

REAL ESTATE DATABASE MANAGEMENT SYSTEM

Marcello Rusciano
SFSU ID: 917213120
GitHub ID: ycram3

Milestone/Version	Date
M2-Checkpoint2	07/19/2022
M2-Checkpoint1	07/19/2022
M1V1	06/23/2022

Table of Contents

Section I: Project Description	3
Section II: Use Cases	4
Section III: Database Requirements (Business Rules)	9
Section IV: Detailed List of Main Entities, Attributes and Keys	13
Section V: Entity Relationship Diagram (ERD)	20
Section VI: Database Model/EER	22

Section I: Project Description

Our real estate database management system helps real-estate owners to find their clients in a very easy and fast way. As an owner, you can manage your properties, no matter the types. We can handle everything, even things that other services don't provide, like tiny houses. As a buyer or possible tenant we assure you that you can find the estate you are looking for just with a few clicks, thanks to our vast and advanced filtering and sorting.

We can also assure you to help you any step of the way in case you need further help.

If you are in the market to buy or sell Real Estates we will provide you, in case you need it, with financing and real estate agents, or even escrow companies, title insurance, property inspectors and Real Estate law firms if that's what you are looking for.

Our service fits even DIY people that like to do everything on their own, giving them a proper guide on all the legal paperwork required on both State and County level.

All this to make the process the smoothest possible.

Finally, if you are looking to rent a property, to give you a piece of mind, we also provide reviews from previous tenants and a walkability score, so you could make the best decision possible without further surprises once you move in.

Section II: Use Cases

1. Use Case: Leasing land

Actor: Land owner, Real Estate Service

Description:

Jose just inherited a plot of land from his family. He realizes that now he has to pay property taxes on it, even if he is not making any profit from it. He is thinking of leasing but he doesn't want to pay hefty fees to a real estate agent. He goes online and finds a real estate service that posts houses on sale at seemingly no cost. Unfortunately this service doesn't include any real estate category that is not residential buildings.

Real Estate Database Management System could help both Jose and the real estate service to solve their problem, adding to their portfolio not only raw land but commercial, industrial and special use as well.

2. Use Case: Pets allowed

Actor: Potential tenants, Real Estate Service

Description:

A couple is looking on craigslist to find an apartment where pets are allowed, since for every post they found on craigslist they had to call to make sure if their beloved cat was allowed to live with them. The process was very lengthy and frustrating.

Real Estate Database Management System could help both, the couple and craigslist. Whenever there is a new listing, the database will require to input if pets are allowed or not.

In this way the couple won't need to contact the owner of the listing every single time to get that specific information.

3. Use Case: Tiny house

Actor: Registered user, Real Estate Service

Description:

Sandra is a very frugal person. She is trying to minimize her expenses because living in San Francisco is very costly. She heard of a new housing movement that involves tiny houses. This type of housing is very minimal and inexpensive because of the low square feet size involved, between 70 and 400 sq/feet. Unfortunately, the online service that she relied on last year doesn't provide her with this option.

Real Estate Database Management System could help one that online service for housing to add tiny houses in their portfolio. Whenever a user will search for a residential housing arrangement, now they can filter it with "tiny house", making their search way faster.

4. Use Case: Agent manages private properties

Actor: Commercial building owner, Real Estate agent, Real Estate Service

Description:

Tom decides to sell his commercial building because he wants to move to another State. He decided to go on some of those websites that allow you to post about your sale without external agents. He signed-up and listed the property but he realized that he is not really good at negotiating, losing all the offers he received at that point. He realizes that he isn't really fit for this task. He started thinking that maybe he could use some help. Now he is looking for a Real Estate agent.

Real Estate Database Management System could help the company where Tom initially signed up to keep him on its platform. The database would connect Tom with an external Real Estate agent. If the agent is not on the platform yet he would need to sign up as well to use the service. Tom needs to delete his post before the agent would post it for him.

5. Use Case: Private Sell

Actor: Home owner, Real Estate website

Description:

Sammy is an independent woman that believes she could do anything in life, without external help. She is the type of person that files taxes by herself. After divorcing, she decides to move out from the house where she used to live with his ex-husband. She went on a Real Estate website posting her house for sale. She also decided to go and look online for all the legal steps she needs to take to finalize the deal but the information found is often very general and confusing.

Real Estate Database Management System could help both Sammy and the Real Estate website to solve their problem. The Database has the ability to store all the information regarding the official paperwork she needs to file for the State and the County, plus external help like Financing agent, Real Estate agent, Escrow company, Title Insurance and Real Estate Law Firms, in case she is actually in need of those for any reason.

6. Use Case: Undecided on potential property

Actor: Potential tenant, Real Estate Service

Description:

Tyron finally received the new raise he was waiting for since last year. Now he feels he is ready to move out from his parents house and go living with his girlfriend. For him this is a big step and he wants everything to go smoothly.

He went on a listing website that rents houses. He searched around and three apartments got his attention but now he is undecided on which one is the best one for him because the apartments are very similar to each other.

Real Estate Database Management System could help Tyron and the listing website by providing a way to store reviews from other registered users that have been living in those apartments before. Plus it could also provide a “walkability score” which is a number attached to every neighborhood and it’s based on how many means of public transportation are available. This means that a group of buildings next to each other might have the same score.

With these features Tyron might have a smoother experience deciding his next housing.

7. **Use Case:** Premium account

Actor: Warehouse owner, Real Estate website

Description:

Luca had a paper company that recently went bankrupt because the use of its products has been declining more and more every year since computers started spreading into society.

Now he has a huge empty warehouse that he is thinking of leasing. He went online and posted his property on a Real Estate website. However, after the first month on the platform he only had one person interested in his listing. He is desperate because the more time he spends without leasing the property, the more money he loses. He is thinking of going to a classic Real Estate agent hoping he will find a buyer faster even if it will be at a hefty price.

Real Estate Database Management System could help Luca finding a client and the Real Estate website not losing a user. The database could help setting up a payment system and a premium account which will allow Luca to have his listing on top of the searches so that he will receive more visibility. It’s a win for Luca because this type of advertisement is cheaper than the fees that a Real Estate agent charges. It’s a win for the Real Estate website because he didn’t lose a user, plus it created a new stream of income.

8. **Use Case:** Property inspector

Actor: Potential buyer, Real Estate website

Description:

Tina just landed a new job which will pay her very well. She decided that it was time to move out from her parents house and buy a house closer to where she works so that commuting wouldn’t be so stressful. She goes on a real estate website looking for 2 bedroom houses. She quickly realizes the area she is looking for, all the houses are quite old. She went to a few open houses but now she is scared to find something wrong

with the house she is interested in after buying it. She needs to find someone that can inspect the house before taking the final decision.

Real Estate Database Management System could help her and the Real Estate website with a problem like that, because they are able to store the contact information of reliable property inspectors so that Tina could potentially connect with them with a classic phone call to schedule an appointment. Hence, Tina will have a peace of mind when she buys a property and won't sweat to find what she was looking for, while the website will provide a simple new feature that will keep the customer on their platform for longer.

9. Use Case: Online backup

Actor: Entire Database, Real Estate website

Description:

"CheapHousing" is a Real Estate service with a platform online that connects buyers to sellers. A couple of months ago part of their hard drives just stopped working. Because of that they lost a big part of their customers' data. After these two months of hard work they were able to retrieve all the data that was lost, however that involved a lot of money and resources. This time they are trying to make it right and look for a better long term solution.

Real Estate Database Management System could help CheapHousing with this problem because they can easily sync all the databases with online services. In this way, whenever there is a power outage, a local hard disk breaks down or any other type of problem related to data storage, the website will still run smoothly without any inconveniences because the data will be safely stored in multiple places.

10. Use Case: Admin

Actor: Administrator, Real Estate website

Description:

George is a 70 years old man that is transferring to a Home Care For The Elderly so he decided to sign up on "CheapHousing.com" to rent his duplex in San Jose in California. After a few hours from the posting he started to receive many offers but all the people he talked to only spoke Spanish and offered to pay in Colón instead of Dollars. Oddly enough, he decided to call customer service to see what was going on and he realized that he set up the location of his property in San Jose, Costa Rica instead of San Jose in California, USA.

Real Estate Database Management System could help CheapHousing.com and George with this problem. The Database is designed in a way the Admin of the website would be able to modify the location of the post on George behalf since he wouldn't do it so easily given his unfamiliarity with the platform.

11. Use Case: System Monitor

Actor: System, Real Estate website

Description:

The real estate service “CheapHousing.com” lately has had some problems with their online servers providers and some malfunctions in their database. They are looking for ways to prevent these types of problems before they occur.

Real Estate Database Management System could help CheapHousing.com with their malfunctions. The database is able to store the entire database online and connect in its entirety to a System that is able to log all the actions in the database. The log then is monitored by an Admin.

Section III: Database Requirements (Business Rules)

1. Estate:
 - 1.1 An estate should have one or more owner (from registered user)
 - 1.2 An estate is one: residential, commercial, industrial, raw land, or special use estate
 - 1.3 An estate can be posted by one and only one owner or agent
 - 1.4 An estate can have many reviews posted by many reviewers
 - 1.5 An estate is contained in one and only one neighborhood
 - 1.6 An estate can be leased by one owner or one real estate agent
 - 1.7 An estate can be sold by one owner or one real estate agent
2. Residential:
 - 2.1 A residential estate could be one tiny house
 - 2.2 A residential estate can be one single family home
 - 2.3 A residential estate can be one Multi family home
 - 2.4 A residential estate can be one apartment
 - 2.5 A residential estate can be one condo
 - 2.6 A residential estate can be one mobile house
 - 2.7 A residential estate is one and only one estate
3. Commercial:
 - 3.1 A commercial estate is one and only one estate
4. Raw Land:
 - 4.1 A raw land estate is one and only one estate
5. Special Use:
 - 5.1 A special use estate is one and only one estate
6. Post:
 - 6.1 A post can be made by one and only one owner or Real Estate Agent
 - 6.2 A post can be modified by one owner or one Real Estate Agent
 - 6.3 A post can be modified by one or many Admins
7. Owner:
 - 7.1 A owner can have one or more estates
 - 7.2 A owner can make one and only one post per estate
 - 7.3 A owner can connect with many Real Estate agents
 - 7.4 A owner can lease many estates
 - 7.5 A owner can sell many estates
 - 7.6 A owner can modify many posts
 - 7.7 A owner is one and only one registered user
 - 7.8 A owner can make one or more posts
 - 7.9 A owner can access to zero or one premium account

8. Tiny houses:
 - 8.1 Tiny house is one and only one residential housing
9. Real Estate Agent:
 - 9.1 A Real Estate agent can be reached out by zero or many registered users
 - 9.2 A Real Estate agent can modify many posts
 - 9.3 A Real Estate agent can lease many estates
 - 9.4 A Real Estate agent can sell many estates
 - 9.5 A Real Estate agent can access to zero or one premium account
 - 9.6 A Real Estate agent can make many posts
 - 9.10 A Real Estate agent is one and only one registered user
 - 9.11 A Real Estate agent can reach out zero or many real estate agents
10. Counties official paperwork:
 - 10.1 Counties official paperwork can be accessed by zero or many registered users
11. States official paperwork:
 - 11.1 States official paperwork can be accessed by zero or many registered users
12. Financing Agents:
 - 12.1 Financing agents information can be reached out by zero or many registered users
13. Escrow companies:
 - 13.1 Escrow companies information can be reached out by zero or many registered users
14. Title insurance:
 - 14.1 Title insurance information can be reached out by zero or many registered users
15. Real Estate Law Firms:
 - 15.1 Real Estate Law Firms information can be reached out by zero or many registered users
16. Registered user:
 - 16.1 A registered user can be zero or one reviewer
 - 16.2 A registered user can be zero or one real estate agent
 - 16.3 A registered user can be zero or one owner
 - 16.4 A registered user can reach out zero or many property inspectors
 - 16.5 A registered user can reach out zero or many Real Estate agent
 - 16.6 A registered user can access information about zero or many Counties official paperwork
 - 16.7 A registered user can access information about zero or many States official paperwork

- 16.8 A registered user can reach out zero or many Financing agents
- 16.9 A registered user can reach out zero or many Escrow companies
- 16.10 A registered user can reach out zero or many Title insurances
- 16.11 A registered user can reach out zero or many Real Estate Law Firms
- 16.12 A registered user is able to access information of one and only one website's customer service

17. Neighborhood:

- 17.1 A neighborhood contains one or many estates

18. Premium account:

- 18.1 A Premium Account can store many bank accounts
- 18.2 A premium Account can be accessed by one and only one owner
- 18.2 A premium Account can be accessed by one and only one real estate agent

19. Reviewer/Buyer/Renter:

- 19.1 A reviewer can review many estates
- 19.2 A reviewer can be one and only one registered user

20. Credit Card:

- 20.1 A Credit Card can be from one and only one bank account

21. Property inspector:

- 21.1 Property inspector information can be reached out by many registered users

22. Entire Database: (aggregate of everything except Log and Actions)

- 22.1 The entire Database can be stored on zero or many cloud services

23. Cloud Service:

- 23.1 A cloud service can store zero or many databases

24. Customer Service contact:

- 24.1 A Customer service can be reached out by zero or many registered users
- 24.2 A Customer service can access one Admin

25. Admin:

- 25.1 An Admin can modify many posts
- 25.2 An Admin can monitor zero or one and Log
- 25.3 An Admin can be accessed by many customer services

26. Log:

- 26.1 A Log can be monitored by zero or many Admins
- 26.2 A Log can record zero or many actions

27. Single Family Home:

27.1 Single Family home is one and only one residential housing

28. Multi Family home:

28.1 Multi Family home is one and only one residential housing

29. Apartment:

29.1 Apartment is one and only one residential housing

30. Condo:

30.1 Condo is one and only one residential housing

31. Mobile Home:

31.1 Mobile home is one and only one residential housing

32. Industrial:

32.1 An industrial estate is one and only one estate

33. Bank Account:

33.1 A Bank Account can be linked to many premium accounts

33.2 A Bank Account has zero or one debit card

33.3 A Bank Account can have zero or many credit cards

34. Debit Card:

34.1 A Debit Card can be from one and only one bank account

35. Actions in database:

35.1 An action is recorded by one log

Section IV: Detailed List of Main Entities, Attributes and Keys

1. Estate (strong):
 - estate_id: key (strong), numeric
 - address: key (strong), composite, alphanumeric
 - owner_id: key (weak), numeric
 - realEstateAgent_id: key (weak), numeric
 - neighborhood_id: key (weak), numeric
 - desc: alphanumeric
 - application_fee: numeric
 - pool: numeric (BIT for boolean)
 - special_feature: alphanumeric
 - wheelchair_access: numeric (BIT for boolean)
2. Owner (weak):
 - owner_id: key(strong), numeric
 - estate_id: key (weak), numeric
 - registeredUser_id: key (weak), numeric
 - post_id: key(weak), numeric
 - premiumAccount_id: key (weak), numeric
 - numberOfOwners: numeric
3. Residential Estate (weak):
 - residential_id: key(strong), numeric
 - estate_id: key(weak), numeric
 - singleHome_id: key(weak), numeric
 - multiFamilyHome_id: key(weak), numeric
 - apartment_id: key(weak), numeric
 - condo_id: key(weak), numeric
 - tinyHouse_id: key(weak), numeric
 - mobileHome_id: key(weak), numeric
 - numberOfStories: numeric
 - greenHomeCertificate: numeric (BIT for boolean)
 - bathrooms: numeric
 - bedrooms: numeric
 - yearConstruction: numeric
 - squareFeet: numeric
 - indoorParking: numeric (BIT for boolean)
 - elevator: numeric (BIT for boolean)
 - fireplace: numeric (BIT for boolean)
 - AC: numeric (BIT for boolean)

4. Commercial Estate (weak):
 - commercial_id: key (strong), numeric
 - estate_id: key(weak), numeric
 - lotSize: numeric
 - greenHomeCertificate: numeric (BIT for boolean)
 - numberOfAreas: numeric
 - bathrooms: numeric
 - yearConstruction: numeric
 - squareFeet: numeric
5. Industrial Estate (weak):
 - Industrial_id: ke(strong), numeric
 - estate_id: key(weak), numeric
 - lotSize: numeric
 - greenHomeCertificate: numeric (BIT for boolean)
 - numberOfAreas: numeric
 - bathrooms: numeric
 - yearConstruction: numeric
 - squareFeet: numeric
6. Raw Land Estate (weak):
 - rawLand_id: key (strong), numeric
 - estate_id: key(weak), numeric
 - lotSize: numeric
 - waterAvailability: numeric (BIT for boolean)
 - electricityAvailability: numeric (BIT for boolean)
7. Special Use (weak):
 - specialUse_id: key (strong), numeric
 - estate_id: key(weak), numeric
 - lotSize: numeric
 - numberOfAreas: numeric
 - bathrooms: numeric
 - yearConstruction: numeric
 - squareFeet: numeric
8. Post (weak):
 - post_id: key (strong), numeric
 - owner_id: key(weak), numeric
 - realEstateAgent_id: key(weak), numeric
 - admin_id: key(weak), numeric
 - title: alphanumeric
 - desc: alphanumeric

9. Registered User (strong):

- registeredUser_id: key (strong), numeric
- name: composite, alphanumeric
- email: key(strong) multivalue, alphanumeric
- address: composite, alphanumeric
- phoneNumber: multivalue, numeric
- reviewerBuyerRenter_id: key(weak), numeric
- realEstateAgent_id: key(weak), numeric
- owner_id: key(weak), numeric
- dob: multivalue, timestamp
- password: alphanumeric (encrypted)

10. Tiny House (weak):

- tinyHouse_id: key (strong), numeric
- residential_id: key(weak), numeric
- lotSize: numeric
- fixSewage: numeric (BIT for boolean)
- fullBathroom: numeric (BIT for boolean)
- backyard: numeric (BIT for boolean)
- frontyard: numeric (BIT for boolean)

11. Single Family Home (weak):

- singleHome_id: key(strong), numeric
- residential_id: key(weak), numeric
- lotSize: numeric
- backyard: numeric (BIT for boolean)
- frontyard: numeric (BIT for boolean)
- basement: numeric (BIT for boolean)

12. Multi Family Home (weak):

- multiFamilyHome_id: key(strong), numeric
- residential_id: key(weak), numeric
- lotSize: numeric
- backyard: numeric (BIT for boolean)
- frontyard: numeric (BIT for boolean)
- basement: numeric (BIT for boolean)
- numberOfHouses: numeric

13. Apartment (weak):

- apartment_id: key(strong), numeric
- residential_id: key(weak), numeric
- floor: numeric
- studio: numeric (BIT for boolean)

14. Condo (weak):

- condo_id: key(strong), numeric
- residential_id: key(weak), numeric
- lotSize: numeric
- backyard: numeric (BIT for boolean)
- frontyard: numeric (BIT for boolean)
- basement: numeric (BIT for boolean)

15. Mobile Home (weak):

- mobileHome_id: key(strong), numeric
- residential_id: key(weak), numeric
- lotSize: numeric
- backyard: numeric (BIT for boolean)
- frontyard: numeric (BIT for boolean)
- manufacturerName: composite, alphanumeric

16. Real Estate Agent (strong):

- realEstateAgent_id: key(strong), numeric
- license: key(strong), alphanumeric
- registeredUser_id: key (weak), numeric
- post_id: key(weak), numeric
- estate_id: key(weak), numeric
- owner_id: key(weak), numeric
- reviewerBuyerRenter_id: key (weak), numeric
- premiumAccount_id: key(weak), numeric
- address: key (strong), composite, alphanumeric
- phoneNumber: multivalue, numeric

17. Counties official paperwork (strong):

- countyPaperwork_id: key(strong), numeric
- hyperLinkPDF: key(strong), alphanumeric
- namePaperwork: composite, alphanumeric

18. State official paperwork (strong):

- statePaperwork_id: key(strong), numeric
- hyperLinkPDF: key(strong), alphanumeric
- namePaperwork: composite, alphanumeric

19. Financing Agent (strong):

- license: key(strong), alphanumeric
- name: composite, alphanumeric
- email: key(strong), multivalue, alphanumeric
- address: key (strong), composite, alphanumeric
- phoneNumber: multivalue, numeric

20. Escrow Company (strong):

- license: key(strong), alphanumeric
- name: composite, alphanumeric
- email: key(strong), multivalue, alphanumeric
- address: key (strong), composite, alphanumeric
- phoneNumber: multivalue, numeric

21. Title Insurance (strong):

- license: key(strong), alphanumeric
- name: composite, alphanumeric
- email: key(strong), multivalue, alphanumeric
- address: key (strong), composite, alphanumeric
- phoneNumber: multivalue, numeric

22. Real Estate Law Firm (strong):

- license: key(strong), alphanumeric
- name: composite, alphanumeric
- email: key(strong), multivalue, alphanumeric
- address: key (strong), composite, alphanumeric
- phoneNumber: multivalue, numeric

23. Property Inspector (strong):

- license: key(strong), alphanumeric
- name: composite, alphanumeric
- email: key(strong) multivalue, alphanumeric
- address: key (strong), composite, alphanumeric
- phoneNumber: multivalue, numeric

24. Neighborhood (strong):

- neighborhood_id: key(strong), numeric
- estate_id: key(weak), numeric
- numberOfTrainStation: numeric
- numberOfBusStation: numeric
- numberOfBikeLane: numeric
- walkabilityScore: derived, numeric

25. Premium Account (weak):

- premiumAccount_id: key (strong), numeric
- bankAccount_id: key(weak), alphanumeric
- owenr_id: key(weak), numeric
- realEstateAgent_id: key(weak), numeric

26. Reviewer/Buyer/Renter (weak):

- reviewerBuyerRenter_id: key(strong), numeric
- registeredUser_id: key(weak), numeric
- estate_id: key(weak), numeric (where it has lived)

27. Review (weak): (Relation becomes an entity)

- estate_id: key(weak), numeric
- reviewerBuyerRenter_id: key(weak), numeric
- numberOfStar: numeric
- desc: alphanumeric

28. Bank Account (strong):

- bankAccountNumber: key(strong), numeric
- premiumAccount_id: key (weak), numeric
- creditCardNumber: key(weak), numeric
- debitCardNumber: key(weak), numeric
- bankName: composite, alphanumeric
- accountOwnerName: composite, alphanumeric

29. Payment Info (weak): (Relation becomes an entity)

- premiumAccount_id: key (weak), numeric
- bankAccountNumber: key(weak), numeric
- paymentOrder_id: numeric
- paymentAmount: numeric
- date: multivalue, timestamp
- desc: alphanumeric

30. Credit Card (weak):

- cardNumber: key(strong), numeric
- bankAccount_id: key(weak), alphanumeric
- expiration: multivalue, numeric

31. Debit Card (weak):

- cardNumber: key(weak), numeric
- bankAccount_id: key(weak), alphanumeric
- expiration: multivalue, numeric

32. Cloud Service (strong):

- cloud_id: key(strong), alphanumeric
- websiteLink: alphanumeric
- name: composite, alphanumeric

33. Customer Service (strong):

- customerService_id: key(strong), numeric
- registeredUser_id: key(weak), numeric
- admin_id: key(weak), numeric
- phoneNumber: multivalue, numeric
- email: multivalue, alphanumeric

34. Admin (strong):

- admin_id: key(strong), numeric
- customerService_id: key(weak), numeric
- log_id: key(weak), numeric

35. Log (strong):

- log_id: key(strong), numeric
- admin_id: key(weak), numeric
- action_id: key(weak), numeric

36. Action in database (weak):

- action_id: key(strong), numeric
- log_id: key(weak), numeric
- desc: alphanumeric

37. Rent (weak):

- estate_id: key(weak), numeric
- owner_id: key(weak), numeric
- realEstateAgent_id: key(weak), numeric
- price: numeric
- securityDeposit: numeric
- petAllowed: numeric (BIT for boolean)
- moveInDate: multivalue, timestamp

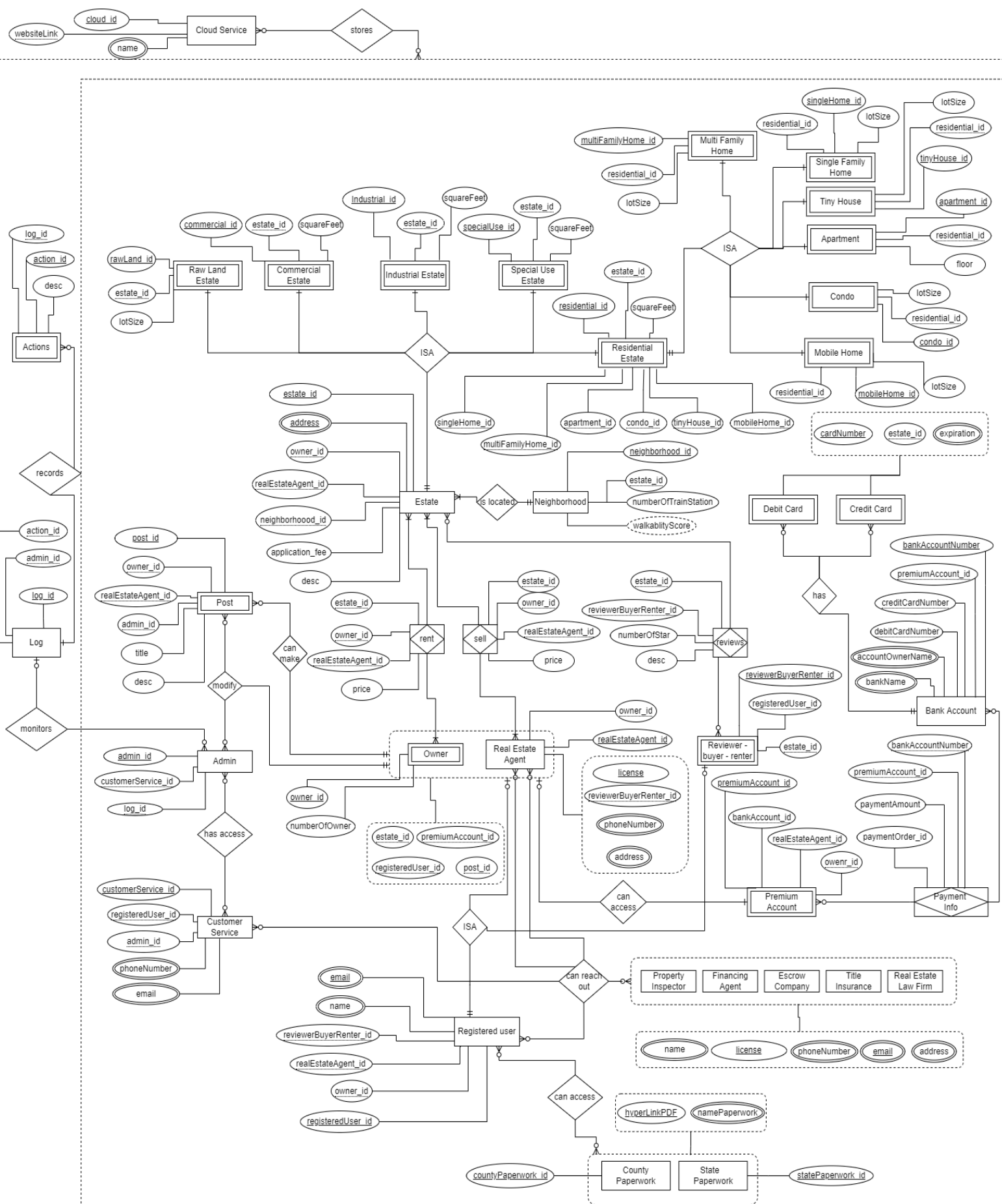
38. Sell (weak):

- estate_id: key(weak), numeric
- owner_id: key(weak), numeric
- realEstateAgent_id: key(weak), numeric
- price: numeric

Section V: Entity Relationship Diagram (ERD)

TO BE NOTED:

- 1) "Real Estate Agent" is a strong entity because it can be reached out by registered users even if the agent is not registered himself.
- 2) The diagram is very cramped because I knew I would have to put it in a PDF.
If I didn't have the requirement to submit it in this way I would have done it bigger and more clearly.
Sorry for the inconvenience.
- 3) The EDR is on the next page.



Section VI: Database Model/EER

Table	FK	ON DELETE	ON UPDATE	COMMENT
Access	countyPaperwork_id	CASCADE	CASCADE	If I delete or update County Paperwork I also need to do it for Access
Access	statePaperwork_id	CASCADE	CASCADE	If I delete or update State Paperwork I also need to do it for Access
Access	registeredUser_id	CASCADE	CASCADE	If I delete or update Registered User I also need to do it for Access
AccessCustomerService	customerService_id	CASCADE	CASCADE	If I delete or update CustomerService I also need to do it for AccessCustomerService
AccessCustomerService	admin_id	CASCADE	CASCADE	If I delete or update CustomerService I also need to do it for AccessCustomerService
Apartment	residential_id	CASCADE	CASCADE	If I delete or update Residential I also need to do it for Apartment
Commercial	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Commercial
CreditCard	bankAccount_id	CASCADE	CASCADE	If I delete or update BankAccountI also need to do it for CreditCard
DebitCard	bankAccount_id	CASCADE	CASCADE	If I delete or update BankAccountI also need to do it for CreditCard
Estate	neighborhood_id	NO ACTION	CASCADE	If I delete Neighborhood I don't need to delete it in the Estate So we set it to NO ACTION because I need to have a neighborhood, but if I update Neighborhood I need to update Estate
Industrial	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Industrial
MobileHome	residential_id	CASCADE	CASCADE	If I delete or update Residential I also need to do it for MobileHome
ModifyFromAdmin	post_id	CASCADE	CASCADE	If I delete or update Post I also need to do it for ModifyFromAdmin
ModifyFromAdmin	admin_id	CASCADE	CASCADE	If I delete or update Admin I also need to do it for ModifyFromAdmin
Monitors	log_id	NO ACTION	NO ACTION	Logs are important and they are not supposed to be deleted or modified for any reasons
Monitors	admin_id	CASCADE	CASCADE	If I delete or update Admin I also need to do it for Monitors

MultiFamily	residential_id	CASCADE	CASCADE	If I delete or update Residential I also need to do it for MultiFamily
Owner	registeredUser_id	CASCADE	CASCADE	If I delete or update RegisteredUser I also need to do it for Owner
PaymentInfo	premiumAccount_id	NO ACTION	CASCADE	If I delete PremiumAccount I don't need to delete the paymentInfo but if it gets updated I do have to do it in PaymentInfo
PaymentInfo	bankAccount_id	CASCADE	CASCADE	If BankAccount is either deleted or updated I need to do the same in PaymentInfo
Post	owner_id	CASCADE	CASCADE	If Owner is deleted or updated, Post should be as well
Post	realEstateAgent_id	CASCADE	CASCADE	If RealEstateAgent is deleted or updated, Post should be as well
PremiumAccount	owner_id	CASCADE	CASCADE	If Owner is deleted or updated, PremiumAccount should be as well
PremiumAccount	realEstateAgent_id	CASCADE	CASCADE	If RealEstateAgent is deleted or updated, PremiumAccount should be as well
RawLand	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for RawLand
Reach	financingAgent_id	CASCADE	CASCADE	If I delete or update Financing Agent I also need to do it for Reach
Reach	escrowCompany_id	CASCADE	CASCADE	If I delete or update Escrow Company I also need to do it for Reach
Reach	titleInsurance_id	CASCADE	CASCADE	If I delete or update Title Insurance I also need to do it for Reach
Reach	realEstateLawFirm_id	CASCADE	CASCADE	If I delete or update Real Estate Law Firm I also need to do it for Reach
Reach	realEstateAgent_id	CASCADE	CASCADE	If I delete or update Real Estate Agent I also need to do it for Reach
Reach	registeredUser_id	CASCADE	CASCADE	If I delete or update Registered User I also need to do it for Reach
Reach	propertyInspector_id	CASCADE	CASCADE	If I delete or update Property Inspector_ I also need to do it for Reach
ReachCustomer	registeredUser_id	CASCADE	CASCADE	If I delete or update Registered User I also need to do it for ReachCustomer
ReachCustomer	customerService_id	CASCADE	CASCADE	If I delete or update Customer Service I also need to do it for ReachCustomer
Real Estate Agent	RealEstateAgent_id	CASCADE	CASCADE	If I delete or update Real Estate Agent I also need to do it for Real Estate Agent

Real Estate Agent	registeredUser_id	CASCADE	CASCADE	If I delete or update Registered User I also need to do it for Real Estate Agent
Rent	realEstateAgent_id	CASCADE	CASCADE	If I delete or update Real Estate Agent I also need to do it for Rent
Rent	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Rent
Rent	owner_id	CASCADE	CASCADE	If I delete or update Owner I also need to do it for Rent
Residential	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Residential
ReviewerBuyerRenter	registeredUser_id	CASCADE	CASCADE	If I delete or update RegisteredUser I also need to do it for ReviewerBuyerRenter
Sell	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Sell
Sell	owner_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Sell
Sell	realEstateAgent_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Sell
SingleFamily	Residential	CASCADE	CASCADE	If I delete or update Residential I also need to do it for SingleFamily
SpecialUse	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for SpecialUse
Review	estate_id	CASCADE	CASCADE	If I delete or update Estate I also need to do it for Review
Review	reviewerBuyerRenter_id	CASCADE	CASCADE	If I delete or update ReviewerBuyerRenter I also need to do it for Review
TinyHouse	residential_id	CASCADE	CASCADE	If I delete or update Residential I also need to do it for TinyHouse

