Dream House Project Report

COMP 231

Centennial College

Professor: Hao Lac

Jake Nesovic

Group Members:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student No. | Section No. | Signature |
| Abhishek Mekwan | 300-734-015 | 060 |  |
| Gowtham Pinnaka | 300-733-643 | 060 |  |
| Mohammed Ekram Ullah | 300-719-918 | 060 |  |
| Pei Wang | 300-712-924 | 060 |  |
| Santhalingam Sivasambu | 300-546-288 | 060 |  |
| Sri Yoga | 300-685-658 | 060 |  |

Date: December 11th, 2013

Table of Contents

[Project Scope 2](#_Toc374544798)

[Problem statement 3](#_Toc374544799)

[Proposed Solution 3](#_Toc374544800)

[High Level Deliverables 3](#_Toc374544801)

[High Level Architecture 3](#_Toc374544802)

[UML diagrams 5](#_Toc374544803)

[Use Case Diagram 5](#_Toc374544804)

[Class Diagram 6](#_Toc374544805)

[ERD Diagram 6](#_Toc374544806)

[Activity Diagram 7](#_Toc374544807)

[Database Table Design and Analysis 12](#_Toc374544808)

[Table 1: HOME 12](#_Toc374544809)

[Table 2: USERDETAILS 12](#_Toc374544810)

[Customer-Friendly UI 13](#_Toc374544811)

[Team Activities 14](#_Toc374544812)

[GitHub Version Control Repository 14](#_Toc374544813)

[Project on Pivotal Tracker 14](#_Toc374544814)

# Project Scope

## Problem statement

Toronto’s housing market is exploding, and with that market real estate companies are advertising their own product everywhere. Real estate companies use some software/web-application to register customer but there are very few software for agent themselves. Actually, most of real estate software is web based basic applications and barely is made for the agent. So, our team will build that software for the real estate company agent where all the information of housing market will be there and real estate agent will search out the appropriate housing for the customer through that software. The efficiency of this application will help agent get rid of communication problems, document management problems, media asset problems and so on.

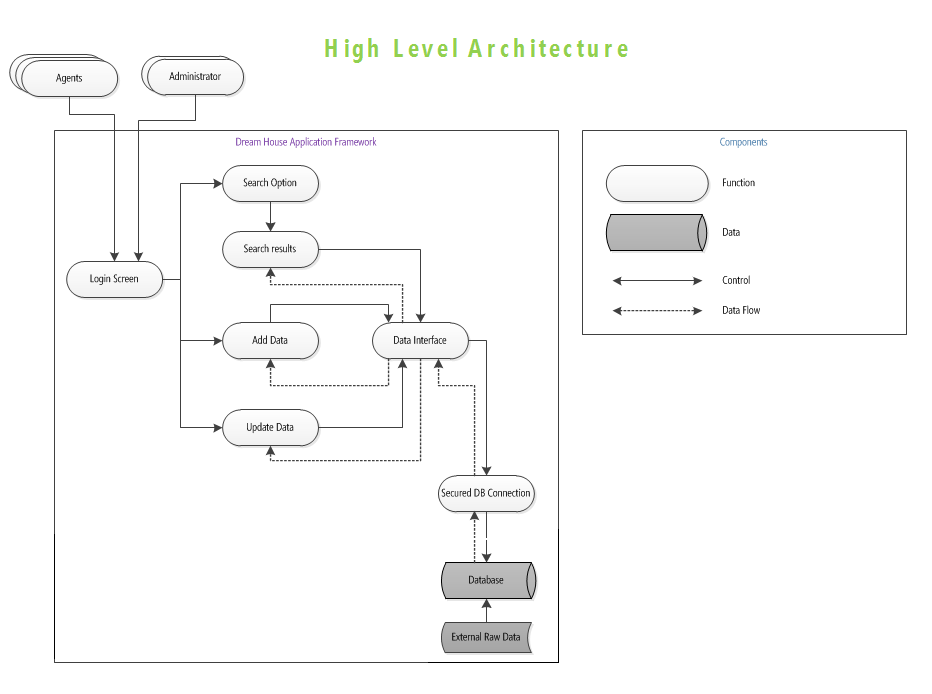
## Proposed Solution

Dream House Project aims at achieving a breakthrough in usability, code portability and performance scalability by providing an improved system which can holds real time data of the real estate market, keep on-sale households information database update and could be used by the real estate company agent to conduct search out the appropriate housing for the customer easily and efficiently.

## High Level Deliverables

* Scope and objectives document
* Analysis and Design
* All Java source code
* Code integration and testing
* An MS-SQL Server database, database schema, etc.
* A .NET application for administrating the data on the server side
* Application Testing
* User manuals/system documentation

# High Level Architecture



**Figure 1:** Architecture drawing of Dream House System

**The description for Figure 1 would be:**

Agent: - is an entity that has access to the system, but cannot make changes in the database.

Administrator: - is an entity that has full access to the system, including authentication to make desirable changes in the database.

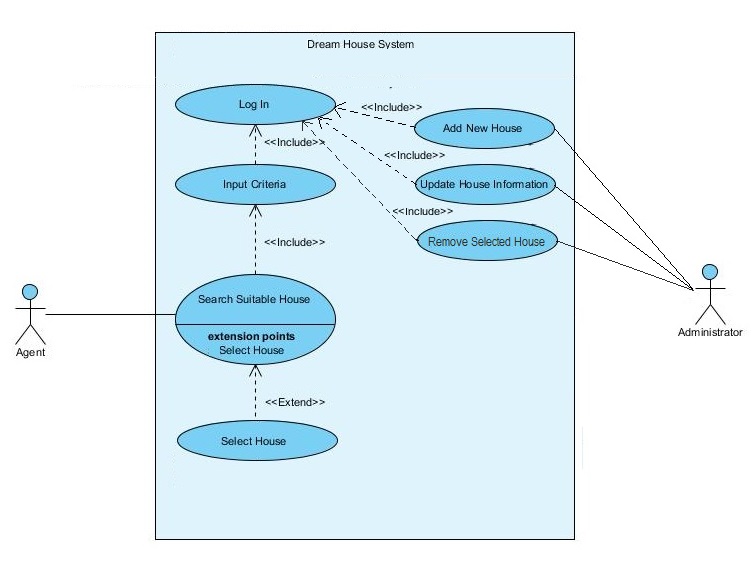
The clients will come to the agents with their particular requirements in mind. The agent will log in the system using his/her credentials. Based on the credentials the system will recognise him/her as agent or administrator, allowing him to access system limited to the credentials. Then the agent will be shown a search page, where the requirements of clients can be configured. After filling required information, the agent will get results matching the client’s requirements. On the search result page agent can browse among the results and show clients available choices.

The administrator has privileges to add new data in the database or update or delete the existing one.

The agent on other hand cannot add or delete records from the database. There are two layers of security in the system.

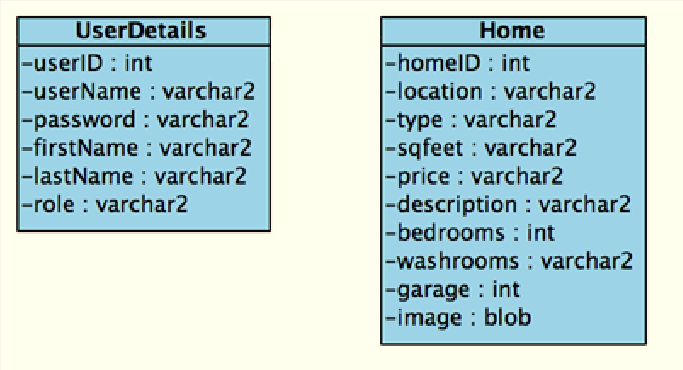
# UML diagrams

## Use Case Diagram



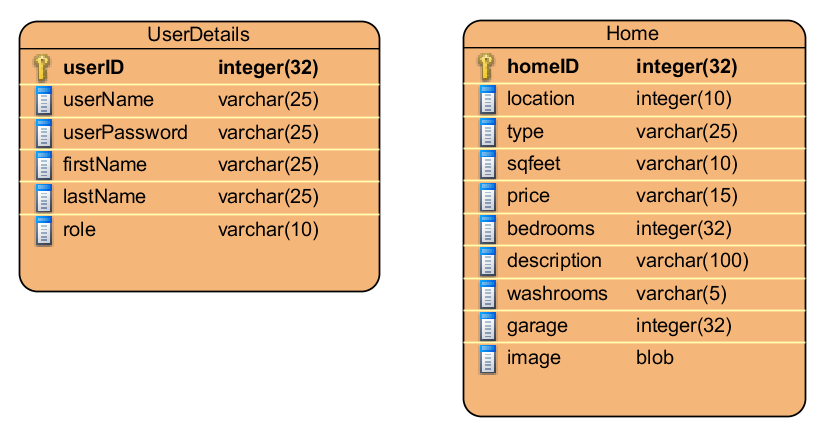
This use case diagram identifies all of the users of the system and the functions they are able to do.

## Class Diagram



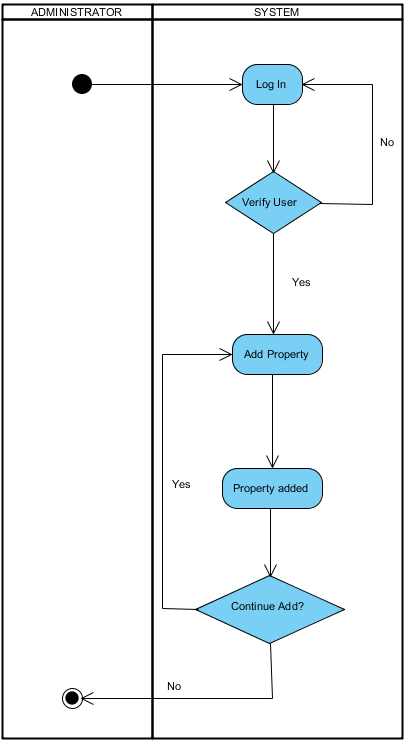
This class diagram show two of the classes used on our program. UserDetails class stores the fields for the agent/admin retrieved from the database while the Home class stores the fields of the property listing.

## ERD Diagram



This is the ERD diagram that represents the database we’ve created. It has two tables corresponding with the two classes used in the program.

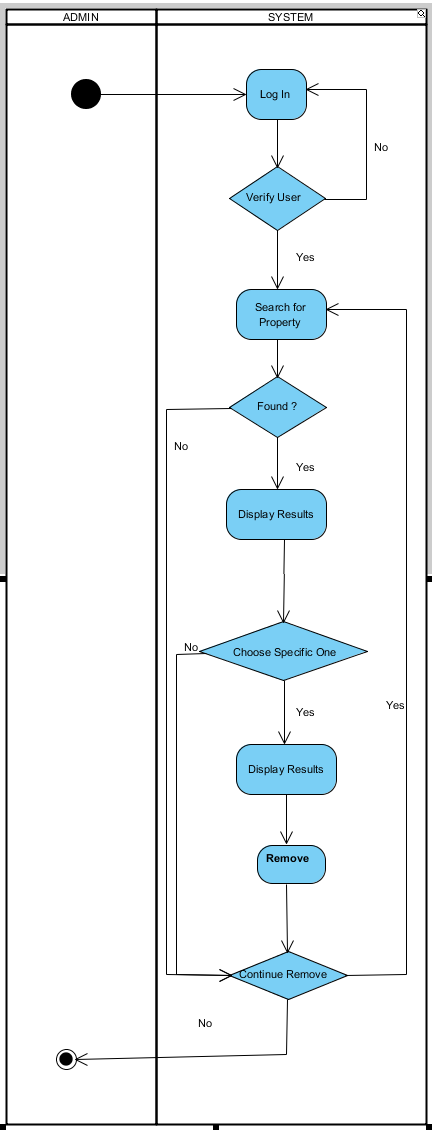
## Activity Diagram



Activities of the Admin – Add Property

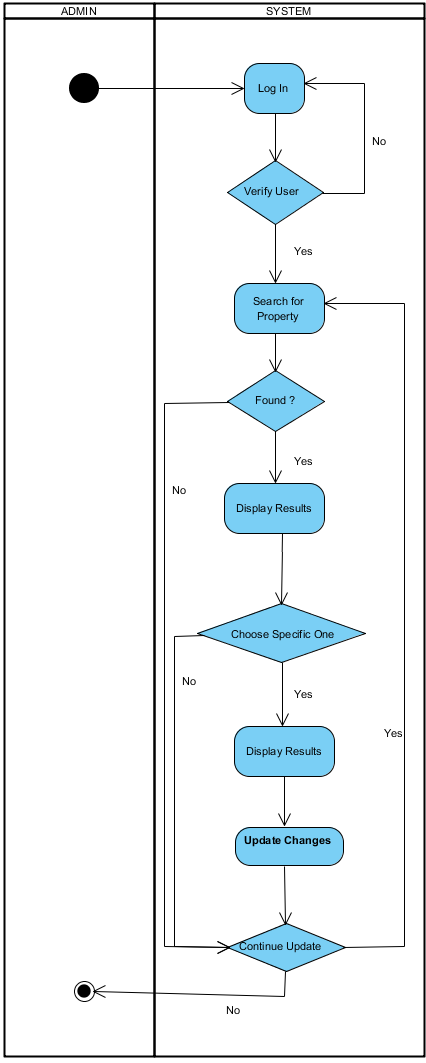
This is the activity diagram that shows the flow of processes that the admin is able to perform to add a property listing.

This is the activity diagram that shows the flow of processes that the agent is able to perform. These processes include searching for a property listings.



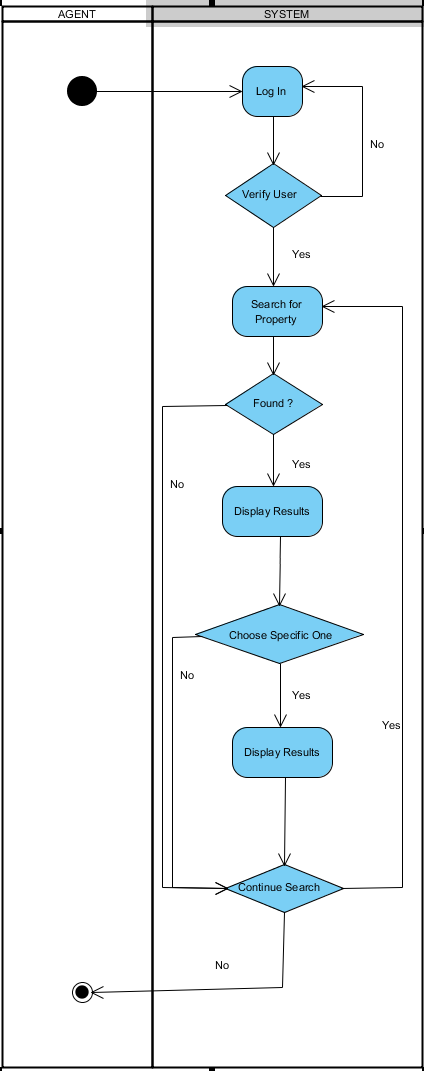
This is the activity diagram that shows the flow of processes that the admin is able to perform to remove a property listing.

Activities of the Admin – Remove Property



This is the activity diagram that shows the flow of processes that the admin is able to perform to update a property listing.

Activities of the Admin – Update Property

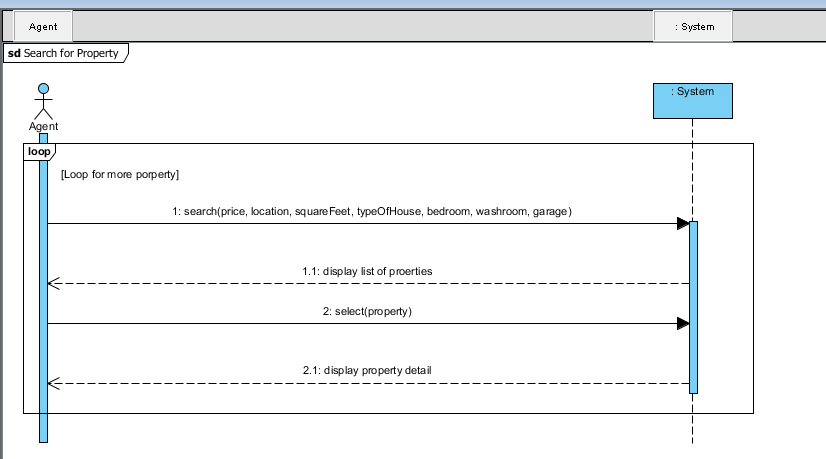


This is the activity diagram that shows the flow of processes that the agent is able to perform to search for a property listing.

Activities of the Admin – Search for Property

**System Sequence Diagram**

This System Sequence Diagram shows, an administrator can add a new property to the database, or update which includes any changes to the property, such as change in price, or remove the property from database, if it is sold.



This System Sequence Diagram shows, an agent can search for property by inputting searchable criteria (price, location, square feet, etc.), and from the list of results (one or more than one), he/she can select a specific one to view more details about the chosen property.

# Database Table Design and Analysis

## Table 1: HOME

|  |  |
| --- | --- |
| Name of Fields | Type of the Fields |
| HOMEID | NUMBER (Mandatory) |
| LOCATION | VARCHAR2(80 BYTE) |
| TYPE | VARCHAR2(50 BYTE) |
| SQFEET | VARCHAR2(80 BYTE) |
| PRICE | VARCHAR2(100 BYTE) |
| DESCRIPTION | VARCHAR2(2000 BYTE) |
| BEDROOMS | VARCHAR2(80 BYTE) |
| WASHROOMS | VARCHAR2(80 BYTE) |
| GARAGE | VARCHAR2(80 BYTE) |
| IMAGE | BLOB |

## Table 2: USERDETAILS

|  |  |
| --- | --- |
| Name of Fields | Type of the Fields |
| USERID | NUMBER (Mandatory) |
| USERNAME | VARCHAR2(40 BYTE) |
| PASSWORD | VARCHAR2(20 BYTE) |
| FIRSTNAME | VARCHAR2(20 BYTE) |
| LASTNAME | VARCHAR2(20 BYTE) |
| ROLE | VARCHAR2(20 BYTE) |

# Customer-Friendly UI

|  |  |
| --- | --- |
| Login Page | Welcome Page |
| Search Page | Add New Home Page |
| Search Results Page | About Us Page |

# Team Activities

## GitHub Version Control Repository

<https://github.com/SriYoga/DreamHouseProject>

## Project on Pivotal Tracker

<https://www.pivotaltracker.com/s/projects/910344>