Online Real Estate System

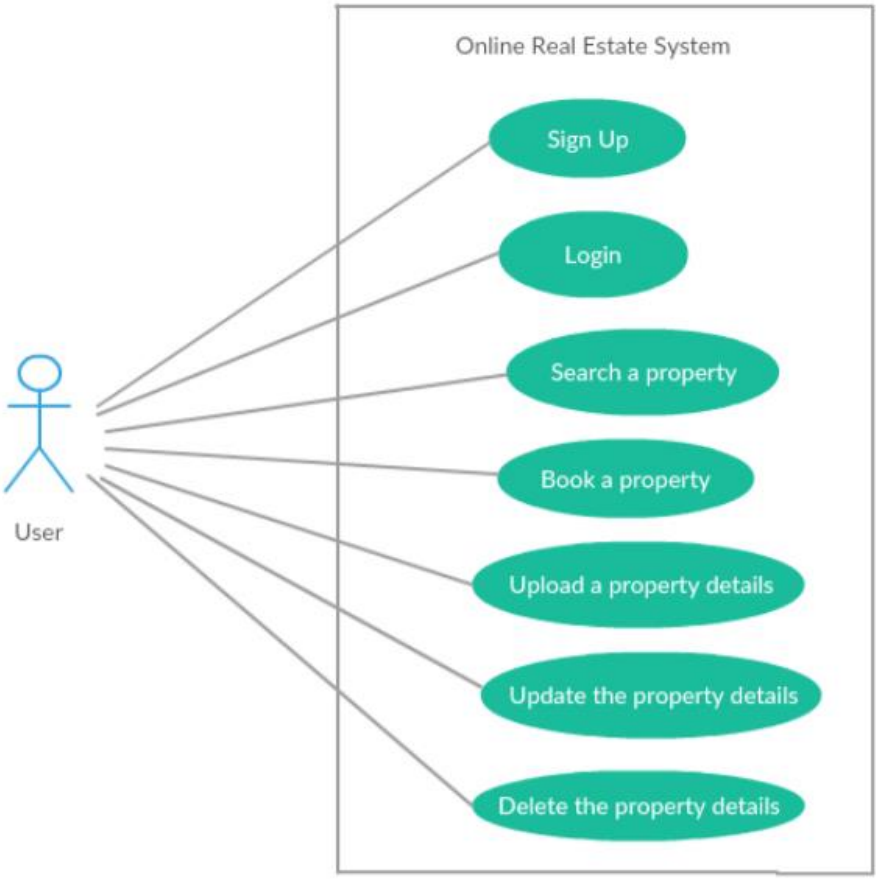
Abstract

Online Real Estate System is a project that will provide a platform to buy and sell the properties of people in a simple and effective way. Customers can search the property and browse the property from the system. Property can be houses, land and apartments. They can also book the property online. This application also allows on maintain and manage the details of the property.

Customer will have to register to the system by signing up and afterwards he/she can login to the system by using his username and password. Customer can view the details about property on the website and update it afterwards. There will be two options, whether to buy or sell the property. If user wants to buy a property, then he/she should contact to the agent and if the user wants to sell a property, then he/she can upload it to the system so that other users can look into their property and buy it with the help of agent.

1. REQUIREMENTS WORKFLOW

The following use case diagram depicts the requirements workflow in case of our project:



Brief Description(Signup)

The signup use case enables user to sign up to the real estate system by providing their identity details such as first name, last name, gender and so on in order to be able to use the interface online.

Step-by-Step description

1. Provide the user details in terms of
    - First name
    - Last Name
    - Gender
    - Date of Birth
    - User Name
    - Password

2. After successful signup process, the user is then able to log in to the system using email and password which was used to sign up.

Brief Description (login)

The login use case prompts the user to provide email and password and if login is successful, when a agents or users logs in, both the Users and agents are given access to an automated online

interface which has various functions and modules within this interface. The users can view the property details and agent contact.

Step-by-Step Description

1. The users or agents logs in by providing their email and password.
2. If the email and password matches, the user is directed to the online portal with various functions.
3. If the details do not match, the user is given a message that says INVALID USERNAME OR PASSWORD.

Brief Description (Search property)

User or clients can search for property after they login to the system. They can search for apartments, houses, or lands.

Step-by-Step Description

1. The user or agents search property either by typing city name, state or by zipcode.
2. If user types other thing than city name, state or zipcode, user will remain in the same page

Brief Description (Book Property)

If a user like or are interested in buying any one of the property, then they can contact agent to book the property. They will have to provide their name, email and phone number to the agent.

Step-by-Step Description

1. User search for interested property
2. User sends message to agent to buy a property

Brief Description (Upload Property Details)

Users who wants to sell their property can upload the details of their property. They can write information like, area, price, location and they can also upload the pictures of their property.

Step-by-Step Description

1. User upload image of the property
2. User types the other details of the property
3. User saves the property

Brief Description (Delete Property Details)

User can also delete the information about their property, that they don't want to sell any more and if it has already been sold.

Step-by-Step Description

1. User delete the property, that he no longer wants to rent or sell or if it has already been sold

2. ANALYSIS WORKFLOW

2.1 Object-Oriented Analysis

- Class Diagram

- CRC Cards
- Sequence Diagram

  The two aims of the analysis workflow are:
  1. Obtain a deeper understanding from the requirements
  2. Describe those requirements in a way that will result in a maintainable design and implementation.

Here, we explain the analysis workflow with the help of the class diagrams which we have described with the help of following two diagrams:

ENTITY CLASS MODELING

Noun Extraction:

It has two-stage process:

Stage 1: Concise problem definition

A user should be able to register his account and login into it. He should be able to search the property details that he is interested to buy. If user wants to sell his/her property, then he/she should also be able to upload the details of his property. If users like to make any changes to the uploaded property, he should able to update that as well. Users should also be able to delete the uploaded property. If a user is interested to buy a property, he should be able to book the property. User should also be able to contact agent by sending a message to an agent.

Stage 2: Identifying the nouns

A user should be able to register his account and login into it. He should be able to search the property details that he is interested to buy. If user wants to sell his/her property, then he/she should also be able to upload the details of his property of different types. He can upload the images of his property and fill out the details. If users like to make any changes to the uploaded property, he should able to update that as well. Users should also be able to delete the uploaded property. If a user is interested to buy a property, he should be able to book the property. User should also be able to contact agent by sending a message to an agent.

All the nouns are underlined above which gives us a way to categorize the three classes:

Entity Class: PropertyProfile, PropertyTypes, Image, register, login, Agent

Boundary Class: upload property, update property, delete property, User

Control Class: Search class

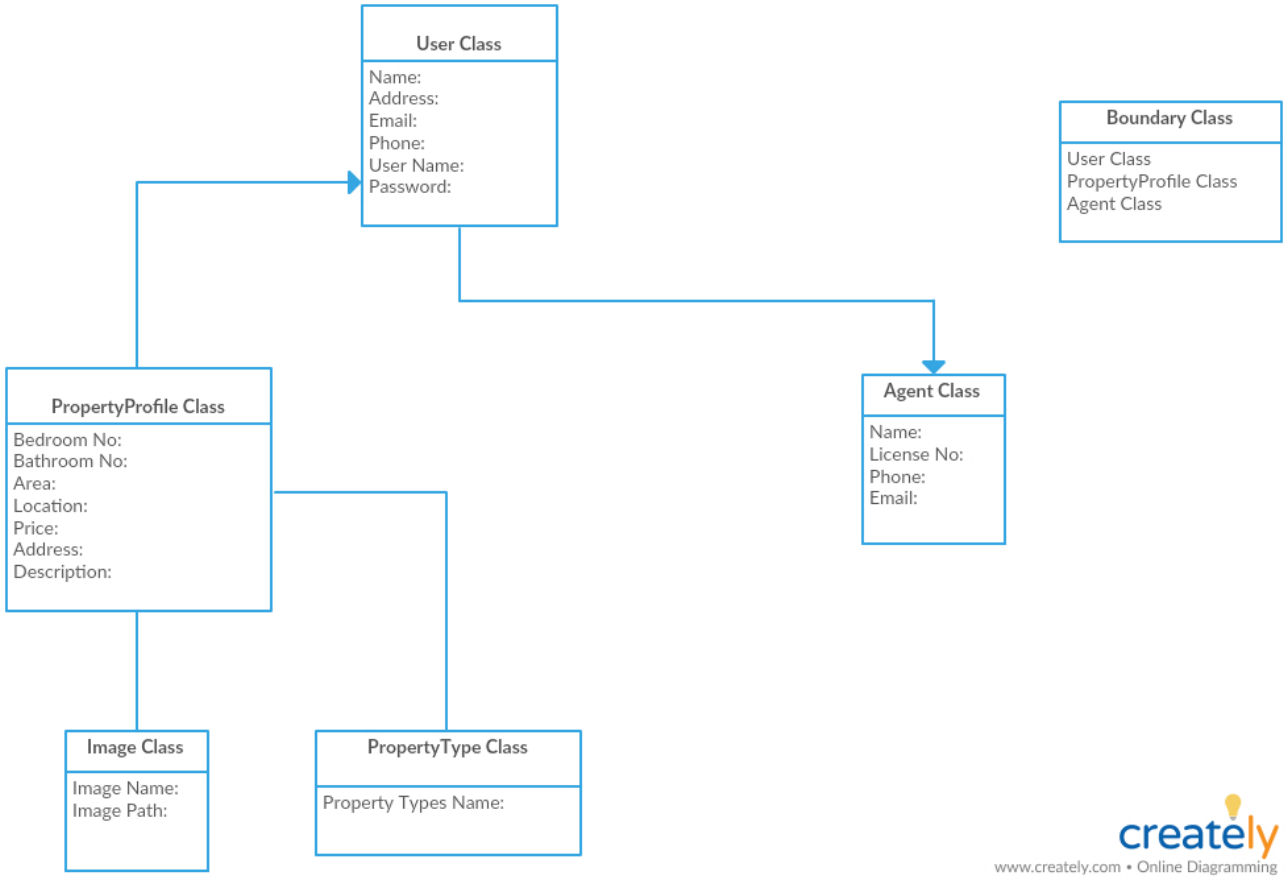
Class diagram

The major classes in our project are as follows:

1. Entity Classes
   - PropertyProfile Class
   - PropertyTypes Class
   - Image Class
   - AgentProfile Class

2. Boundary Classes
   - Agent interface Class
   - User Interface Class

Real Estate System Initial Class Diagram



CRC Card

CLASS **PropertyProfile Class**
RESPONSIBILITY Provides Images of property to users Determines Property types Gives other details of property
COLLABORATION Image Class PropertyTypes Class User Class

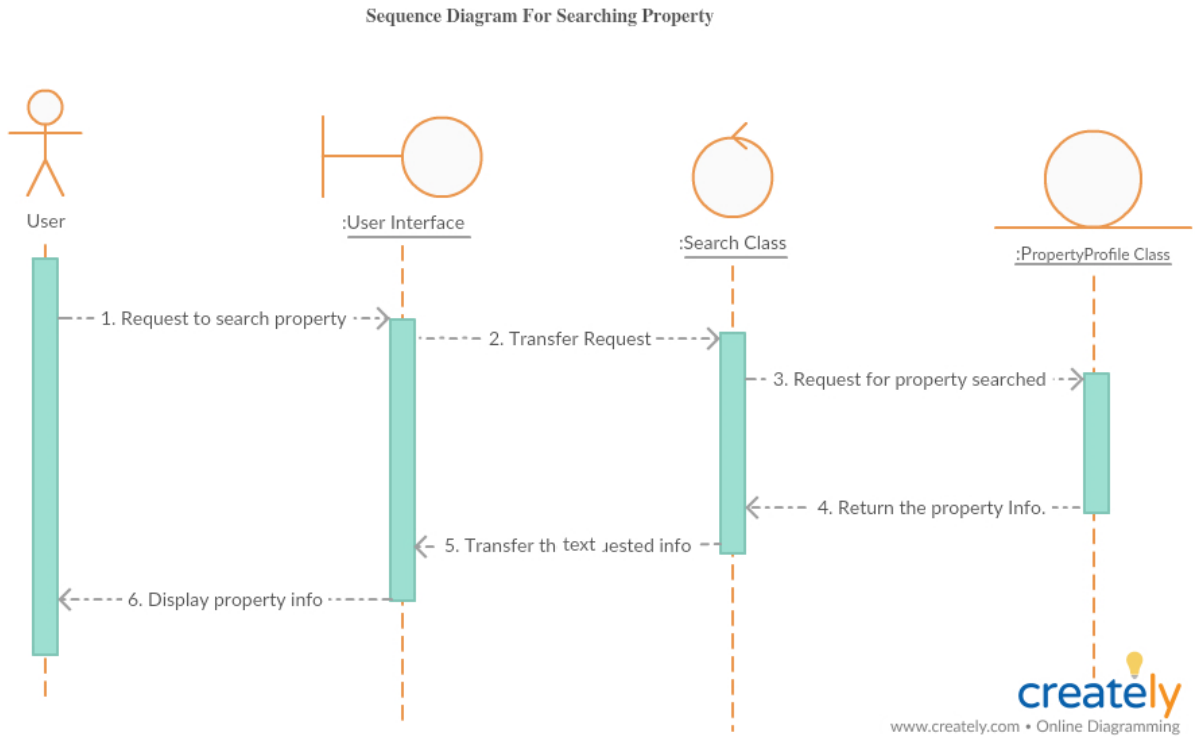
PropertyTypes Class
Determines the type of property
PropertyProfile Class User Class

Image Class
Provides multiple images of property
PropertyProfile Class User Class

AgentProfile Class
Gets request from users Contact users back to sell the property
User Class

Sequence Diagram
Sequence Diagram is equivalent to the collaboration diagram which shows the objects as well as the messages, numbered in the order in which they are sent in the specific scenario.

Search Property Details



The user want to make a request for searching property details. In the sequence diagram, this is modeled by message

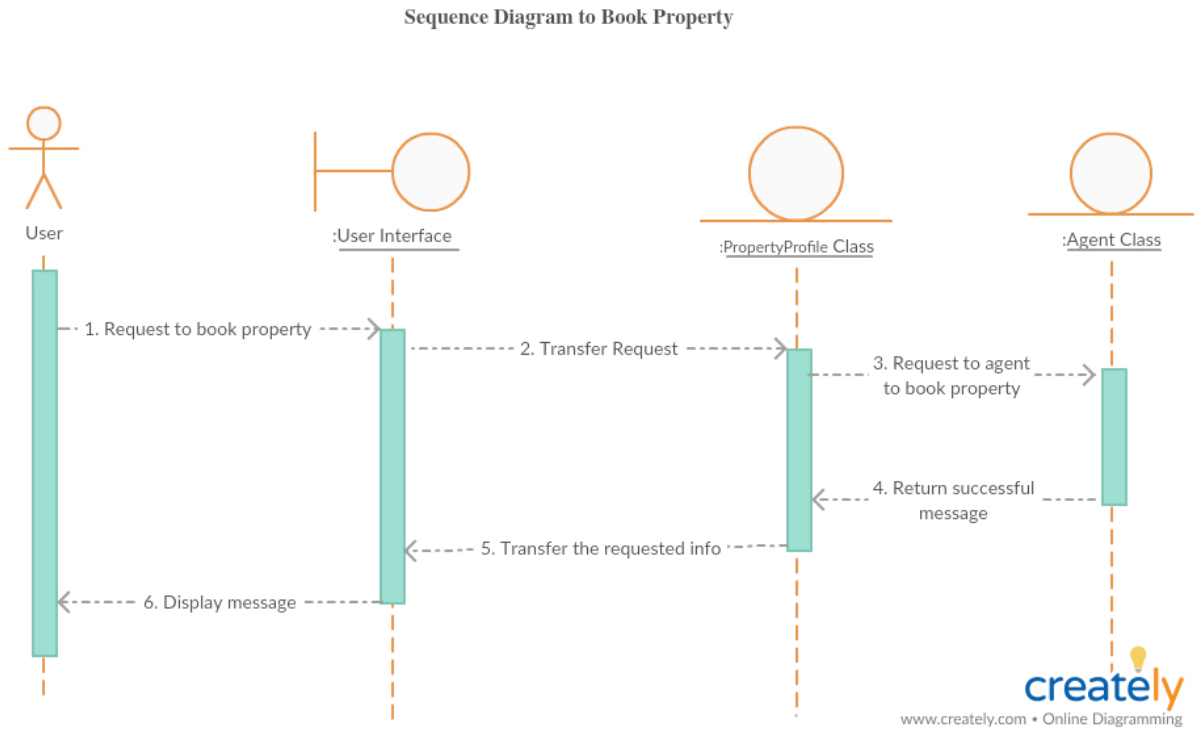
1. Request to search property details
   from USER to :User Interface Class an instance of User Interface Class
2. Transfer Request

This request is passed on to :Search Class , an instance of the control class PropertyProfile Class which does the actual searching inside by working with PropertyProfile Class.

The Search Class then makes a request to PropertyProfile Class. The message is then passsed from PropertyProfile Class to Search Class and then to User via User Interface Class in form of acknowledgement. This is modeled by the following messages:

3. Request for property searched
4. Return the property information
5. Transfer the property information
6. Display the property information

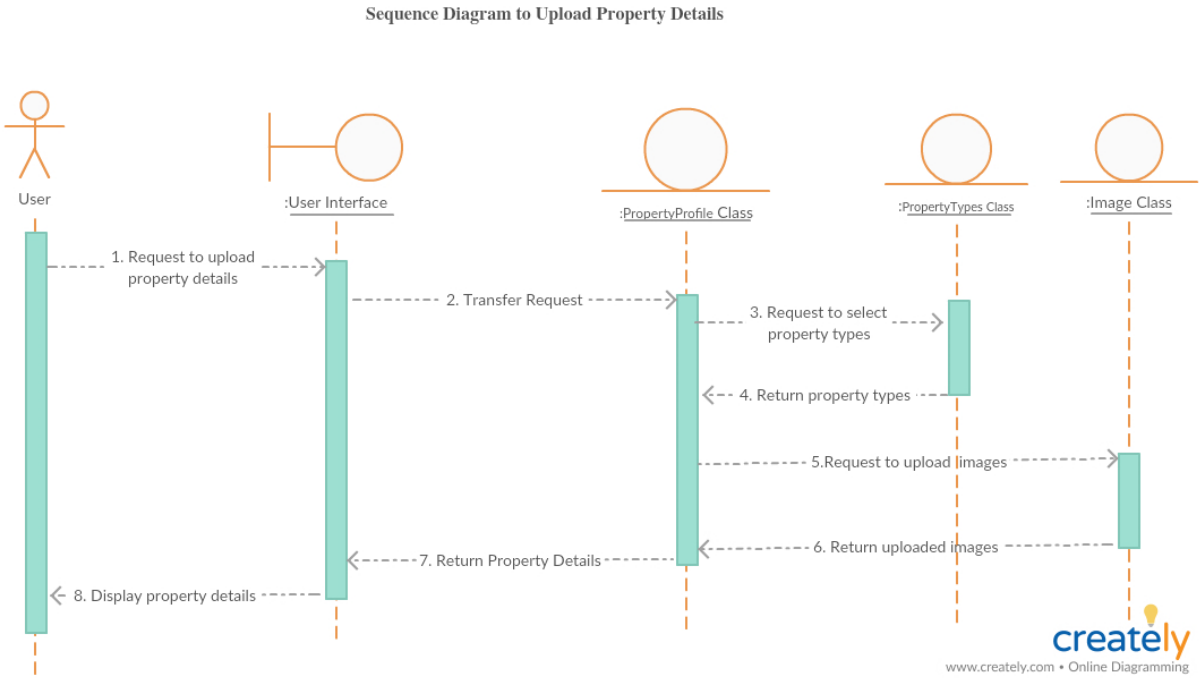
Book a property



If a user wants to buy a property, then first of all he should book it. Booking a property is modeled by message

1. Request to book a property
   from USER to :User Interface Class an instance of User Interface Class
2. Transfer request to :PropertyProfile Class
3. Request to an agent to book property
4. Return successful message
5. Transfer successful requested information
6. Display message

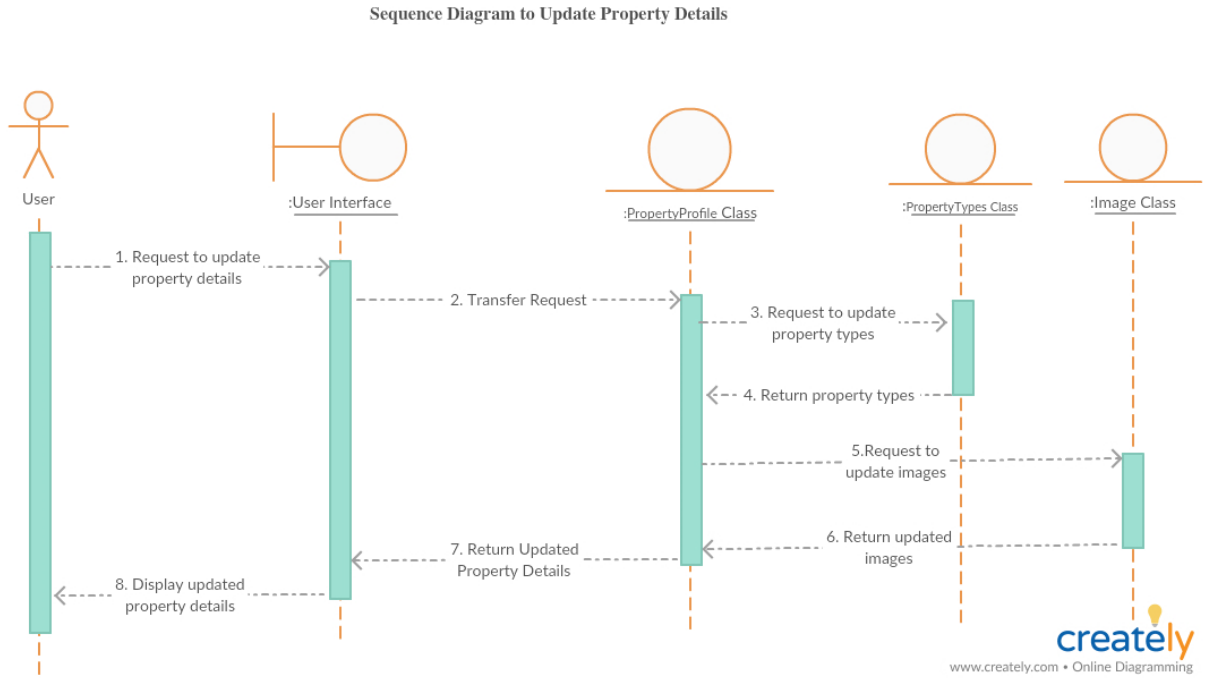
Upload Property Details



A user may want to upload his/her property details in order to sell it. In the sequence diagram, this is modeled by message

1. Request to upload property details
2. Transfer request
3. Request to select property types
4. Return property types
5. Request to upload images
6. Return uploaded images
7. Return property details
8. Display property details

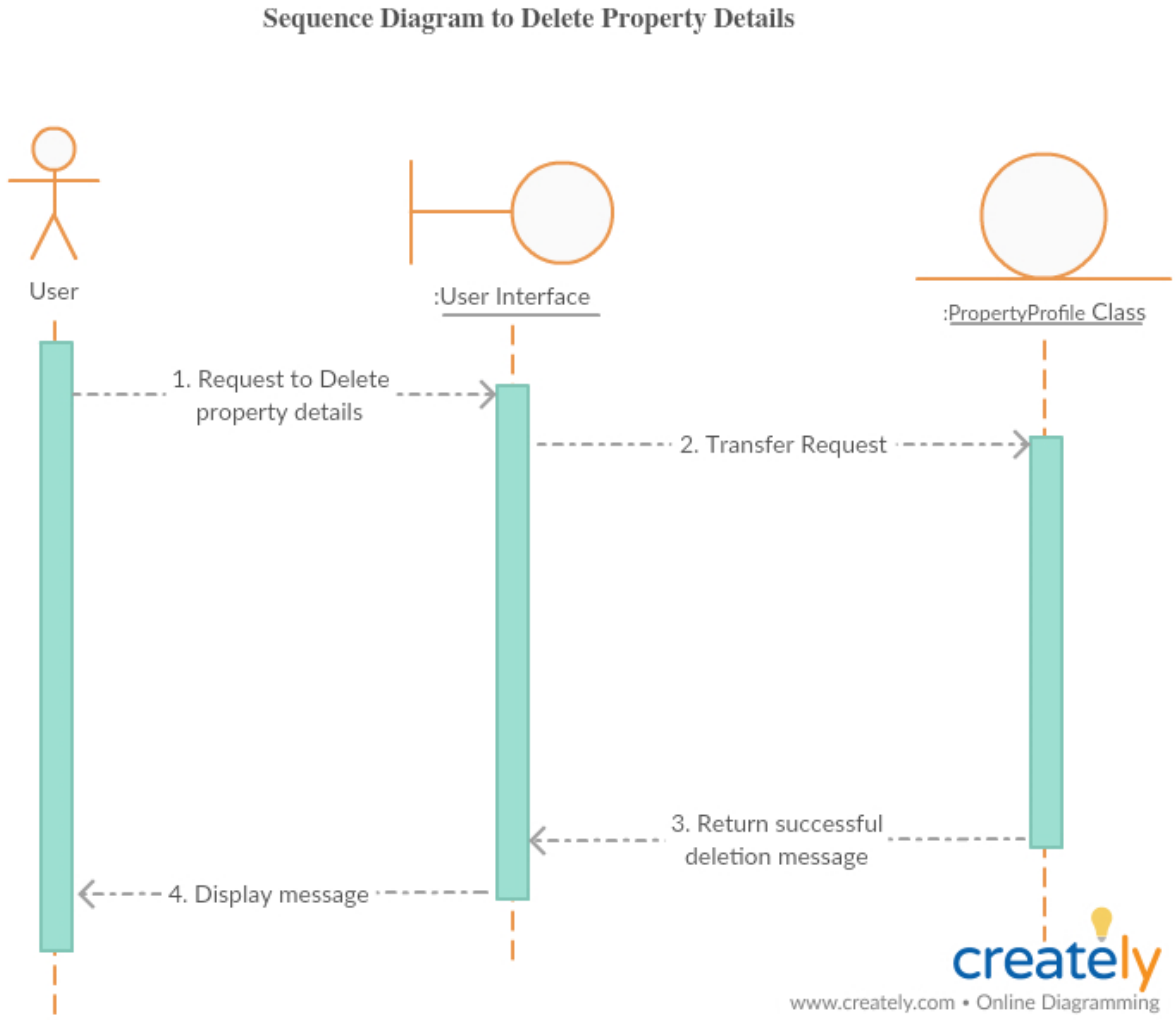
Update Property Details



User may want to edit the property details that has already been uploaded. In the sequence diagram, this is modeled by message

1. Request to update the property details
2. Transfer request
3. Request to update property types
4. Return updated property types
5. Request to update images
6. Return updated images
7. Return updated property details
8. Display property details

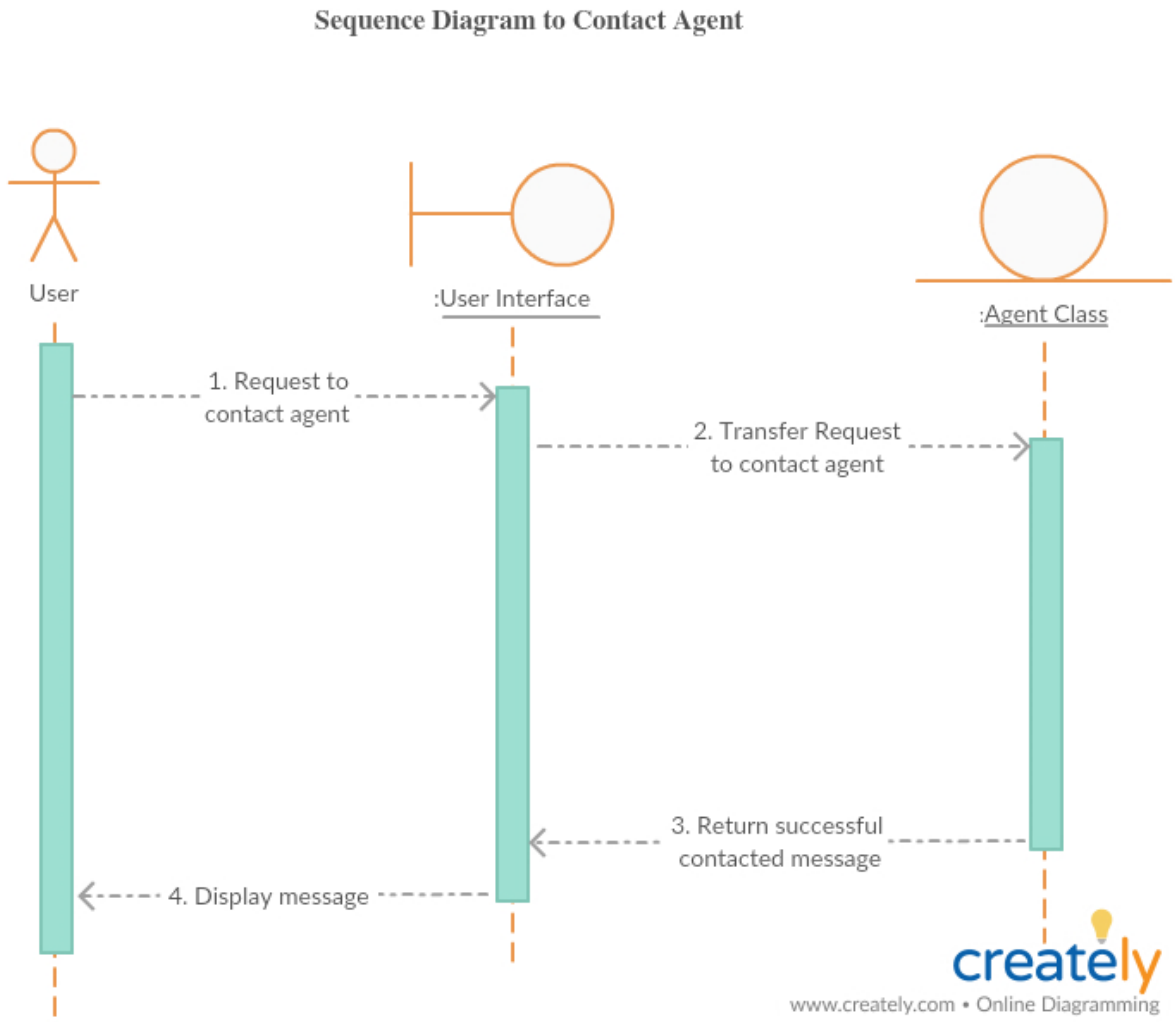
Delete Property Details



User may want to delete his/her property details that has already been uploaded. In the sequence diagram, this is modeled by message

1. Request to delete property details
2. Transfer request
3. Return successful deletion message
4. Display message

Contact Agent



User may want to take a help from an agent to search the required property. In the sequence diagram, this is modeled by message

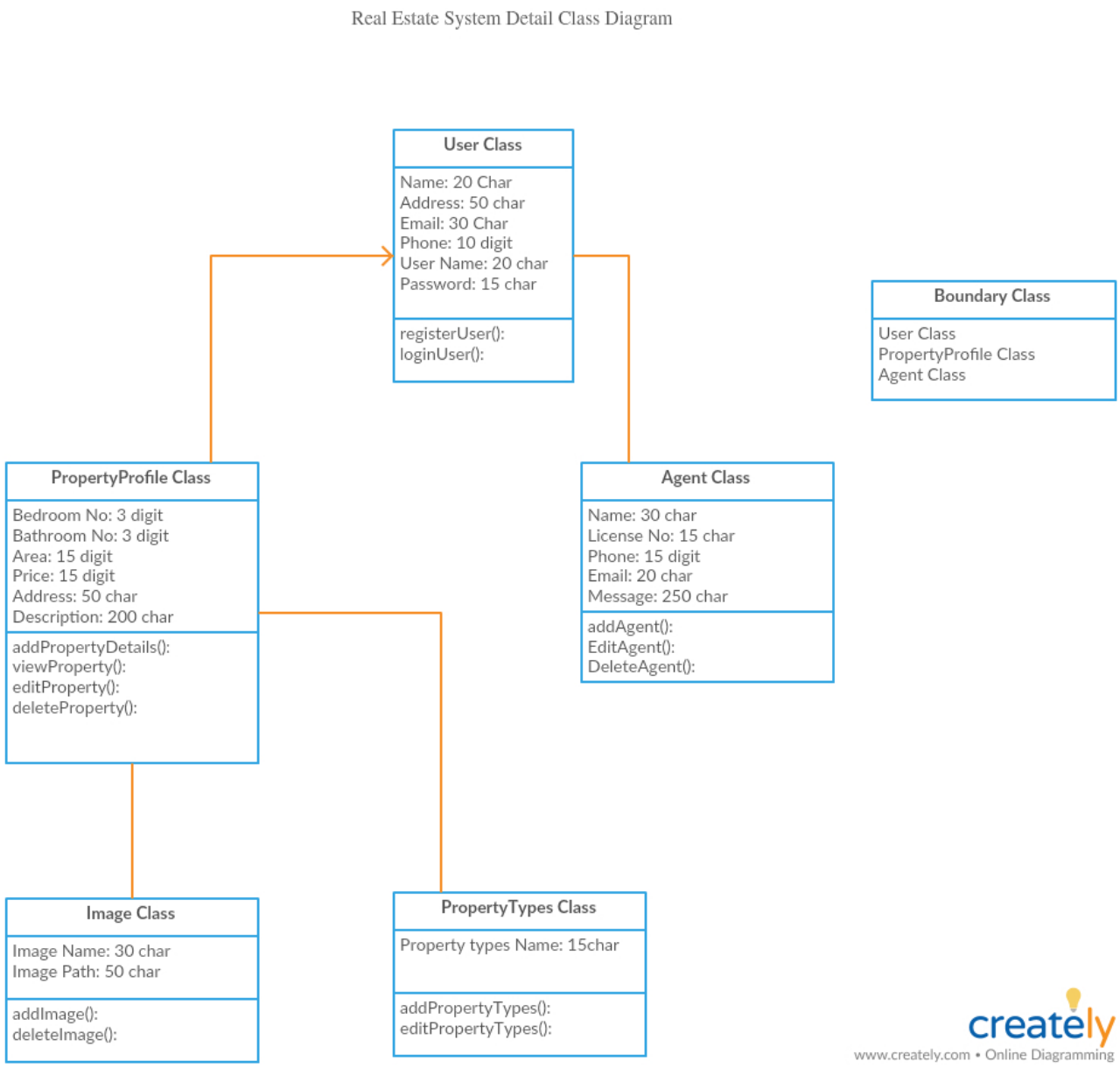
1. Request to contact agent
2. Transfer request to contact agent
3. Return successful contacted message
4. Display message

3. DESIGN WORKFLOW

3.1 Object Oriented Design

- Completed Class Diagram
- Pseudocode

Completed Class Diagram



Pseudocode

Registering User

```
Public boolean RegisterUser()
//This method created a new user in the system.
{
String Lastname;
String FirstName;
String Email;
String Password;

// Check if email already exists
if(email already exists){
        return false;
}

//Create new user with the email if email do not exists before
else{
User.Create();
        return true;
}
}
```

Login User

```
Public boolean LoginUser()
// This method validates the user and sets user in the session and returns true.
{
String email;
String password;
//Check if the user with the given email and password exists in the database and sets session for
that user.
If (email and password exists in the database){
        session[:current user] = user (user_id,token);
        return true;
}
else{
        return false;
}
}
```

Upload Property Details

```
public class PropertyProfile
{
```

```
String PropertyImages;
String PropertyTypes;
String BedroomNo;
String BathorromNo;
String Area;
String Price;
String Address;

//Upload PropertyDetails
Create ProppertyProfile (PropertyImage, PropertyTypes, Area, Price, Address)
        return PropertyDetails;
}
```

View Property

```
getProperty()
{
        string= property.ID;
        var property= Property details where property ID is equal to property.ID;
        return Property;
}
```

Search For Organ

```
Public List<Property> searchProperty()
//This method returns all the properties searched with particular city, state or zipcode
{
        //Get all the Property from the database with particular address
        List<Property> PropertyList= Property.all property searched
        return propertyList;
}
```

3. TEST WORKFLOW

Test Cases

First Name/Last Name

1. Non Alphabetic Character	Error(123, @!#%$)
2. 0 Character	Error (white character)
3. Between 0-20 Character	Acceptable(R, Ram, Ramsay)
4. 20 Char	Acceptable
5. >20 Character	Error

Password

1. 0 Character	Error (white character)
2. <6 Char	Error(abvd, 1254)
3. Between 5-20 char	Acceptable
4. 20 char	Acceptable
5. >20 Char	Error

Phone

1. Non numeric characters	Error (abd, @#$)
2. 0 digit	Error (white character)
3. <10 digit	Error (123456789)
4. 10 digit	Acceptable (6602385861)
5. >10 digit	Error

Functional Testing

1. Register User
2. Login User
3. Register Agent
4. Upload Property
5. Edit Property
6. Delete Property
7. Search Property
8. View Property
9. Contact agent