

Real Estate Database Management System

By: Andy Almeida

ID: 922170012

Github Username: pie240

Checkpoint #	Date Submitted
Checkpoint 1	2/15/2023
Checkpoint 2	2/28/2023
Checkpoint 3	3/14/2023
Checkpoint 4	
Checkpoint 5	
Checkpoint 6	
Checkpoint 7	

Table of Contents

Project Description	2
Use Cases	3
Use Case 1 : Owning/Managing Multiple Properties	3
Use Case 2 : Marketing a Commercial Property	4
Use Case 3 : Marketing a Property for Rent/Purchase	5
Use Case 4 : Property Search - Filtered Searches	5
Use Case 5 : Property Search - Comparing Properties	6
Database Requirements	6
Functional Database Requirements	6
Non-Functional Database Requirements	9
Entity Relationship Diagram	10

Project Description

Employing users with the most pertinent information to suit their needs would allow for buyers/renters to make well-informed decisions. The Real Estate Database Management System aims to create a seamless and accessible program to allow for users to get the most important information pertinent to suit their needs. This system is deemed extremely valuable for many levels of users and companies. Valuable real estate information is necessary for users such as people looking to rent an apartment, families searching for their next home, and business owners looking for a location for their next place of operations.

A well constructed Real Estate Database Management application would allow for less roadblocks slowing down the Real Estate market. The aspects provided would prove extremely useful to concurrent owners of Real Estate, allowing for easier and more seamless forms of connection between people looking to buy/rent property and the real estate owners who are looking to sell/rent. With quick access to available commercial locations, users looking for tangible locations for their businesses would spend less time focused on researching and comparing properties, and more time on other pressing matters.

The Real Estate Database Management System will allow for real estate owners to input their available properties, along with the property's important characteristics, such as location, pricing, images, and descriptions, as well as potentially including their contact information. The property would then be put into a system where users could access the needed information internally through the application, and if the owners would like to opt in, ads could be generated for other popular open-market sites allowing for a wider range of potential customers. External users would be redirected to the source ad. Internal users looking to purchase or rent will utilize a simple to use user interface that would allow for data querying utilizing specific filters such as type of property, location, pricing, and feature desires. This system would also include a comparison feature, allowing for a user to select two properties and view them side-by-side to make for an improved and streamlined decision making process. When a buyer is prepared to move to the next step, they would be able to fill out an application, allowing the property owner a system to help filter out inapplicable buyers. The property owner would then be able to either chat with the user using an internal messaging system, or respond via email with either a pregenerated response or a personalized response. The overall focus of this project is to ease all aspects of the transactional aspects of real estate to help the players within the market respond much faster and allow for business to be much more accessible to a larger number of users.

Use Cases

Use Case 1 : Owning/Managing Multiple Properties

Actor: Real Estate Owner (Samuel)

Description: Samuel is a Real Estate Owner that owns 12 different properties all across California. He has information about each of the properties in his own personal system, however, when it comes to an online presence, his properties are scattered across a handful of websites. Some properties are currently available for rent/sale, and the rest are currently unavailable, but they could be within the near future. Samuel has an issue in that there is no online place for all of his properties to stay on. When a property is unavailable, he can only merely remove the ads on the websites that he posts them on, if he can even remember all. Whenever those properties become available once more, he must recreate those ads and redisperse them throughout the internet. When this happens, it can take up a large chunk of time since he would have to revisit the information that he requires in order to fill out and generate those ads. He even occasionally gets contacts on properties from ads that he failed to remember to take down, adding to the confusion.

The Real Estate Database Management System would allow Samuel a place to enter all of his properties in one place. They would be updatable at any moment. Fields such as price point, features, and availability would be easily accessible. Changing the availability of the property would immediately regenerate the ads that he requires and disperse them across the internet. The ads would display the up-to-date information, and help redirect clients and potential customers back to Samuel within a short time. Disabling a property's availability would also remove any ads that were created by the system, removing the need to search around for loose ads that were left behind, and removing confusion through ads that are still up on properties that are not available.

Use Case 2 : Marketing a Commercial Property

Actor: Real Estate Owner (Raul),

Description: Raul is a commercial property owner in San Francisco. He is struggling to find a way to easily market his property to people looking to start a business. Commonly known websites are proving to be difficult in his venture. Raul does not want to only rely on Craigslist due to its poor social perception and overcrowded/muddy market. Zillow, whilst a great tool, is centered around homeowners and apartment owners, thus not supporting his needs. Raul is open-minded and relatively tech savvy, but doesn't have all the time or energy to spend on

seeking out people to look into his property since he owns multiple other properties. He needs some form of outlet to allow him to quickly research the properties in his area in order to accurately value his storefront. He also needs a good platform that will quickly connect him with people who are searching for properties such as his own. He would like to market his property at a fair price whilst not wasting too much time on obtaining potential clients.

The Real Estate Database Management System will help the Real Estate owner, which in this case is Raul, by giving him a platform to market his property fairly quickly. Raul would be given a system that helps him compare his property to others in his area to better value the storefront he owns. People looking to rent his property for the sake of their business would have an easy means of connecting with Raul, allowing him to focus less time on finding potential clients, and more time meeting with clients to potentially put him one step closer to closing a deal.

Use Case 3 : Marketing a Property for Rent/Purchase

Actor: Property Manager (Sarah),

Description: Sarah is a property manager who works under someone who owns a handful of properties. Sarah is in charge of advertisements, client acquisition, and client vetting. Sarah is very comfortable with most forms of technology, but has found it to be rather time consuming when it comes to creating ads for multiple sites. She has noticed that copying and pasting ads on multiple websites is extremely time consuming, and requires a lot of effort. She has also noticed there is no generalized system for each of these individual market sites to assist her in selecting the right clients to proceed forward with contacting. Overall, her job is not very streamlined and there are many aspects to her work that prove to be more busy-work than she'd prefer.

The Real Estate Database Management System will remove much of the busy work that is troubling this property manager. To mitigate the time spent on marketing, Sarah would only need to input the property's information into this single system, and it would in turn, generate ads for multiple potential participating websites such as Facebook Marketplace, Craigslist, Offer-up, and Zillow. Users could then be redirected to the system, which would allow them to apply for Sarah's contact information. Sarah would then be given a very friendly user interface that would help her quickly release information to clients that meet her needs via their application. Her ads would reach a wider range of potential customers in much less time due to the use of multiple external websites. She would spend much less time creating ads and filtering out spam and inapplicable messages, and more time closing the deal on with her next renter/owner.

Use Case 4 : Property Search - Filtered Searches

Actor: A Couple looking for a new place to rent in a new city (Janine and Chris)

Description: Janine and Chris are a young couple looking to move to the city of San Francisco. They have a fairly average income, they both work remotely, and they have a beautiful German Shepard named Wiggles. In their search through Craigslist and Facebook marketplace, they have found that finding places that fit within their filter have been proven to be fairly frustrating. They always find places that initially look perfect for their new home, but when reading through descriptions, they find something that does not support them and their needs. They find a place that they can afford, has enough rooms for their work-from-home situation, and in a nice area, but upon closer inspection, they find that the place does not allow for dogs. In another situation, they find a place that supports dogs, has enough rooms, but is extremely out of their budget. They get extremely frustrated with the difficulty of their search.

The Real Estate Database Management System would remove the difficulties of searching for a place that meets all of their criterias by allowing for searches based on specific queries. Specifications in the database would include factors such as price points, locations, rooms, building size, if it includes a front/backyard, if pets are allowed, if appliances are included, and much more. This would allow for users to filter out the places that do not meet their needs, and help them get set up with a place that is much more applicable for their situation much quicker.

Use Case 5 : Property Search - Comparing Properties

Actor: An older single gentleman searching for a home to retire in (Rajan)

Description: Rajan is a well off 65 year old man searching for a place to call his home for the rest of his life. He is not too tech savvy, but knows what he wants and what he doesn't want when it comes to where he wants to live. In his search for properties, he has found himself struggling using the technologies in today's modern market. He has some knowledge with his computer, and can get around, but overall his experience is quite time consuming and limited. He has found a handful of places that seem to work for him, but when he is comparing the two, he finds that it is quite difficult to constantly click between the two pages. With older age, he struggles with his memory, and comparing the two places with a detailed/keen eye is getting frustrating.

The Real Estate Database Management System would allow for two properties to be compared to each other within the same window. With the click of a button, he could compare the two side-by-side. Rajan could compare pictures, prices, details, and locations between the two properties at the same time within a simple to use user interface. This would allow Rajan to make a well informed decision to get him into his perfect retirement home.

Database Requirements

Functional Database Requirements

1. General User
 - 1.1. A user shall create only one account
 - 1.2. A user shall have at least one role
 - 1.3. A user shall be able to check out multiple properties at a time
 - 1.4. A user shall be able to check out multiple locations at a time
 - 1.5. A user shall be able to create multiple applications
2. Actions
3. Account
 - 3.1. An account shall be created by only one user
 - 3.2. An account shall have at least one role
 - 3.3. An account can be linked to a Firm
 - 3.4. An account can be linked to multiple properties
 - 3.5. An account can create multiple ads
 - 3.6. An account can create multiple applications
 - 3.7. An account can receive multiple applications
4. Role
 - 4.1. A role can be linked to many users
 - 4.2. A role can be linked to many accounts
5. Permissions
6. Firm
 - 6.1. A Firm can be linked to many accounts
 - 6.2. A Firm shall have one location
7. Property
 - 7.1. A property can be viewed by many users
 - 7.2. A property shall be owned by one account
 - 7.3. A property shall can have multiple locations linked to it
 - 7.4. A property can have multiple buildings
 - 7.5. A property can have multiple apartments
 - 7.6. A property can be a home
 - 7.7. A property can be linked to multiple ads
 - 7.8. A property can receive multiple applications
 - 7.9. A property can have one set of amenities
8. PropertyType
9. Property List
10. Location
 - 10.1. A location can be accessed by multiple users

- 10.2. A location can be linked to one Firm
- 10.3. A location can be linked to one property
- 10.4. A location can be linked to one building
- 10.5. A location can be linked to one apartment
- 10.6. A location can be linked to one home
- 11. Building
 - 11.1. A building can be linked to a single property
 - 11.2. A building can be linked to one location
 - 11.3. A building can hold many apartments
 - 11.4. A building can be a home
 - 11.5. A building can hold many rooms
 - 11.6. A building can be linked to multiple ads
 - 11.7. A building can receive multiple applications
 - 11.8. A building can have one set amenities
- 12. ApartmentList
- 13. Apartment
 - 13.1. An apartment must be linked to a property
 - 13.2. An apartment must be linked to a location
 - 13.3. An apartment can be linked to a building
 - 13.4. An apartment can hold many rooms
 - 13.5. An apartment can be linked to multiple ads
 - 13.6. An apartment can receive multiple applications
 - 13.7. A building can have one set of amenities
- 14. Home
 - 14.1. A home can be a property
 - 14.2. A home can be linked to a property
 - 14.3. A home can be linked to a location
 - 14.4. A home can be a building
 - 14.5. A home can hold many rooms
 - 14.6. A home can be linked to multiple ads
 - 14.7. A home can be receive multiple applications
 - 14.8. A home can have one set of amenities
- 15. Commercial Building
- 16. RoomList
- 17. Room
 - 17.1. A room can be linked to a building
 - 17.2. A room can be linked to an apartment
 - 17.3. A room can be linked to a home
 - 17.4. A room can have a set of amenities

- 18. AdList
- 19. Ad
 - 19.1. An ad can be linked to an account
 - 19.2. An ad can be linked to a property
 - 19.3. An ad can be linked to a building
 - 19.4. An ad can be linked to an apartment
 - 19.5. An ad can be linked to a home
 - 19.6. An ad can be linked to multiple external e-commerce websites
 - 19.7. An ad can receive multiple applications
- 20. External e-Commerce Websites
 - 20.1. An External e-Commerce Website can be linked to multiple ads
- 21. Application List
- 22. Application
 - 22.1. An application can be linked to only one user
 - 22.2. An application can be linked to only one account
 - 22.3. An application can be linked to only one property
 - 22.4. An application can be linked to only one building
 - 22.5. An application can be linked to only one apartment
 - 22.6. An application can be linked to only one home
 - 22.7. An application can be linked to only one ad
- 23. Amenities
 - 23.1. A set of Amenities can be linked to a single property
 - 23.2. A set of Amenities can be linked to a single building
 - 23.3. A set of Amenities can be linked to a single apartment
 - 23.4. A set of Amenities can be linked to a single home
 - 23.5. A set of Amenities can be linked to a single room
- 24. CDN (Content Delivery Network)
- 25. Region
- 26. MultiMediaContentList
- 27. MultiMedia Content
- 28. Video
- 29. Photo

Non-Functional Database Requirements

1. System Requirements
2. Performance
 - 2.1. The database system shall support concurrent transactions
 - 2.2. The database system should be able to process queries and transactions in reasonable amount of time
3. Storage
 - 3.1. The database system shall assign 10 MB of memory per table
 - 3.2. The database system should support persistent storage
4. Security
 - 4.1. Only encrypted passwords shall be supported by the database system
 - 4.2. All the values inserted into the database shall be consistent with the attribute's data type and domain
 - 4.3. The database shall be automatically backed up everyday at 11:59pm
5. Environmental
 - 5.1. The database system should be located on a cloud based system
6. Legal
 - 6.1. A database system shall have a license attached to it.
 - 6.2. The contributors linked to this project shall be properly cited
 - 6.3. This project shall not be stolen and used without permission of the primary owner
 - 6.4. The database should be in compliance with data protections, privacy, and data retention legislation.
7. Content
 - 7.1. The database system shall be able to handle improper data correctly
8. Compatibility
 - 8.1. The database system should be compatible with most concurrent hardwares, softwares, and network systems.
 - 8.2. The database system should be accessible from a browser and handheld device
 - 8.3. The database system should be accessible through at least 3 major search engines
 - 8.4.
9. Organizational
 - 9.1. The database system will fill follow a relational hierarchical entity convention that prevents repetitive data
10. Scalability
 - 10.1. The database system should be able to handle a large number of users, data, and transactions without struggling with performance
 - 10.2. The database system should be able to be accessed by at least 20 concurrent users at one time.

Entity Relationship Diagram

