CHAPTER-1

INTRODUCTION

1.1 PURPOSE

Real estate appraisal, which is the process of estimating the price for real estate properties, is crucial for both buys and sellers as the basis for negotiation and transaction. Real estate plays a vital role in all aspects of our contemporary society. In a report published by the European Public.

Real Estate Association , it was shown that real estate in all its forms accounts for nearly 20% of the economic activity. Therefore, accurate prediction of real estate prices or the trends of real estate prices help governments and companies make informed decisions. On the other hand, for most of the working class, housing has been one of the largest expenses. A right decision on a house, which heavily depends on their judgement on the value of the property, can possibly help them save money or even make profits from their investment in their homes. From this perspective, real estate appraisal is also closely related to people's lives. Current research from both estate industry and academia has reached the conclusion that real estate value is closely related to property infrastructure, traffic, online user reviews and so on. Generally speaking, there are several different types of appraisal values. In particular, we are interested in the market value, which refers to the trade price in a competitive auction setting.

Today, people are likely to trade through real estate brokers, who provide easy access online websites for browsing real estate property in an interactive and convenient way. Fig. 1 shows an example of house listing from Realtor (http://www.realtor.com/), which is the largest real estate broker in North America. From the figure, we see that a typical piece of listing on a real estate property will introduce the infrastructure data in text for the house along with some pictures of the house. Typically, a buyer will

look at those pictures to obtain a general idea of the overall property in a selected area before making his next move.

1.2 SCOPE

We intend to employ the pictures for the task of real estate price estimation. We want to know whether visual features, which is a reflection of a real estate property, can help estimate the real estate price. Intuitively, if visual features can characterize a property in a way similar to human beings, we should be able to quantify the house features using those visual responses. Meanwhile, real estate properties are closely related to the neighborhood. In this work, we develop algorithms which only rely on 1) the neighbor information and 2) the attributes from pictures to estimate real estate property price. To preserve the local relation among properties we employ a novel approach, which employs random walks to generate house sequences. In building the random walk graph, only the locations of houses are utilized. In this way, the problem of real estate appraisal has been transformed into a sequence learning problem. Recurrent Neural Network (RNN) is particularly designed to solve sequence related problems. Recently, RNNs have been successfully applied to challenging tasks including machine translation, image captioning and speech recognition. Inspired by the success of RNN, we deploy RNN to learn regression models on the transformed problem.

CHAPTER-2

FEASIBILITY STUDY

2.1 INTRODUCTION

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.1.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user.

CHAPTER-3

SYSTEM ANALYSIS

3.1 INTRODUCTION

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

3.2 EXISTING SYSTEM

Current research from both estate industry and academia has reached the conclusion that real estate value is closely related to property infrastructure, traffic, online user reviews and so on. Generally speaking, there are several different types of appraisal values. In particular, we are interested in the market value, which refers to the trade price in a competitive auction setting Traditionally, both real estate industry professionals and researchers have relied on a number of factors, such as economic index, house age, history trade and neighborhood environment and so on to estimate the price. Indeed, these factors have been proved to be related to the house price, which is quite difficult to estimate and sensitive to many different human activities. The current algorithms are 1). Regression Models and 2). Deep Walk. Regression model has been employed to analyze real estate price index. Recently, the results in Fu et al. show that sparse regularization can obtain better performance in real estate ranking. Thus, we choose to use LASSO which is an l1-constrained regression model, as one of our baseline algorithms. Deep Walk is another way of employing random walks for unsupervised feature learning of graphs. The main approach is inspired by distributed word representation learning. In using Deep Walk, we also use _-neighborhood graph with the same settings with the graph we built for generating sequences for B-LSTM. The learned

features are also fed into a LASSO model for learning the regression weights. Indeed, deep walk can be thought as a simpler version of our algorithm, where only the graph structures are employed to learn features. Our framework can employ both the graph structure and other features, i.e. visual attributes, for building regression model.

DRAWBACKS OF EXISTING SYSTEM

- The existing system is quite difficult to estimate and sensitive to many different human activities. There are lot of difficult works have been done with the existing systems to measure the number of factors such as economic index, house age, history trade and neighborhood environment.
- Current research from both estate industry and academia has reached the conclusion that real estate value is closely related to property infrastructure, traffic online user Reviews and so on.

3.3 PROPOSED SYSTEM

We intend to employ the pictures for the task of real estate price estimation. We want to know whether visual features, which are a reflection of a real estate property, can help estimate the real estate price. Intuitively, if visual features can characterize a property in a way similar to human beings, we should be able to quantify the house features using those visual responses. Meanwhile, real estate properties are closely related to the neighborhood. In this work, we develop algorithms which only rely on 1) the neighbor information and 2) the attributes from pictures to estimate real estate property price To preserve the local relation among properties we employ a novel approach, which employs random walks to generate house sequences. In building the random walk graph, only the locations of houses are utilized. In this way, the problem of real estate appraisal has been

transformed into a sequence learning problem. Recurrent Neural Network (RNN) is particularly designed to solve sequence related problems. Recently, RNNs have been successfully applied to challenging tasks including machine translation, image captioning, and speech recognition. Inspired by the success of RNN, we deploy RNN to learn regression models on the transformed problem. The main contributions of our work are as follows: To the best of our knowledge, we are the first to quantify the impact of visual content on real estate price estimation. We attribute the possibility of our work to the newly designed computer vision algorithms, in particular Convolutional Neural Networks (CNNs). We employ random walks to generate house sequences according to the locations of each house. In this way, we are able to transform the problem into a novel sequence prediction problem, which is able to preserve the relation among houses. We employ the novel Recurrent Neural Networks (RNNs) to predict real estate properties and achieve accurate results.

ADVANTAGE:

- A picture is worth a thousand words. One advantage with images and videos is that they act like universal languages. For the given house pictures, people can easily have an overall feeling of the house, e.g. what is the overall construction style, how the neighboring environment looks like. These high-level attributes are difficult to be quantitatively described
- Map Based Location information are most commonly effective than the viewing in raw details. The most accurate details can be viewed in simple steps
- The proposed algorithms are very effective than the existing algorithms such as LASSO and Deep Walk.

3.4 USER REQUIREMENTS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well order and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

3.5 REQUIREMENT SPECIFICATION

Functional Requirements

Graphical User interface with the User.

Requirements:

1. Python
2. Django

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

3.6 MODULES DESCRIPTION

3.6.1 MODULES

In this project there are four modules present as listed in the below

- Property module
- Adding location details
- Price negotiation
- Geometric analysis

PROPERTY ADDITION

The property addition is the main initiative module for the project.Once authorized user login into the system, they can perform their activity as per their wish. In this module, User must have interested in selling the property which they own. The Property details such as Location, Address, and Facilities that the households are need to add to the cloud where everything that seller uploads can viewable to buyer and agent.

ADDING LOCATION DETAILS

In this module user that is seller need to upload the details of their location as well as their neighboring facility location such as schools, colleges and medical etc., In previous modules also user need to add the location that are into the raw typed format but here in this module we can upload the location details in maps and map formats. Spotting these locations can be very handy for agents or users to get to know about the details of property and neighboring details.

PRICE NEGOTIATION

This module is mainly designed for buyers and agents. Firstly, buyer sends the request to agents along with the cost of expectations and other query details about property. Once agents view the request from the buyer, Agent can decide the price according to the merit of location and both the buyer and seller. This module designed like chat. Dual way communication can be accomplished among the various users.

GEOMETRICAL ANALYSIS

The Geometrical analysis of given data set is done by charts. Here in this project there are two graphs have been plot between numbers of locations versus city. The pie chart and line charts are established in this project in order to analysis the data effectively.

3.6.2 INPUT AND OUTPUT REPRESENTATION

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objective.

CHAPTER-4

SYSTEM ARCHITECTURE

4.1 INTRODUCTION

A **system architecture** or **systems architecture** is the conceptual model that defines the structure, behaviour and more views of a system.[1] An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages

4.2 URL PATTERN

ARCHITECTURE

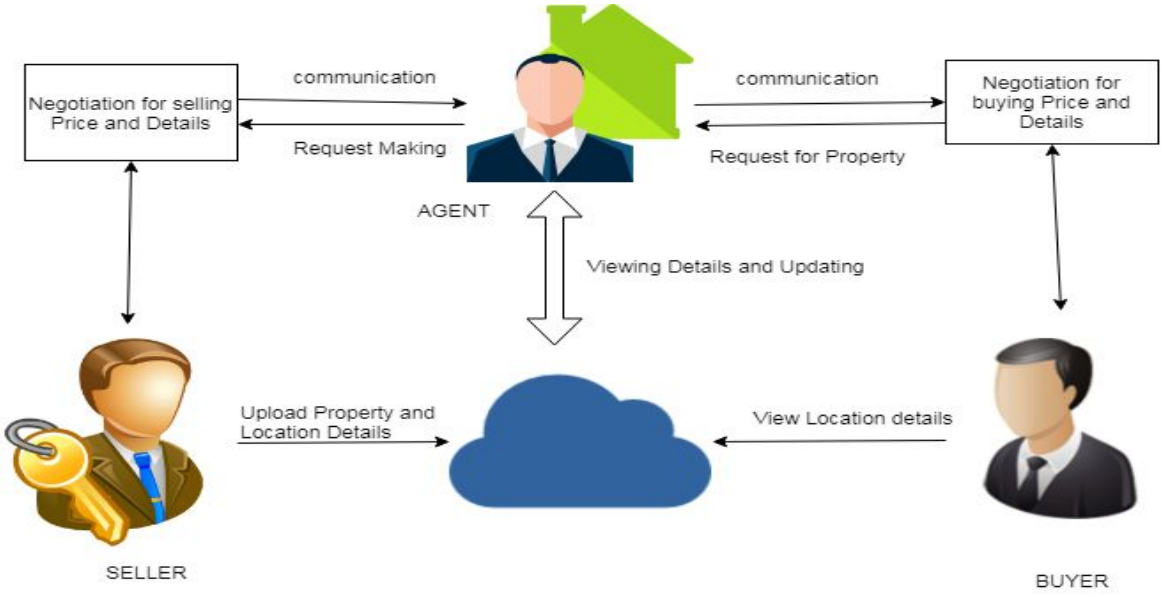


FIG NO. 4.2 ARCHITECTURAL DIAGRAM

4.3 SOFTWARE REQUIREMENT SPECIFICATION

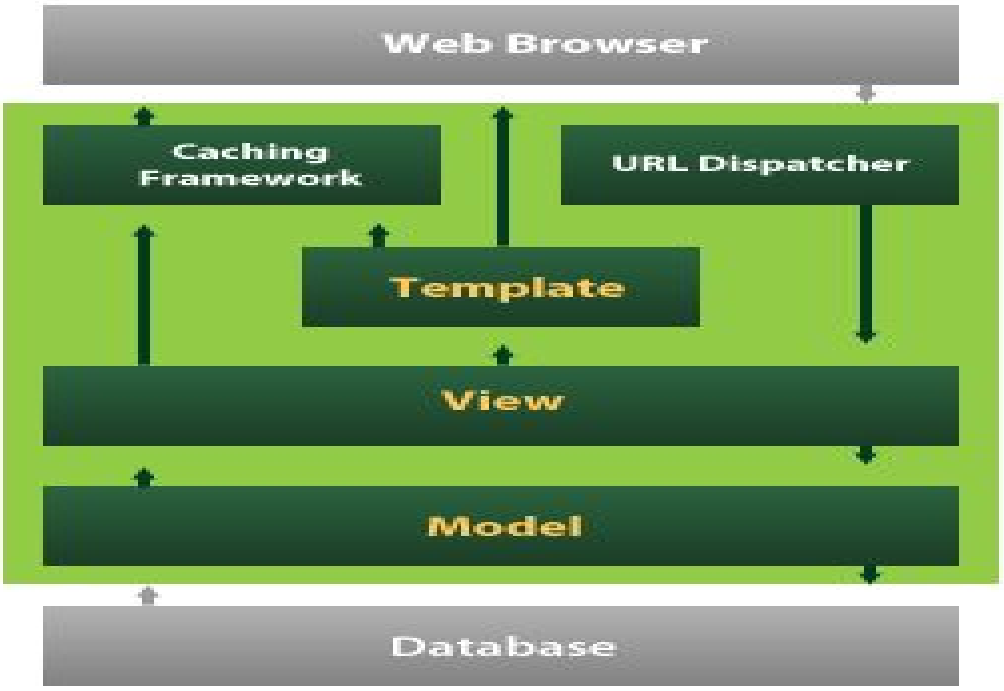
PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library

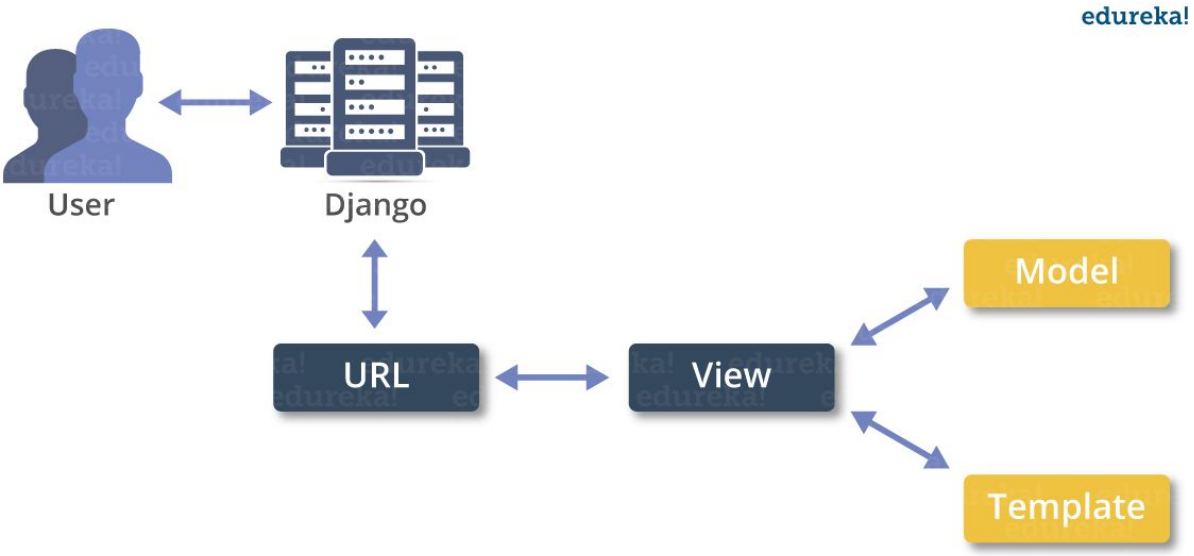
DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models



4.4 FUNCTIONAL REQUIREMENTS

Functional software requirements help you to capture the intended behaviour of the system. This behaviour may be expressed as functions, services or tasks or which system is in software engineering, a functional requirement defines a system or its component. A function is nothing but

inputs, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification.

4.4.1 SOFTWARE REQUIREMENTS

A software requirements specification (SRS) is a description of a software system to be developed. It is modeled after business requirements specification (CONOPS), also known as a stakeholder requirements specification. It should also provide a realistic basis for estimating product costs, risks, and schedules.

- **Operating system** : Windows 7 Ultimate.
- **Coding Language** : Python.
- **Front-End** : Python.
- **Designing** : Html, CSS, Java script.
- **Data Base** : MySQL (WAMP Server).

4.4.2 HARDWARE REQUIREMENTS

- Processor (CPU) with 2 gigahertz (GHz) frequency or above
- A minimum of 2 GB of RAM
- Monitor Resolution 1024 X 768 or higher
- A minimum of 20 GB of available space on the hard disk
- Internet Connection Broadband (high-speed) Internet connection with a speed of 4 Mbps or higher
- Keyboard and a Microsoft Mouse or some other compatible pointing device
- Sound card
- Speakers or headphone.

CHAPTER-5

SYSTEM DESIGN

5.1 UML DIAGRAM

5.1.1COMPONENT DIAGRAM

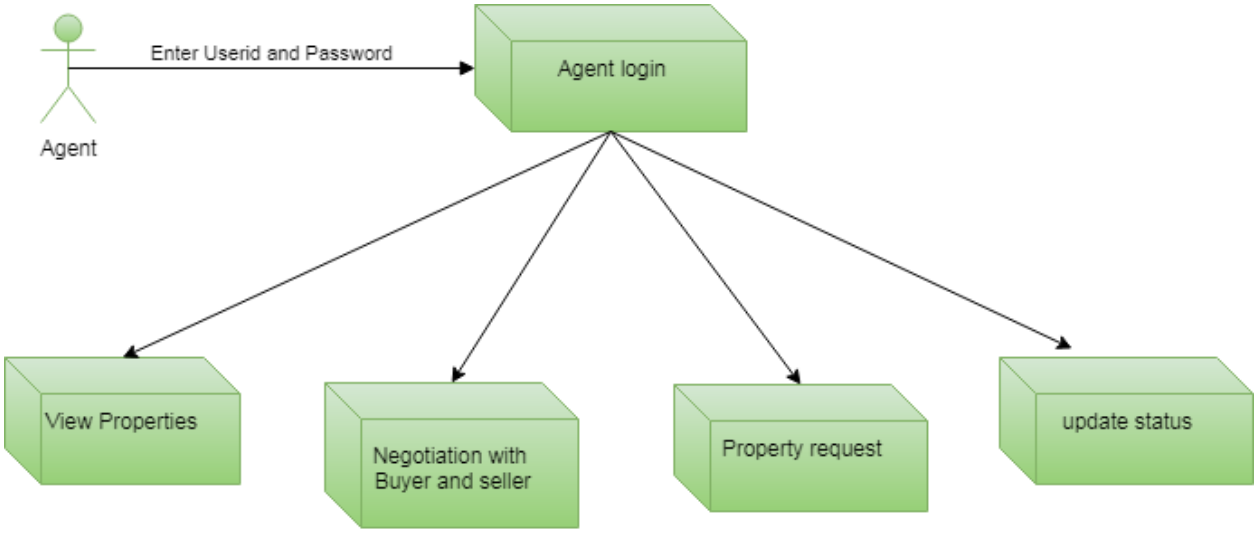


FIG 5.1.1 a) COMPONENT DIAGRAM OF AGENT

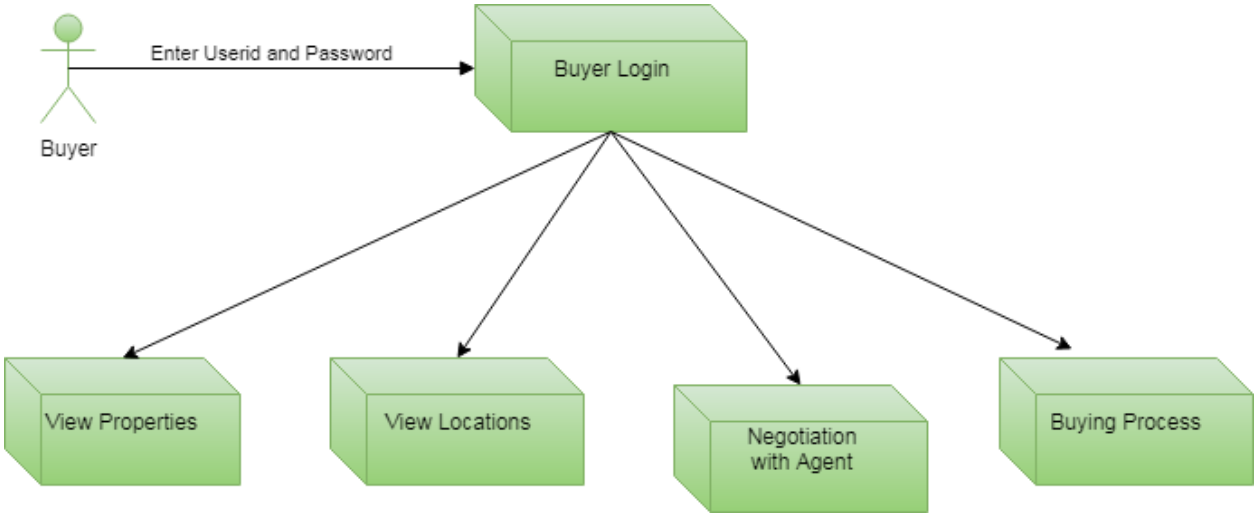


FIG 5.1.1 b) COMPONENT DIAGRAM OF BUYER

5.1.2 ER DIAGRAM

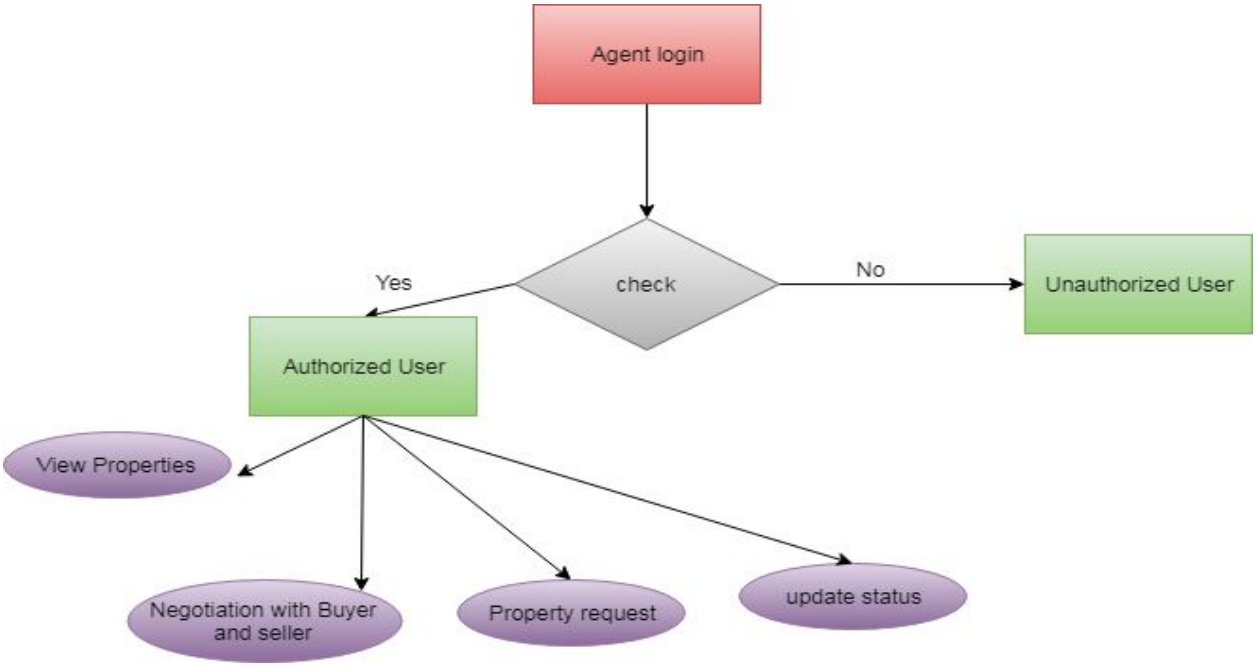


Fig 5.1.2 a) ER Diagram for Agent

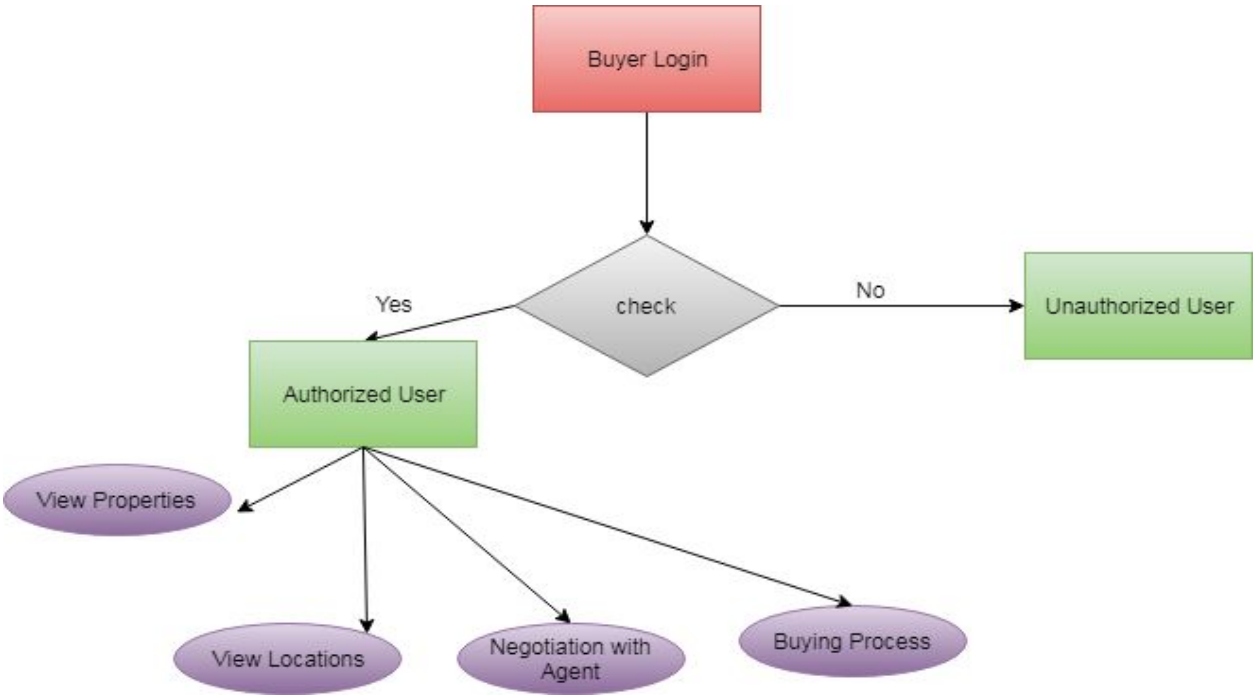


Fig 5.1.2 b) ER Diagram for Buyer

5.2 USECASE DIAGRAM

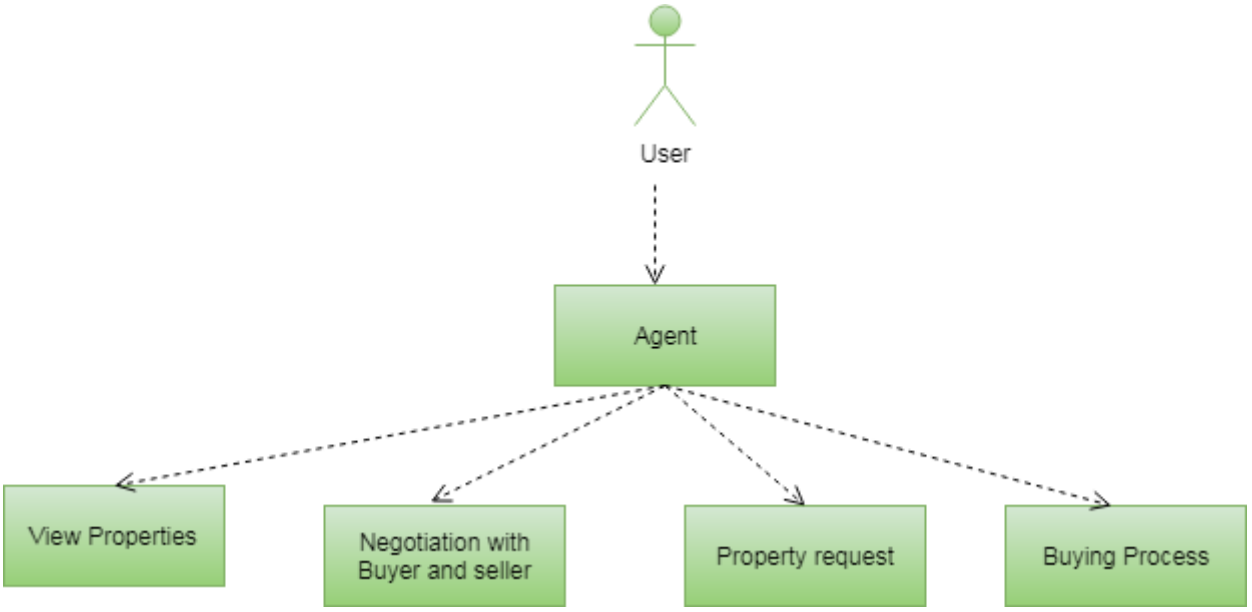


Fig 5.2.1 a)Usecase Diagram for Agent

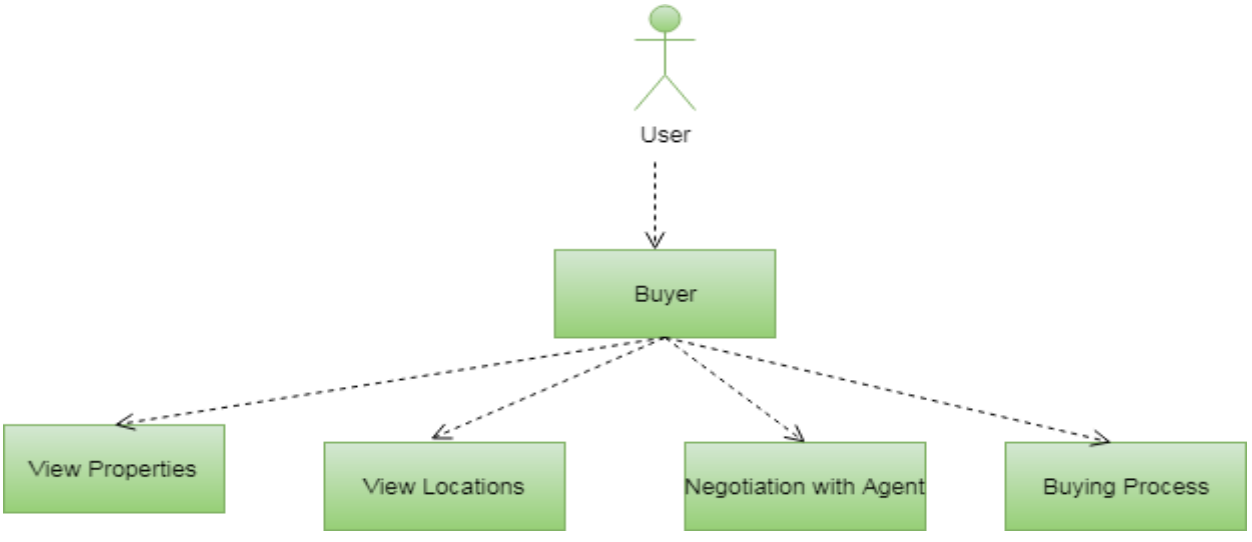


Fig 5.2.1 b)Usecase Diagram for Agent

5.3 CLASS DIAGRAM

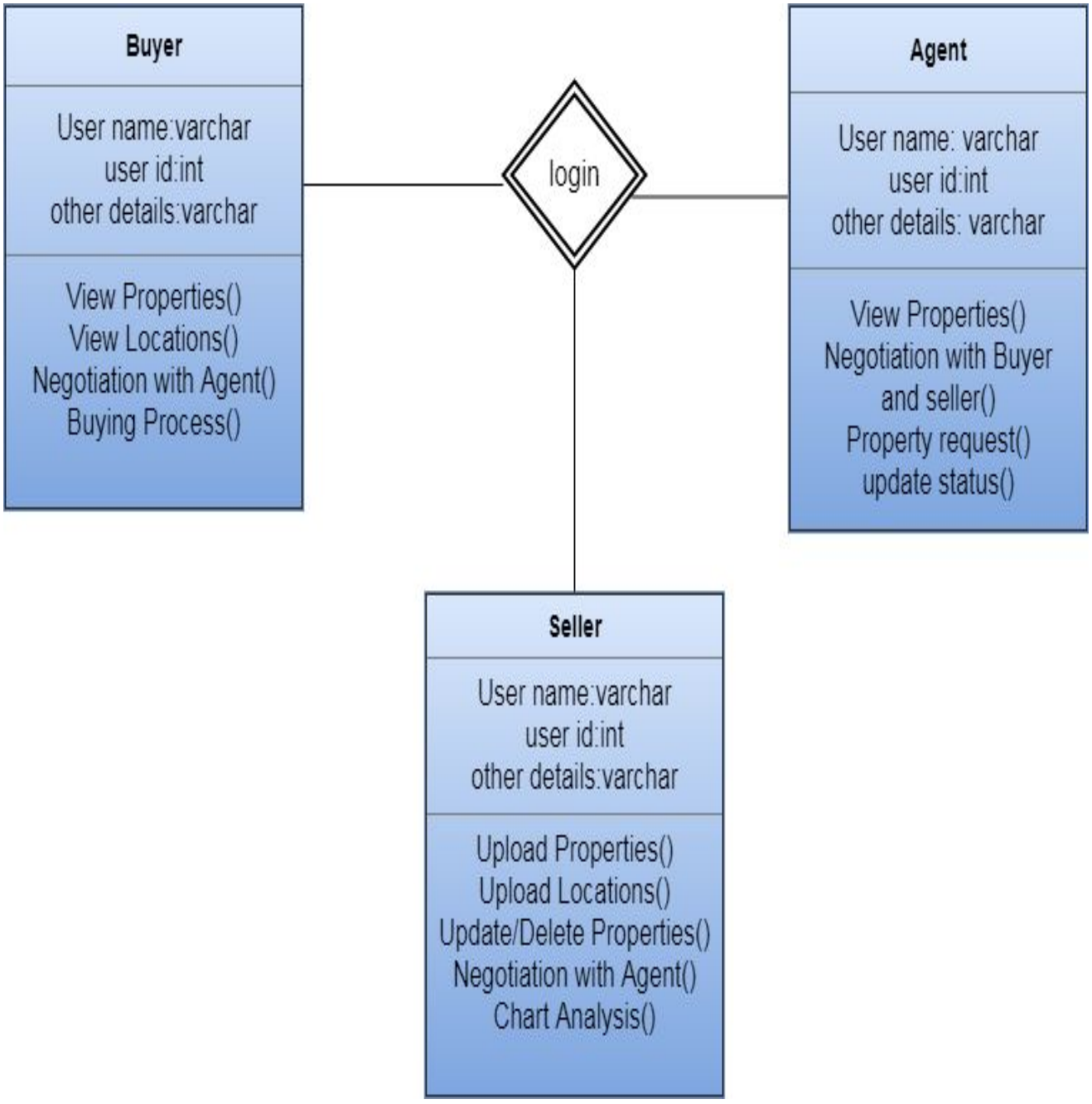


Fig 5.3 Class Diagram

5.4 FLOW DIAGRAM

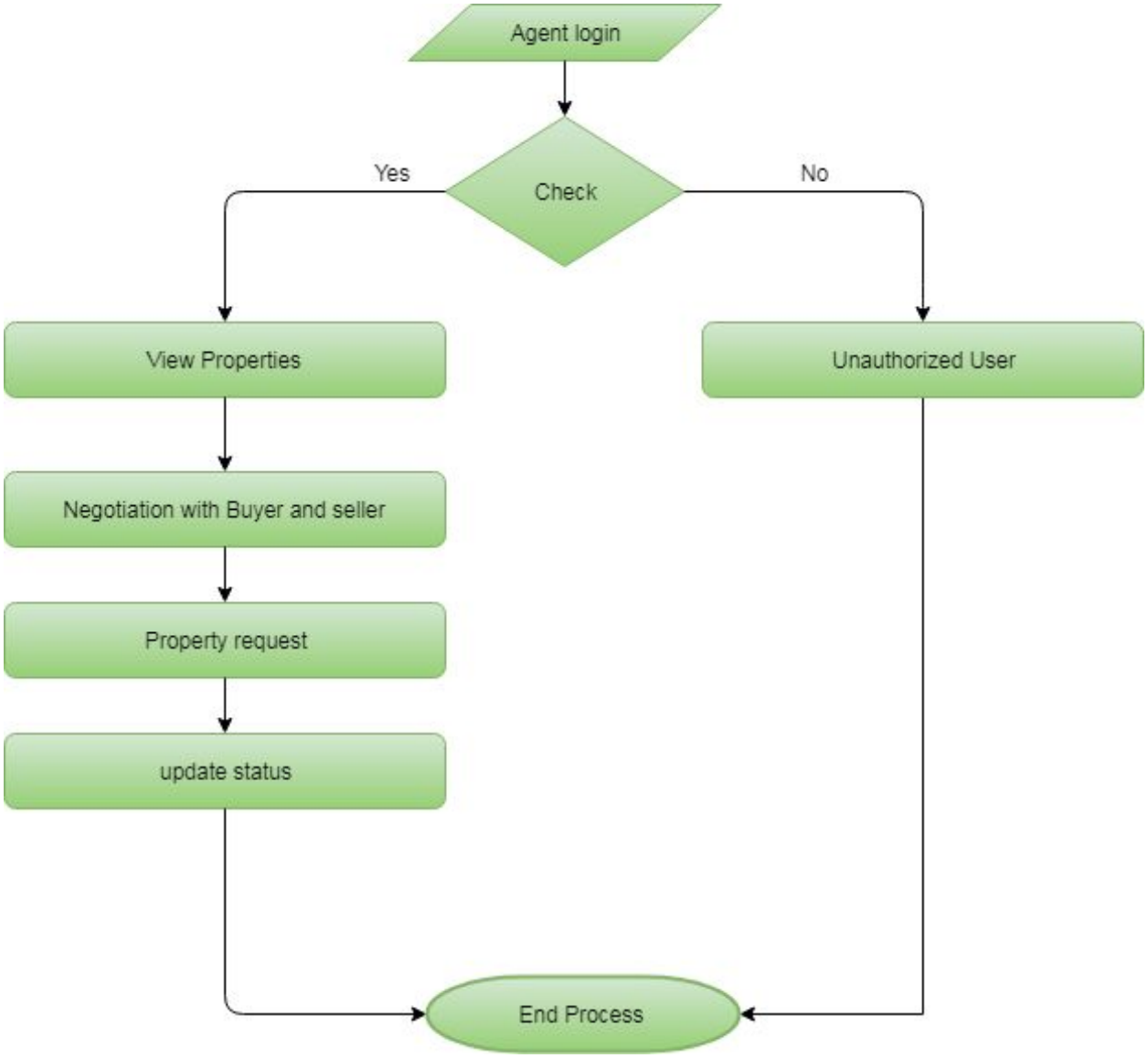


Fig 5.4 Flowcase diagram for Agent

5.5 ACTIVITY DIAGRAM

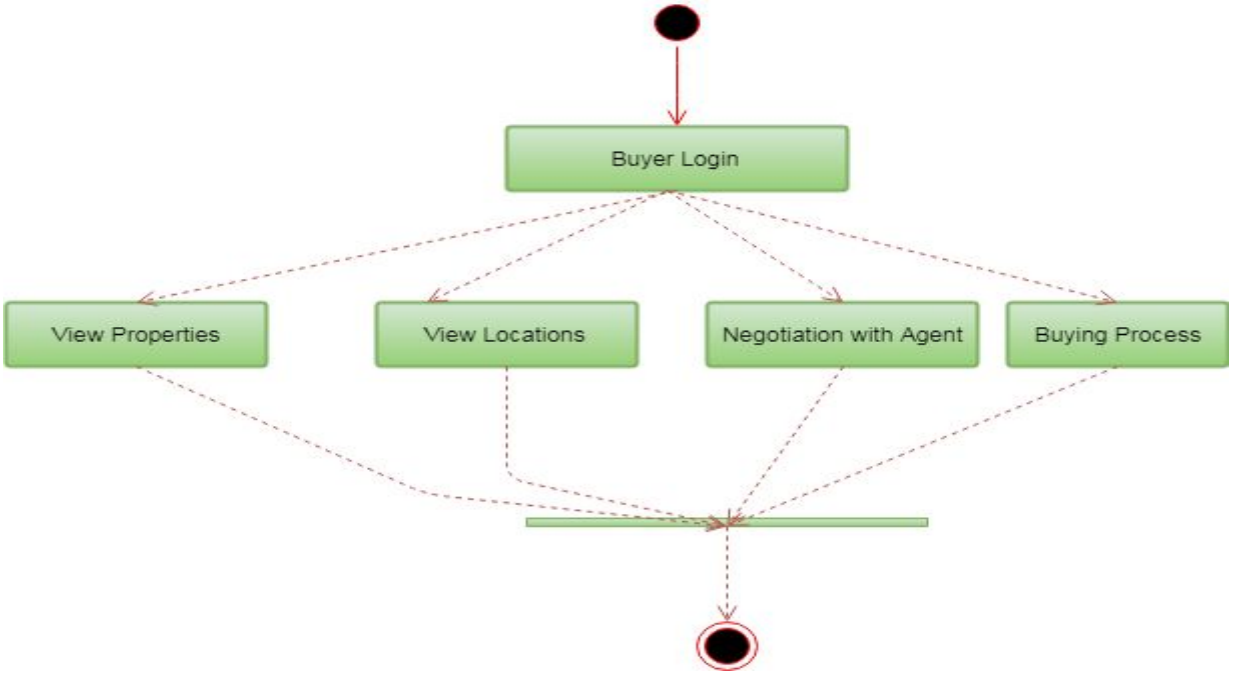


Fig 5.5.1 Activity diagram for Buyer

5.6 SEQUENCE DIAGRAM

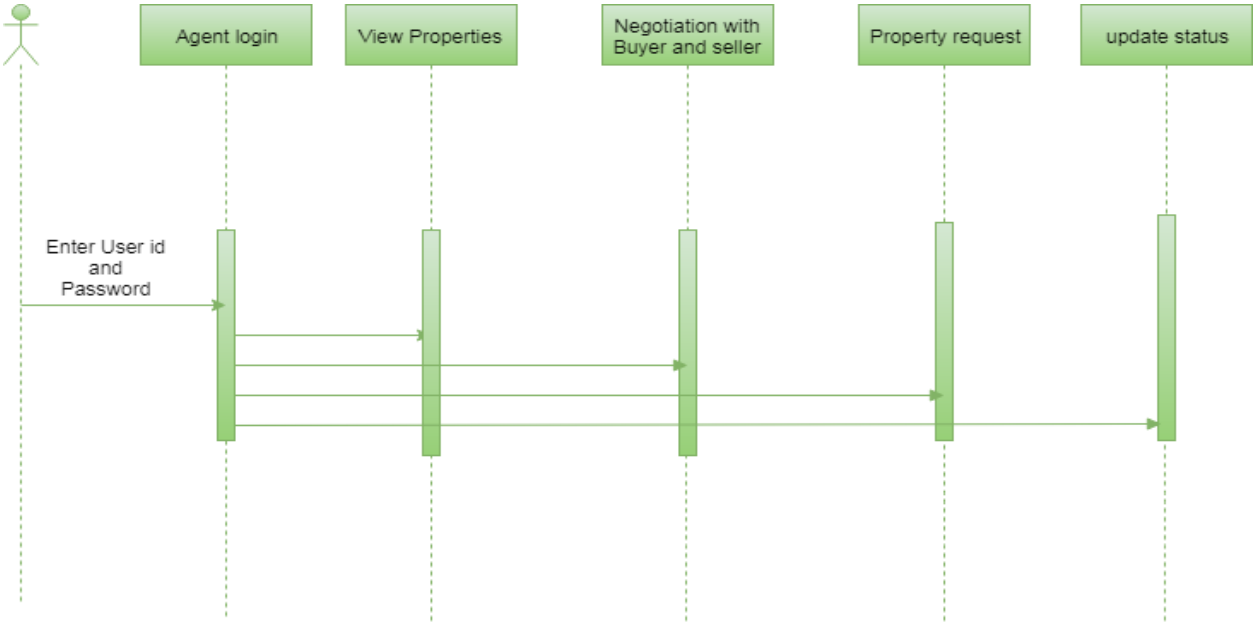


Fig 5.6.1 Sequence diagram for Agent

CHAPTER-6

SOURCE CODE

```
-- phpMyAdmin SQL Dump

-- version 4.0.4

-- http://www.phpmyadmin.net
-- Host: localhost
-- Generation Time: Dec 19, 2017 at 12:05 PM
-- Server version: 5.6.12-log
-- PHP Version: 5.4.16
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
Database: `real_estate`
CREATE DATABASE IF NOT EXISTS `real_estate` DEFAULT CHARACTER
SET latin1 COLLATE latin1_swedish_ci;
USE `real_estate`;
-- Table structure for table `agent_chat`
CREATE TABLE IF NOT EXISTS `agent_chat` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `message` varchar(400) NOT NULL,
  `receiver` int(11) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
```

```
  PRIMARY KEY (`id`),
  KEY `agent_chat_user_id_0dcfd75a_fk_general_profile_id` (`user_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=43 ;
Dumping data for table `agent_chat`
INSERT INTO `agent_chat` (`id`, `message`, `receiver`, `user_id`) VALUES
(1, 'asdasd', 2, 3),
(2, 'hi gokul,', 12, 3),
(3, 'dsfsdf', 4, 3),
(4, 'xzdc', 5, 3),
(5, 'drrfg', 7, 3),
(6, 'dzfsdf', 4, 3),
(7, '5452', 5, 1),
(8, 'fdfg', 5, 1),
(9, 'dsafr', 3, 1),
(10, 'saasd', 1, 3),
(11, 'sdf', 3, 1),
(12, 'gokul is good sdf sdf sdfs sdf sdefesd  sdfdsf sdfsdf sdf sdf sdf sdf sdfsdf sdfs sdfsdf sdf sdfsdfsd fsdfsd sdf sdf fsd sd gokul is good sdf sdf sdfs sdf sdefesd  sdfdsf sdfsdf sdf sdf sdf sdf sdfsdf sdfs sdfsdf sdf sdfsdfsd fsdfsd sdf sdf fsd sdv gokul is good sdf sdf sdfs sdf sdefesd  sdfdsf sdfsdf sdf sdf sdf sdf sdfsdf sdfs sdfsdf sdf sdfsdfsd fsdfsd sdf sdf fsd sd', 3, 1),
(13, 'ravi is bad', 5, 1),
(14, 'Suresh is very good boy', 6, 1),
(15, 'w re', 3, 1),
(16, 'Hi gokul how  r u?', 1, 3),
(17, 'i am fine', 3, 1),
(18, 'what r u doing?', 3, 1),
(19, 'im also working onle..', 1, 3),
(20, 'when is ur marriage?', 1, 3),
(21, 'then go and work da..', 3, 1),
(22, 'the day after yours', 3, 1),
(23, 'hmm ok enjoy da', 1, 3),
```

```
(24, 'how is ur work?', 1, 3),
(25, 'u enjoy da..', 3, 1),
(26, 'it is going good', 3, 1),
(27, 'wat abt yours? da', 3, 1),
(28, 'pothum poda', 3, 1),
(29, 'hmmm ok da', 1, 3),
(30, 'sdf', 0, 2),
(31, 'is that negotiable price?', 1, 3),
(32, 'if it is negotiable, how much it can be?', 1, 3),
(33, 'if negotiable how much it will be?', 1, 3),
(34, 'it can be up to 1000rs..', 3, 1),
(35, 'ok..we will confirm', 1, 3),
(36, 'what happend to my property?', 3, 2),
(37, 'ya..', 2, 3),
(38, 'your property is sold by other buyer and we can proceed further', 2, 3),
(39, 'how much it can be negoiable?', 1, 3),
(40, 'it is to be confirmed', 1, 3),
(41, 'up to 1000rs', 3, 1),
-- Table structure for table `auth_group`
CREATE TABLE IF NOT EXISTS `auth_group` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(80) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT
 Table structure for table `auth_group_permissions`
CREATE TABLE IF NOT EXISTS `auth_group_permissions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `group_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
```

```
  UNIQUE KEY
`auth_group_permissions_group_id_permission_id_0cd325b0_uniq`
(`group_id`,`permission_id`),
  KEY `auth_group_permissio_permission_id_84c5c92e_fk_auth_perm`
(`permission_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT
 Table structure for table `auth_permission`
CREATE TABLE IF NOT EXISTS `auth_permission` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `content_type_id` int(11) NOT NULL,
  `codename` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY
`auth_permission_content_type_id_codename_01ab375a_uniq`
(`content_type_id`,`codename`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=31
 Dumping data for table `auth_permission
INSERT INTO `auth_permission` (`id`, `name`, `content_type_id`, `codename`)
VALUES
(1, 'Can add log entry', 1, 'add_logentry'),
(2, 'Can change log entry', 1, 'change_logentry'),
(3, 'Can delete log entry', 1, 'delete_logentry'),
(4, 'Can add permission', 2, 'add_permission'),
(5, 'Can change permission', 2, 'change_permission'),
(6, 'Can delete permission', 2, 'delete_permission'),
(7, 'Can add group', 3, 'add_group'),
(8, 'Can change group', 3, 'change_group'),
(9, 'Can delete group', 3, 'delete_group'),
(10, 'Can add content type', 4, 'add_contenttype'),
(11, 'Can change content type', 4, 'change_contenttype'),
(12, 'Can delete content type', 4, 'delete_contenttype'),
```

```
(13, 'Can add session', 5, 'add_session'),
(14, 'Can change session', 5, 'change_session'),
(15, 'Can delete session', 5, 'delete_session'),
(16, 'Can add user', 6, 'add_profile'),
(17, 'Can change user', 6, 'change_profile'),
(18, 'Can delete user', 6, 'delete_profile'),
(19, 'Can add location details', 7, 'add_locationdetails'),
(20, 'Can change location details', 7, 'change_locationdetails'),
Table structure for table `buyer_propertyrequests`
CREATE TABLE IF NOT EXISTS `buyer_propertyrequests` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `requestedProperty_id` int(11) NOT NULL,
  `status` varchar(400) NOT NULL,
  `price` varchar(400) NOT NULL,
  `message` varchar(400) NOT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `buyer_propertyrequests_user_id_0ac8a8e5_fk_general_profile_id`
(`user_id`),
  KEY `buyer_propertyrequests_requestedProperty_id_a5b42a0c`
(`requestedProperty_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
 Dumping data for table `buyer_propertyrequests`
INSERT INTO `buyer_propertyrequests` (`id`, `requestedProperty_id`,
`status`, `price`, `message`, `user_id`) VALUES
(1, 2, 'sold', '1250', 'Looking for good price', 1),
(2, 5, 'sold', '1523', 'price is negotiable',
Table structure for table `django_admin_log`
CREATE TABLE IF NOT EXISTS `django_admin_log` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `action_time` datetime(6) NOT NULL,
  `object_id` longtext,
```

```
  `object_repr` varchar(200) NOT NULL,
  `action_flag` smallint(5) unsigned NOT NULL,
  `change_message` longtext NOT NULL,
  `content_type_id` int(11) DEFAULT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `django_admin_log_content_type_id_c4bce8eb_fk_django_co`
(`content_type_id`),
  KEY `django_admin_log_user_id_c564eba6_fk_general_profile_id` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=
Table structure for table `django_content_type
CREATE TABLE IF NOT EXISTS `django_content_type` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `app_label` varchar(100) NOT NULL,
  `model` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `django_content_type_app_label_model_76bd3d3b_uniq`
(`app_label`,`model`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMEN
 Dumping data for table `django_content_type`
INSERT INTO `django_content_type` (`id`, `app_label`, `model`) VALUES
(1, 'admin', 'logentry'),
(9, 'agent', 'chat'),
(3, 'auth', 'group'),
(2, 'auth', 'permission'),
(10, 'buyer', 'propertyrequests'),
(4, 'contenttypes', 'contenttype'),
(6, 'general', 'profile'),
(7, 'seller', 'locationdetails'),
(8, 'seller', 'propertydetails'),
(5, 'sessions', 'session
 Table structure for table `django_migrations
```

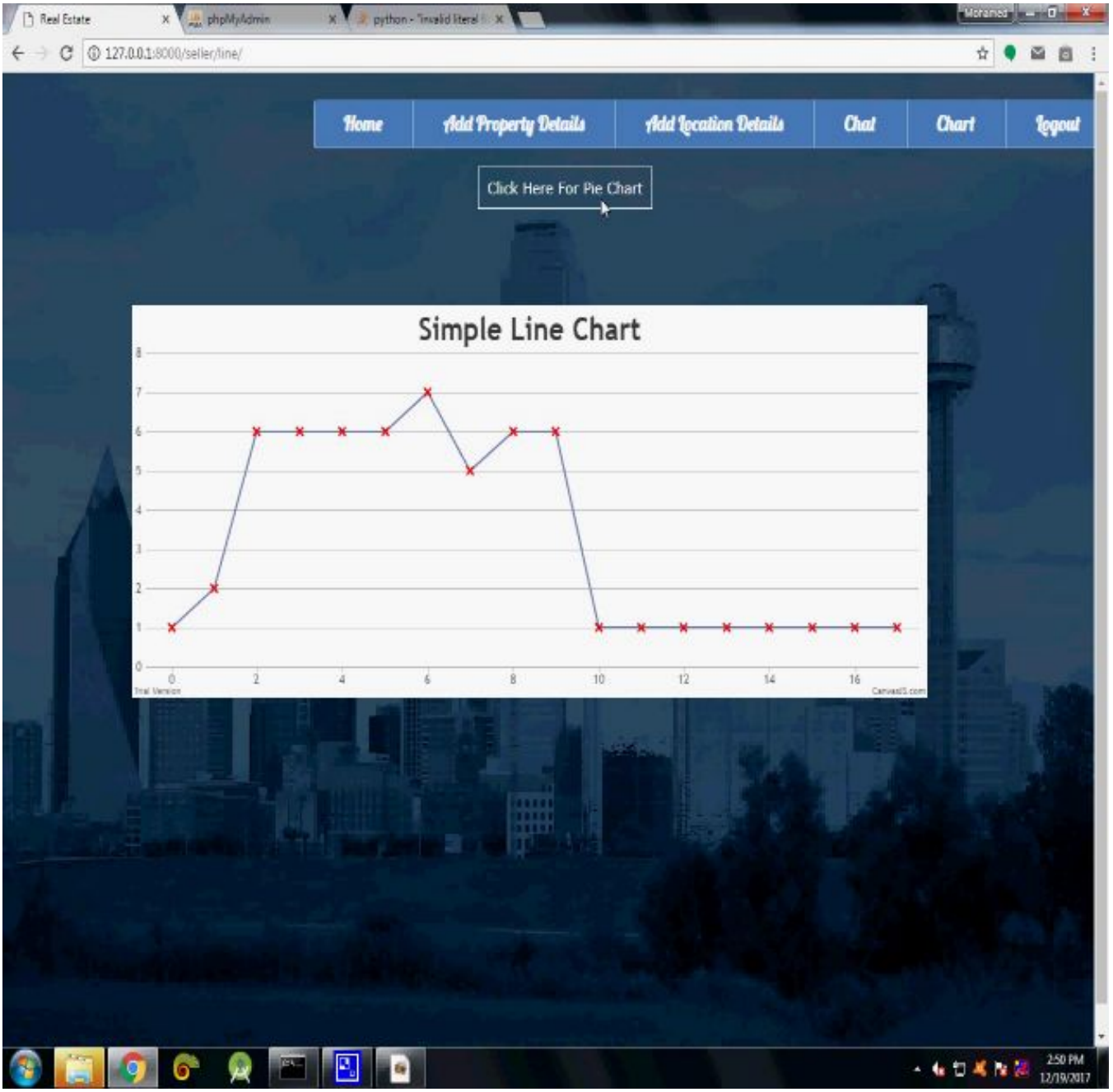
```
CREATE TABLE IF NOT EXISTS `django_migrations` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `app` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `applied` datetime(6) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=26 ;
Dumping data for table `django_migrations`
INSERT INTO `django_migrations` (`id`, `app`, `name`, `applied`) VALUES
(1, 'contenttypes', '0001_initial', '2017-12-08 19:26:21.477539'),
(2, 'contenttypes', '0002_remove_content_type_name', '2017-12-08
19:26:22.833007'),
(3, 'auth', '0001_initial', '2017-12-08 19:26:27.474
 Table structure for table `general_profile_user_permissions`
CREATE TABLE IF NOT EXISTS `general_profile_user_permissions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `profile_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY
`general_profile_user_per_profile_id_permission_id_4ec35cbe_uniq`
(`profile_id`,`permission_id`),
  KEY `general_profile_user_permission_id_d7c45d82_fk_auth_perm`
(`permission_id`)
)NGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT
 Table structure for table `seller_locationdetails`
CREATE TABLE IF NOT EXISTS `seller_locationdetails` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `locaddress` varchar(400) NOT NULL,
  `locationusage` varchar(4) NOT NULL,
  `locationname` varchar(400) NOT NULL,
  `lati` double NOT NULL,
```

```
  `longi` double NOT NULL,
  `otherdetails` varchar(400) NOT NULL,
  `propid_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `seller_locationdetai_propid_id_e883f790_fk_seller_pr` (`propid_id`),
  KEY `seller_locationdetails_user_id_d9b2cdd1_fk_general_profile_id`
(`user_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=62 ;
 Dumping data for table `seller_locationdetails`
INSERT INTO `seller_locationdetails` (`id`, `locaddress`, `locationusage`,
`locationname`, `lati`, `longi`, `otherdetails`, `propid_id`, `user_id`) VALUES
(3, 'sharanpur', 'prop', 'sharanpur', 29.967079, 77.5510172, 'It is holy god
rama place', 2, 2),
(4, 'delhi', 'near', 'sharanpur', 28.7040592, 77.1024902, 'It is holy god rama
place', 2, 2),
(5, 'sholinganallur', 'prop', 'sholinganallur', 12.9009877, 80.2279301, 'It is
surrounded by IT parks,schools and many good hospitals and temples', 3,
2),
(6, 'ISKCON Chennai, Sri Sri Radha Krishna Temple.', 'near',
'sholinganallur', 12.9059915, 80.2417354, 'It is surrounded by IT
parks,schools and many good hospitals and temples', 3, 2)
```

CHAPTER-7

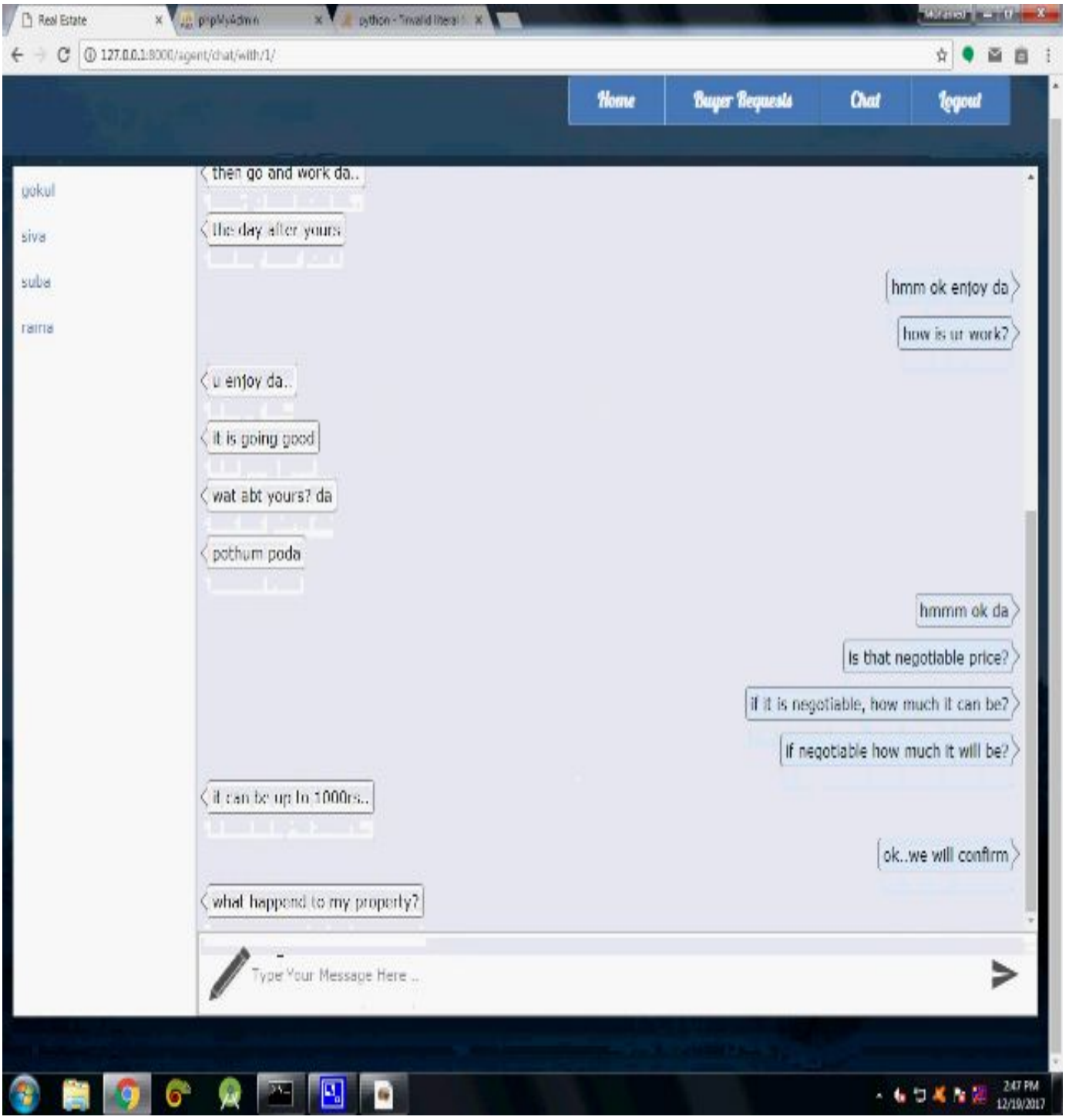
OUTPUT SCREENS

LINE GRAPH:



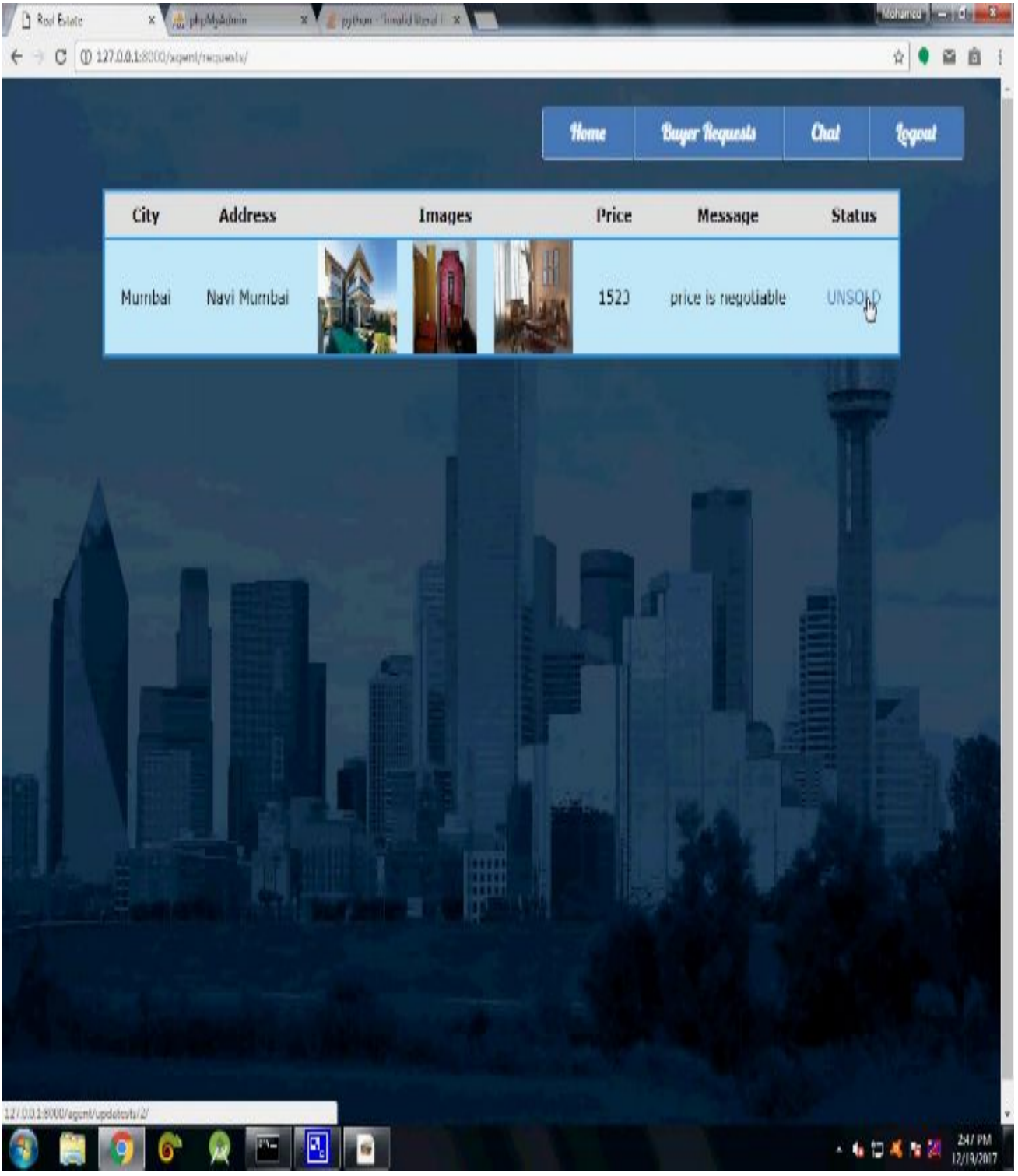
SCREEN:7.1 LINE GRAPH

CHAT:



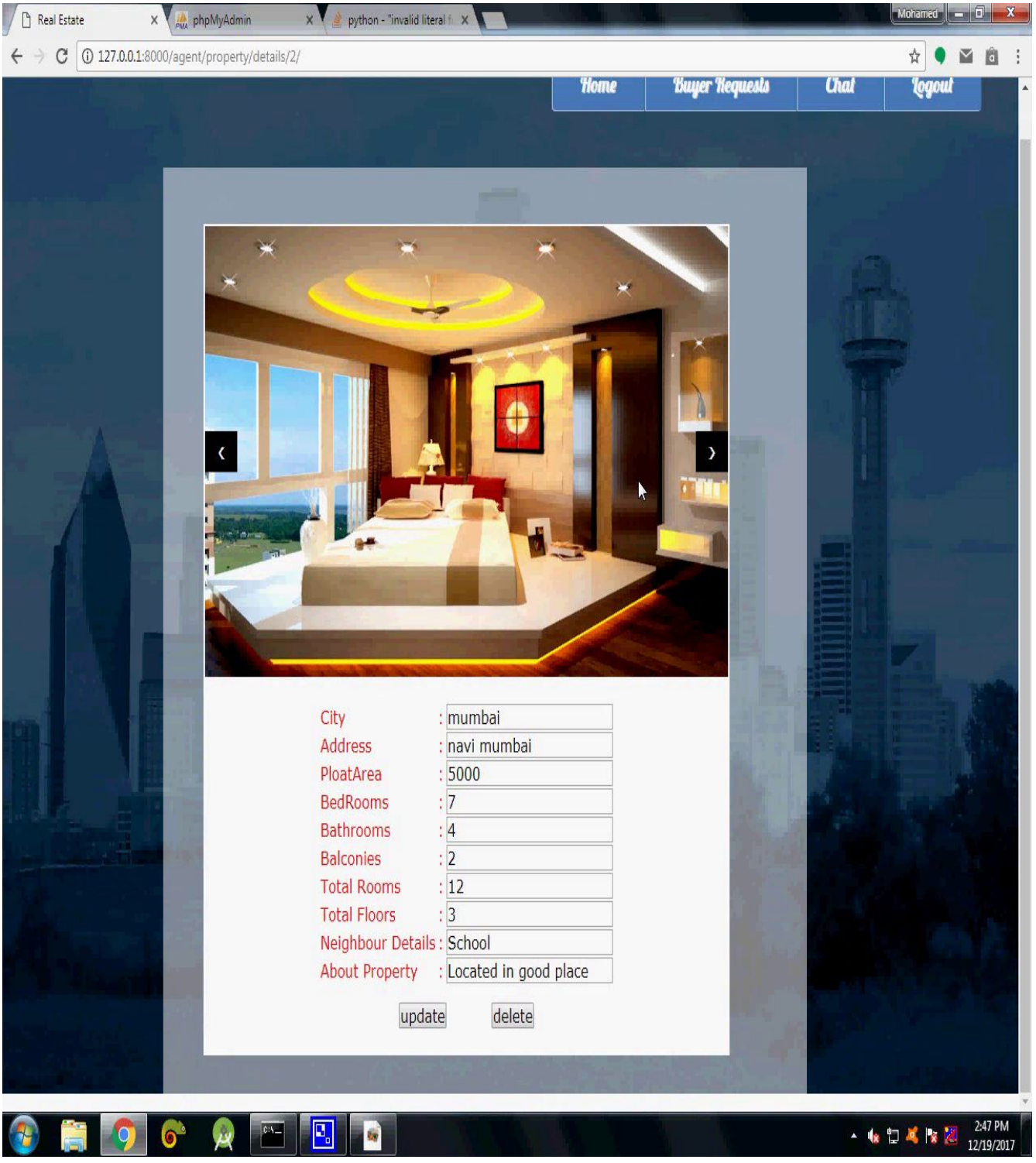
SCREEN 7.2 CHAT

AGENT VIEW:



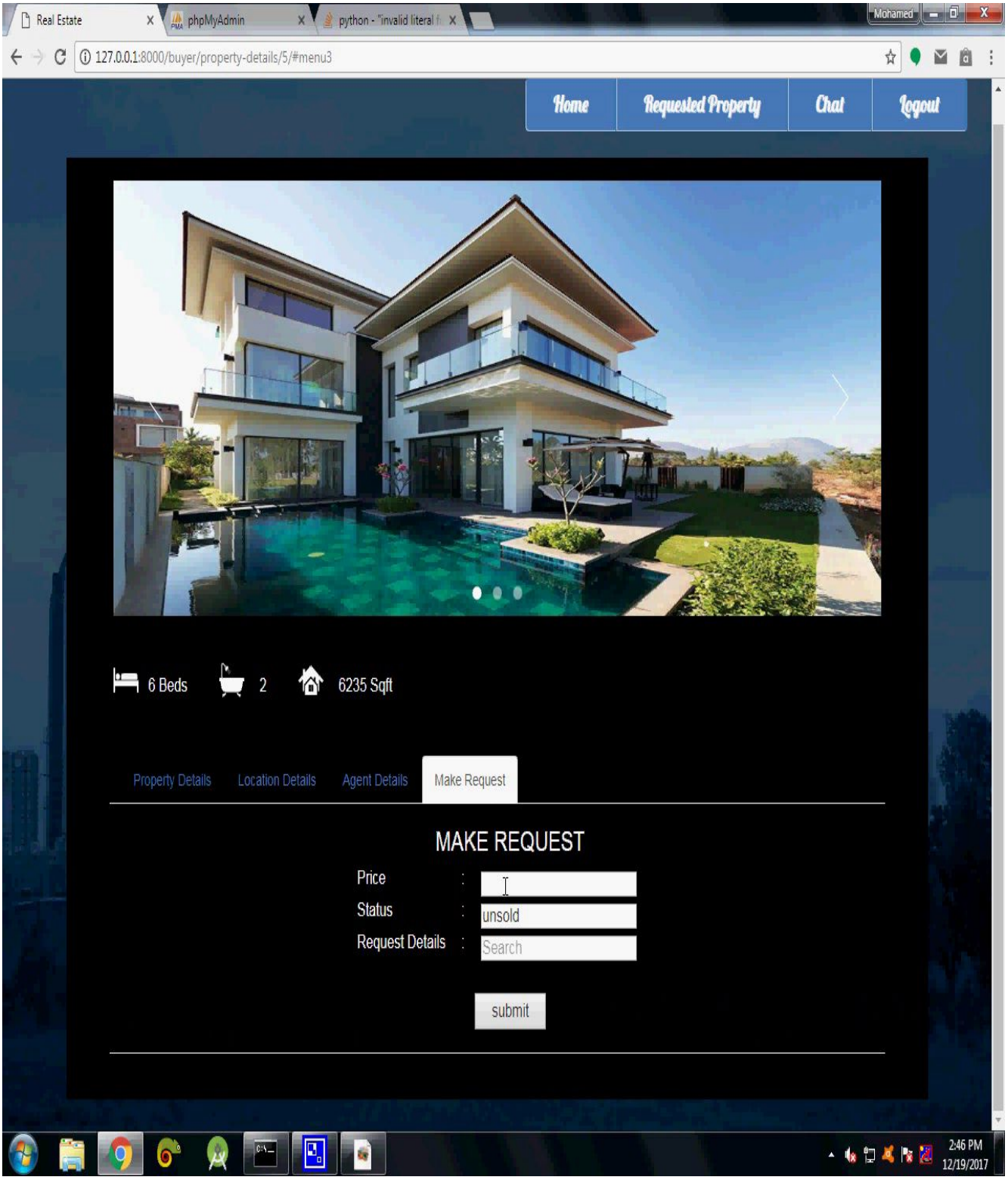
SCREEN 7.3 AGENT VIEW

PROPERTY VIEW:



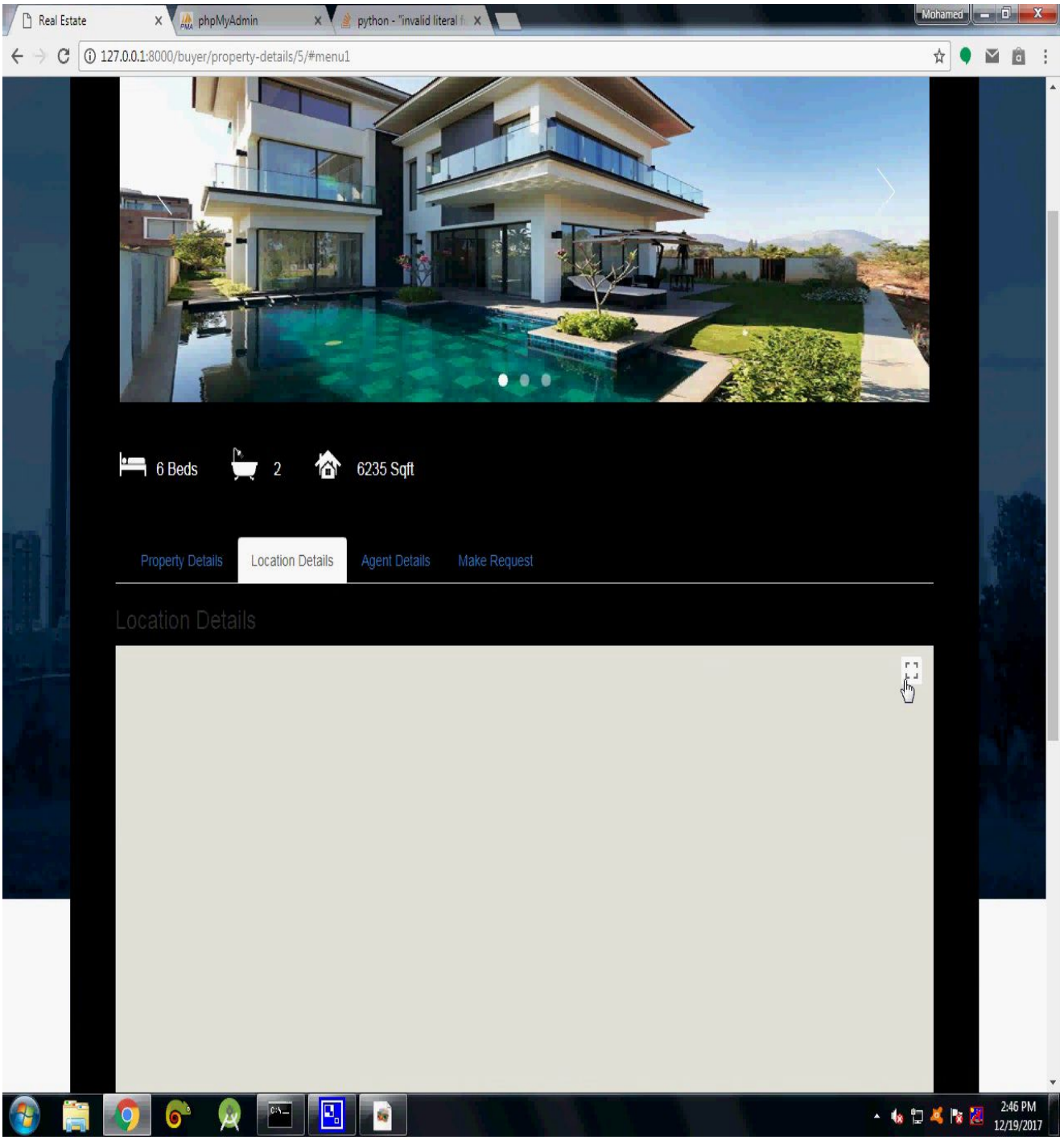
SCREEN 7.4 PROPERTY VIEW

BUYER VIEW:



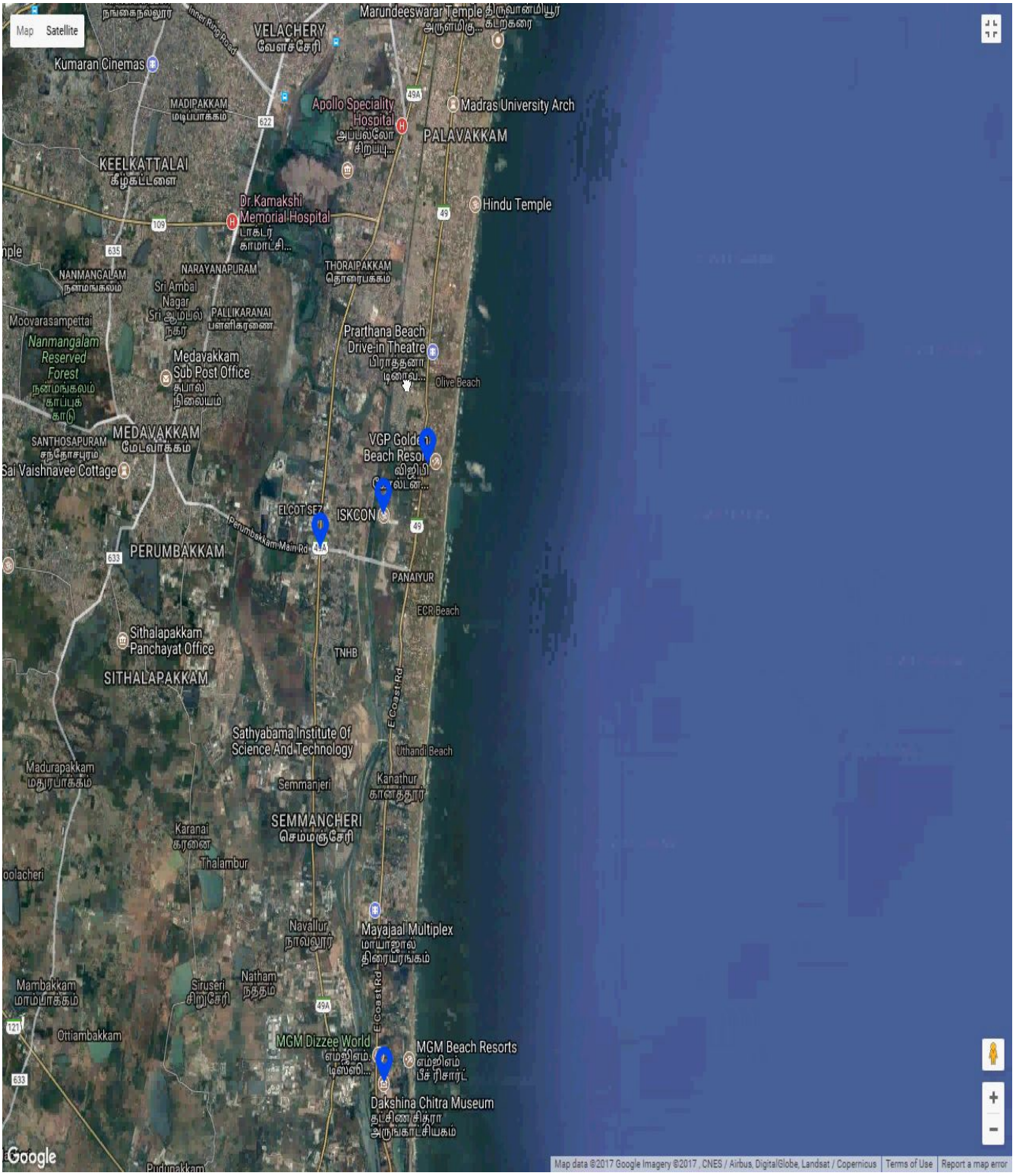
SCREEN 7.5 BUYER VIEW

BUYER PROPERTY DETAILS:



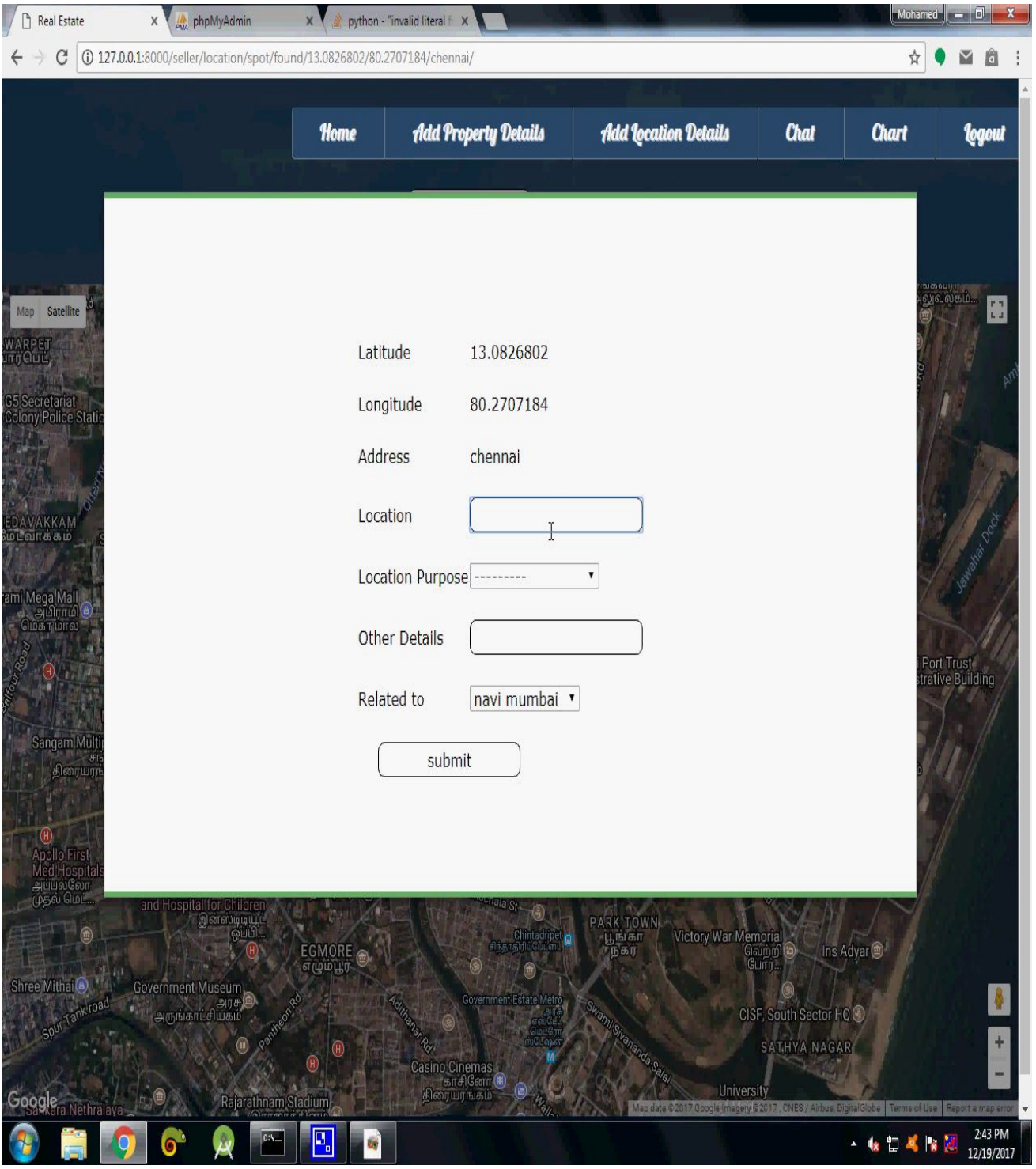
SCREEN 7.6 BUYER PROPERTY DETAILS

ADDRESS DETAILED VIEW:



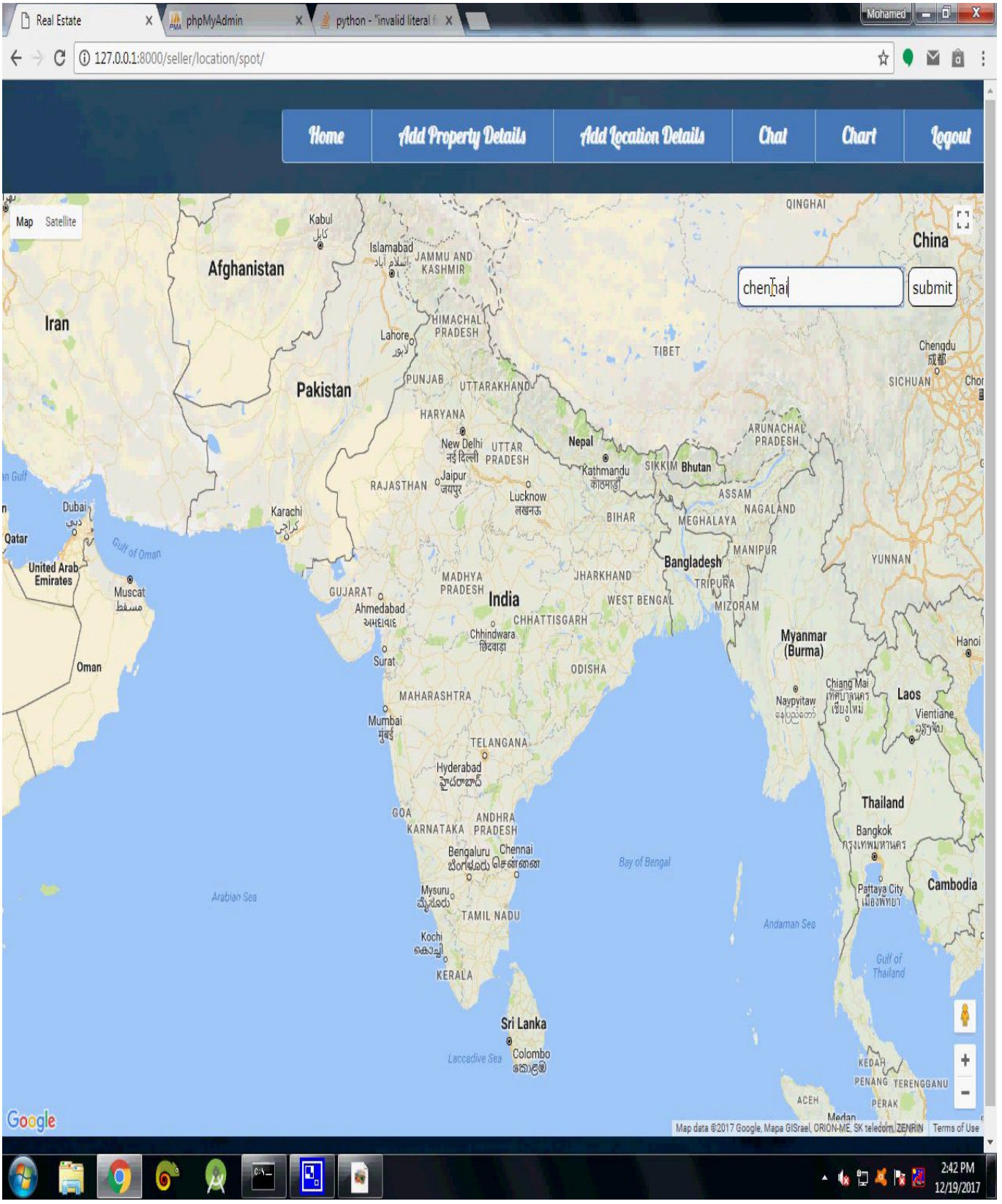
SCREEN 7.7 ADDRESS DETAILED VIEW:

LOCATION SPOT:



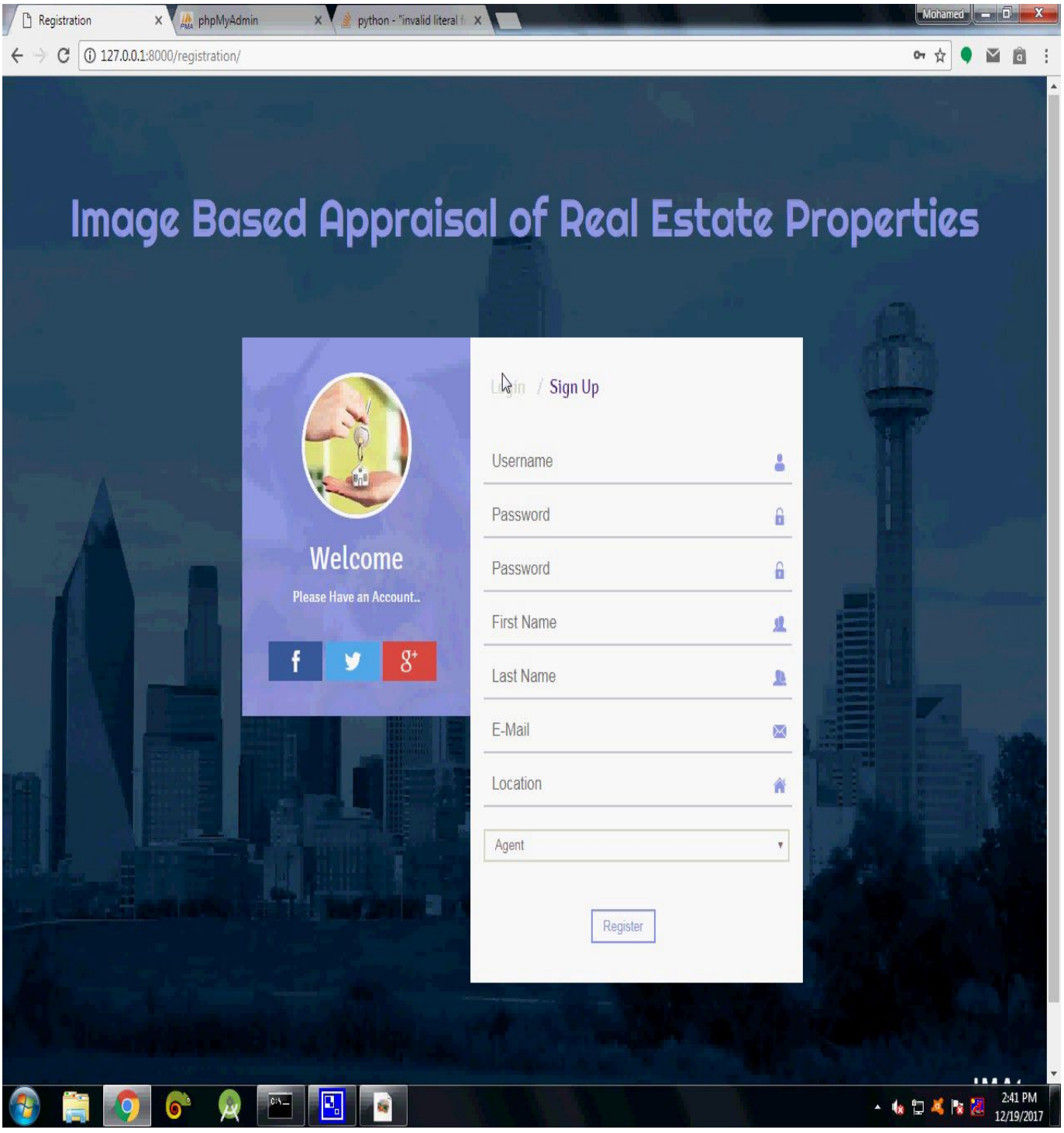
SCREEN 7.8 LOCATION SPOT

SELLER LOCATION SPOT:



SCREEN 7.9 SELLER LOCATION SPOT

REGISTRATION PAGE:



SCREEN 7.10 REGISTRATION PAGE

CHAPTER-8

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the

combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs expected.

Systems : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing: User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER-9

CONCLUSION

we propose a novel framework for real estate appraisal. In particular, the proposed framework is able to take both the location and the visual attributes into consideration. The evaluation of the proposed model on two selected cities suggests the effectiveness and flexibility of the model. Indeed, our work has also offered new approaches of applying deep neural networks on graph structured data. We hope our model can not only give insights on real estate appraisal, but also can inspire others on employing deep neural networks on graph structured data.

CHAPTER-10

FUTURE ENHANCEMENT

The main contributions of our work are as follows:

• To the best of our knowledge, we are the first to quantify the impact of visual content on real estate price estimation. We attribute the possibility of our work to the newly designed computer vision algorithms, in particular Convolutional Neural Networks (CNNs).

• We employ random walks to generate house sequences according to the locations of each house. In this way, we are able to transform the problem into a novel sequence prediction problem, which is able to preserve the relation among houses.

• We employ the novel Recurrent Neural Networks (RNNs) to predict real estate properties and achieve accurate results.

CHAPTER-11

REFERENCES

[1] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and Z.-H. Zhou, “Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering,” in SIGKDD. ACM, 2014, pp. 1047–1056.

[2] K. Wardrip, “Public transits impact on housing costs: a review of the literature,” 2011.

[3] F. T. Wang and P. M. Zorn, “Estimating house price growth with repeat sales data: what’s the aim of the game?” Journal of Housing Economics, vol. 6, no. 2, pp. 93–118, 1997.

[4] A. Beja and M. B. Goldman, “On the dynamic behavior of prices in disequilibrium,” The Journal of Finance, vol. 35, no. 2, pp. 235–248, 1980.

[5] E. L’Eplattenier, “How to run a comparative market analysis (cma) the right way,” http://fitsmallbusiness.com/comparative-market-analysis/, 2016.