Spring 2019

Home Away From Home

Malcolm Washington
Governors State University

Home Away from Home

By

Malcolm Washington
A.S., Prairie State College, 2007
B.S., Governors State University, 2018

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University
University Park, IL 60484

2019

ABSTRACT

This project addresses the need for a housing market for the on the go person. It is important because there is a need for fully furnished convenient housing around the globe.

Our website will include an admin page and section. The admin will be able to view transactions, property listings, bookings, and ratings.

Owners accounts will include uploading their properties, set payment settings and ratings.

Renters accounts will include ratings, and surveys.

Fully furnished convenient housing is the perfect solution for business travelers, relocating employees, and project teams needing short or long-term housing around the globe. Whether you arrange temporary living for mobile employees or need temporary housing yourself, convenient housing delivers temporary living for the on the go person.

We are enhancing an existing service to address short-term and long-term housing. We are improving the convenient housing market through an easy process by offering a single source solution service platform.
We intend to build this web application within a window of January 2019 to May 2019 and release on May 8, 2019.

We will be building this project using C#, HTML, CSS JavaScript, jQuery, Bootstrap, AJAX, LINQ, XML, TSQL and MSSQL Server.

Table of Content

1 *Project Description*

A short-term rental, or vacation rental, is the renting out of a furnished home, apartment or condominium for a short-term stay. The owner of the property usually will rent out on a weekly basis, but some vacation rentals offer nightly rates. The owner of the property will rent their space when they are not using it - though some rentals are shared spaces - especially during peak holiday seasons like in December or for New Years and times over the summer including the Fourth of July when higher rates can be applied. It's also become common for people traveling for festivities like music festivals or sports events like the Super Bowl to rent houses or condos rather than pay for over-the-top expensive hotel rates. The short-term rental market has limitations in choices, so we decided to enter this market and add our input. We built a website that can make the process smoother by allowing a user to make a quick selection, pay and go without limitations.

2 *Project Technical Description*

The following are the pieces that are required to build and implement the project:

The project has 3 main databases for SQL database & programming [GLAccounts], [Vendors], [Terms]

//This is the [dbFiles] database which is the main database for the home page and property-owners

```
CREATE TABLE [dbo].[ GLAccounts](
	[AccountNo] [int] IDENTITY(1,1) NOT NULL,
	[AccountDescription] [varchar](50) NOT NULL,
	[ContentType] [nvarchar](200) NOT NULL,
 CONSTRAINT [PK_ GLAccounts] PRIMARY KEY CLUSTERED
```

//This is the [Vendors] database which is the main database for the vacationers-renters.

```
CREATE TABLE [dbo].[Vendors](
	[VendorID] [int] IDENTITY(1,1) NOT NULL,
	[VendorName] [varchar](50) NOT NULL,
	[VendorAddress2] [varchar](50) NOT NULL,
	[VendorAddress2] [nvarchar](100) NOT NULL,
	[VendorCity] [nvarchar](100) NOT NULL,
	[VendorState] [nvarchar](100) NOT NULL,
	[VendorZip] [nvarchar](10) NOT NULL,
	[VendorPhone] [nvarchar](100) NOT NULL,
	[VendorContactLName] [nvarchar](100) NOT NULL,
	[VendorContactFName] [datetime] NOT NULL,
	[DefaultTermsID] [datetime] NOT NULL,
	[DefaultAccountNo] [nvarchar](200) NOT NULL,
 CONSTRAINT [PK_Vendors] PRIMARY KEY CLUSTERED
```

//This is the [Terms] database which is the main database for the home page and property-owners.

```
CREATE TABLE [dbo].[tblComment](
	[TermsID] [int] IDENTITY(1,1) NOT NULL,
	[TermsDescription] [varchar](50) NOT NULL,
	[TermsDueDays] [nvarchar](200) NOT NULL,
 CONSTRAINT [PK_Terms] PRIMARY KEY CLUSTERED
```

A Models folder containing .cs files for database connectivity,

The database connectivity to the interface starts with implementing Category.cs.

This class contains the table names that can be referenced in the DB.cs class that handles select, insert and delete functions.

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;

public class Category
{
    public string id { get; set; }
    public string Name { get; set; }
    public string ContentType { get; set; }
    public string Data { get; set; }
    public string FileName { get; set; }
    public string Email { get; set; }
    public string City { get; set; }
    public string State { get; set; }
    public string Zip { get; set; }
    public string DateArrive { get; set; }
    public string DateDepart { get; set; }
    public string Guests { get; set; }
    public string Price { get; set; }
    public string Address { get; set; }
    public string Notes { get; set; }
    public string HousingType { get; set; }
    public string Rooms { get; set; }
    public string Baths { get; set; }
    public string SqFt { get; set; }
}
```

the DB.cs class that handles select, insert and delete functions is CategoryDB.cs

//This is the select area of the class

```
List<Category> categoryList = new List<Category>();
        string sql = "SELECT id, ContentType, Data, Name, FileName, Email, City, State, Zip,
Guests, Price, Address, Notes, HousingType, Rooms, Baths, SqFt "
            + "FROM tblFiles ORDER BY Name";
```

//insert

```
public static void InsertCategory(Category category)
    {
        string sql = "INSERT INTO tblFiles "
            + "(id, Name, ContentType, Data, FileName, Email, City, State, Zip, Guests, Price,
Address, Notes, HousingType, Rooms, Baths, SqFt) "
            + "VALUES (@id, @Name, @ContentType, @Data, @FileName, @Email, @City, @State, @Zip,
@Guests, @Price, @Address, @Notes, @HousingType, @Rooms, @Baths, @SqFt)";
```

//delete

```
public static int DeleteCategory(Category category)
    {
        int deleteCount = 0;
        string sql = "DELETE FROM tblFiles "
            + "WHERE id = @id "
            + "AND Name = @Name "
            + "AND ContentType = @ContentType "
            + "AND Data = @Data "
            + "AND Email = @Email "
            + "AND City = @City "
            + "AND State = @State "
            + "AND Zip = @Zip "
            + "AND Guests = @Guests "
            + "AND Price = @Price "
            + "AND Address = @Address "
            + "AND Notes = @Notes "
            + "AND HousingType = @HousingType "
```

```
            + "AND Rooms = @Rooms "
            + "AND Baths = @Baths "
            + "AND SqFt = @SqFt";
```

Database interface connectivity is handled by a data source object within each interface.

```
<asp:ObjectDataSource runat="server" ID="ObjectDataSource3"
                DataObjectTypeName="Category" DeleteMethod="DeleteCategory"
                InsertMethod="InsertCategory"
                OldValuesParameterFormatString="original_{0}"
                SelectMethod="GetCategories" TypeName="CategoryDB"
                UpdateMethod="UpdateCategory"
                ConflictDetection="CompareAllValues"
                OnDeleted="ObjectDataSource1_GetAffectedRows"
                OnUpdated="ObjectDataSource1_GetAffectedRows">
                <UpdateParameters>
                    <asp:Parameter Name="original_Category" Type="Object"></asp:Parameter>
                    <asp:Parameter Name="category" Type="Object"></asp:Parameter>
                </UpdateParameters></asp:ObjectDataSource>
```

We have some folders that contain some dot aspx files.
These files contain standard html5 and html4 and database connectivity,
They are listed as follow:
Three Account folders that contain pages for login accounts and management,
Login/registration management is handled by the dot net built in data structure.
A single Property/Owner folder containing property owner pages,
A single Vacationers folder containing vacationers' pages,
A Comments.aspx page,
A Default.aspx for home page,
A View.aspx for property query.

This is the E-R diagram of our database

Figure 1

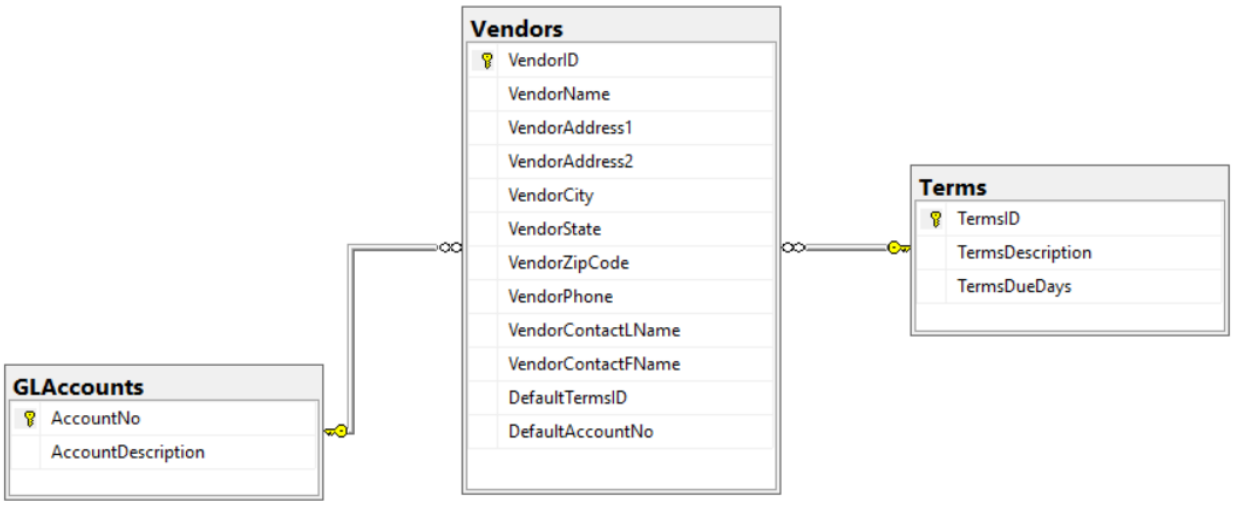
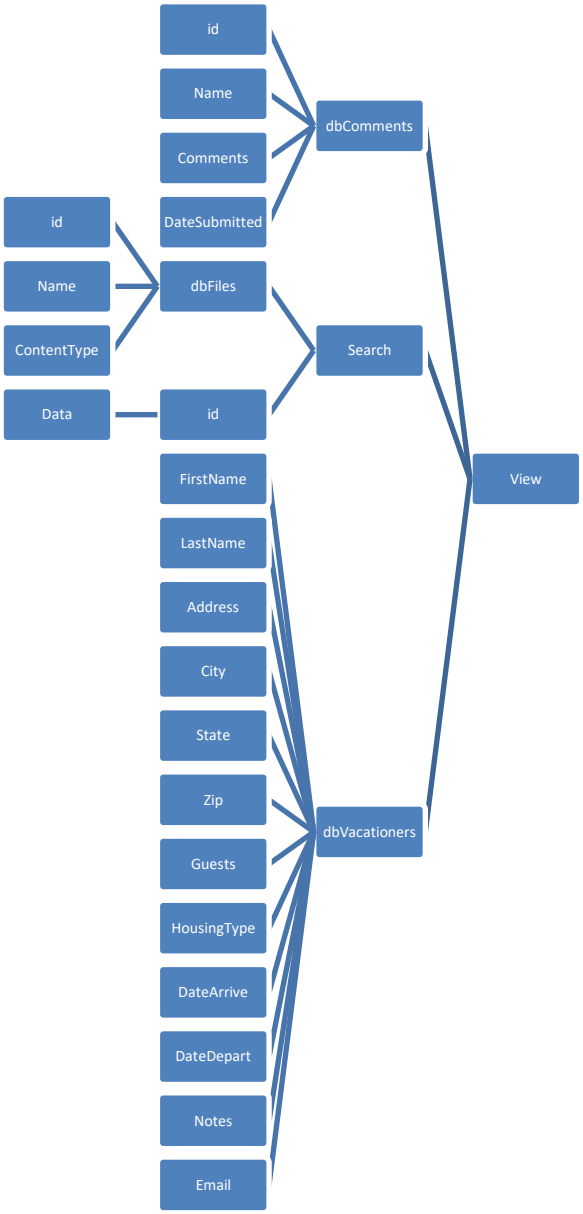


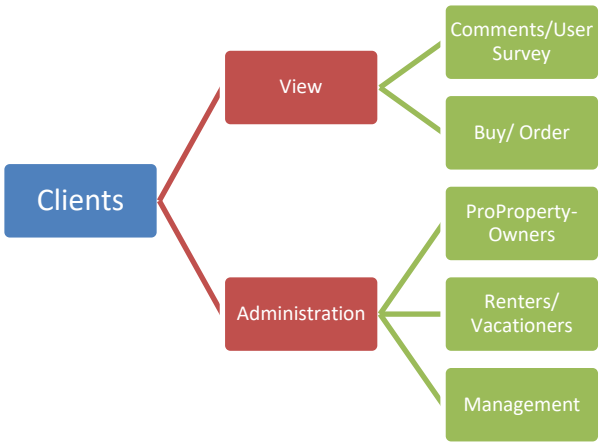
Figure 2



2.1 Application Architecture

This project is written in C# and using the .Net structure and follows this architecture flow. We rely on the dot net framework and below are the interactions of its components.

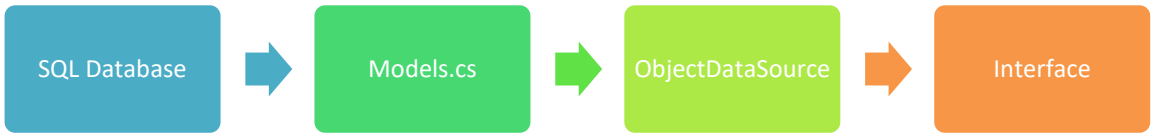
Figure 3



2.2 Application Information flows

Users can run this project from notepad or visual studio. The user must have the dot (.)NET Framework installed on their computer. To run in notepad, copy and paste the class into notepad. Save the file as a .cs or .aspx file and run the code from the default open as box at the top in the folder structure. To run in visual studio, click the debug tab and choose "start debugging" to run.

Figure 4



2.3 Capabilities

This project requires the dot (.)NET Framework installed on their computer.

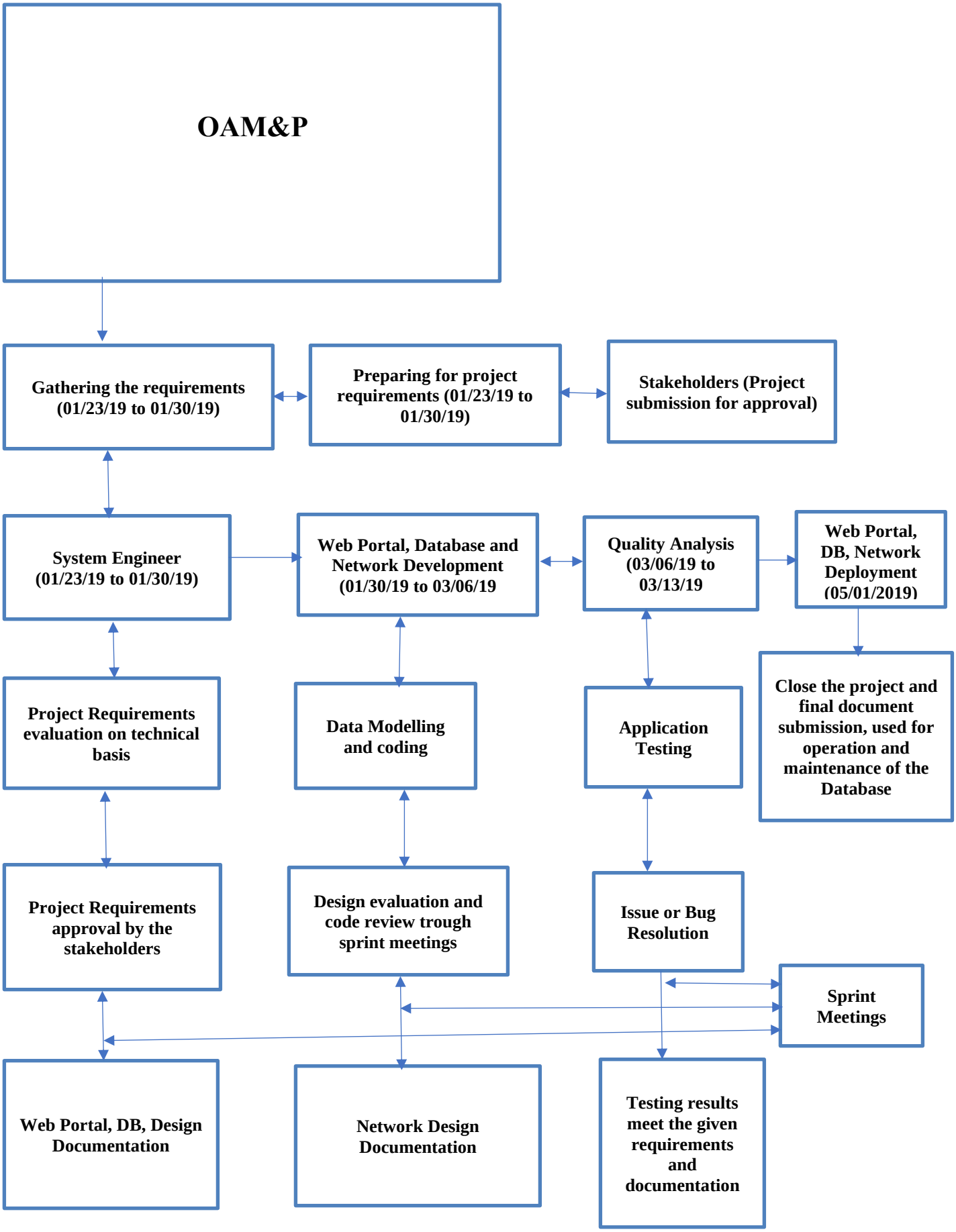
3 Project Requirements

3.1 Identification of Requirements

- **S**pecific – Allow images to be downloaded from a link. Allow images to be shared on social media from a link
- **M**easurable – Allow images to be downloaded from a link. Allow images to be shared on social media from a link and increase shared by 50%
- **A**chievable – Adding the correct code logic to add the extra functionality.
- **R**ealistic – Adding the new code is a subset of the code already in place.
- **T**ime-related – The results should be realized with two to three weeks.

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

This project OAM&P is outlined below.
Figure 5


OAM&P
Gathering the requirements (01/23/19 to 01/30/19)
Preparing for project requirements (01/23/19 to 01/30/19)
Stakeholders (Project submission for approval)
System Engineer (01/23/19 to 01/30/19)
Web Portal, Database and Network Development (01/30/19 to 03/06/19
Quality Analysis (03/06/19 to 03/13/19
Web Portal, DB, Network Deployment (05/01/2019)
Project Requirements evaluation on technical basis
Data Modelling and coding
Application Testing
Close the project and final document submission, used for operation and maintenance of the Database
Project Requirements approval by the stakeholders
Design evaluation and code review trough sprint meetings
Issue or Bug Resolution
Sprint Meetings
Web Portal, DB, Design Documentation
Network Design Documentation
Testing results meet the given requirements and documentation

3.3 Release and Transition Plan

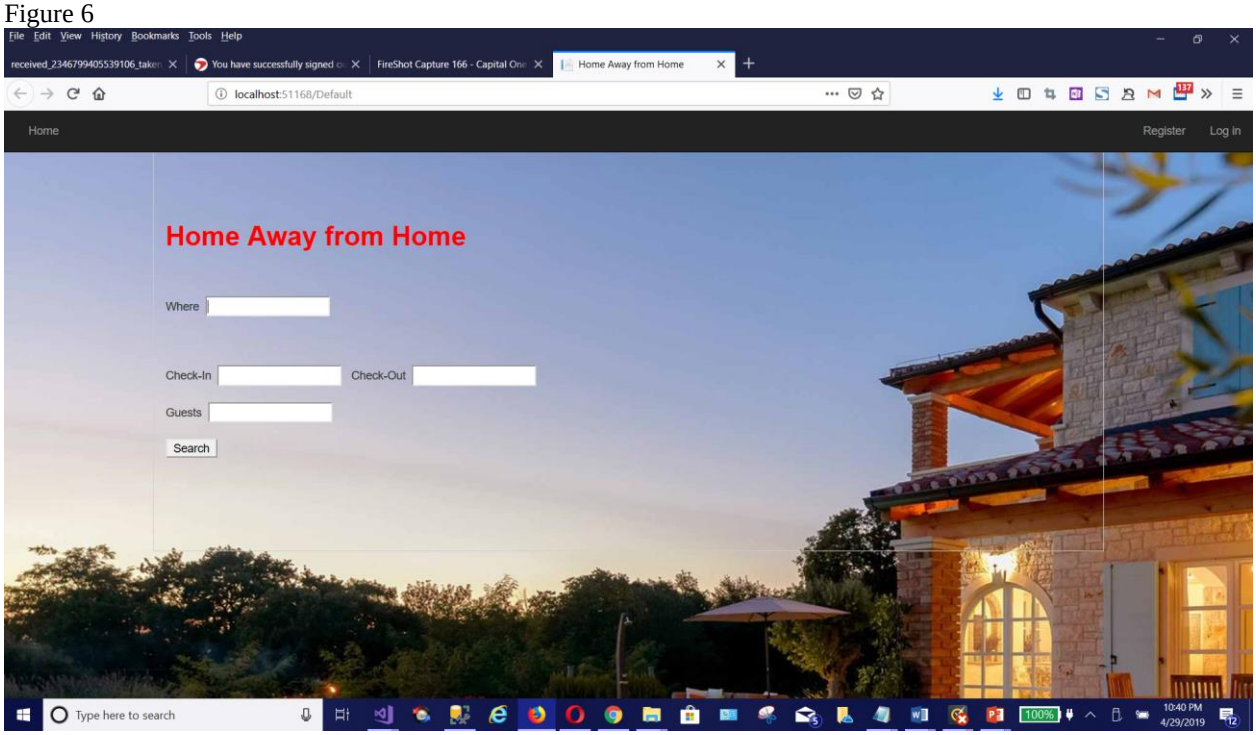
This project can be delivered via a website or can be packaged into a console program that can be hosted on a website and downloaded by a link to the user's computer.

4 Project Design Description

The following slides show what the design pieces are.

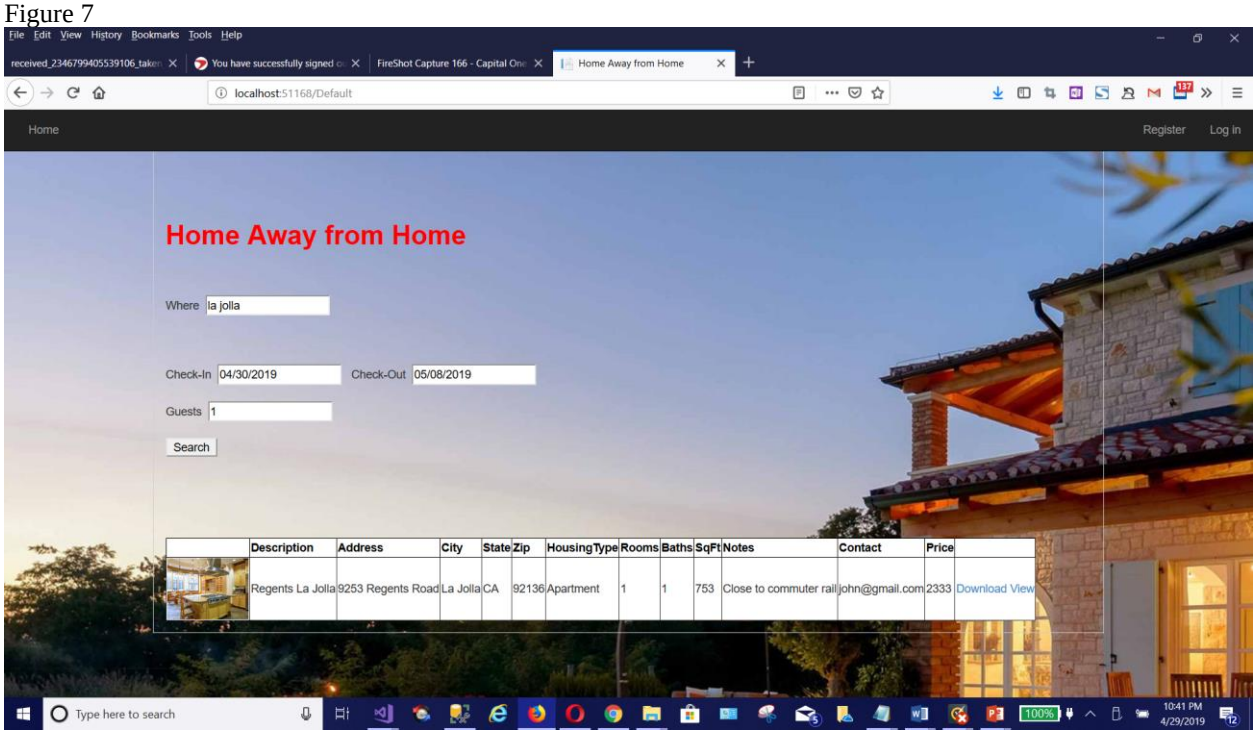
This shows our version front page to show with Search box.

Figure 6



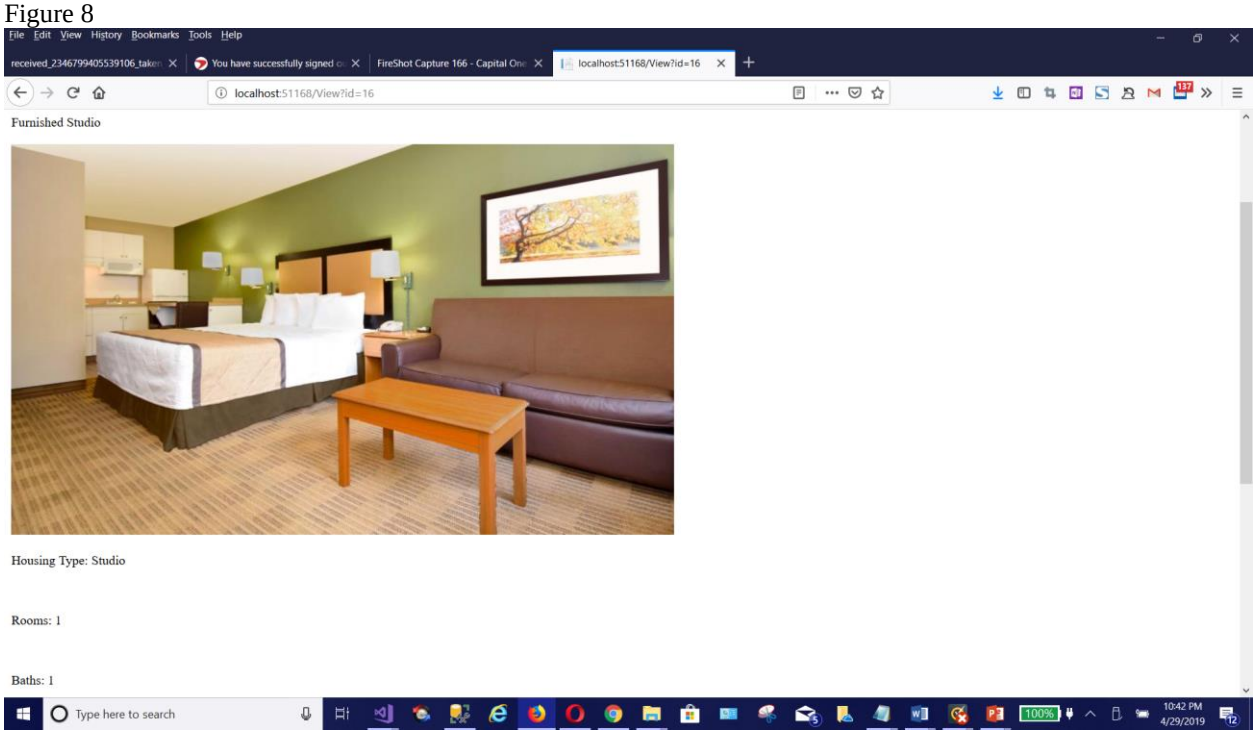
This shows our version front page to show different sections based on types of properties and search result listings showing hyperlinked thumbnails of properties.

Figure 7



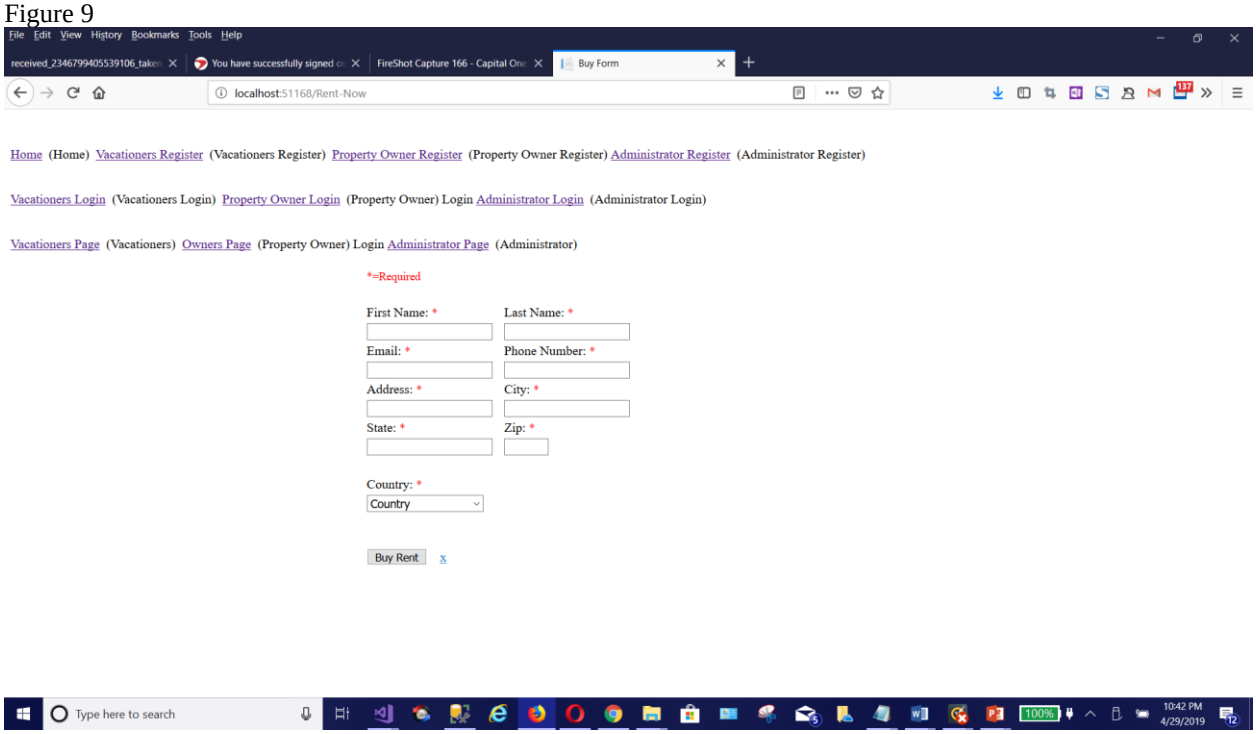
This shows our version of property page showing photos and full details of properties, availability, amenities, etc.

Figure 8



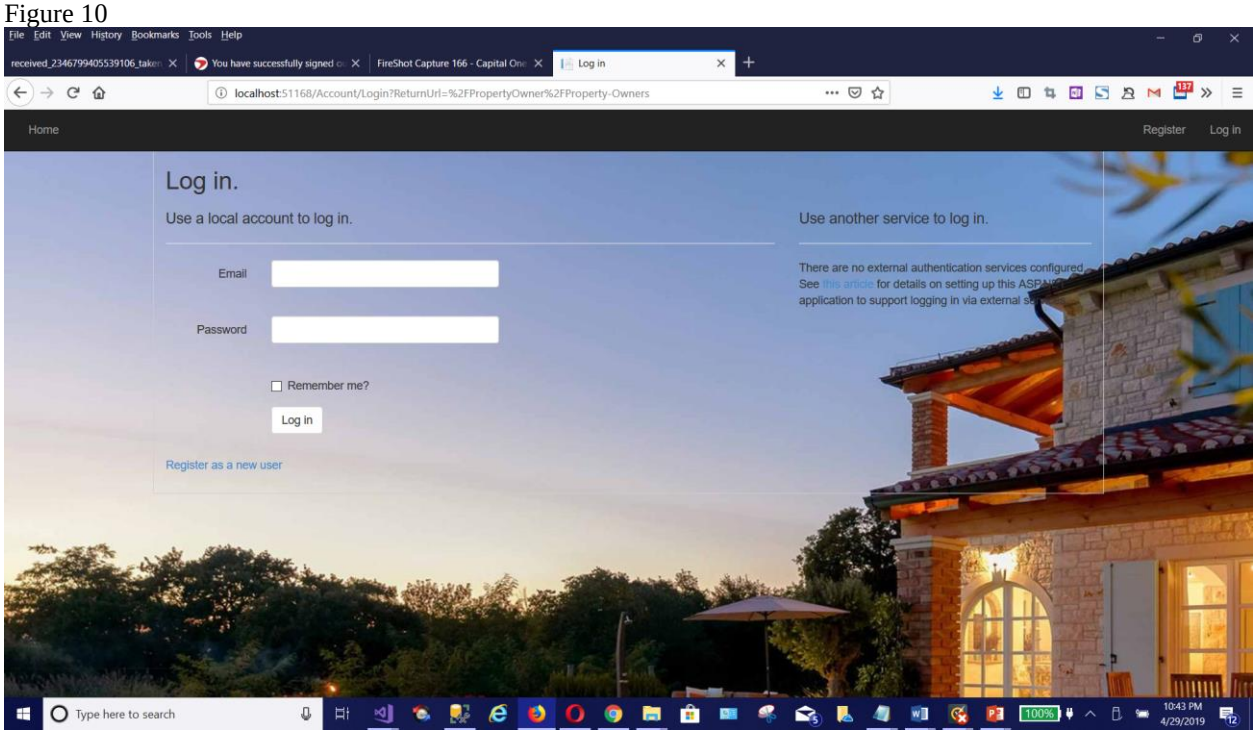
This shows our version of a booking form and payment and process.

Figure 9



This shows a version of the 3 types of accounts: vacationers, property owners and administrator.
An Owner console for managing their properties and renters.
An Admin console for managing properties and accounts, etc.
These features are accessed by using the login to access these areas of the website.

Figure 10



This shows our version of User survey and comment.

Figure 11