ONLINE PROPERTY ADVERTISEMENT SYSTEM

Developed by

Muhammad Farooq

Supervised by

Dr. Rabeeh Ayaz Abbasi

Department of Computer Science

Quaid-i-Azam University

Islamabad, Pakistan

Session 2010-2012

A report is submitted to the

Department of Computer Science,

Quaid-i-Azam University, Islamabad

as a partial fulfilment of the requirements

for the award of the degree of

M.Sc. Computer Science.

Dedicated to…

My loving parents, family and well wishers,

Specially my Uncle **Barkat Ali** and teachers

Mr. Zafar Saeed and **Dr. Rabeeh Ayaz Abbasi.**

Acknowledgment

First of all, I would like to express my sincere and humble gratitude to Almighty Allah, whose blessings and guidance has been a real source of all the achievements in my life. I am thankful to my Uncle, parents, brothers, sisters and grand-parents for their prayers and moral & financial support during my educational career.

It is my bounden duty to pay tributes to my worthy teachers and staff members of Department of Computer Science, Quaid-i-Azam University Islamabad, who have molded a lump of clay and made him into what he is today. Special thanks to **Dr. Rabeeh Ayaz Abbasi, Dr. Onaiza Maqbool, Dr. Shuaib Karim, Mr. Zafar Saeed** and **Mr. S. M. Naqi** for their kind help and guidance.

At the end, I would like to thanks to **Malik Akhtar, Arshad Ali, Shahid Hussain** and my class fellows **Fazal Shahzad, Fareed Ahmed, Mudassar** and **Muzaffer Abbas** for encouraging me.

Muhammad Farooq

Abstract

Online Property Advertisement System provides the facility of advertising the properties as plots, commercials, residential and agricultural lands to their registered members. Members are allowed to upload the specification of property/land along with its pictures, videos and location on map. System maintains separate listings for each user, from where they can change/remove the property. If someone marks a property as favourite, it will be stored in his/her favourite list. Every member can rate a property, maximum to five stars. A good feature of this system is "comments", where different members can comment on a property to ask question or share information about property.

There are also two kinds of property searching, one is through list and other is through map. In list searching, users can specify search criteria according to which search would be done. And in map searching, users can browse the map to view properties on map. They can also specify the name of city, region or state to go to that location, where all available properties will be displayed as markers, and by clicking on a marker, the details could be gotten of a property.

Table of Contents

Chapter # 1

Introduction

This chapter provides an overview of the report and the proposed system and describes why the proposed system is needed, what its advantages are, and what is new in this system. This chapter also describes the motivation for this project. Finally, a brief overview of all chapters is given.

1.1 Problem Definition

Many people face difficulties to search property as per their requirements. For property buyers, buying is a challenge because they don't know every property which is available for them in their areas of interest and if they know, they do not know the specifications of that property. If they want to visit a property with their families, it is almost impossible for them. For property sellers, advertising their properties is a big challenge, they can't place banners or posters at each and every spot of the city/town, which is also costly. The intermediary person called agent/broker handles this work and owner of property pays the broker. The timing also matters; no one (broker) can meet whole the time. To overcome these problems, and facilitate both (buyers and sellers), there should be a common solution. The solution is **OPAS (Online Property Advertisement System).** This system helps the buyers to search for properties of their interest, and sellers to advertise their properties without any intermediate person, at any desired time.

1.2 Related Work

There are many different websites which allow their users to sell and buy properties, in Pakistan there are two such popular websites, they are www.bastee.pk and www.zameen.com. These websites are providing the facility of uploading property's specifications (as number of rooms, washrooms, garage, drawing room, and some extra features), its pictures and location on map and owners details only. But there is no facility to rate a property, comment on a property, uploading a video related to the property, browse the map to search the properties, and follow a user.

1.3 Motivation

The motivation behind creating this system is to facilitate the end users by providing more information about their concerned properties and making system more secure and practical by blocking those users who post/advertise illegal property/area (as governmental or intelligence agencies). If someone (who is registered with system) wants to comment on an advertised property that something is wrong in this advertisement (as neighbors know correct information about that place) or any comments related to price, what can he/she do? Moreover if someone likes the property and wants to rate the property, and follow his/her favorite user, there should be facility provided by the system. If the owner has a video about property, uploaded on YouTube, he/she should be able to share that link. These functionalities are not provided in systems implemented before. So there should be a system which provide and user will access easily these all facilities.

1.4 Proposed Solution

The system to be developed will be different from current systems in a way that it will also allow the registered users to upload the photos and movies related to their property, can comment on an existing property, rate the property, can report to admin if someone posts an illegal area/property (as governmental or intelligence agencies) and admin will block that user after receiving more than 5 complaints from different users, can view more ads of a particular advertiser if they like his post and system will also display the days of an ad that how long it is on the website. Some features are chosen from existing systems, as this system will also tells the location of property on a map, owners details, facility of uploading pictures and specification of the property. Furthermore this system will be easily extendable to add a new service, with few changes.

1.5 Outline of Report

Chapter 2

This chapter describes purpose, scope and objective of Online Property Advertisement System. It also specifies requirements with different perspectives. At last summary of major functions is given under product functions section.

Chapter 3

This chapter describes requirements in the form of use cases. All uses cases are written in detail, where detail includes uses case description, primary actor, input, output, pre-condition, post-condition and extensions. After use cases, a use case diagram shows all use cases and their interaction with primary and secondary actor(s). Further chapter contains sequence diagrams built on the basis of use cases described, before to show interaction of classes through their methods.

Chapter 4

This chapter contains detailed design of Online Property Advertisement System in the form of diagrams. At start architecture diagram shows basic structure of system which is helpful in understanding system at abstract level. Furthermore class diagram is shown, which shows all classes used in system and also their interaction with each other. At the end of chapter some design highlights are being discussed.

Chapter 5

This chapter is related to system implementation. This chapter mentions the tools, framework and API's used to develop Online Property Advertisement System.

Chapter 6

This chapter contains User Acceptance Test specification which is built against use cases as stated in Requirement Analysis.

Chapter 7

This chapter summarizes Online Property Advertisement System regarding what functionality system is currently providing and what should be future enhancements.

Definitions, Acronyms and Abbreviations

Definition of terms, acronyms and abbreviations used in chapters is given in this section.

References

At last references related to each chapter which were helpful in system documentation development are given.

Chapter # 2

Requirements Specification

This chapter describes SRS purpose, objective and scope of OPAS (Online Property Advertisement System). It also specifies requirements with different perspectives. At the end of the chapter, summary of major functions is provided.

2.1 Introduction of SRS

This section describes SRS purpose, Objective and scope of OPAS (Online Property Advertisement System).

2.1.1 SRS purpose

Purpose behind developing this SRS is, let other developers understand basic requirements of OPAS. This SRS will be helpful to both customer and developer of the system to agree upon requirements of system to avoid any ambiguities/confusion.

2.1.2 Objective

Main objective of this system is to facilitate end users in such a way that they can be able to search/advertise their properties easily, any time, from anywhere and without any cost.

2.1.3 Scope

Scope of this system includes:

- Allow users to register with the system.
- Registered users will post their properties, specifications, and system will store along with their names.
- Allow the registered users to share a video link or photos related to their properties.
- Registered users will be able to comment on an existing property.

- Registered users will be able to give rating to any property.
- Registered users can report any advertisement to administrator as abuse.
- System will show the location of property on a map.
- Search properties
- Browse properties on the map
- Following a particular user(seller)

2.1.4 Overview

This SRS contains system requirements in an organized way to help developers better understand the system functionality. Next sections include overall description of the system from product perspective. Also summary of major functions of system is provided under product functions section with user characteristics at last.

2.2 Product Perspective

2.2.1 System Interfaces

System uses APIs of Google Maps and YouTube, to retrieve property's geographical location on map and video contents associated with it.

2.2.2 User Interfaces

Following user interfaces will be provided by the system:

- Map based interface for browsing properties on the map.
- Non map based interface for browsing search results based on different criteria like price range, location, etc.

2.2.3 Hardware Interfaces

Following are minimum hardware requirements for SUD (system under discussion) which are also minimum hardware requirements of Google Maps & YouTube:

- CPU: Pentium 3, 500MHz.
- RAM: 256MB
- Hard Disk: 300MB free space
- Network Adapter.
- Screen: 1024x768, 16-bit High Color

2.2.4 Software Interfaces

- System requires one of following Web browsers to run on, based on requirements of Google maps and YouTube:
    - IE 8.0+
    - Firefox 11.0+
    - Safari 3.2.1+
    - Chrome 4.0.249+
    - Netscape 7.1+
    - Opera 9.0+

- System requires Windows XP, Windows Vista or Windows 7 operating system to run on.

2.2.5 Communication Interfaces

System requires internet connectivity to interact with API's server (Google Maps and YouTube).
(Minimum 256 kbps, Recommended 1 Mbps).

2.2.6 Memory Constraints

System requires 256 MB or more RAM to run Google Maps on any compatible browser mentioned in software interfaces.

2.3 Product Functions

Following are major functions which system should:

- Provide registration for users.
- Provide an interface to upload the property's features.
- Be able to store pictures of the concerned property.
- Also be able to store the video links of the concerned property.
- Provide maps for browsing and searching properties.
- Provide a map where the exact location of property is displayed.
- provide zooming facility on the map

- provide navigation facility to navigate to different areas on map
- Provide an interface to comment on a property.
- Display all the comments made on a property to each registered user.
- Provide rating facility to rate properties.
- Provide an option to report about an abuse/illegal property.
- Provide an option to administrator to block the user, after receiving a certain number of complaints from different users.
- Provide search facility based on different criteria (...).
- Rank properties based on the time they were posted.
- Provide an option to view more ads of a particular advertiser, if someone likes his post.
- Provide an option to follow a user.

2.4 User Characteristics

User of the system should be an internet literate.

2.5 Constraints

Following are some constraints which restricts developer's options:

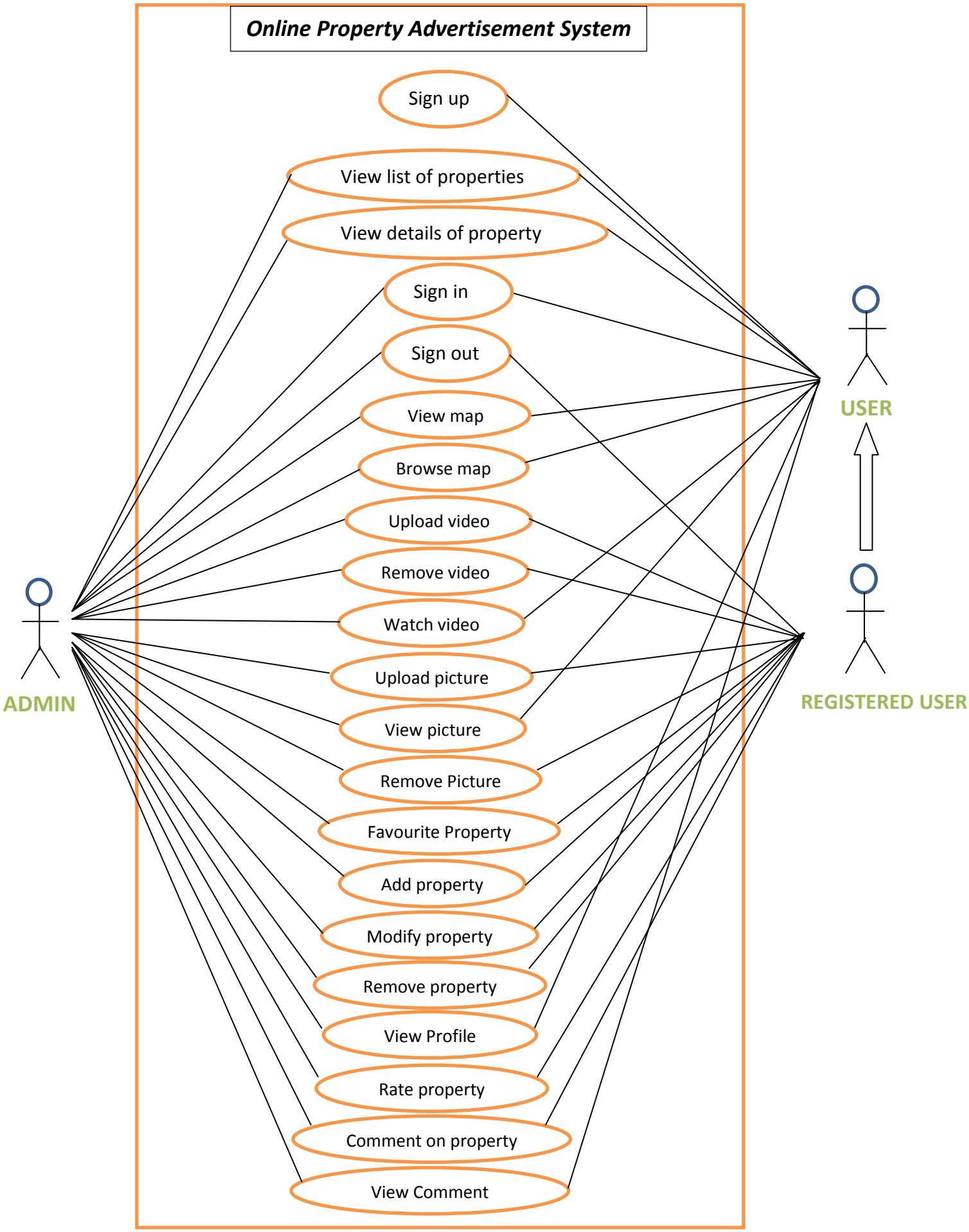
- Availability of some contents and efficiency of system services depends on backend API's services which system is using.
- Internet availability is must, to run system.

Chapter # 3

Requirements Analysis

This chapter describes requirements in the form of use cases. A use case diagram shows all use cases and their interaction with primary and other actors. After use case diagram, all use cases are written in detail, where detail includes use case name, primary actor, pre-condition, post-condition, steps, extensions and actions for those extensions.

3.1 Use case diagram:



3.2 Use cases Description

U.C No.	1
Name	Sign up.
Primary actor	User
Pre-condition	---
Post-condition	User has been registered.
Steps	1. User selects sign up. 2. System displays registration form. 3. User enters full name. 4. User enters email address. 5. User enters password. 6. Re-enters the password for confirmation. 7. User enters contact number. 8. User enters country. 9. User enters city. 10. After submitting the form, system will store the user's information.
Extensions	i- User may leave some important information. ii- User may enter invalid values or already registered.
Actions	i- System will alert again null values. ii- System will alert against invalid information, and if user is already registered, the registration process will be cancelled.

U.C No.	2
Name	View list of properties
Primary actor	User, Administrator.
Pre-condition	-
Post-condition	User/admin has viewed the list of properties.
Steps	1. User/admin opens the website. 2. User/admin selects the search by type. 3. System displays the list of properties.
Extensions	-

U.C No.	3
Name	View details of property
Primary actor	Registered user, admin.
Pre-condition	User/admin must be signed In.
Post-condition	User/admin has viewed the details of property.
Steps	1. User/admin selects the search by type. 2. System displays the list of properties. 3. User/admin selects a property for details. 4. System displays all the information stored about property.
Extensions	-

U.C No.	4
Name	Sing in
Primary actor	Admin, registered user
Pre-condition	User/Admin is registered.
Post-condition	User/admin is signed in.
Steps	1. Registered user/Admin enters email address. 2. Registered user/Admin enters password. 3. Registered user/Admin selects Sign in. 4. System will display the new interface.
Extensions	1. Email address is invalid. 2. Password is incorrect.
Actions	1. System will display an appropriate error message. 2. System will display an appropriate error message.

U.C No.	5
Name	Sign out
Primary actor	Registered user/Administrator.
Pre-condition	Administrator/registered user is signed in.
Post-condition	Signed out successfully.
Steps	1. Registered user/admin selects sign out.
Extensions	---

U.C No.	6
Name	Add property.
Primary actor	Registered user.
Pre-condition	User must be signed in.
Post-condition	Property is added successfully.
Steps	1. User selects add property. 2. System displays a form. 3. User enters property type. 4. User enters area of property. 5. User enters price. 6. User enters specification of the property. 7. User enters location of the property. 8. User selects Add property. 9. System stores the property details.
Extensions	-User enters invalid or miss some information.
Actions	-System will alert against invalid values.

U.C No.	7
Name	Modify property.
Primary actor	Registered user.
Pre-condition	Registered User has performed U.C No.3.
Post-condition	New data is stored successfully.
Steps	1. System displays the property details. 2. User changes the necessary information. 3. User selects 'Update property'. 4. System stores the details.
Extensions	---

U.C No.	8
Name	Remove property.
Primary actor	Registered user/admin.
Pre-condition	Registered User/admin must be signed in.
Post-condition	Property is deleted.
Steps	1. Registered user/admin selects property list. 2. System displays property list. 3. User/admin selects the desired property. 4. User/admin selects 'delete property'.
Extensions	---

U.C No.	9
Name	View map.
Primary actor	Registered User/admin.
Pre-condition	Registered user/admin has performed U.C No.3
Post-condition	Map is displayed.
Steps	1. Registered user/admin selects property's details. 2. System displays the map along with details.
Extensions	---

U.C No.	10
Name	Browse map.
Primary actor	Registered User/admin.
Pre-condition	Registered user/admin has performed U.C No.9
Post-condition	Map contents changed.
Steps	1. Registered user/admin selects map. 2. Registered user/admin spins the map. 3. System shows the new location on map.
Extensions	---

U.C No.	11
Name	Upload video.
Primary actor	Registered User.
Pre-condition	User has performed U.C No.6
Post-condition	Video is uploaded.
Steps	1. Registered user selects 'Upload video'. 2. User will provide a video on Youtube, and give the URL. 3. System stores the video URL.
Extensions	2. User may provide a URL, which does not exists.

U.C No.	12
Name	Watch video.
Primary actor	Registered User/admin.
Pre-condition	User has performed U.C No.11.
Post-condition	Video is watched successfully.
Steps	1. Registered user/admin selects property's details. 2. System displays the details along with video. 3. Registered user/admin selects 'play video'. 4. System starts playing video.
Extensions	2. System will not display the video.
Actions	2. Video is not uploaded.

U.C No.	13
Name	Remove video.
Primary actor	Registered User/admin.
Pre-condition	User/admin must be signed in.
Post-condition	Video is deleted.
Steps	1. Registered user/admin selects property's details. 2. System displays property's details along with video. 3. Registered user/admin selects 'delete video'. 4. System removes the video.
Extensions	2. System will not display the video.
Actions	2. Video is not uploaded.

U.C No.	14
Name	Upload picture(s).
Primary actor	Registered User.
Pre-condition	User has performed U.C No.6
Post-condition	Picture(s) is/are uploaded.
Steps	1. Registered user selects 'browse picture' 2. System displays an interface to upload picture. 3. User provides a picture. 4. User selects 'Upload picture'. 5. System stores the picture(s).
Extensions	3. User may provide another picture.
Actions	3. System will store all the subsequent pictures.

U.C No.	15
Name	View picture(s).
Primary actor	Registered User/admin.
Pre-condition	User has performed U.C No.14.
Post-condition	Picture(s) is/are viewed successfully.
Steps	1. Registered user/admin selects property's details. 2. System displays the details along with picture(s). 3. Registered user/admin selects picture.
Extensions	2. System will not display the picture, if not uploaded.

U.C No.	16
Name	Delete picture(s).
Primary actor	Registered User/admin.
Pre-condition	User has performed U.C No.14.
Post-condition	Picture is deleted.
Steps	1. Registered user/admin selects property's details. 2. System displays property's details along with pictures. 3. Registered user/admin selects a picture. 4. Registered user/admin selects 'delete picture'. 5. System removes the picture.
Extensions	2. System will not display the picture, if not uploaded.

U.C No.	17
Name	Favourite Property.
Primary actor	Registered User.
Pre-condition	User has performed U.C No.3.
Post-condition	User is followed.
Steps	1. Registered user/admin selects property's details. 2. System displays property's details. 3. Registered user selects 'Favourite Property'.
Extensions	---

U.C No.	18
Name	View Profile.
Primary actor	Admin.
Pre-condition	---
Post-condition	User is blocked.
Steps	1. Admin searches the user. 2. System displays the user. 3. Admin selects 'View Profile'. 4. System will display the user.
Extensions	3. User is already blocked.

U.C No.	19
Name	Rate property.
Primary actor	Registered User.
Pre-condition	User has performed U.C No.3.
Post-condition	Property is rated.
Steps	1. Registered user selects property's details. 2. System displays property's details. 3. Registered user gives the rating to property.
Extensions	3. user will give no ratings.
Actions	3. No effect on property's ratings.

U.C No.	20
Name	Comment on property.
Primary actor	Registered User.
Pre-condition	User has performed U.C No.3.
Post-condition	Comment added.
Steps	1. Registered user selects property's details. 2. System displays property's details. 3. Registered user writes comments on property. 4. Registered user selects 'Comment'. 5. System stores the comments.
Extensions	3. User does not write comments.
Actions	3. System will alert for null values.

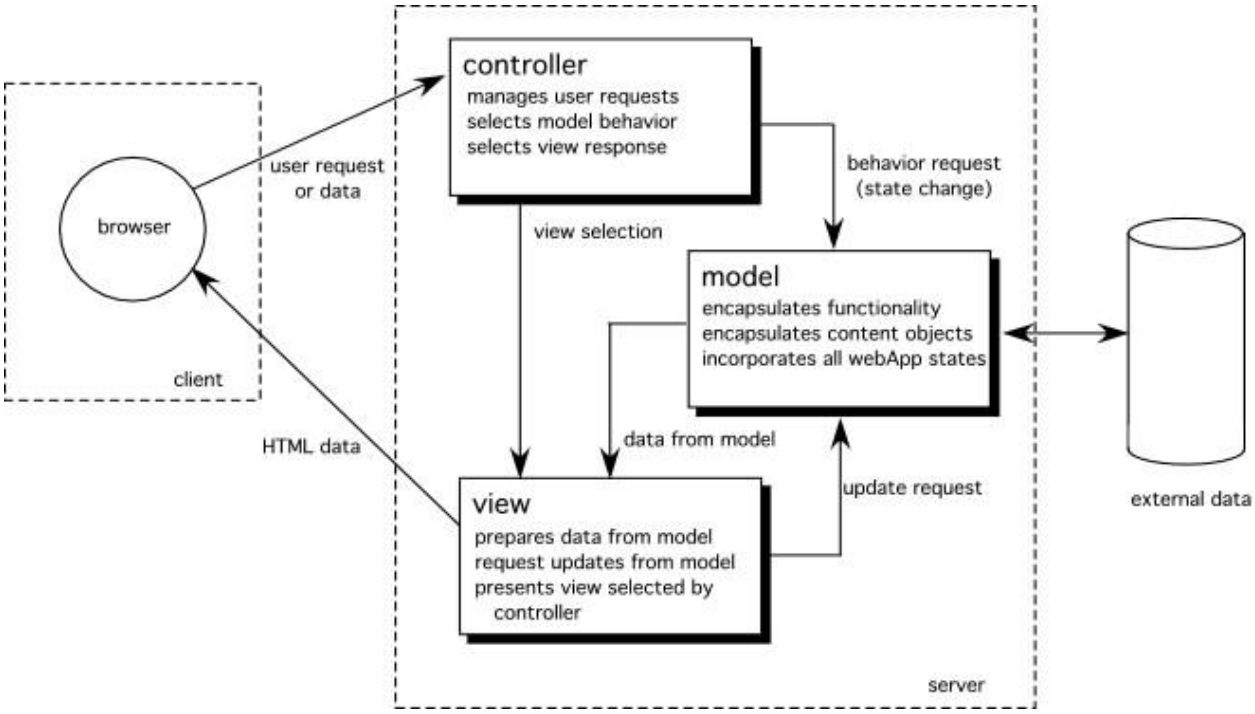
U.C No.	21
Name	View comments.
Primary actor	Registered User/admin.
Pre-condition	User/admin has performed U.C No.3.
Post-condition	Comment viewed.
Steps	1. Registered user/admin selects property's details. 2. System displays property's details along with all comments. 3. Registered user/admin views comments.
Extensions	---

Chapter # 4

System Design

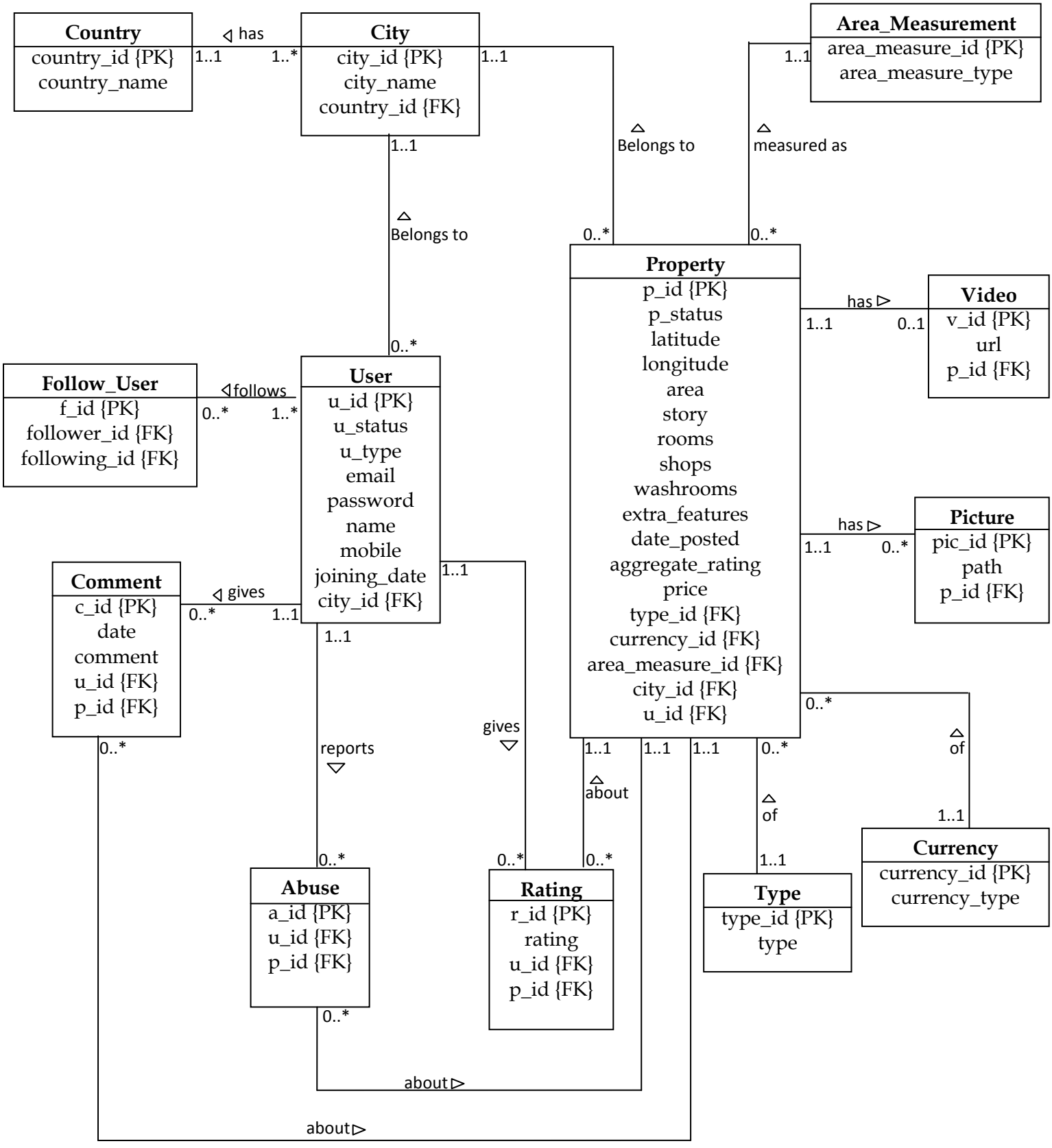
This chapter contains design of OPAS in the form of diagrams. At start architecture diagram shows basic structure/architecture of system and some classes are identified, which are helpful in understanding system at abstract level. Furthermore, class diagram is shown, which shows all classes used in system and also their interaction with each other, after that Entity Relationship Diagram and Logical Relational Design is shown, which shows entities/tables and attributes used in the database of this system and their relations with each other. At the end of chapter some design highlights are being discussed.

4.1 Architecture Diagram

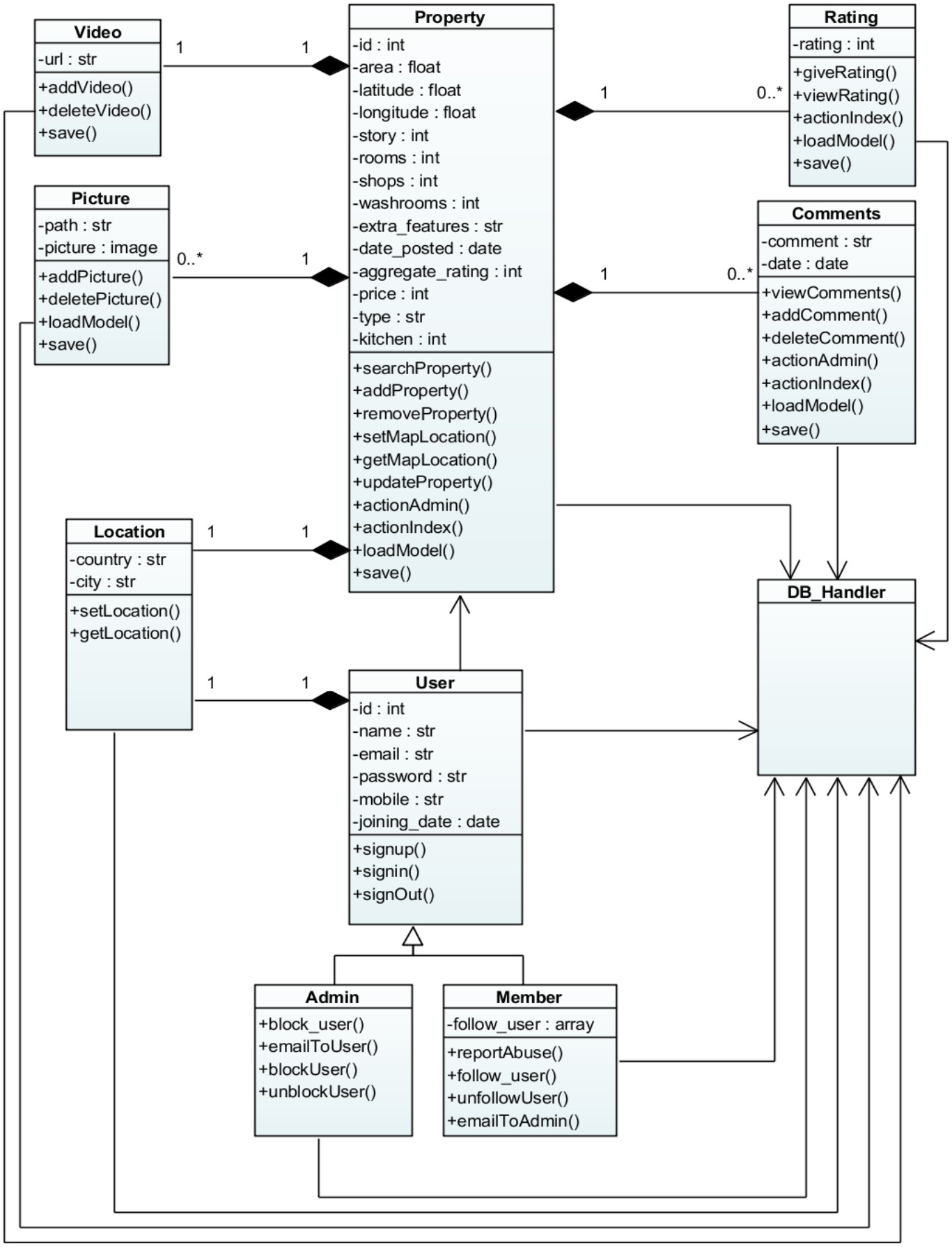


MVC Architecture Diagram [Pressman 2008]

4.2 Entity Relationship Diagram



4.3 Class Diagram



4.4 Logical Relational Design

Property (p_id, p_status, latitude, longitude, area, story, rooms, shops, washrooms, extra_features, date_posted, aggregate_rating, price)

Primary key p_id

Foreign key city_id **references** City(city_id)

Foreign key u_id **references** User(u_id)

Foreign key area_measure_id **references** Area_Measurement(area_measure_id)

Foreign key currency_id **references** Currency(currency_id)

Foreign key type_id **references** Type(type_id)

User (u_id, u_status, u_type, email, password, name, mobile, joining_date)

Primary key u_id

Foreign key city_id **references** City(city_id)

Country (country_id, country_name)

Primary key country_id

City (city_id, city_nane)

Primary key city_id

Foreign key country_id **references** Country(country_id)

Area_Measurement (area_measure_id, area_measure_type)

Primary key area_measure_id

Video (v_id, url)

Primary key v_id

Foreign key p_id **references** Property(p_id)

Picture (pic_id, path)

Primary key pic_id

Foreign key p_id **references** Property(p_id)

Currency (currency_id, currency_type)

Primary key currency_id

Type (type_id, type)

Primary key type_id

Rating (r_id, rating)

Primary key r_id

Foreign key p_id **references** Property(p_id)

Foreign key u_id **references** User(u_id)

Abuse (a_id)

Primary key a_id

Foreign key p_id **references** Property(p_id)

Foreign key u_id **references** User(u_id)

Comment (c_id, date, comment)

Primary key c_id

Foreign key p_id **references** Property(p_id)

Foreign key u_id **references** User(u_id)

Follow_User (f_id)

Primary key f_id

Foreign key follower_id **references** User(u_id)

Foreign key following_id **references** User(u_id)

4.5 Design Highlights

While making system it is desirable that system will tolerate changes efficiently and easily. So keeping that in mind there is a need of an approach which keeps presentation logic, business logic and data logic separate. Therefore MVC approach is considered to implement this system.

4.6 MVC Approach

According to this approach system is divided into three tiers Model, View and Controller. View encompasses those classes to which user interacts directly, which handles events generated by user and gives response to user after interacting with other tiers. Controller helps Interface Handler to communicate to Model. Model encompasses those classes which interact with database. Controller directly interacts to model classes to communicate with database. Model classes communicate with database. Database gives results to model class which passes those results to Controller and finally Controller passes response to View (user interface).

Chapter # 5

System Implementation

This chapter is related to system implementation. This chapter mentions the tools, framework and API's used to develop OPAS. Also some code screen shots are given in the end.

5.1 Introduction

After the design phase, the implementation phase comes. In this phase we decide how to implement our design and which techniques to use.

5.2 Coding Guidelines

For implementation following coding guidelines are used:

- **Pascal Case**

  Pascal case means that the first letter in each word in an identifier is capitalized. As a general rule Pascal case is used everywhere except for parameters, functions and attributes names. Here Pascal case is used for class names. As,

  PropertyDescription

  FollowProperty

- **Camel Case**

  Camel case is similar to Pascal case except the first letter is in lower case. Camel case is used for parameters, functions and attributes names. As,

  propertyDescription

  followProperty

5.3 Language Selection

PHP is selected to implement system.

PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page.

Why PHP?

Easy

PHP pages are easy to write and maintain because the source code and HTML are together.

5.4 Framework Selection

Yii (Yes It Is) framework v1.1.10 is used to develop the system. Yii is a high performance PHP framework best for developing Web 2.0 applications. Yii comes with rich features: MVC, DAO/ActiveRecord, caching, authentication and role-based access control, scaffolding, testing, etc. It can reduce development time significantly.

5.4.1 Fast

Yii only loads the features that we need. It has powerful caching support. It is explicitly designed to work efficiently with AJAX.

5.4.2 Secure

Security comes as standard with Yii. It includes input validation, output filtering, SQL injection and Cross-site scripting prevention.

5.4.3 Professional

Yii helps to develop clean and reusable code. It follows the MVC pattern, ensuring a clear separation of logic and presentation.

5.4.4 Caching

The source code is compiled the first time the page is requested. The server saves the compiled version of the page to use next time the page is requested.

5.5 Software Used

Notepad++

Notepad**++** is a text editor and source code editor for Windows. One advantage of Notepad++ over the built-in Windows text editor, Notepad, is tabbed editing, which allows working with multiple open files.

MySQL

MySQL is a relational database server. It was first developed by Sun Microsystems and currently being managed by Oracle Corporation who bought Sun Microsystems. It is open source database software whose primary function is to store and retrieve data as requested by other software applications.

Wamp Server

WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser.

5.6 Code Snapshots

Video Model

```php
<?php

class Video extends CActiveRecord
{

    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }


    public function tableName()
    {
        return 'video';
    }


    public function rules()
    {
        return array(
        array('v_id, url, p_id', 'safe', 'on'=>'search'),
        );
    }


    public function relations()
    {
        return array(
            'property' => array(self::BELONGS_TO, 'Property', 'p_id'),
        );
    }


    public function attributeLabels()
    {
        return array(
            'v_id' => 'Video ID',
            'url' => 'YouTube Video Url',
            'p_id' => 'Property ID',
        );
    }


    public function search()
    {

        $criteria=new CDbCriteria;

        $criteria->compare('v_id',$this->v_id);
        $criteria->compare('url',$this->url,true);
        $criteria->compare('p_id',$this->p_id);

        return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
        ));
    }
}
```

Property Model

```php
class Property extends CActiveRecord
{



    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    public function tableName()
    {
        return 'property';
    }

    public function rules()
    {
        return array(
            array('status, latitude, longitude, type_id, area, date, aggregate_rating, price, u_id', 'required'),
            array('status, type_id, story, rooms, shops, washrooms, aggregate_rating, price, 'numerical', 'integerOnly'=>true),
            array('latitude, longitude, area', 'numerical'),
            array('extra_features', 'safe'),
            array('p_id, status, latitude, longitude, type_id, area, story, rooms, shops, washrooms, price, 'safe', 'on'=>'search'),
        );
    }


    public function relations()
    {
        return array(
            'abuses' => array(self::HAS_MANY, 'Abuse', 'p_id'),
            'comments' => array(self::HAS_MANY, 'Comment', 'p_id'),
            'pictures' => array(self::HAS_MANY, 'Picture', 'p_id'),
            'user' => array(self::BELONGS_TO, 'User', 'u_id'),
            'currency' => array(self::BELONGS TO, 'Currency', 'currency id'),
            'type' => array(self::BELONGS_TO, 'Type', 'type_id'),
            'city' => array(self::BELONGS_TO, 'City', 'city_id'),
            'area_measurement' => array(self::BELONGS_TO, 'AreaMeasurement', 'area_measurement_id'),
            'ratings' => array(self::HAS_MANY, 'Rating', 'p_id'),
            'videos' => array(self::HAS_MANY, 'Video', 'p_id'),
        );
    }


    public function attributeLabels()
    {
        return array(
            'p_id' => 'Property ID',
            'status' => 'Status',
            'latitude' => 'Latitude',
            'longitude' => 'Longitude',
            'type_id' => 'Type',
            'area' => 'Area',
            'story' => 'Story',
            'rooms' => 'Rooms',
            'shops' => 'Shops',
            'washrooms' => 'Washrooms',
            'extra_features' => 'Extra Features',
            'date' => 'Date & Time',
            'aggregate_rating' => 'Aggregate Rating',
            'price' => 'Price',
            'currency_id' => 'Currency',
            'city_id' => 'City',
            'area_measurement_id' => 'Area Measured In',
            'u_id' => 'User',
        );
    }
```

```php

    public function search()
    {

        $criteria=new CDbCriteria;

        $criteria->compare('p_id',$this->p_id);
        $criteria->compare('status',$this->status);
        $criteria->compare('latitude',$this->latitude);
        $criteria->compare('longitude',$this->longitude);
        $criteria->compare('type_id',$this->type_id);
        $criteria->compare('area',$this->area);
        $criteria->compare('story',$this->story);
        $criteria->compare('rooms',$this->rooms);
        $criteria->compare('shops',$this->shops);
        $criteria->compare('washrooms',$this->washrooms);
        $criteria->compare('extra_features',$this->extra_features,true);
        $criteria->compare('date',$this->date,true);
        $criteria->compare('aggregate_rating',$this->aggregate_rating);
        $criteria->compare('price',$this->price);
        $criteria->compare('currency_id',$this->currency_id);
        $criteria->compare('area_measurement_id',$this->area_measurement_id);
        $criteria->compare('city_id',$this->city_id);
        $criteria->compare('u_id',$this->u_id);

        return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
        ));
    }


    public function save()
    {
        $this->setAttribute('u_id', Yii::app()->user->id);
        $this->setAttribute('aggregate_rating', 0);
        $this->setAttribute('date', date('Y-m-d H:i:s', time()));
        $this->setAttribute('status', 1);
        $this->setAttribute('latitude' , 1234.123456);
        $this->setAttribute('longitude', 2134.214365);

        return ( parent::save() );
    }
}
```

Property Controller

```
public function actionCreate()
{
    $model=new Property;
    $picture = new Picture;
    $video = new Video;



    if(isset($_POST['Property']))
    {
        $model->attributes=$_POST['Property'];

        if($model->save())
        {
            if( isset($_POST['Video']) )
            {
                $url = $_POST['Video']['url'];
                $pattern = '/^(http(s)?:\/\/)?www\.youtube\.com\/watch\?v=[\w-]+(&feature=\w+)?$/';

                if( preg_match($pattern , $url) )
                {
                    $result = preg_split('/[=&]/' , $url);

                    $video->url = $result[1];
                    $video->p_id = $model->p_id;
                    $video->save();
                }
            }

            $images = CUploadedFile::getInstancesByName('images');  //getting images uploaded by user...

            if(isset($images) && count($images) > 0)
            {
            $dir = Yii::getPathOfAlias('webroot').'/images/'.$model->p_id.'/';
            $this->createDIR($dir);

            foreach( $images as $i => $image )
            {

                if( $image->saveAs($dir.$image->name) ) //this strores the picture in every loop's interation, and returns true...
                {
                    $picture2 = new Picture;            //CActiveRecord stores one row object in one model... untill it is unset and set again.
                    $picture2->path ='images/'.$model->p_id.'/'.$image->name;
                    $picture2->p_id = $model->p_id;
                    $picture2->save();

                }

            }

        }

        $this->redirect(array('view','id'=>$model->p_id));

    }

}

$this->render('create',array(
    'model'=>$model,
    'picture'=>$picture,
    'video'=>$video,
));
```

Chapter # 6

System Testing

This chapter contains User Acceptance Tests, which are built against use cases stated in Requirement Analysis.

6.1 Objectives

This UAT (User Acceptance Test) will be helpful to verify requirements finalized in SRS and check whether all the requirements have been fulfilled or not.

6.2 Scope

- UAT will cover all requirements specified in the form of use cases.
- UAT will provide test cases against each use case.
- Requirements described in SRS can be cross checked using this specification
- UAT will provide: test case ID, description, date against each test case, input, expected output, actual output, procedure to perform test case, and pass/fail status.

6.3 Test Cases

Test Case ID	TC01 (Sign up)
Description	This use case allows user to register with the system.
Tester	User
Date	17-7-2012
Setup	Open the 'Register' page from the OPAS website.
Procedure	1- Type an email address e.g. "abc@xyz.com" in the field. 2- Type a password "user". 3- Type contact number "03001234567". 4- Select country name. 5- Select city name. 6- Press 'Register' button.
Expected Result	User is registered successfully.
Actual Result	User got registered.
Status (Pass/Fail)	Passed

Test Case ID	TC02 (View list of properties)
Description	This use case allows user to view the properties of his/her interest.
Tester	User
Date	17-7-2012
Setup	Open the OPAS website, click on property tab.
Procedure	1. Specify search criteria e.g. "house". 2. Specify the location "Lahore". 3. Specify the price range "200000 – 500000". 4. Press search button.
Expected Result	House in Lahore with a price range of 200000 – 500000.
Actual Result	All houses list.
Status (Pass/Fail)	Failed.

Test Case ID	TC03 (View details of property)
Description	This use case allows user to view details of a particular property.
Tester	User
Date	17-7-2012
Setup	Properties' list has been displayed.
Procedure	1. Select a property for details. 2. Click on 'details'.
Expected Result	Property's detail has been showed.
Actual Result	Property's detail has been showed.
Status (Pass/Fail)	Passed

Test Case ID	TC04 (Sign in)
Description	This use case allows user to enter in the members area of the system.
Tester	User
Date	17-7-2012
Setup	Open website.
Procedure	1. Type a username e.g. “abc@xyz.com” in the field. 2. Type a valid password “user”. 3. Press login button.
Expected Result	User signed in.
Actual Result	User signed in.
Status (Pass/Fail)	Passed

Test Case ID	TC05 (Sign out)
Description	This use case allows user to log out of the system.
Tester	User
Date	17-7-2012
Setup	User signs in himself/herself.
Procedure	1. Select ‘logout’ button.
Expected Result	Logged out from the system.
Actual Result	Logged out from the system.
Status (Pass/Fail)	Passed

Test Case ID	TC06 (View map)
Description	This use case allows user to view properties on a map.
Tester	User
Date	18-7-2012
Setup	Open the OPAS website.
Procedure	1. Select ‘property’ tab. 2. Select ‘search’ button.
Expected Result	Map interface
Actual Result	Map interface
Status (Pass/Fail)	Passed

Test Case ID	TC07 (browse map)
Description	This use case allows user browse map for search properties through map.
Tester	User
Date	17-7-2012
Setup	Properties are searched and map is displayed.
Procedure	1. Hold left mouse button on map. 2. Navigate to change the contents/browse the map.
Expected Result	Map interface
Actual Result	Map interface
Status (Pass/Fail)	Passed

Test Case ID	TC08 (Upload video)
Description	This use case allows user to upload video along with property's specification.
Tester	User
Date	17-7-2012
Setup	Sign in to the system. And 'add property' is selected.
Procedure	1. Open YouTube video of your concern. 2. Copy the URL from the browser. 3. Paste the URL in the video field, provided by the system. 4. Press 'Add property' button.
Expected Result	Video showed on page.
Actual Result	Video showed on page.
Status (Pass/Fail)	Passed

Test Case ID	TC09 (Remove video)
Description	This use case allows user to remove video from property's specification.
Tester	User
Date	17-7-2012
Setup	Sign in to the system. Select 'my listings'.
Procedure	1. Open property whose video needs to be removed. 2. Selects 'remove video' link.
Expected Result	Video removed.
Actual Result	Video removed.
Status (Pass/Fail)	Passed

Test Case ID	TC10 (Watch video)
Description	This use case allows users to watch video of a property.
Tester	User
Date	17-7-2012
Setup	Search properties.
Procedure	1. Selects 'details' of a property whose video is available. 2. Click on play icon of YouTube player.
Expected Result	Video on player
Actual Result	Video on player
Status (Pass/Fail)	Passed

Test Case ID	TC11 (Upload picture)
Description	This use case allows user to upload picture(s) along with property's specification.
Tester	User
Date	15-7-2012
Setup	Open 'add property' form.
Procedure	1. Select 'image' button. 2. Browse picture to be uploaded. 3. Select 'ok'. 4. Select 'add property'.
Expected Result	Picture(s) uploaded and displayed.
Actual Result	Picture(s) uploaded and displayed.
Status (Pass/Fail)	Passed

Test Case ID	TC12 (Remove picture)
Description	This use case allows user to remove picture(s) from his/her uploaded property.
Tester	Sign in to the system. Select 'my listings'.
Date	15-7-2012
Setup	Select
Procedure	1. Open property whose picture(s) needs to be removed. 2. Selects 'remove picture' link along with each picture.
Expected Result	Picture is removed.
Actual Result	Picture is removed.
Status (Pass/Fail)	Passed

Test Case ID	TC13 (View picture)
Description	This use case allows users to view the pictures .
Tester	User
Date	15-7-2012
Setup	Search properties.
Procedure	1. Selects 'details' of a property whose pictures are available. 2. Click on a picture to show.
Expected Result	Picture
Actual Result	Picture
Status (Pass/Fail)	Passed

Test Case ID	TC14 (Follow user)
Description	This use case allows users to follow other users, for email alerts when they post a new property.
Tester	User
Date	15-7-2012
Setup	Search properties to show property's list.
Procedure	1. Select property's details. 2. Click on 'follow user' button.
Expected Result	User is followed.
Actual Result	User is followed.
Status (Pass/Fail)	Passed

Test Case ID	TC15 (Add property)
Description	This use case allows user to upload his/her property's specification for advertising.
Tester	User
Date	15-7-2012
Setup	Sign in to the system.
Procedure	1. Select 'Property' tab. 2. Select 'add property'. 3. Type the necessary information in the fields. 4. Click on 'add property' button.
Expected Result	Property added and showed.
Actual Result	Property added and showed.
Status (Pass/Fail)	Passed

Test Case ID	TC16 (Remove property)
Description	This use case allows user to remove the property, uploaded by him/her.
Tester	User
Date	15-7-2012
Setup	Sign in to the system.
Procedure	1. Select 'My listings'. 2. Select the 'details' of the property, which needs to be deleted. 3. Click on 'delete' link.
Expected Result	Property removed, 'My listing' showed.
Actual Result	Property removed, 'My listing' showed.
Status (Pass/Fail)	Passed

Test Case ID	TC17 (Modify property)
Description	This use case allows user to modify property's details.
Tester	User
Date	15-7-2012
Setup	Sign in to the system.
Procedure	1. Select 'My listings'. 2. Select the 'details' of the property, which needs to be updated. 3. Modify the details of property with new values. 4. Select 'update property'.
Expected Result	Property updated, 'My listings' showed.
Actual Result	Property updated, 'My listings' showed.
Status (Pass/Fail)	Passed

Test Case ID	TC18 (Block user)
Description	This use case allows administrator to block the users, who posts illegal properties.
Tester	User
Date	15-7-2012
Setup	Administrator login.
Procedure	1. Select 'Abuse Reports' from menu. 2. List of users is showed. 3. Click on 'Block user' in the list of users.
Expected Result	User blocked.
Actual Result	User blocked.
Status (Pass/Fail)	Passed

Test Case ID	TC19 (Rate property)
Description	This use case allows user to rate an existing property.
Tester	User
Date	15-7-2012
Setup	Property is uploaded, and user is signed in.
Procedure	1. Select 'Property' tab. 2. Select 'details' of a property. 3. Select stars to rate the property. 4. click on 'Rate' button.
Expected Result	Rated stars.
Actual Result	Rated stars.
Status (Pass/Fail)	Passed

Test Case ID	TC20 (Comment on property)
Description	This use case allows user to comment on an existing property.
Tester	User
Date	15-7-2012
Setup	User is signed in and selects 'Property'.
Procedure	1. User selects property's details, on which he/she wants to comment. 2. User writes comments in a text area. 3. Selects 'Comment'.
Expected Result	New Comment displayed.
Actual Result	New Comment displayed.
Status (Pass/Fail)	Passed

Test Case ID	TC21 (View Comments)
Description	This use case allows user to view comments given on a property.
Tester	User
Date	15-7-2012
Setup	Selects 'Property' tab.
Procedure	1. Selects 'details' of a property. 2. All the comments about property are shown.
Expected Result	Comments
Actual Result	Comments
Status (Pass/Fail)	Passed

Test Case ID	TC22 (Report abuse)
Description	This use case allows user to report to admin about an abuse property.
Tester	User
Date	15-7-2012
Setup	User is logged in and selects an abusive property.
Procedure	1. Selects the 'Report as abuse' button. 2. A report is forwarded to admin.
Expected Result	Confirmation message.
Actual Result	Confirmation message.
Status (Pass/Fail)	Passed

Chapter # 7

Summary

This chapter summarizes OPAS regarding what functionality system is currently providing and what should be future enhancements.

7.1 Conclusion

OPAS is an Online Property Advertisement System, where different users from different countries can share their properties' information/specification with other interested users at any time, from anywhere, at one platform. It saves a lot of time and money of both parties, and information is available 24/7 days (24 hours and 7 days of a week). This system is developed to reduce the problems which users face when they want to buy a property.

In this system every registered and non-registered user can view all the information, but add/change is only allowed to registered users. Every registered user can upload their property's specification along with pictures and a YouTube video associated with the property. User can simply add pictures from his/her PC and video by providing the URL of YouTube video clip. Moreover, every registered user can comment on any property and see all comments made by himself/herself or by others. The purpose of comments is to share some information which is hidden from others by owner. Users can rate a property, if they like. If a user posts abusive things e.g. picture or video, he/she will be blocked by the administrator after receiving at least 5 complaints to that user. If someone has followed a user, whenever that user will post a property, an email will be sent to the follower.

Another major feature of the system is, Google Map. Map shows those properties in its region whose latitude and longitude resides inside its boundary. Users can search the properties through the map also. Whenever user selects a property marker from the Map, the details of the property will be shown. Moreover, if users navigate/browse the map, new properties will be shown according to the information of lat/long which are stored in OPAS database.

7.2 Future Enhancements

This system can be enhanced further in a lot of ways. As present system is using YouTube player & URL for videos, these videos can be stored in system's own database and its own player can be made with a lot of new features e.g. extract frames/snapshots from the video. Users will also be able to upload multiple videos along with a single property. Furthermore, system can be made to be able to communicate owners through SMS, as owners' cell number is stored in the system's database, using that number system will be able to send SMS from web to mobile. This feature will be useful for interested users.

7.3 Definitions, Acronyms and Abbreviations

Word	**Definition**
Lat/long	Latitude/longitude
YouTube	Online videos service provider
Google Maps	Online world map service
WAMP	A web server
Marla	Area measurement unit
Kanal	Area measurement unit

Abbreviation	**Meaning**
SRS	Software Requirements Specification
UC	Use Case
API	Application Programming Interface
SUD	System Under Discussion
UAT	User Acceptance Test
OPAS	Online Property Advertisement System

7.4 References

Chapter 1

- http://www.google.com/
- http://www.zameen.com/
- http://www.bastee.pk/
- http://maps.google.com/

Chapter 2

SRS is developed using IEEE recommended practices for SRS. This system uses API's i.e. Google Maps API provided by Google Inc, YouTube API provided by Google Inc.

Chapter 3

Larman, Craig - Applying UML and Patterns, 2nd Edition

Chapter 4

http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/mvc.html

Chapter 5

http://www.yiiframework.com/doc/

http://php.net/manual/en/

http://www.YouTube.com/

http://maps.google.com/

Chapter 6

www.bced.gov.bc.ca/imb/downloads/**uat**-test-process.pdf

http://www.tbs-sct.gc.ca/emf-cag/acceptance/proc/atp-per-eng.asp