Development of the User Interface for a City of White Plains Parking App

Ari Butowsky, Keke Gai, Mike Coakley, and Meikang Qiu
Seidenberg School of CSIS, Pace University, White Plains, New York

Abstract

White Plains City (WPC) contains a residential population of 57,000 people, which rises by 225,000 people every day who commute in from the outlying suburbs of White Plains to work and shop in the city. Due to the traffic created, along with WPC's large number of parking locations and how spread out they are, an app must be created that will allow people to find viable parking spaces easily. It will navigate customers to parking structures in conjunction with their destination location(s). The app's user interface (UI) will be constructed using HTML, CSS, and JavaScript. It will have a tag-based network structure to facilitate exploratory behavior. The research methodology used will involve action research and case studies. The expected results of this project are the completed UI, a report detailing an explanation of its perceived benefits, an analysis of the difficulties experienced in developing the UI, an evaluation of it and its application, an outline methodology for future UI projects, and a discussion and evaluation of alternative tools and methods for developing UIs.

Keywords: action research, Cascading Style Sheet (CSS), case study, Hypertext Markup Language (HTML), parking app, programming language, web application, tag-based network structure, user interface (UI)

I. Introduction

White Plains City (WPC) is a growing central business and commercial city with a residential population of approximately 57,000 people. Known as one of the top suburban office and retail centers in the United States, the city boasts a massive number of public and private entities that draw commuters to it [2].

A. Reasons for Daily Population Increases

In fact, WPC's population has been known to increase by 225,000 people every day due to the influx of workers, shoppers, and visitors arriving by various means of transportation, especially by automobile [2]. This is because the city has the following features:

1. Various establishments like restaurants, arts and cultural destinations, boutiques, and many others that attract both visitors and residents alike [2].

2. Downtown White Plains being a business hub, bringing law firms, major corporations, colleges/universities, and high technology firms to its doorstep [2].

3. Schools that provide top quality education and services for a colossal student population [2].

4. Twenty electric vehicle charging stations in parking areas throughout the city [7].

5. The Transit Center serves as a travel hub for commuters and serves as a gateway from New York City to Upstate New York.

These and other features illustrate why WPC is such an attractive city for commuters and why there was the need to build so many parking centers and parking garages throughout the entire city.

B. Problems associated with Parking Locations

One of the problems with WPC's parking locations today is that each of them are scattered all over the place within an extremely large area. As a result of this, as well as due to the influx of commuters no doubt creating heavy traffic in the city all the time, it is often problematic for the average person to travel to different places within the city, much less find a place to park.

C. Parking App's Necessity

The problems described in the previous section illustrate why there is a need for a parking app that will facilitate and efficiently assist these people in finding a viable parking space.

This user-friendly app will assist customers by primarily locating the appropriate parking location. It will navigate customers to the nearest parking structure in conjunction with their destination location.

Coming up with an idea for an app is one thing. Building the interface for it, however, is another. Luckily, as the following sections illustrate, there is a way to create the best interface possible for a mobile app of this kind.

II. Comparison to Similar Works

A. Apps that Perform Similar Functions

There is already a mobile app for MapQuest that does something similar to the app that is going to be built to address the WPC parking issue. For example, one types in a certain address and the app takes the user to that specific location, but data about pricing is not listed when it comes to parking locations.

Another app is called BestParking, which provides a map of the parking areas in WPC. However, unlike MapQuest, this app does list the prices associated with each parking area. In addition, both MapQuest and BestParking can be used on various types of mobile devices, including Android, BlackBerry 10, iPhone, and iPad [3].

Lastly, there is the WPC website, which states all kinds of information about the city. In spite of that, however, it is only a website and not a mobile app.

B. How Will WPC App be Similar?

The WPC parking app will be similar to all of the apps described in the previous section by containing data on the parking facilities in the city, including their locations, the prices of parking, and so on. However, the app will be different in that it will have a GPS function that will allow people in WPC to navigate through the city easier in order to find the nearest parking facilities much quicker than normal.

III. What Consumers Want from the App

Before getting right into the topic of the app interface itself, however, it is worth noting what's expected from such apps by today's mobile phone users and how difficult it is to meet these expectations.

A. Mobile Users' Expectations

They expect to find speedy transitions from one Web page to the next, smooth load experiences, and application-centric error messages. In addition, mobile phone apps are expected by users to be responsive even when network connectivity is spotty and the network is not highly responsive itself [12].

B. Difficulty in Meeting Consumers' Demands

Meeting these demands is tough for any app developer today because it involves a new way of thinking. The old rules for creating and implementing websites on desktop computers will not work when they are applied to creating and implementing mobile web applications [12]. This is because not only could doing so degrade the performance of the mobile applications themselves, but it could also lead to the intended benefits of using the old approaches not being seen as opposed to when using them for creating websites on desktop computers [12]. Therefore, new methods need to be created in order to make sure that mobile web applications are able to function properly.

IV. Why a Web Application?

The reason for this app being a web application is due to the pros of developing it as such far outweigh the cons. (i.e. Look at Table 1 in the Appendix for more details on the pros and cons of building a web application as opposed to a native application.)

In fact, the only cons associated with building a web application include having to come up with one's own payment system for the app, being unable to access some of the hardware features of the phone, and finding it difficult to create sophisticated effects on the user interface [1].

In the grand scheme of things, these cons seem to be a small price to pay for designing a web application. After all, based on the information provided in the table about the pros and cons of both native and web applications, it becomes clear that web applications allow more flexibility for the app developer in a wide variety of ways. For example, becoming a Web application developer does not require any money, one's current web design and development skills are enough for the job, one's current authoring tools can be used, the app will run on ANY device with a web browser instead of just on an Android phone, bugs in the app can be fixed at any time, and the development cycle for the app is faster than that for native applications. Simply put, web applications are more flexible and easier to build than native applications which is why a web application will be built for this study [1].

V. Hardware and Software Details

The interface for the WPC parking app will be constructed so that it can be used on any type of mobile device. In order to do that, it must contain a wide variety of attributes that would make it adaptable to any situation that comes its way.

A. The Programming Languages Used

The interface will be constructed using three computer languages: Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), and JavaScript. An HTML document would serve as the underlying structure of the webpage created by it, CSS would be applied to the document in order to define its visual representation, and JavaScript code would be added to it in order to make the webpage more interactive and convenient for the user. In other words, web pages are simply HTML documents that are given visual representation by CSS and that become more interactive and convenient for users because of JavaScript. Therefore, with regards to the WPC parking app's interface, neither HTML, CSS, nor JavaScript must be kept out of the equation when constructing it [1].

B. Maximizing User Interfaces Performance

As for how to make the interface effective in terms of its performance as a mobile app, Hypertext Transfer Protocol (HTTP) requests and round trips will have to be reduced significantly [12].

1. What are HTTP Requests?

HTTP requests are sent by the web browser to the web server for a certain Web page in the URL when the page is loaded. When the HTML that structures the page is delivered, the web browser analyzes it and searches for more requests for images, scripts, and CSS, among other things. The more of these there are, the more HTTP requests there will be and the more time it will take for the Web page to load. Therefore, minimizing these HTTP requests is necessary in order for the WPC parking app to function effectively [11].

2. How to Reduce HTTP Requests

This can be done through the use of image maps, CSS sprites, and through combining various files together. Image maps are clickable regions on images that make links more visually appealing and the Web page more interesting [11]. By combining these images into a CSS sprite (i.e. Splits a large image into many smaller images.), all site maintenance and navigation buttons will be scrunched up together into one file [5]. With regards to using CSS, although using style sheets and scripts is necessary to keep various files from increasing the amount of time needed for a Web page to load, the app will have only one style sheet and script. This is because style sheets and scripts can interfere with the loading time of a Web page as well in when in greater numbers. In order to make this possible, the files containing each aspect of the app's Web page will have to be combined into one [11].

3. What is a packet?

Before going any further, it is worth talking about what a packet is. It's a unit of data that is part of a single file that is sent from one destination to another. As the file is sent, it is split up into individual packets that each take different routes before combining into their original form upon reaching their destination [13].

4. Round Trips

Round trips are the process in which packets go from one destination to the other [14]. The time it takes for these packets to undergo this process should be reduced in order to make the mobile web application more efficient. In order to do this, all code, assets, and CSS for the website must load in just one round trip. Additionally, dynamic data will be loaded separately and HTML5 storage APIs should be used to store/cache follow-on content [12].

5. Tag-Based Network Structure

Even though HTTP requests will be greatly reduced through these methods, it will all mean nothing without the proper network structure. For the WPC parking app, it will be a tag-based network structure as opposed to the folder-based hierarchical structure seen on typical user-interfaces seen on mobile phones. (i.e. For more information on each interface, see Table 2 in the Appendix.) A tag is a user-generated descriptor that refers to an aspect of content. It is usually visualized in a tag cloud, which is a depiction of frequently used tags. Clicking on a tag will lead to a colossal amount of content being displayed in association with that tag. However, this will only be possible if the tag-based network is installed on a D-button phone, a multi-display button phone in which each display button can present either content or contextual information individually and be pressed physically. A D-button phone has content or contextual information both on the main screen and the display buttons. It also provides preview thumbnails of media content while detailed content is shown on the main screen. This way, exploratory browsing depends on chance and it is affected by context since the D-button phone displays both a view of the context surrounding the content presented and a view of the actual content itself [9].

VI. Benefits of the WPC Parking App

The benefits of the WPC parking app are impossible to overestimate. It will provide people in White Plains with the ability to find the nearest parking locations and make navigation throughout WPC a whole lot easier due to the GPS navigation feature installed into the app. Since this feature will function using the data obtained from WPC, it will almost certainly provide accurate information on where various parking centers can be located. The data will also help in providing information to users on parking fees and where various electric car chargers are located, among other things.

VII. Research Methodology

In order to complete this app, a case study, and a significant amount of action research, must be performed.

A. Action Research

Action research is the study of an attempt to solve a problem or change a situation. It involves working with a certain subject and evaluating the results. In the case of the WPC parking app, the action research will be conducted with various people, more specifically with a number of Ph.D graduate students from Pace University. The results of working with these students will then be evaluated as part of the creation of the app itself [6].

B. Case Study

A case study involves the investigation of a certain problem, situation, or company(s). This will be done directly through the study of WPC official documentation concerning data on parking in the city. This documentation can include where specific parking places are, how much it costs to park in those areas, and so on [6].

VIII. Desired Outcome

The outcome of the creation of the WPC parking app is not simply its construction, evaluation, and effective performance. It will also involve a written report with an explanation of the perceived benefits of this user interface, an analysis of the difficulties experienced in developing the user interface, a critical evaluation of the user interface and its application, an outline methodology for future user interface projects, and a discussion and evaluation of alternative tools and methods for developing user interfaces.

VIV. Progress

So far, the web interface's homepage and its main webpage have been created, but they are incomplete and are riddled with problems.

A. Interface Homepage

The homepage was created using only HTML because that was the only programming language needed for making a simple introductory webpage. (i.e. For more details on the code that was typed to create the homepage, please refer to Fig. 1 in the Appendix.)

As seen in the screenshot of the homepage that is provided in Fig. 2 in the Appendix, there are five hyperlinks on the top called HOME, ABOUT US, DEVELOPERS, PARKING OPERATORS, and CONTACT US. Each of these hyperlinks are set up to direct the user to different webpages related to the interface itself.

There are also four pictures at each of the far corners of the homepage that are related to the concept of parking. (**Note:** Due to various reasons, only one of these pictures was able to be placed in the screenshot in Fig. 2 in the Appendix.)

Lastly, there is a hyperlink named WELCOME TO MY PARKING SYSTEM!, which leads to the interface's main webpage when clicked on.

B. Homepage Problems

The interface's homepage is still imperfect and is riddled with all kinds of problems. One issue is that the webpages the user is sent to by clicking on ABOUT US and/or CONTACT are incomplete with regards to information and presentation. Another issue is that clicking on DEVELOPERS and PARKING OPERATORS leads to an error message since no HTML files have been created for the webpages those hyperlinks were supposed to send the user to. In other words, the homepage's main problem is that it is incomplete.

C. Interface Main Webpage

The main webpage was created using both HTML and JavaScript. For more details on the code, refer to Fig. 3 in the Appendix. It boasts a wide variety of features as seen in Fig. 4 in the Appendix, including:

1. Five hyperlinks on the top of the screen named HOME, ABOUT US, DEVELOPER, PARKING OPERATORS, and CONTACT.

2. Textboxes where users enter in various kinds of data, including the distance between a parking

location and the user, the time period when a parking lot is available, whether or not a certain type of parking is public or on-street, and departure and arrival times.

3. A "search" function that looks for which parking areas are available during specific times based on the data entered in the textboxes by the user.

4. A Google Maps function that acts based on the data entered in the textboxes by the user.

D. Problems with Main Webpage

Just like the interface's homepage, however, its main webpage is far from perfect. First of all, the hyperlinks at the top of the webpage do not lead to other webpages like they should be doing when clicked on. This is because no HTML files were created yet for the webpages the hyperlinks are supposed to direct users to upon being clicked. Although, in the case of HOME, that was not the problem. The real problem for HOME was that a route was not established to take users to the interface's homepage upon clicking on this particular hyperlink. Another problem is that the "search" button does not do anything when clicked on. This happens no matter what information is entered into the textboxes above it.

E. Recommendations for Future Work

In the future, this interface is expected to be completed, with a more attractive looking homepage with all of its hyperlinks fully operational, among other things. It will also contain more pictures, some animations, and maybe some movies as well.

As for the interface's main webpage, its hyperlinks and "search" function are expected to be fully operational.

HTML and JavaScript are expected to play huge roles in helping to make these objectives realized. Some more action research and case studies are also expected to play a role as well in completing the interface. Until then, the problems associated with the interface as of today will continue.

Appendix

Table 1: Native vs Web Applications [1]

Native Applications		**Web Applications**	
Pros	**Cons**	**Pros**	**Cons**
Credit card owners are one click away from activating the app.	Becoming an Android app developer requires payment.	Web developers can use their current authoring tools.	One has to roll one's own payment system if one wants to charge for the app.
All hardware features of the phone are accessible.	The app will only run on Android phones.	The app will run on any device with a web browser.	Not all hardware features of the phone are accessible.
N/A	The app can only be developed using Java.	One's current web design and development skills can be used to develop the app.	It is difficult to achieve sophisticated user interface effects.
N/A	The development cycle is slow and requires a multitude of steps that need to be repeated before being officially completed.	The development cycle is extremely fast.	N/A
N/A	N/A	Bugs in the app can be fixed at any time.	N/A

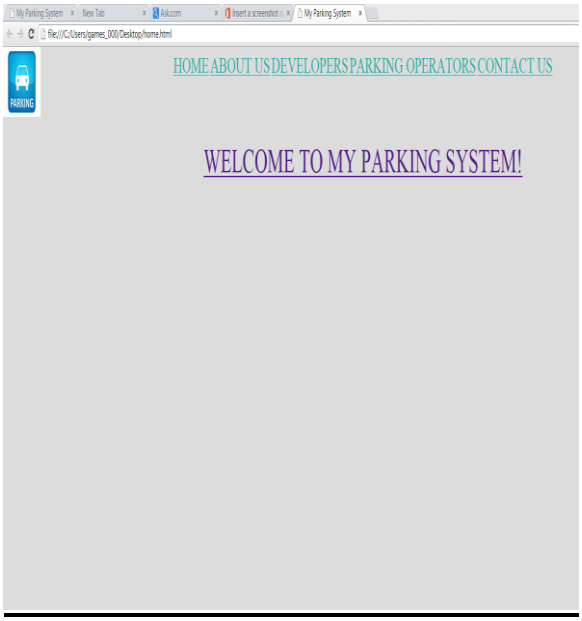
Table 2: Folder-Based Hierarchical Structure vs. Tag-Based Network Structure [9]

Folder-Based Hierarchical Structure	Tag-Based Network Structure
Does not lead to exploratory browsing because users are forced to navigate through a fixed and rigid structure.	Leads to exploratory browsing because of multiple tags attached to various user-generated content stimulating users' curiosity.
User-generated content in the same folder belong to the same category, failing to spark users' curiosity due to them already foreseeing what content the category contains.	Selecting a single tag in a tag cloud leads to an array of content associated with that tag, leading to higher chances of discovering meaningful content by chance, thus sparking users' curiosity.
Content is limited to one display, preventing users from viewing other content or contextual information at the same time.	Content is not limited to one display.

Fig. 1: Interface Homepage - Code

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://w3.org/1999/xhtml"><head><BODY bgcolor="#DCDCDC"><title>My Parking System</title><meta http-equiv="Content-Type" context="text/html; charset=utf-8" /><style type = "text/css"></style></head><body><center><a href = "home.html"><font name = "Calibri" size = 6 color = "#20B2AA">HOME</font></a><a href = "about.html"><font name = "Calibri" size = 6 color = "#20B2AA">ABOUT US</font></a><a href = "developer.html"><font name = "Calibri" size = 6 color = "#20B2AA">DEVELOPERS</font></a><a href = "parking-operators.html"><font name = "Calibri" size = 6 color = "#20B2AA">PARKING OPERATORS</font></a><a href = "contact.html"><font name = "Calibri" size = 6 color = "#20B2AA">CONTACT US</center></font></a><br><br><br><br><br><center><a href="mainpage.html"><font name = "Calibri" size = 7>WELCOME TO MY PARKING SYSTEM!</a></font></center><div style="position: absolute; left: 0px; top: 0px;"><img src="C:\Users\games_000\Pictures\parking.png" alt="Parking Sign" height="100" width="100"></div>
<div style="position: absolute; right: 0px; top: 0px;">
<img src="C:\Users\games_000\Pictures\parking.png" alt="Parking Sign" height="100" width="100"></div>
<div style="position: absolute; left: 0px; bottom: 0px;">
<img src="C:\Users\games_000\Pictures\parking.png" alt="Parking Sign" height="100" width="100"></div>
<div style="position: absolute; right: 0px; bottom: 0px;">
<img src="C:\Users\games_000\Pictures\parking.png" alt="Parking Sign" height="100" width="100"></div></body></html>
```

Fig. 2: Interface Homepage - Screenshot



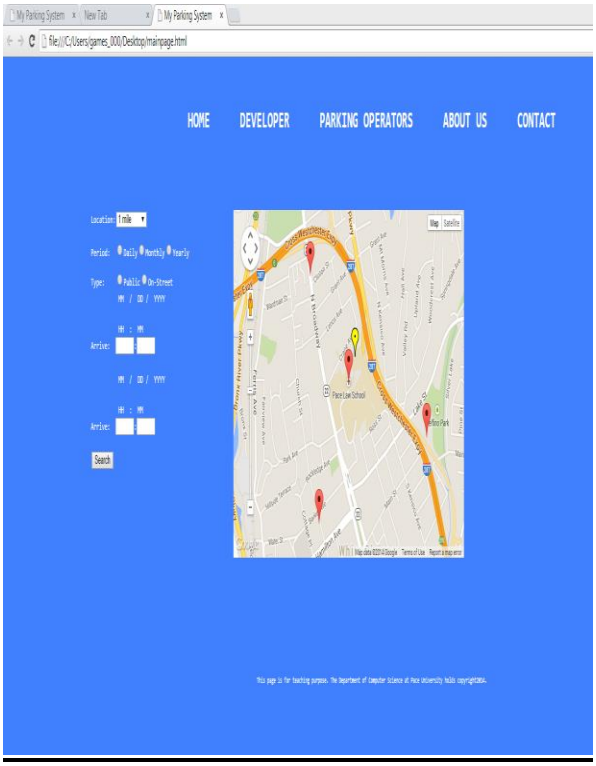
(**Note:** This screenshot was unable to capture all of the illustrations created by the HTML code.)

Fig. 3: Interface Mainpage – Code

```
<html><head><title>My Parking System</title><style>
#wrap{width: 1200px; height: 550px;}#textboxes{
width: 1000px; height: 400px; position: absolute; left: 0px;
float: left;}#map_canvas{width: 600px; height: 400px;
position: absolute; left: 600px;}</style><script
src="https://maps.googleapis.com/maps/api/js"></script>
<script>function initialize(){var mapCanvas =
document.getElementById('map_canvas'); var mapOptions
= {center: new google.maps.LatLng(41.0402,-73.7647),
zoom: 15, mapTypeId:
google.maps.MapTypeId.ROADMAP }var map = new
google.maps.Map(mapCanvas, mapOptions);
var contentString = '<font color="#000000">'+
'<table>'+'<tr>'+'<th>Hours Parked</th>'+'<th>Rate</th>'+
'<th>Hours Parked</th>'+'<th>Rate</th>'+'</tr>'+'<tr>'+
```

```
'<td>1 hour</td>'+'<td>$2</td>'+'<td>5
hours</td>'+'<td>$10</td>'+'</tr>'+'<tr>'+'<td>2
hours</td>'+'<td>$4</td>'+'<td>6
hours</td>'+'<td>$12</td>'+'</tr>'+'<tr>'+ '<td>3
hours</td>'+'<td>$6</td>'+'<td>7
hours</td>'+'<td>$12</td>'+'</tr>'+'<tr>'+'<td>4
hours</td>'+'<td>$8</td>'+'<td>8
hours</td>'+'<td>$12</td>'+ '</tr>'+'</table>'+'</font>';
var infowindow = new google.maps.InfoWindow({
content: contentString}); var marker = new
google.maps.Marker({position: new
google.maps.LatLng(41.0402,-73.7647), map: map, });
var parking2 = new google.maps.Marker({ position: new
google.maps.LatLng(41.0442,-73.7690), map: map, });
var parking3 = new google.maps.Marker({
position: new google.maps.LatLng(41.0350,-73.7680),
map: map, }); var parking4 = new google.maps.Marker({
position: new google.maps.LatLng(41.0382,-73.7560),
map: map, });
google.maps.event.addListener(marker, 'click', function() {
infowindow.open(map,marker); }); var pinColor =
"FFFF00"; var pinImage = new
google.maps.MarkerImage("http://chart.apis.google.com/ch
art?chst=d_map_pin_letter&chld=%E2%80%A2|" +
pinColor, new google.maps.Size(21, 34), new
google.maps.Point(0,0), new google.maps.Point(10, 34));
var pinShadow = new
google.maps.MarkerImage("http://chart.apis.google.com/ch
art?chst=d_map_pin_shadow", new google.maps.Size(40,
37), new google.maps.Point(0, 0), new
google.maps.Point(12, 35)); var marker = new
google.maps.Marker({position: new
google.maps.LatLng(41.0412,-73.7640), map: map, icon:
pinImage, shadow: pinShadow }); }
google.maps.event.addDomListener(window, 'load',
initialize); </script></head><body bgcolor="#4080ff">
<br><br><pre><center><strong><a href =
"temporary.html" style = "text-decoration:none"><font
size = 6  color = "#ffffff" >HOME</font></a><a href=
"temporary.html" style = "text-decoration:none"><font
size = 6  color = "#ffffff" >DEVELOPER</font></a>
<a href= "temporary.html" style ="text
decoration:none"><font size = 6  color = "#ffffff"
>PARKING OPERATORS</font></a><a href=
"temporary.html" style = "text-decoration:none"><font
size = 6  color = "#ffffff" >ABOUT US</font></a>
<a href= "temporary.html" style = "text-
decoration:none"><font size = 6  color =
"#ffffff">CONTACT</font></a></strong></center><pre>
<div id = "wrap"><div id = "textboxes"><font size = 3
color = "#ffffff" >Location:</font><select><option
value="1" type="text">1 mile</option><option value="5"
type="text">5 miles</option><option value="10"
type="text">10 miles</option> </select><br><font size = 3
color = "#ffffff" >Period:  <input type="radio" name="a"
value="1" id="a1">Daily</input><input type="radio"
name="a" value="2" id="a2">Monthly</input><input
type="radio" name="a" value="2"
id="a2">Yearly</input></font><br><font size = 3  color =
"#ffffff" >Type: <input type="radio" name="b" value="1"
id="b1">Public</input><input type="radio" name="b"
value="2" id="b2">On-Street</input><br><font size = 3
color = "#ffffff" >MM  /  DD /  YYYY </font>
<font size = 3  color = "#ffffff" >Arrive: </font> <input
name="" type="text"  style="width:48px;"/>/<input
name="" type="text" style="width:48px;"/>/<input
name="" type="text" style="width:100px;"/><br><font
size = 3  color = "#ffffff" >HH  :  MM </font><font size =
3  color = "#ffffff" >Arrive: </font><input name=""
type="text"  style="width:48px;"/>:<input name=""
type="text" style="width:48px;"/><br>
<font size = 3  color = "#ffffff" >MM  /  DD /
YYYY</font><font size = 3  color = "#ffffff"
>Depart:</font><input name="" type="text"
style="width:48px;"/><input name="" type="text"
style="width:48px;"/><input name="" type="text"
style="width:100px;"/><br><font size = 3  color = "#ffffff"
>HH  :  MM </font><font size = 3  color = "#ffffff"
>Arrive: </font><input name="" type="text"
style="width:48px;"/>:<input name="" type="text"
style="width:48px;"/><br><font size = 3  color = "#ffffff"
><button type="button"/>Search</button></font><br>
</div><div id ="map_canvas"></div></div><center><font
size = 2  color = "#ffffff" >This page is for teaching
purpose. The Department of Computer Science at Pace
University holds
copyright2014.</font></center></body></html>
```

Fig. 4: Interface Mainpage - Screenshot



References

[1] "1. Getting Started." *Safari*. Web. 27 Oct. 2014. https://www.safaribooksonline.com/library/view/building-android-apps/9781449327996/ch01.html

[2] "About White Plains." *White Plains, New York*. 27 October, 2014. http://www.cityofwhiteplains.com/index.aspx?nid=248

[3] "BestParking.com: Parking Finder | Find. Compare. Save."*BestParking.com: Parking Finder | Find. Compare. Save.* Web. 27 Oct. 2014. http://www.bestparking.com/

[4] "Best Practices for Speeding Up Your Web Site." *YAHOO! Developer Network.* YAHOO!, Web. https://developer.yahoo.com/performance/rules.html

[5] "CSS Sprites." *About Technology*. Web. 28 Oct. 2014. http://webdesign.about.com/od/advancedcss/ss/css_sprites_buttons.htm

[6] Dawson, Christian. *Projects in Computing and Information Systems: A Student's Guide – Second Edition*. England: Pearson Education Limited, 2009. Print.

[7] "Electric Vehicle Charging Stations in White Plains." *White Plains, New York*. 27 October, 2014. http://www.cityofwhiteplains.com/index.aspx?nid=368

[8] Johansson, Johan. "How To Make Your Websites Faster On Mobile Devices." *Smashing Magazine*. 3 April, 2013, 24 October, 2014.

[9] Kim, Seongwoon, Inseong Lee, Kiho Lee, Seungki Jung, Joonah Park, Yeun Bae Kim, Sang Ryong Kim, and Jinwoo Kim. "Mobile Web 2.0 with Multi-Display Buttons." *Communications of the ACM* 53.1 (2010): 136-141.

[10] "MapQuest Maps - Driving Directions - Map." *MapQuest*. Web. 27 Oct. 2014. http://www.mapquest.com/

[11] "Minimize HTTP Requests to Speed Up Page Load Times." *About Technology*. Web. 28 Oct. 2014. http://webdesign.about.com/od/speed/qt/minimize-http-requests.htm

[12] Nicolau, Alex. "Best Practices on the Move: Building Web Apps for Mobile Devices." *Communications of the ACM* 56.8 (2013): 45-51. Web. http://queue.acm.org/detail.cfm?id=2507894

[13] "Packet." *COMPUTER Hope*. Web. 28 Oct. 2014. http://www.computerhope.com/jargon/p/packet.htm

[14] "Round-trip Time (RTT)." *TechTarget*. Web. 28 Oct. 2014. http://searchnetworking.techtarget.com/definition/round-trip-time