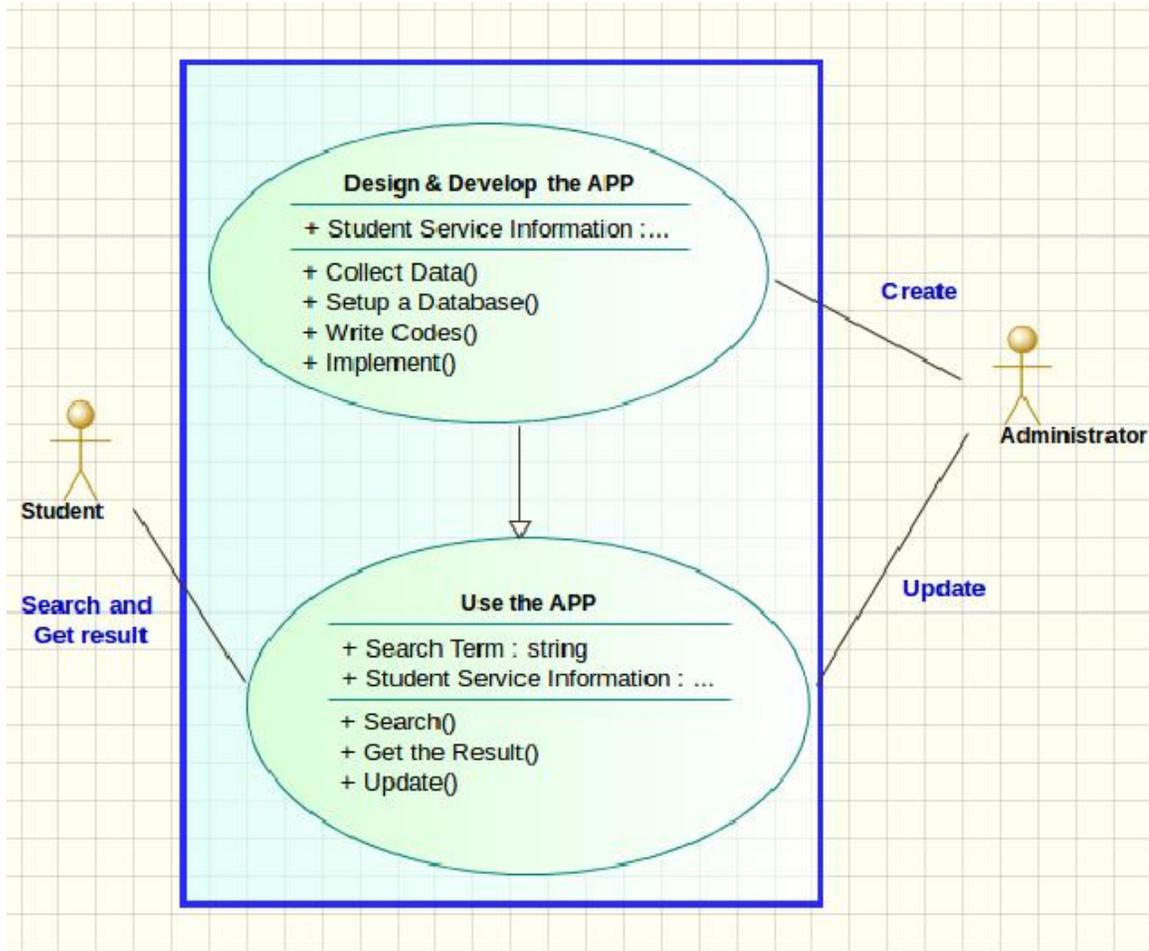


Figure 1: The use case diagram of the student services web app

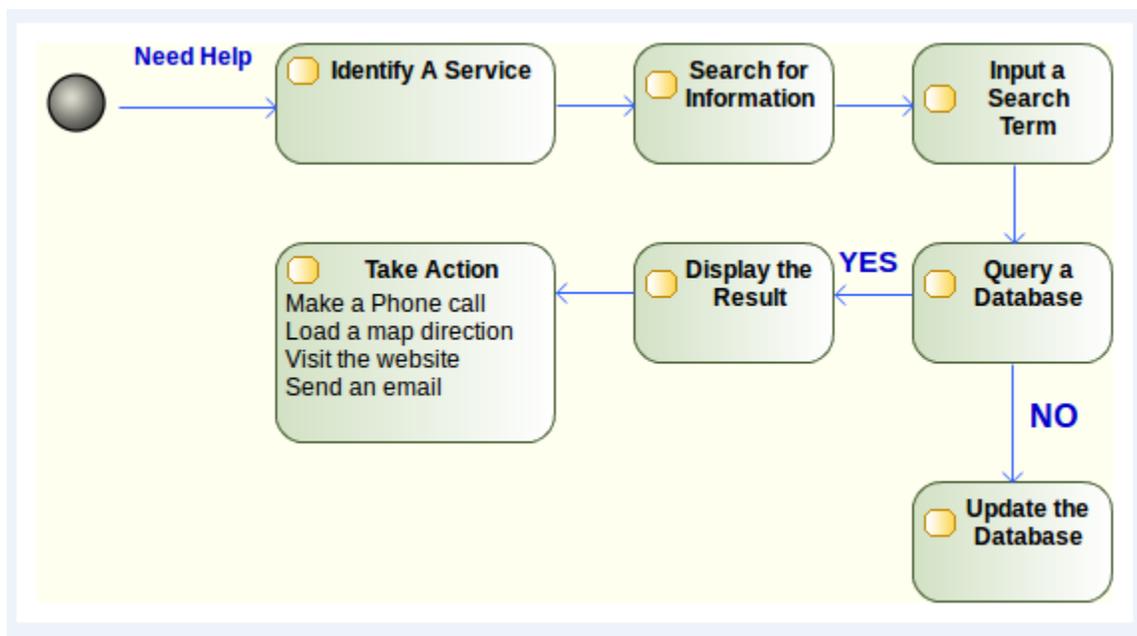


Figure 2: Activity diagram of the web app

To develop our student services web app, we made use of several open-source software and tools. Among them are the server running Ubuntu v.14.04, MySQL, Apache, and PHP. We also added JQuery and JQuery Mobile to make our web interface mobile-friendly. Table 1 provides a brief description of these software and tools.

Table 1: Description of open-source software/tools used	
SOFTWARE/TOOLS	DESCRIPTION
Ubuntu v.14.04	Ubuntu is an open-source operating system. It is a Linux-based OS and can be obtained free from the URL below. We use the server version 14.04. For more information, visit http://www.ubuntu.com/
MySQL	MySQL is an open-source database engine. We use it to store student services information and make the information accessible via PHP and SQL code.
Apache	Apache is the well-known open-source web server. We use it to make our website available to the world.
PHP	PHP is a server-side scripting language designed for web development. We use it to write codes to access database, retrieve information, and display results.
JQuery	jQuery is a fast, small, and feature-rich JavaScript library. We use it in conjunction with JQuery Mobile to create some of the mobile-friendly features on our app.
JQuery Mobile	jQuery Mobile is a touch-optimized web framework. Basically, it provides JavaScript and CSS library. We use it to create some of the mobile-friendly features on our app.

Step 2: Setting up an Online Database for the Student Services Web App

After we identified and collected data on student services available on our campus, we input the information and saved it in an Excel spreadsheet. Next, we began to transfer the information from the Excel spreadsheet to an online MySQL database. From our experience, the best format for importing Excel data onto MySQL is to save it in OpenDocument spreadsheet format. The reason is because MySQL can read and import OpenDocument spreadsheet more efficiently and accurately especially when the data contains foreign fonts or special characters. This was done by using “save as type” in Excel to save the data in OpenDocument spreadsheet format as shown in Figure 3.

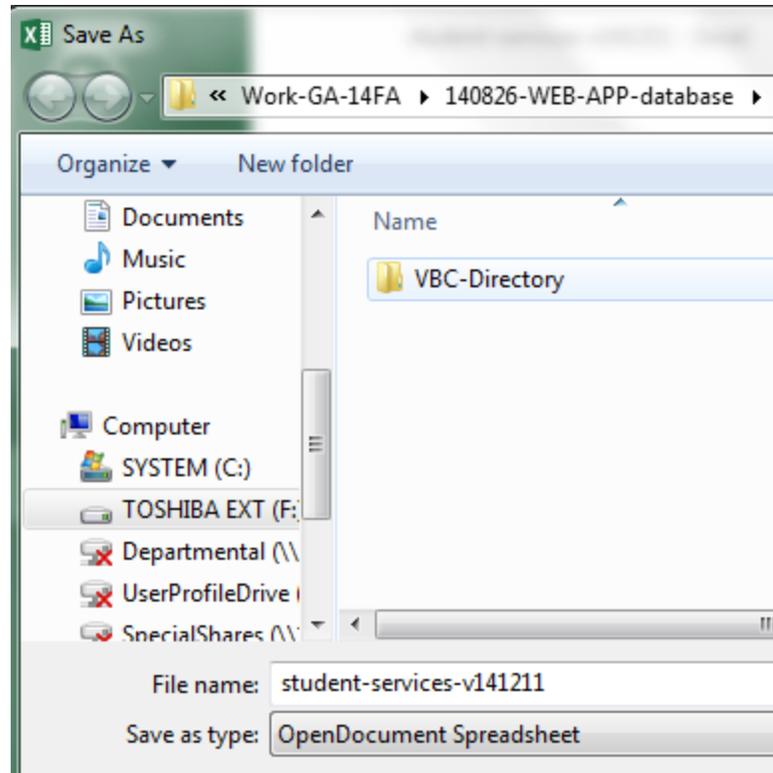


Figure 3: Save as type “OpenDocument Spreadsheet”

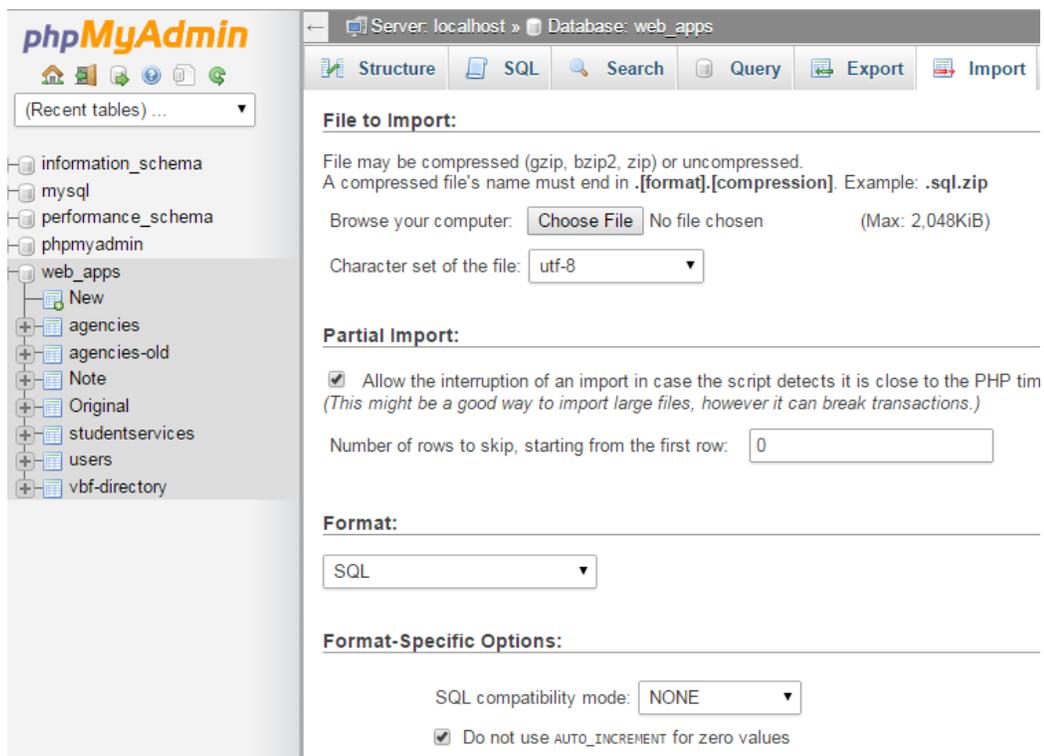


Figure 4: Using phpMyAdmin to import data from the OpenDocument spreadsheet to create a table in MySQL

Next, we logged on our Ubuntu server and accessed MySQL via the program called PhpMyAdmin. We ran PhpMyAdmin from a web browser using this url: <http://localhost/phpmyadmin>. Through PhpMyAdmin, we used the function “Import” to read the data from the OpenDocument spreadsheet and save them into a table in MySQL database on an Ubuntu server as shown in Figure 4.

Step 3: Coding in PHP and Integrating JavaScript Library from jQuery and jQuery Mobile

In the coding process for the student services web-app, we created several important files in PHP. Table 2 has the description of the main functions of these PHP files. To illustrate the inner working of the app, we included the actual codes of selected files in the appendix A for reference. Along with the codes presented are the explanations of the functions and the design logic. These explanations could be useful for anyone who is interested in adopting and using this web app for teaching purpose or for further developing this app to fit in his/her own context.

FILE NAME	DESCRIPTION	REMARK
index.php	This file is loaded when the user initiates the application. It decides whether to show mobile view or normal PC view depending on the device being detected. For instance, when a user opens the app from a mobile device, this file invokes “home.php”, which is the mobile friendly homepage of the application.	> See the appendix A for the actual codes.
home.php	This file loads the homepage of the mobile application. It includes connection to the database, CSS (Cascade Style Sheet) and JS (JavaScript) files, top bar, main content and footer of the application. It provides dropdown menu for selecting the options that the user wants to search the data by. It also has a search bar where user can search the service based on the search category they select from the dropdown menu.	
info.php	When the user selects the service from the list, the information related to a particular service is rendered by “Info.php”. This file includes connection to the database, loads all the CSS and JS files, and displays information related to the requested service in a proper format. Another function of this file is to define the data types that are used in the database. As a result, the user can define his/her own data types.	
map.php	The map related to a particular service is shown to the user using Google Map API. The code responsible for requesting data from the API resides in “map.php” file. It converts the string address to latitude and longitude and loads the map. Based on the derived latitude and longitude, “map.php” displays the current default location which is set to be the central position of the map on the screen regardless of screen sizes or types.	
phone.php	Since our application is designed for mobile devices, the display does not fit well if it is viewed on a large screen size. To aid users who are using desktop computers, we created “phone.php”. It displays the application in a mobile screen view. In our case, we chose iPhone 5. This way, users would be able to experience the look and feel of the application even when they are on a desktop computer.	

core/ config.php	Using this file, the administrator of the app can configure connection to the database. Database connection parameters include server, username, password, database name and table name. Furthermore, the file also contains configuration for text or image that goes on the top bar and the homepage of the application. Footer text such as copyright notice can also be configured in this file.	
core/ connec- tion.php	This file is responsible for establishing the connection to the MySQL database. Database parameters are retrieved from “config.php”. Different database queries are performed to fetch different data such as fieldnames, name list, searchable data types etc. All these information are stored in different arrays so that they can be used by other pages as long as those pages contain “connection.php” in their “require_once” statements. Furthermore, “connection.php” is the only file supporting database access and running queries.	
core/ header.php	This file loads all the CSS and JavaScript files for the application. Some of these file are a part of jQuery UI (user interface) library whereas others are custom created by the administrator of the app to render various UI functionalities. Note: This part requires the download of jQuery and jQuery mobile. _jQuery Core Library: jquery-1.11.1.min.js _jQuery Mobile Framework: jquery.mobile-1.4.5.min.js; jquery.mobile-1.4.5.min.css; all file inside “images” folder.	
core/ topbar.php	This file displays either text of image that is configured for the application in the “config.php” file. It generally contains application name and logo.	
core/ home- content.php	This file is responsible for displaying the text of image on the homepage of the application according to the configuration present in the “config.php” file.	
core/ footer.php	This file displays the text in the footer of the application. Usually, information like application name or copyright notice for the application is appropriate for displaying in this area. Furthermore, the footer also contains two buttons on its either side: the home and the back button. These buttons help make the application simple to navigate.	
core/ helpers.php	This file contains all the data validation and conversion methods for the application. As of now, a function to amend the URL and a function to check if the URL exists lives in this file. More functions can be added in this file to meet the application needs.	

Since all the codes for the student services web app were already written and tested, they could be copied and adapted easily for creating any other similar web app. As a result, it may not be necessary to know in-depth the programming languages of PHP, JavaScript, CSS, etc. in order to adopt this app and implement it in another context. All one needs to do is to obtain the source codes and set up a new database. Then, the next steps are to modify several parts in the codes and to configure the codes properly for a new server environment.

We used GitHub repository hosting service to store our application codes. The source codes for the student services web app as well as the documentations are accessible to the public and available for download at this url: <https://github.com/mqhuynh/Mobile-CMS>. Appendix B provides a screen shot and a brief instruction to download the source codes.

Step 4: Exploring Different Ways to Help Students Become Aware of and Access the Student Services Web App

To search for a native app, a user just comes to a respective app stores to download. For instance, if a user looks for an iPhone app, he/she would click on the App Store and then search the app by keyword. This is an advantage of a native app. On the other hand, to find a web app, a user has to know what he/she is looking for. A web app is basically an interactive website so it could be found through these channels including a search engine, a link on other websites, a share on social media, words of mouth, and/or advertising. Therefore, for us this means we need to make students aware of our web app so they can access and download it using any one of the channels above. In addition, we came up with two additional ways to make access to the student services web app easy and convenient. The first is to use QR code. We used QR generator website at <http://www.qr-code-generator.com/> to create a QR code. Figure 5 shows the QR code for the student services web app.



Figure 5: The QR code for <http://cob.cdcs.selu.edu/WEB-APP/Student-Services/>

Simply scanning the QR code above would allow users to get access to our app. Any typical mobile device with a camera would be capable of scanning and decoding a QR code. If not, then a QR reader app could easily be obtained from the app stores.

The other way to run our student services web app is to save the URL on the home screen of the mobile device. Here is how we did it from Safari web browser on an iPhone. We first ran Safari and then typed in the URL. After the page was displayed, we clicked the icon  and selected the option “Add to Home Screen”. An icon would be saved on the Home Screen just like a regular app. We only need to add to the Home Screen once. After this, we just clicked the icon to invoke the browser and to go directly to the URL embedded. For an Android device, the process is similar. This is a convenient way to run the app on a mobile device.

Understanding how the Student Services Web App Works

Here is the description of what happens when the student services web app is running. After the user types in the URL for the student services web app, the “index.php” file is loaded. Behind the scene are a series of events taking place. First, the code inside the “index.php” fetches the list of all the services from the database and their corresponding ID (primary key) associated with a specific service.

When the user types in letter(s), the app displays the list of student services that matches the input letter(s). For instance, when letter “p” was typed in, the list of all services containing the letter “p” would be displayed. This is possible because we wrote the code to filter the service names that contain the letter(s) being typed and display the service names that match the letter(s). (For more detail see the codes in Appendix A).

To continue with the example of the input letter “p” above, if “Police Department” is the service of interest, then the user could click it to obtain more information. This event would open a new page to display the desired information about Police Department.

From here, further possible actions for the user include:

1. Click for map – to invoke a map with the pin location of the address shown.
2. Click the phone number – to make a phone call to the number shown.
3. Click the email address – to send an email to the address shown.
4. Visit the URL – to open a page at the URL given.
5. Play video/audio – to view or listen to interactive elements such as video clips or audio files.

Figure 6 illustrates the interface of our student services web app and the major functions of our current version of student services web-app. It starts with a student who has a mobile device. The user would run our app in order to find the information about a specific service. The user identifies a service needed and then searches for information about the service. To input a search term, the user would type a letter or more and the web app responds with the listing of all services containing the input letter(s). From the list, the user would select one of the desired listings and this action triggers a query to the database. The database would retrieve a record and display the information of the selected service. If the service does not exist in the database, then the listing would be empty. In this case, the administrator of the app would have to get the information and update the database.

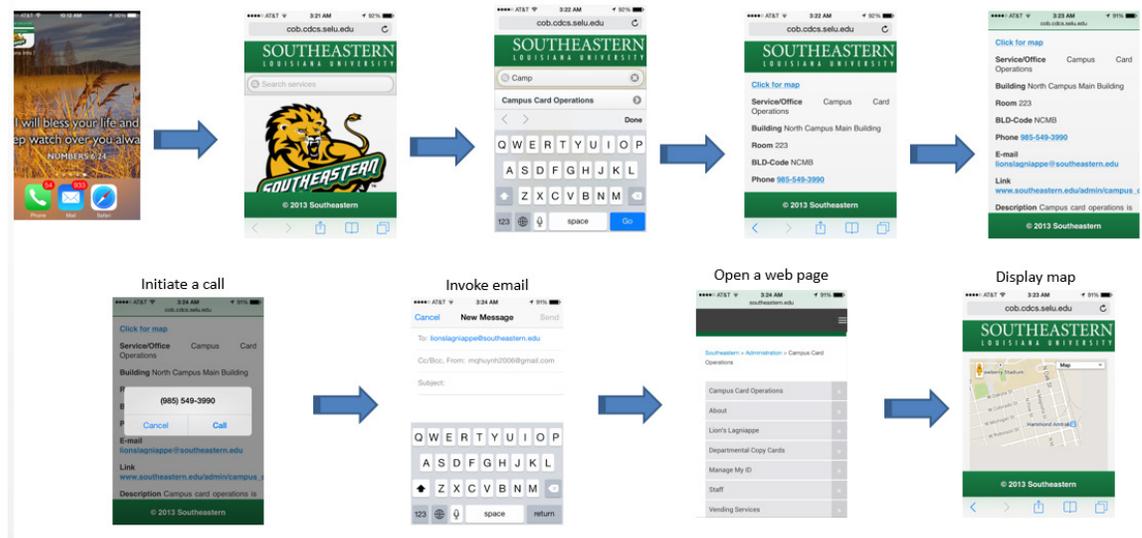


Figure 6: Interface and functions of student services web app

Discussion

Those involved in teaching information technology, information system development, and programming know firsthand the challenge of coping with the fast pace of changes in technology. Seven years ago, with almost four billion wireless connections worldwide, the cell phone industry generated vast services and large job markets. Fast forward to the present time, the number of mobile devices is thought to have increased twofold. The advance in technology has replaced many cell phones with smartphones. With added features and capabilities, these mobile devices rapidly become pervasive. As a result, there is always need for apps.

The struggle for IT educators is how to help our students to understand the processes involved in the development of an app and to appreciate the technologies supporting these processes. Such knowledge can help students to be innovative and productive in contributing their ideas to the development of future useful apps. Furthermore, having the knowledge, experience and skills in developing successful cross-platform apps is highly desirable. Yet, the reality is that it is difficult to teach app development for mobile devices to our students. As Lutes (2012) pointed out, teaching mobile app development courses presents many challenges given that mobile device technologies are changing at a blistering pace and that there is no clear smartphone market leader. However, as someone who has taught mobile app development since 2002, Lutes recommended that by utilizing the web development technologies of HTML5, CSS, and JavaScript along with native web app developer tools such as PhoneGap, a reasonable cross-platform toolset and mobile app development course is both feasible for the instructor and fun for the students (Lutes, 2012). Others, like Akopian, Melkonyan, Golgani, Yuen, and Saygin (2013), proposed a template-based short course concept as a way to teach mobile app development. As demonstrated in their report (Akopian et al., 2013), using template-based learning helped shorten the learning cycle because of these factors: ready-to-use tools, existing code templates, reusable codes, and integration process. At the other end, instead of focusing on coding, those such as Chao, Parker, and Davey (2013) put an emphasis on using application development frameworks to teach software design concepts. Their rationale is that students are often worried about learning programming language syntax, and therefore they fail to see the big picture of applying that language to solving real problems (Hundley, 2008). This is why they introduced the Model-View-Controller (MVC) pattern and chose PHP framework as the foundation for teaching app development course.

Conclusion

Recognizing the challenges in teaching mobile app development, what we hope to do in this paper is to share our own experience in the actual development of a mobile web app. From our experience, we can offer some practical guidance to lower the barriers in the development processes. In the previous sections, we have described the development processes as well as the functionalities of the student services web app that we created. Up to this point, we were successful in developing and delivering a functional web app. As we look back at the entire development processes, we realize that developing a mobile app requires an elaborate planning. To be successful, a number of processes have to come together to form a harmonious project. As in our case, our project started with an idea for an app. We next decided on the approach for the development. For us, the approach was the web app option using HTML/JQuery and CSS. After that, we then proceeded to the planning, app designing, coding, testing, and finally deploying the app to run on cross-platform mobile devices. However, the most critical part was to decide on the proper approach. We chose to follow the web app path to create and deliver something that could support cross-platforms. Our Web app was implemented with HTML, CSS, and JavaScript. This is well-aligned with Lutes (2012)'s suggestion for using web development technologies. We relied on the browser to provide a runtime environment and to capitalize the browser support of multiple platforms. As a result, we were able to implement our app capable of running on various mobile devices. This optimization accounted for the different screen size and usage of mobile devices. Although mobile devices are built differently, their browsers are similar because these browsers are based on common technologies with open and mature standards and protocols. Our student services web app also does not require advanced programming knowledge. Hence, our app constitutes an ideal starting point for cross-platform development because it can be easily portable to other cross-platform approaches (Heitkötter et al., 2013).

Teaching Implications

To develop our student services web app, we made use of several open-source software and tools as displayed in Table 1. Among them are the server running Ubuntu v.14.04, MySQL, Apache, and PHP. We also added JQuery and JQuery Mobile to make our web interface mobile-friendly. To successfully develop the student services web app from the scratch, it is necessary to know how to write code in HTML, PHP, and JavaScript. In addition, a basic knowledge of JQuery and JQuery Mobile is needed in order to control the display of information on mobile device. Finally, because our app involves the use of databases, some knowledge of SQL, database, and Linux commands is also helpful. Possessing the technical skills above is necessary to write codes for and set up our web-app to run on the server. Yet, we realized that not everyone is proficient in all these programming languages. As a result, when we developed the student service web app, we had in mind making the web app flexible, adaptable, and sharable. In our coding, we tried to make the codes easy to adopt, use, and modify. We specifically designed our programs in such a way that they could be used in various contexts.

From this project, we are able to provide some useful guidance for those who are interested in developing mobile-friendly web app. As we have described throughout this paper, it is possible to teach and learn the skills in developing a mobile-friendly web app. Learning to create a simple web app using open-source software and tools is feasible and makes good sense. Our functional student services web app provides an excellent introduction to the technical foundation needed for a web app developer. One reason is that the development of this web app involves the use of HTML, CSS, PHP, and some JavaScript. These are some of the popular programming languages for web application development. A basic competency in these languages is useful for IS students. Another reason is that there is a series of interactivities between the web app and the database. Data need to be stored and made accessible online. Then SQL needs to be used to retrieve records. Mobile device are used to display the results. Therefore, this web app provides a useful illustration on how different pieces work together. By using this web app and looking into its code, students can gain understanding of the processes involving the creation of a web app and specifically in how a web app for database access can be put together without extensive knowledge in programming languages.

Future Development Work

Our student services web app is fully functional. Its URL is <http://cob.cdcs.selu.edu/WEB-APP/Student-Services/>. Currently, it is designed to run from our own server and to deliver mainly student services information. In the next phase of our development, we plan to modify the current codes in the app and extend our design concept to create a more flexible, shareable, and easy-to-adopt source codes. The goal is to allow others to easily and quickly develop more web apps specifically involved accessing database online. In addition, we would add more enhancements including the following capabilities:

- Feature to allow users specify their own content in the database,
- Feature to support a flexible number of data fields and different user-defined data types,
- Feature to display maps for location and direction,
- Functionalities like supporting video and audio, and allowing more searchable fields,
- A simplified process for data input such as allowing users to input data directly onto the server,
- Support for international characters/fonts.

We also hope to make this source code easy for others to adopt and use. Anyone who is interested in using the codes can download and modify them to fit his/her needs. Our goal is to make the code adoption process simple so that there is no need for extensive technical knowledge. Any

typical user can take the codes, install, configure, and run them with minimal modifications. Ultimately, we hope to contribute these source codes for others to use in teaching, learning, understanding of mobile-friendly web app development.

Limitations

Since this paper focuses primarily on our first development work, it is not without limitations. A number of important elements have not been addressed. The first one is the lack of insight on the effectiveness and usability of our student services app. None of the feedback from users was collected. Hence, in the next phase, we plan to include a way to collect feedback from users. As we turn this experience into a class project, we may be able to conduct actual research and to collect data to measure not only how students evaluate the app but how much they learn from it. The second element that we did not look into is the quality and the maintenance aspects of our app development work. For instance, the paper does not address points related to metrics, quality control, and long-term maintenance. These are important considerations in the system development in general and in the cross-platform mobile app development in particular given the fact that the mobile device technologies are changing at a blistering pace.

Despite the aforementioned limitations, we hope that by reading this paper, others can gain some understanding of not only how our student services web app works and how it was developed, but more importantly how the codes and the ideas could be taken and adapted for teaching purpose and for use in other contexts.

References

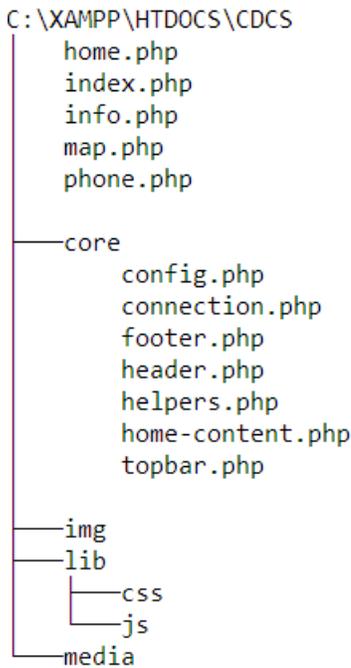
- Agrawal, M., & Gill, G. (2013). Mobile application development strategy at Vology1. *Journal of Information Technology Education: Discussion Cases*, 2, Case Number 13. Retrieved February 19, 2015 from <http://www.jite.org/documents/DCVol02/DCv02TOC.pdf>
- Akopian, D., Melkonyan, A. Golgani, S., Yuen, T., & Saygin, C. (2013). A template-based short course concept on Android application development. *Journal of Information Technology Education: Innovations in Practice*, 12, 13-28. Retrieved February 19, 2015 from <http://www.jite.org/documents/Vol12/JITEv12IIPp013-028Akopian1196.pdf>
- Caverly, C. D., Ward, R. A., & Cavarly, A. (2009). Techtalk: Mobile learning and access. *Journal of Developmental Education*, 33(1), 32-39.
- Charland, A., & Leroux, B. (2011). Mobile application development: Web vs. native. *Communications of the ACM*, 54(5), 49-53.
- Chao, J., Parker, K., & Davey, W. (2013). Navigating the framework jungle for teaching web application development. *Issues in Informing Science and Information Technology*, 10, 95-109.
- Heitkötter, H., Hanschke, S., & Majchrzak, T. (2013). Evaluating cross-platform development approaches for mobile applications. In J. Cordeiro and K. Krempels (eds), *Proceeding of Web Information Systems and Technologies. 8th International Conference, WEBIST 2012, Porto, Portugal, April 18-21, 2012*. Retrieved October 1, 2014 from <http://www3.nd.edu/~cpoellab/teaching/cse40814/crossplatform.pdf>
- Hinchly, G. (2014, January 20). Financial Times weighs in on Web vs. native app debate, how technology has shaped its own digital strategy. *INMA*. Retrieved October 1, 2014 from <http://www.inma.org/blogs/keynote/post.cfm/financial-times-weighs-in-on-web-vs-native-app-debate-how-technology-has-shaped-its-own-digital-strategy#ixzz3JMdp124z>
- Hundley, J. (2008). A review of using design patterns in CS1. *Proceedings of the 46th Annual Southeast Regional Conference on XX*, Auburn, CA, 30–33.
- Lutes, K. (2004). *Software development for mobile computers*. IEEE Pervasive Computing, July-September 2004.

Lutes, K. (2012). Cross-platform mobile app software development in the curriculum. *Issues in Informing Science and Information Technology*, 9, 115-124. Retrieved February 19, 2015 from <http://iisit.org/Vol9/IISITv9p115-124Lutes120.pdf>

Zickuhr, K. (2012). *Generations and their gadgets*. Pew Research Center. Retrieved January 20, 2013, from <http://pewinternet.org/Reports/2011/Generations-and-gadgets>

Appendix A

This is the file and folder structure of the student services app.



Below are the actual codes of major PHP files in the student services app. We also included brief explanations for the codes and the design logic behind the codes.

Home.php	COMMENTS
<pre> <?php require_once("core/connection.php"); ?> <?php require_once("core/header.php"); ?> <div data-role="page" data-theme="a"> <div data-role="main" class="ui-content main-content"> <?php require_once('core/topbar.php'); ?> <select id="searchby"> <option value="" selected disabled>Search by ... </option> <?php foreach(\$search_keys as \$key){ </pre>	<p>- Importing header.php to load CSS and JS files and connection.php to establish connection to the database.</p> <p>- Loading topbar of the app and dropdown for searchable fields.</p> <p>- Service name typed in the search box is filtered from the hidden list and displayed. When a service from the list is clicked, it opens information</p>

Develop a Cross-Platform Web App

<pre> echo "<option value=" . \$key . ">" . \$fieldname[\$key] . "</option>"; } ?> </select> <form class="ui-filterable"> <input id="autocomplete-input" data-type="search" placeholder=<?php echo "Search by ' . \$fieldname[\$search_category] . "'"; ?>> </form> <ul class="data-list" data-role="listview" data-filter="true" data-filter-reveal="true" data- input="#autocomplete-input" data-inset="true"> <?php if(\$search_category!=NULL){ \$i = 0; foreach (\$all_data as \$names){ echo "<li id=" . \$name_keys[\$i] . ">" . \$names[\$search_category] . "
<i> .\$names[\$name_index] . "</i>"; } } ?> </div> <div style="text-align:center"> <?php require_once('core/home-content.php');?> </div> <script type="text/javascript"> \$("#searchby").change(function(){ window.location.href= "home.php?s="+\$(this).val(); }); </script> <?php require_once("core/footer.php"); ?> </pre>	<p>related to that particular service.</p> <p>- Configurable content from home-content.php is imported and displayed.</p>
--	---

Info.php	COMMENTS
<pre> <?php require_once ('core/connection.php'); ?> <?php require_once('core/header.php'); ?> <?php require_once('core/topbar.php'); ?> <div class="info main-content"> <?php </pre>	<p>- Including connection.php for accessing database variables, topbar.php to load topbar of the app and header.php so that all the CSS, JavaScript and database variables will be available in the page.</p>

<pre> \$info_id = isset(\$_GET['id'])?\$_GET['id']:NULL; if(\$info_id != NULL){ foreach (\$all_data as \$record) { if(\$record[0] == \$info_id){ \$i = 0; foreach (\$record as \$data) { if(strpos(\$datatype[\$i],"name") !== false){ echo '<h3>'.\$data.'</h3>'; } else if(strpos(\$datatype[\$i],"map") !== false){ echo ''.\$data.''; } else if(strpos(\$datatype[\$i],"phone") !== false){ echo ''.\$data.''; } else if(strpos(\$datatype[\$i],"email") !== false){ echo ''.\$data.''; } else if(strpos(\$datatype[\$i],"url") !== false){ echo ''.linkify(\$data).''; } else if(strpos(\$datatype[\$i],"image") !== false){ if(is_good_link(\$data)){ \$data = linkify(\$data); echo '<div class="center image"></div>'; } } else if(strpos(\$datatype[\$i],"video") !== false){ if(is_good_link(\$data)){ \$data = linkify(\$data); echo '<div class="center video">'; </pre>	<p>- Handling information from the database table according to different data-types configured.</p> <p>- Null values in the database table are discarded.</p>
--	---

Develop a Cross-Platform Web App

```
        if(strpos($data, ".") != false){
            echo
            '<video controls>'
            '<source src="'. $data. '" type="video/mp4">'
            '<source src="'. $data. '" type="video/ogg">'
            'Your browser does not support the video tag.'
            '</video>';
        }
        else{
            echo '<embed src="http://www.youtube.com/v/'. $data. '">';
        }
        echo '</div>';
    }

}

else if(strpos($datatype[$i], "audio") != false){
    if(is_good_link($data)){
        $data = linkify($data);
        echo
        '<div class="center audio"><audio controls>'
        '<source src="'. $data. '" type="audio/ogg">'
        '<source src="'. $data. '" type="audio/mpeg">'
        'Your browser does not support the audio element.'
        '</audio></div>';
    }
}

else if(strpos($datatype[$i], "text") != false){
    echo $data;
}

else{
}

if(strlen($data)>1){
    echo "<hr>";
}

$i++;
}

}
```

<pre> } } ?> </div> <?php require_once("core/footer.php"); ?> </pre>	
---	--

Map.php	COMMENTS
<pre> <?php require_once('./core/connection.php'); require_once('core/header.php'); \$info_id = urldecode(\$_GET['id']); \$record = \$all_data[\$info_id]; \$map_index = 0; foreach (\$datatype as \$key => \$value) { if(strpos(\$value,"map") !== false){ \$map_index = \$key; break; } } \$address = \$record[\$map_index]; \$name = \$name_list[\$info_id]; \$request_url = "http://maps.googleapis.com/maps/api/geocode/xml?address=".\$address."&sensor=true"; \$xml = simplexml_load_file(\$request_url) or die("Google Map did not"); \$status = \$xml->status; \$lat = ""; \$lng = ""; if(\$status == "OK"){ \$lat = \$xml->result->geometry->location->lat; \$lng = \$xml->result->geometry->location->lng; } ?> <!DOCTYPE html> <html> <head> </pre>	<p>- Primary key obtained from the URL parameter is used to query the address from the database variables.</p> <p>- The map is displayed using Google Maps API.</p>

Develop a Cross-Platform Web App

```
<script src="http://maps.googleapis.com/maps/api/js?key=AlzaSyBvA7Bh-
sg13p3xi9ZrllqVAmsLuOZzYVc"></script>
<script>
var name = "<?php echo $name;?>";
var address = "<?php echo $address;?>";
var lat = <?php echo $lat; ?>;
var lng = <?php echo $lng; ?>;
var latLng = new google.maps.LatLng(lat,lng);
function initialize() {
  var mapProp = {
    center:latLng,
    zoom:18,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };
  var map=new google.maps.Map(document.getElementById("googleMap"),mapProp);
  var marker = new google.maps.Marker({
    position: latLng,
    map:map,
  });
  marker.setMap(map);
  google.maps.event.addListener(marker, 'click', function() {
    map.setCenter(marker.getPosition());
    var info = new google.maps.InfoWindow();
    info.setContent(name+" "+address);
    info.open(map,marker);
  });
}
google.maps.event.addDomListener(window, 'load', initialize);

</script>
</head>
<body>
  <?php require_once('core/topbar.php');
      echo '<h3 style="text-align:center">'.$name.'</h3>';
  ?>
  <div class="main-content" id="googleMap" style="width:100%; text-
align:center"></div>
```

<pre><?php require_once("core/footer.php"); ?> </body> </html></pre>	
--	--

Config.php	COMMENTS
<pre><?php \$server = "localhost"; \$username = "xxxxx"; \$password = "xxxxx"; \$databse = "web_apps"; \$table = "agencies"; \$topbar_text = "Southeastern LA University"; \$home_content_image = "img/cdcs-home.png"; \$footer_text = "&copy CDGS 2016"; ?></pre>	<p>- The variables namely \$server, \$username, \$password, \$databse, and \$table should be configured according to the user's server information.</p> <p>- Text that goes on the topbar and footer as well as homepage image for the app can be configured using this file.</p>

Connection.php	COMMENTS
<pre><?php // connection setting require_once('core/config.php'); require_once('core/helpers.php'); \$connection = new mysqli(\$server, \$username, \$password, \$databse); if(mysqli_connect_errno()){ echo "Connection could not be established"; exit(); } \$url_id = isset(\$_GET["id"])?\$_GET["id"]:NULL; \$query = 'SELECT * FROM '.\$table; \$result = \$connection->query(\$query); \$all_data = array(); \$specific = array(); \$name_list = array(); \$fieldname = array();</pre>	<p>- This file established connection to the database server, retrieves the data from the specified table and stores in different variables to reuse in different pages within the application.</p> <p>- Different array variables are declared and initialized (as empty array) to store the information queried from the database.</p> <p>The value stored by each variable is described below.</p> <p>\$all_data : It stores all the rows and columns of the table in a two dimensional array.</p> <p>\$specific : It stores a specific</p>

<pre> \$datatype = array(); \$searchable = array(); \$search_keys = array(); \$name_keys = array(); \$i = 0; \$m = 0; while (\$row = \$result->fetch_array(MYSQLI_NUM)) { if(\$i == 0){ \$fieldname = \$row; } else if(\$i == 1){ \$datatype = \$row; } else{ \$all_data[\$row[0]] = \$row; \$name_keys[\$m++] = \$row[0]; } \$i++; } foreach (\$all_data as \$each_service){ if(!strcmp(\$url_id, \$each_service[0])){ \$specific = \$each_service; break; } } for(\$i = 0; \$i < count(\$datatype); \$i++){ if(strpos(\$datatype[\$i], "search_")!== FALSE){ \$searchable[\$i] = \$datatype[\$i]; \$search_keys[\$i] = \$i; } } \$name_index = 0; foreach (\$datatype as \$key) { if(strpos(\$key,"name") !== false){ break; } </pre>	<p>data row from the table according to request query. It differs accordingly with the value of “id” that comes in from the url.</p> <p>\$name_list: It stores values from the column that is marked as datatype “name”.</p> <p>\$fieldname: It stores name of the fields of each columns as provided in first row of the table.</p> <p>\$datatype: It stores datatypes of all the columns of the table that are provided in second row of the table.</p> <p>\$searchable: It stores fields that are marked as “searchable” in second row.</p> <p>\$search_keys: It stores primary keys of searchable fields.</p> <p>\$name_keys: It stores primary keys of all the rows in the table.</p>
---	--

<pre> \$name_index++; } foreach (\$all_data as \$key=>\$value) { \$name_list[\$key] = \$value[\$name_index]; } \$connection->close(); ?> </pre>	
header.php	COMMENTS
<pre> <!DOCTYPE html> <html> <head> <title>CDCS App</title> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <link rel="stylesheet" href="lib/css/customCdc.css" /> <link rel="stylesheet" href="lib/css/jquery.mobile.icons.min.css" /> <link rel="stylesheet" href="lib/css/jquery.mobile.structure- 1.4.5.min.css" /> <link href="lib/css/custom.css" rel="stylesheet" type="text/css"/> <script src="lib/js/jquery-1.11.1.min.js" type="text/javascript"></script> <script src="lib/js/jquery.mobile-1.4.5.min.js"></script> <script src="lib/js/custom.js" type="text/javascript"></script> </head> <body> <?php \$search_category = isset(\$_GET['s'])? \$_GET['s']:\$name_index; ?> </pre>	<p>- The header.php files contains link to local external libraries JavaScript and CSS files. It includes reference to custom JavaScript and CSS files. It also includes link to optional CSS theme file.</p> <p>- Custom theme can be applied by replacing the link to customCdc.css with the link to the file downloaded from JQuery Mobile Themroller.</p>

Appendix B

We used GitHub repository hosting service to store our application codes. The source codes for the student services web app as well as the documentations are accessible to the public and available for download at this url: <https://github.com/mqhuynh/Mobile-CMS>

Below is the screen shot of our GitHub repository page. Note that at the bottom right corner of the screen shot is the button to download ZIP file. Click on it to download all the source codes. After downloading, unzip the file. Open the source codes using any HTML editors.

The screenshot shows the GitHub repository page for 'mqhuynh / Mobile-CMS'. The repository has 2 commits, 1 branch, 0 releases, and 1 contributor. The commit history table is as follows:

File	Commit Message	Time
core	Initial Commit.	2 months ago
img	Initial Commit.	2 months ago
lib	Initial Commit.	2 months ago
README.md	Readme File	2 months ago
home-content.php	Initial Commit.	2 months ago
index.php	Initial Commit.	2 months ago
info.php	Initial Commit.	2 months ago
phone.php	Initial Commit.	2 months ago
home.php	Initial Commit.	2 months ago
map.php	Initial Commit.	2 months ago

On the right sidebar, there are buttons for 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', and 'Graphs'. At the bottom right, there are buttons for 'Clone in Desktop' and 'Download ZIP'.

Biographies



Minh Q. Huynh is an Associate Professor at Southeastern Louisiana University. He received his Ph.D. from State University of New York at Binghamton, his B.S. in Computer Sciences from University of Maryland University College, his B.A. in Physics from Franklin & Marshall College. His teaching interests are in the areas of E-commerce, Web design, Introduction to MIS. Research interests include software development using open source, web database implementation, web apps, technology supported distance education, IT for small businesses. His publications appear in such journals as the Communications of ACM, Journal of AIS, Communications of AIS, European Journal of IS, Journal of Electronic Commerce in Organizations, Journal of Information Systems Education, Business Studies Journal.



Prashant Ghimire is pursuing a Bachelor of Science degree in Computer Science at Southeastern Louisiana University. Prashant is familiar with Java, PHP, C#, Python, JavaScript, CSS and HTML. He has worked on various projects using development frameworks like ASP.NET MVC, CakePHP, jQuery, and PhoneGap and has actively involved in a research project on web apps development. His work on “Linking QR code to service learning” was featured at the IARSLCE-Research poster session. His research on QR Code/Barcode processor was published on Business Studies Journal. He also presented this research at the 6th Annual General Business Conference at Sam Houston State University. He is currently working in systems administrations and client connectivity for Southeastern as an undergraduate technical student assistant.