

My Real Estate Web-Application

Andrés Garrote Aparicio

Computer Engineering

Jeffrey L. Duffany, Ph.D.

Electrical and Computer Engineering and Computer Science Department

Polytechnic University of Puerto Rico

Abstract — *This report is focused on explaining the aim of the My Real Estate web application. Details about the decisions made about the design and development of the tool with the capability to integrate the different tasks that are generally done when buying a property will be discussed. This paper is aimed to the end user who can be the system administrator or future users of the application. Real estate is at a key moment for buyers and properties analysis and comparisons are a key factor to distinguish between a good deal and a bad one. The objective of this report is to demonstrate how de developed application facilitates the monitoring of the real estate market and the decision-making when buying a property.*

Key Terms — *AJAX, Real Estate, Web-Application, Web Scraping*

INTRODUCTION

The economy on these days is not good and does not appear to be getting better, but this is not a bad sign for everyone. Right now might not be the best moment to sale a house but it is the best moment for people to buy one. Not only the real estate prices are down but also the interest are low which make a great combination to buy a property, it is possible to buy a better property cheaper than it would have been a few years ago. Technology is helping people to find their dream home; the Internet has change the real estate market it has leveled the playing field. The Internet has given homeowners a way to compete with realtors, since it gives them a great opportunity to easily give free exposure to their property.

Exposure is a great thing for the sellers and buyers but it also generates new challenges for the buyers. In a few hours a homebuyer can see hundreds of properties but only a few of them really gather the buyers attention. To remember this

properties can be some what difficult, this becomes even harder when it is taken into consideration that those houses that generated interest on the buyer might not be on the same website. When someone sees a property physically it's easier to remember the details of the house, the location, the prices, specs and monthly fees than it is when seeing so many online. The web-application discussed in this document tries to solve this issue by providing the user a way to store all the information of the houses the user under the same application. This web application let's the user save the all the detail of the houses that gathered his attention with just a few clicks of the mouse.

MY REAL ESTATE WEB-APPLICATION

My Real Estate Web-Application is a tool specifically designed and developed to facilitate the real estate data recompilation, comparison and analysis. As part of the design several approaches where taken into consideration. All of the approaches needed to provide the user with a way to store the important details of properties, such as: location, city, zip code, space, price, description and monthly payment. Al of the approaches also needed to provide a mechanism for the user to see and analyze all the stored data, all of this while requiring as little input as possible.

The first design idea was to make a website where the user could login and after login in allowing the user to go to different real estate websites to view the properties. As the user is browsing through the real estate websites the web-application would be storing all the details of the properties the user is viewing. This approach was discarded because the web-application will store data the user does not necessarily wants, since the user does not necessarily likes all the properties he

sees. This approach didn't have a way of determining if the user liked the property.

The second design idea was to create a toolbar that is embedded to the website, this way the user might provide inputs about which properties he had interest on. This idea solved the deficiency the first design had, but also had a flaw. To make this embedded toolbar work the user will need to modify it's browsing habits, since for the embedded toolbar to work the real estate websites need to be accessed through the web-application. The user needs to login to the web-application and then click a link for the real estate website, if the user want's to change websites he would have to navigate back to the web-application an click the link of the desired real estate website. If the user writes the real estate website address on the address bar the embedded bar will disappear. If a user needs to change its browsing habits its more likely that he won't use the application because it doesn't adapt to him.

The third and final design idea was to make a browser dependent toolbar, this convers all the flaws the previews ideas had. This browser dependent bar can gather the user input and does not require the user to modify its browsing habits. This decision comes with a cost, for the web-application to reach a greater market several toolbars will need to be designed, developed and maintained for different browsers (since this project it's a proof of concept the toolbar will only be developed for the Firefox web-browser which is available for all the major operating systems). The toolbar will provide a way for the user to login and notify when he or she likes a property for the web-application to store it. The toolbar itself only provides a way for the user to make an input to notify he has interest on a property. A tool has to be made to scrape (extract) all the information about the property. Also a website has to be created for the user to view all the data stored.

TOOLBAR

The toolbar is one of the major components of the web-application; its design was made to be as simple as possible. The toolbar main job is to provide the users a mechanism to notify when they would like to store a property. The toolbar has four buttons, the first button is the home button, which redirect the users to the web-application where he can see the properties that he has stored. The second button is a login button, which enables the user to logon to its account without having to go to the website. When the user logs on, the login button disappears and the third and fourth button appears (as shown on Figure 1). The third button is the like button, which the user presses every time he likes a property and wants to store it. The fourth button is the logout button that the user clicks to logout of his/hers account.



Figure 1
Firefox Toolbar on Logged in Sate

When the login button is pressed a window pops out displaying a dialog where the user types its username and password. Before submitting the credentials the user has the option to save the credentials, if this option is selected the username and MD5 hashed password will be saved on Firefox. The hash prevents other users from seeing the real password if they where looking through Firefox preferences. Thanks to the use of sessions on the authentication method if the users logs in using the toolbar, when the user enters My Real Estate website the website will recognize the user and will redirect the user to the main page instead of to the login page.

When the like button is pressed it validates that the user is logged in, in case that the user is not logged in the login pop up is showed. Once the user is logged in, the address of the current page is sent to the scraper.

The logout button when pressed logs out the user, form the toolbar and website. After logging out credentials will need to be provided in order to access the website once more.

WEB SCRAPER

The web scraper is the second major component of the web-application; Web Scraping refers to an application that processes the HTML of a Web page to extract data for manipulation such as converting the Web page to another format [1]. Just as the definitions suggest the web scraper job is to get the HTML code (Hypertext Markup Language) of the website sent through the toolbar using AJAX (Asynchronous JavaScript And XML) and stores it on a MySQL database, where the data is later retrieved by the website and displayed in other format.

The language selected to develop the scraper and website was Perl, known as the Swiss Army chainsaw of scripting languages: powerful and adaptable [2]. Perl was selected for its power for manipulating strings given by regular expressions and for libraries such Mechanize for viewing websites and DBD MySQL for storing and retrieving data on MySQL database. The Figure 2 shown bellow illustrates an extract of the scrape code, this specific extract shows how the HTML code is extracted and how regular expressions are used to extract several specifications of the property from the HTML code.

```
my $scrape = WWW::Mechanize->new();
my $link = $POST{'id'};
$scrape ->get($link);
my $source = $scrape->content;

my $pueblo;
my $urb;
my $zipCode;
my $temp = $source;
if($temp =~ m/<title>(.*?)\s*(.*?)\s*PR\s*(\d+)?\s*\s*zillow</title>/gs)
{
    $urb = $1;
    $pueblo = $2;
    $zipCode = $3;
    $urb =~ s/'//gs;
    $pueblo =~ s/'//gs;
    $zipCode =~ s/'//gs;
}
```

Figure 2
Scraper Code Extract

All websites have different structures and HTML codes, for this reason a different script has to be developed for each website. To reduce the scope of this project the scraper only extracts data from two very different websites.

The first website supported is Clasificados Online, the most popular real state website in Puerto Rico. This website isn't well structured and provided a challenge when extracting the data. The second supported website is Zillow which a Real Estate website from the United States. Zillow is a very well structured website and uses technologies newer than the ones used by Clasificados Online. Both of this websites worked great for the project since they help showcase the power of the Perl scraping tool, which is shown to work on modern and structured websites and old and less structured websites. Although Clasificados Online isn't well structured, all websites have some kind of structure since they all are based on HTML that has a structure itself, which is leveraged by the scraper.

DATABASE

As mentioned earlier in the document the database used for this project is MySQL. This database was selected because it's free, Perl can easily write to it and read from it and MySQL allows relational tables, which make sense to use for the application. A typical relational database has anywhere from 10 to more than 1,000 tables. Each table contains a column or columns that other tables can key on to gather information from that table. A typical large database will contain hundreds or thousands of tables like this all used together to quickly find the exact information needed at any given time. [3]. The database diagram used for the project can be seen bellow in Figure 3.

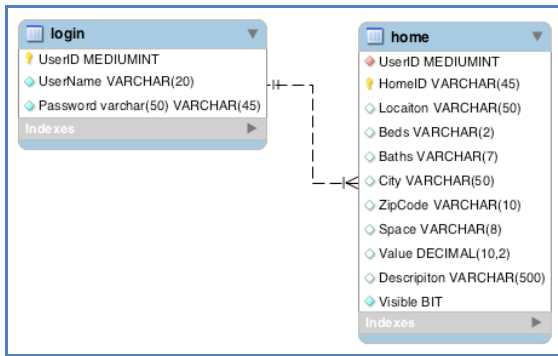


Figure 3
Database Diagram

As seen on Figure 3 the database for this project consists only of two tables, but it made sense to create a relational model instead of just one table. By creating a relational model the username and password are just stored in one record on the login table, without a relational database the user name and password would be repeated many times on the home table. Besides of the user name and password being repeated, a relational database gives flexibility and allows the website to grow without restrictions. For example in a future a table of messages can be added and it can be related to the home table by its primary key (HomeID), this message table could contain comments the user enters about the property. Other table can be created to store pictures, or picture links and can also be linked to the home table by its primary key (HomeID). For this to happen the scraper might be edited, with a few libraries more and a few lines of code, the pictures of the houses might be scraped and stored on the My Real Estate website. These two scenarios are quite real and the database structure will easily allow them to happen.

THE WEBSITE

My Real Estate website was designed and developed with simplicity and performance in mind, its made only of two pages, the login page and the properties page. The login page can be seen below on Figure 4; it has two fields, user name and password and two buttons, one for registering and one for login in. If the credentials are correct, the page will redirect the user to the property page, a message will appear. If the user logs through the toolbar he/she will be redirected to the property page, without having to enter the user name and password again. If the user takes advantage of the toolbar and saves his credentials on the toolbar, he will never have the necessity of entering a user name and password or using this login page. To improve the performance of the login page all of the login validations are made using AJAX. AJAX allows pages to request small bits of information from the server instead of entire pages. This incremental updating of pages eliminates the page refresh problem and slow response that have plagued Web applications since their inception [4].

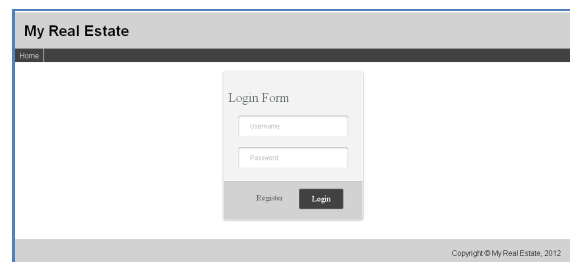


Figure 4
Login Page

The second page is the property page, which is the main page, where the users will spend most of their time. As seen below in Figure 5 the property page is composed of two main objects, a grid (shown at the left of Figure 5) and the mortgage settings (shown at the right of Figure 5).

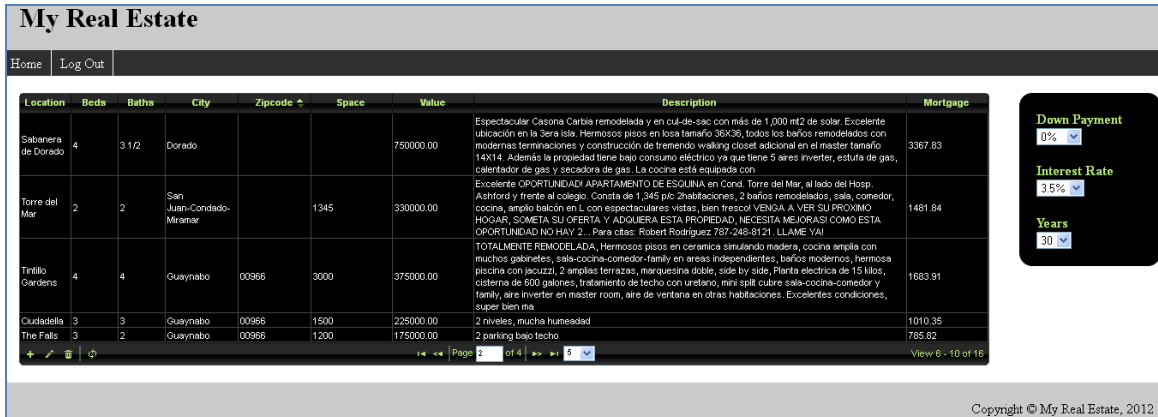


Figure 5
Property Page

The grid displays all of the data of the properties that he user liked using the toolbar and that the scraper extracted. The grid used for this application was JQGrid that is made on JQuery, a JavaScript library that is freely available on the Internet. This grid was selected because of its versatility and performance. The grid main performance comes thanks to the use of AJAX to retrieve the data from the MySQL Database. To use this grid JavaScript libraries need to be inserted in to the HTML code of the website and some JQuery lines of code have to be inserted into the body of the page. The lines of code inserted are the grid itself, on those lines, the columns of the grid, pages, rows per pages, data types of columns, and other properties of the grid are specified. This HTML code controls how the grid will behave when the user is using it, but there are other components of the grid that run on the server side.

The server side of the grid was also done in Perl; the server side of the grid is a script that enables the retrieval, insertion, modification and removal of the data from the database. All of these cases had to be developed on the Perl script. JQGrid has three methods of retrieving the data; none of them access the data directly from the database. The method used for My Real Estate was XML, when the grid retrieves the data it calls a Perl Script that interacts with the MySQL database and returns the data formatted as XML. The XM is then

interpreted by the JQGrid and displayed as a grid. The Perl code responsible for generating the XML data for the grid can be sown bellow on Figure 6.

```
# we should set the appropriate header information. Do not forget this.
use CGI;
my $cgi = new CGI;
print $cgi->header("Content-type: text/xml;charset=utf-8");

my $s = "<?xml version='1.0' encoding='utf-8'?>";
$s .= "<rows>";
$s .= "<pages>". $page. "</pages>";
$s .= "<total>". $total_pages. "</total>";
$s .= "<records>". $count. "</records>";

# the actual query for the grid data
$sql = "select HomeID,Location,Beds,Baths,City,Zipcode,Space,Value,Description";
$stmt = $dbh->prepare($sql);
$stmt->execute;

# be sure to put text data in CDATA
while(@row = $stmt->fetchrow_array())
{
    $s .= "<row id='". $row[0]. "'>";
    $s .= "<cell>". $row[1]. "</cell>";
    $s .= "<cell>". $row[2]. "</cell>";
    $s .= "<cell>". $row[3]. "</cell>";
    $s .= "<cell>". $row[4]. "</cell>";
    $s .= "<cell>". $row[5]. "</cell>";
    $s .= "<cell>". $row[6]. "</cell>";
    $s .= "<cell>". $row[7]. "</cell>";
    $s .= "<cell>". $row[8]. "</cell>";
    $s .= "<cell></cell>";
    $s .= "</row>";
}
$s .= "</rows>";
```

Figure 6
XML Code Generator

JQGrid was selected mainly because of how the user may interact with it. I didn't want the page to be just a plain table showing the saved data. This grid provides de user with the capability to delete record, insert new records (if the user sees a house, or knows of a house that is for sale he can add it manually to the website and will be able to compare it with all the saved houses). Other futures the grid provides is sorting, the capability to select how many rows the user wants to be displayed, the

capability to change between pages on the grid and the capability to modify rows. Modifying rows is very important for this website since this websites depends on the data of other websites, and this data can be entered incorrectly or in the wrong fields (for example the value can be entered in the description instead of in the value field). By using the Edit Row function of JQGrid (show in Figure 7) the user can edit all the data of the property.



Figure 7
Edit Record

The grid also displays the Mortgage, which is calculated on the client side (the values are calculated instantly, no loading happens) using JQuery. Equation (1) [5] shown bellow is executed on the client side to display the mortgage of each property.

$$\frac{\left(\frac{\text{Price} - \text{DownPayment}}{12} * \frac{\text{InterestRate}}{100} \right)}{\left(1 - \left(1 + \frac{\text{InterestRate}}{100} \right)^{-1 * (\text{Years} * 12)} \right)} * 100 \quad (1)$$

The users can manipulate the values that affect the Mortgage by selecting different percent of Down Payment, Interest Rate and a different term of years. This can be done in the mortgage settings, as seen in Figure 8. Selecting a different value from

the drop down menu is the only action the user needs to perform to change the mortgage, after selecting a different value the Mortgage on the grid will change instantly.



Figure 8
Mortgage Settings

CONCLUSION

Having designed and developed My Real Estate Web-Application was a project in which I could use many techniques learned in class such as network security (hashes), communication protocols, programming languages and database design. This project was also challenging and interesting since it was made of three components, the toolbar, scraper and website. These components where extremely different and had there unique challenges and solutions. The solutions where complex and required the use of new technology such as AJAX, JQuery and the use of old one such as Perl which was created more than twenty-five (25) years ago.

The purpose of this project was to help users in the organization and analysis of real estate. During the previous months I have been looking at the Real Estate market and with so many choices and so many websites it was hard to remember all the details about the properties without mixing them together. It got worst when I needed to go and review the property because I forgot some details about them, on what website I saw the property.

With this Web-Application I have achieved the main goal of having all the information I need for the analysis of the properties in one single place, without passing too much trouble. During the testing phase of the Web-Application I used the application a lot since I have been looking for properties myself and the application served its purpose, after browsing the homes on the websites I

didn't even had to go to Excel to make analysis since the website provides all the information I need to consider the properties, even the mortgage calculations which I can change on real-time to see if I can afford the property.

In conclusion technology solves problems but also creates new ones, but that same technology with a good analysis and design puts you in a good position to solve these new problems. Although new technology is good and exiting we must not forget older technology since it could also provide solutions to our problems, not all problems can be resolved using the same concepts and technology and forcing them to do so might not bring the best results. Analyzing and merging technologies, using only the best tools, new or old, for the right work will yield amazing results.

REFERENCES

- [1] "What is Web Scraping?", Retrieved on Jan 16, 2013, http://www.webopedia.com/TERM/W/Web_Scraping.html
- [2] "Beginner's Introduction to Perl", Retrieved on Jan 16, 2013, <http://www.perl.com/pub/2000/10/begperl1.html>
- [3] "What are relational databases?", Retrieved on Jan 17, 2013, <http://computer.howstuffworks.com/question599.htm>
- [4] "Advantages of AJAX", Retrieved on Jan 18, 2013, <http://www.javabeat.net/qna/431-advantages-of-ajax/>
- [5] "JavaScript Mortgage Calculator", Retrieved on Jan 19, 2013, <http://www.mcfedries.com/creatingawebpage/mortgage.htm>