University of Calgary

Fall 2017

CPSC 471: Database Management Systems

Final Report (DRAFT)

Group **8**

Property Rental Website

Ruhma Faraz
Masih Sadat
Niroojen Thambimuthu

T.A.: Kashfia Sailunaz

Contents

1 Summary of System Requirements

Overview and definitions

Definitions. The following terms may be used throughout this final report and are used with regard to our project. They are defined as follows:

- *Rent:* the act of the user paying someone for the use of their place.
- *Owners/landlords:* Whoever owns the piece of property to rent.
- *Post:* Advertisement of property given by the owner.
- *Renters/tenants:* Whoever is interested in renting a property from an owner/landlord.
- *Budget:* The user's willingness to pay when renting out a place.
- *Convenient:* A spot with amenities which perfectly fits with the user's needs or wants.
- *Housing/place/spot/property:* Refer to a particular rental location on the market
- *Lifestyle:* The user's favorite leisure activities or pastimes in which significant time and money is spent.
- *Need/necessity(s)*: When the user requires essentials amenities for survival
- *Want/desires(s)*: When the user wishes to have non-essentials amenities to incorporate into their lifestyle.

It's difficult to find an organized and cost-effective implementation where both owners can post their ads regarding their property, and that renters can effectively browse through those ads. Thus, our group has chosen to work on a database system that will assist its users(tenants/landlords) in being able to find or post a place to rent, while using clever searching criteria which will satisfy user necessity and desires. With this project, we plan to help any users search for any specific type of spot to rent or advertise in any particular convenient area. As such, we aim to create a website where renting users can quickly access an extensive database from which they can easily choose between numerous rental property of their liking and budget. Our website will also incorporate an extensive database for property landlords where they can efficiently upload and update their place for the renters to see. Since it's very difficult to either post or find a lower budget and reasonable rental spot using local methods, our implementation will provide an extensive interface to create a convenient and user-friendly environment where they can easily decide or post any relevant property.

Problem description

Our project's purpose finds its problem in the notion that all renters and landlords are very restricted with the resources and assistance to being able to, easily and quickly, find or post a rental with their specifications. Furthermore, most sources of help are offered on back ended sites with very poor detailing and usually in brochures which takes too much time, and is limited in rentals offered. Interestingly, the market is over-saturated with countless properties that are often priced higher than most spots, and users are limited to what they can choose during time to time. Despite that setback, naturally there will be people who are very specific in spot rentals and want one that meets their needs but often compromise. Also, for property owners, the procedure of putting up their investment rental spot is a tedious and risky endeavor because it's not always clear as to what exactly needs to be prepared. Furthermore, scammers reduce the efficiency of users who wants valid and reliable information posted. As such, we find that its very intimidating task to look or publish for a rental place, and those who gives an attempt usually want one where they can specify some legit and definite criteria and have an extensive search option. The problem occurs when users are searching different websites in order to find or report a housing rental that fits their necessities. Currently, there are many different sites that offer housing rental. They will provide a list of rentals offered in a specific area. However, there are very limited websites that provide such users with detailed and accurate search engines, including the option for landlords to conveniently advertise their rental property. In particular, we have found a website that is quite similar to our proposed project, which is "rentfaster.ca". That website contains a system where housing seekers find legitimate postings, and owners can display their ads with a fee; yet the website can be improved such as removing the costs.

Solution (and system requirements)

The aim of this project is to be able to build a functional and user-friendly website that will provide users who are willing to find a housing or other spot rental, which also has an accessible database with lots of options to choose from to satisfy their needs. Also, a platform for landlords to post advertisements and details about their property which will be available to be viewed and searched by seeking rentals, including an in-depth search criterion (Area, community, No. Of rooms, budget and so forth). We will deliver to users an informative, organized website where they will be able to navigate throughout any pages with relative ease and satisfaction. We will have many different features of this website and its accompanying database will be as follows:

Relational Database:

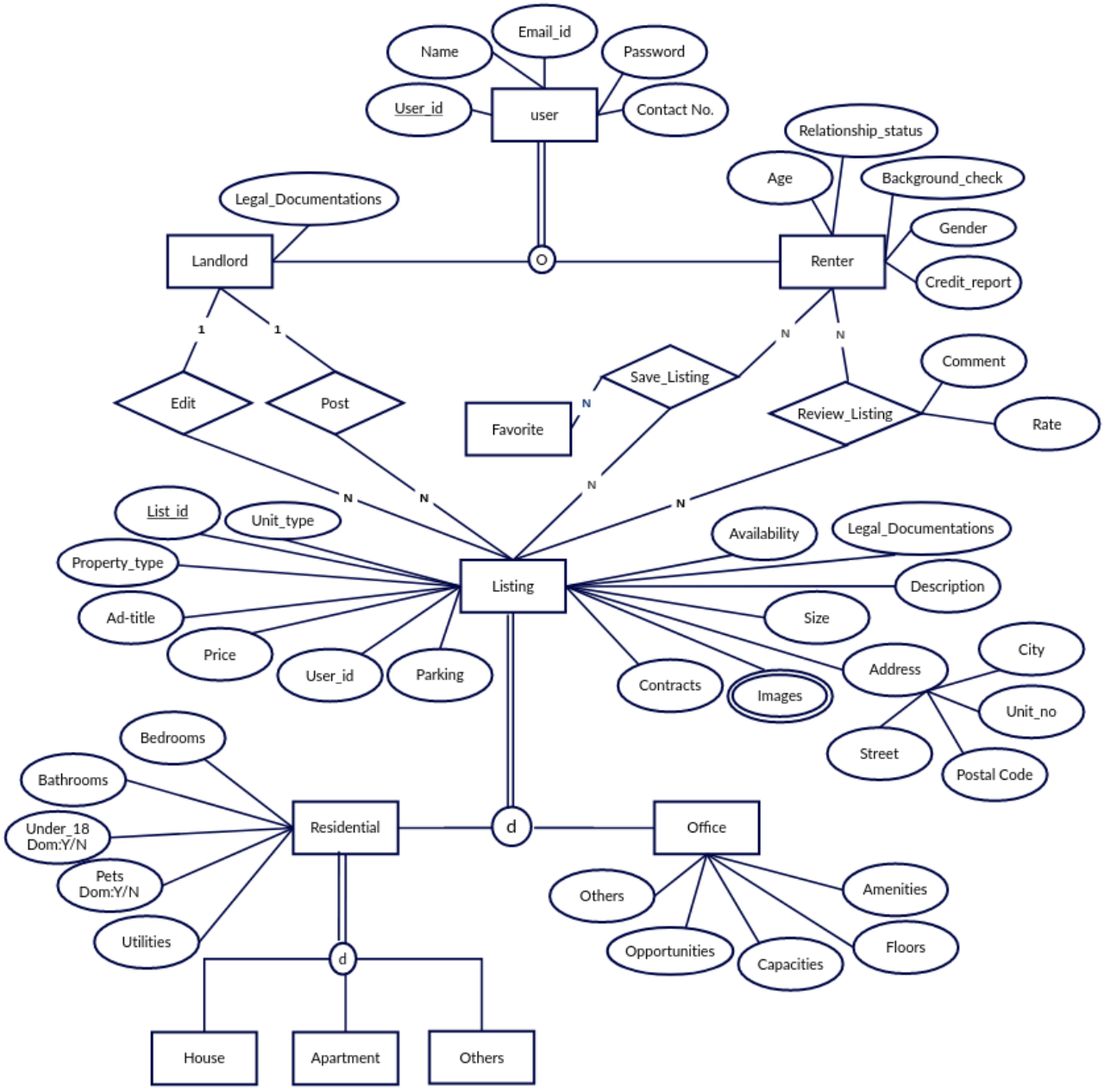
- User profiles
- AD details or House/apartment info(No. Of rooms, Bathrooms, square footage)
- Amenities details
- Geographic info(Address, Community)

GUI/User Friendly Interface :

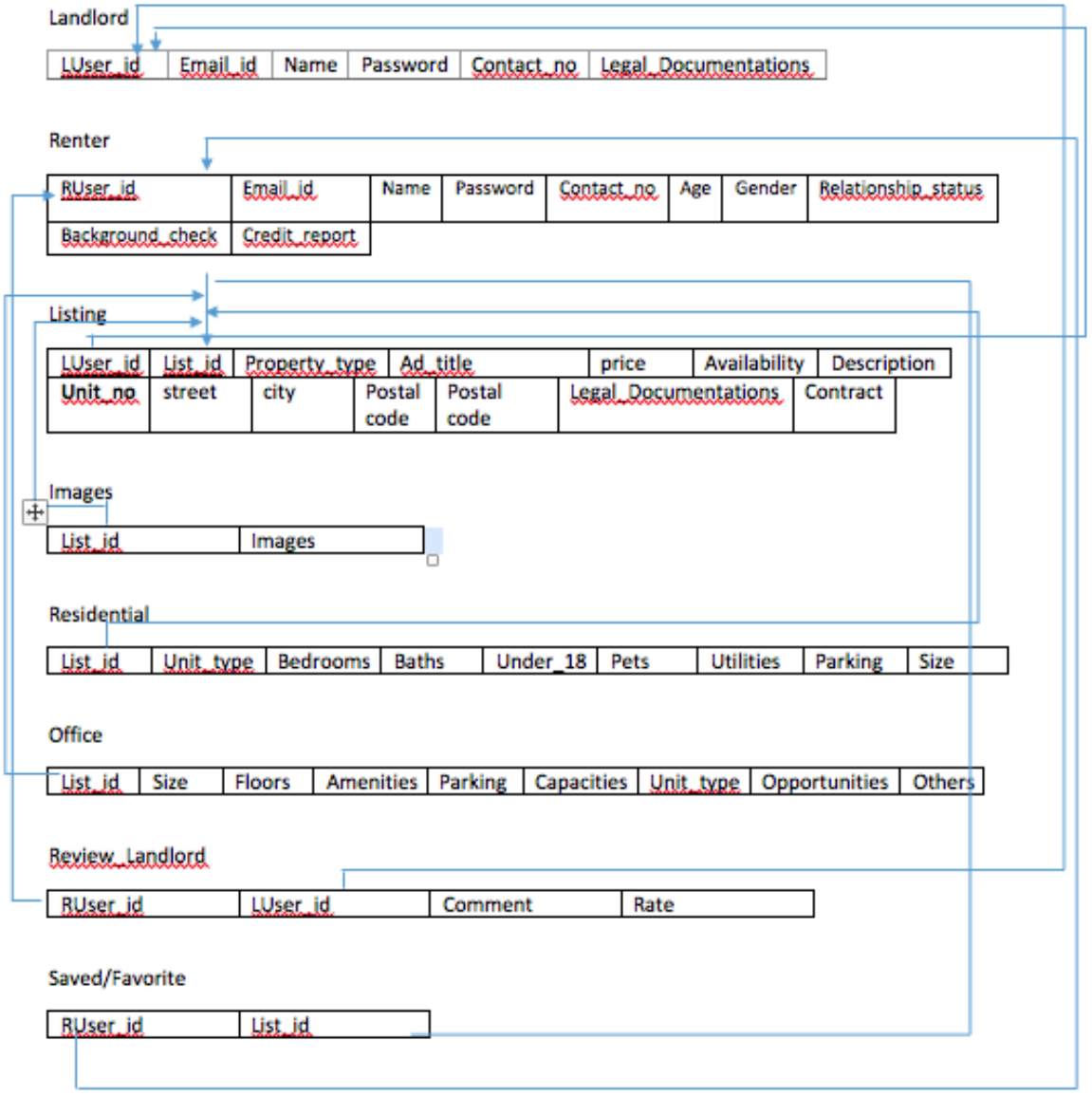
- Post ads for rental property
- View or search House/Apartment listings
- Ability to filter listings
- Check availability
- Set favorites
- Contact landlord

2 Various Models/Diagrams

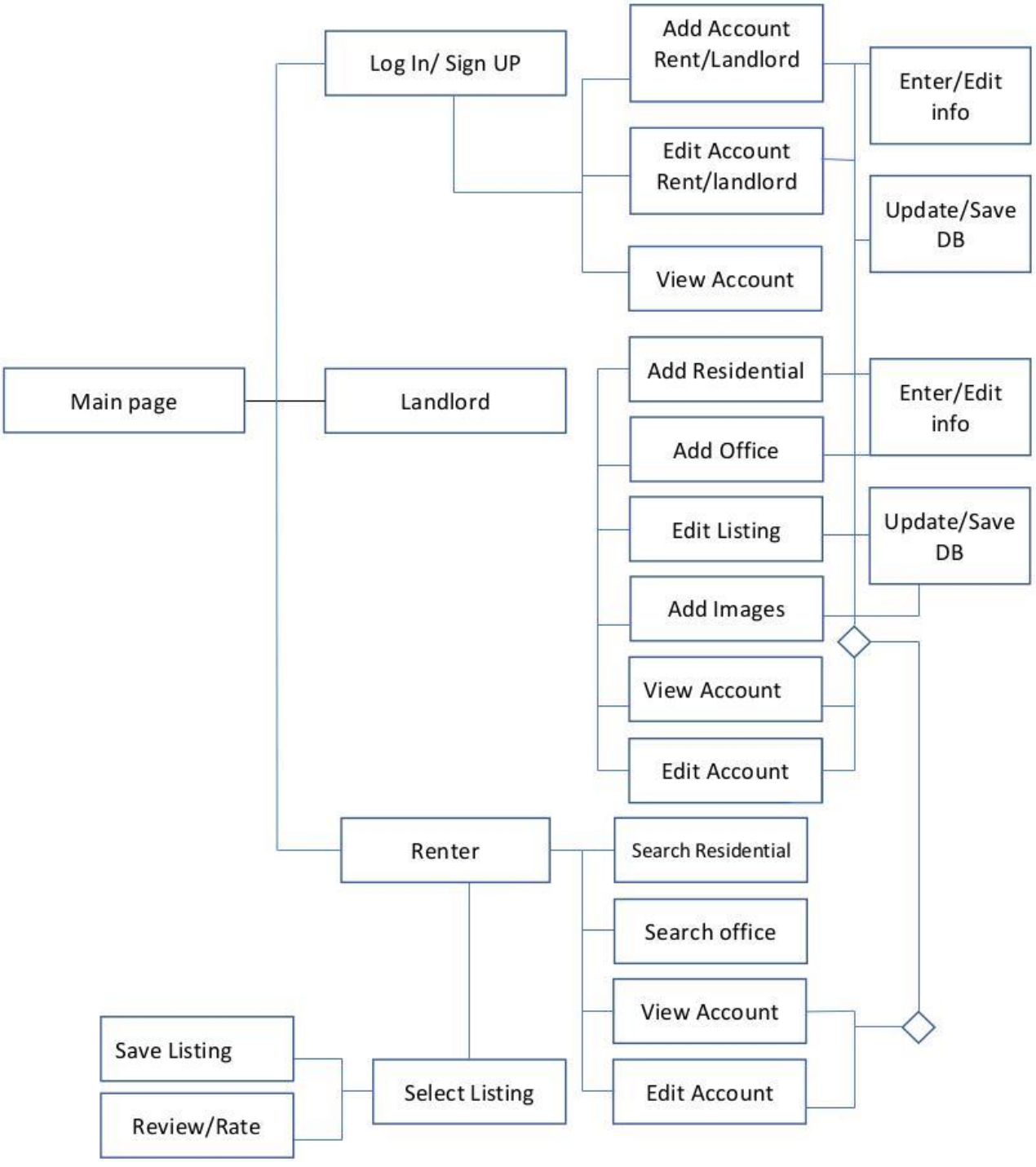
Entity-relationship diagram



Relational model (draft to be completed)



Hierarchy input process output model



Functions and pseudocode:

ADD ACCOUNT LandLord:

Inputs: @name, @Email_id, @password, @contact_no, @legaldoc

Outputs: None

Pseudocode: Connect to the database

Query = INSERT INTO Landlord values(@name, @Email_id, @password, @contact_no, @legaldoc);

Parse Query

Execute Query

Close connection to the database

ADD ACCOUNT Renter:

Inputs: @name, @Email_id, @password, @contact_no, @age, @gender, @relationship, @background_check, @credit_report

Outputs: None

Pseudocode: Connect to the database

Query = INSERT INTO Renter values (@name, @Email_id, @password, @contact_no, @age, @gender, @relationship, @background_check, @credit_report);

Parse Query

Execute Query

Close connection to the database

EDIT ACCOUNT Landlord:

Inputs: @ Luser_id, @name, @Email_id, @password, @contact_no, @legaldoc

Outputs: None

Pseudocode: Connect to the database

Query =

UPDATE Landlord

Set name=@name, Email_id=@Email_id, password=@pasword, contact_no=@contact_no, legaldoc=@legaldoc

where Luser_id=@Luser_id;

Parse Query

Execute Query

Close connection to the database

EDIT ACCOUNT Renter:

Inputs: : @Ruser_id,@name, @Email_id, @password, @contact_no, @age, @gender, @relationship, @background_check, @credit_report

Outputs: None

Pseudocode: Connect to the database

Query =

UPDATE Renter

Set name=@name, Email_id=@Email_id, password=@pasword, contact_no=@contact_no, age=@age, gender=@gender, relationship=@relation, credit_report=@creditreport

where Ruser_id=@Ruser_id;

Parse Query

Execute Query

Close connection to the database

VIEW ACCOUNT LANDLORD:

Inputs: @Email_id, @password

Outputs: @name, @Email_id, @password, @contact_no, @legaldoc

Pseudocode: Connect to the database

Query =

Select *

From Landlord

Where (Email_id=@Email_id) And (password=@password);

Parse Query

Execute Query

Close connection to the database

VIEW ACCOUNT Renter:

Inputs: @ Email_id, @passowrd

Outputs: @name, @Email_id, @password, @contact_no, @age, @gender, @relationship, @background_check, @credit_report

Pseudocode: Connect to the database

Query =

Select *

From Renter

Where (Email_id=@Email_id) And (password=@password);

Parse Query

Execute Query

Close connection to the database

ADD Residential Listing:

Inputs: @Email, @Luser_id, @property_type, @Ad_title, @price, @availibilty, @Description, @street, @city, @postalcode, @legaldoc, @contract

@unit_type,@bedroom, @baths, @under_18, @pets, @utilities, @parking, @size

Outputs: None

Pseudocode: Connect to the database

Query =

Begin

INSERT INTO Listing Values ((select Luser_id from Landlord where Email_id=@email_id), @property_type, @Ad_title, @price, @availibilty, @Description, @street, @city, @postalcode, @legaldoc, @contract)

Select Last_Insert_id() INTO @Var

INSERT INTO Residential values (@var, @unit_type, @bedroom, @baths, @under_18, @pets, @utilities, @parking, @size)

End;

Parse Query

Execute Query

Close connection to the database

ADD Office Listing:

Inputs: @ @Email, @Luser_id, @property_type, @Ad_title, @price, @availibilty, @Description, @street, @city, @postalcode, @legaldoc, @contract

@size, @floor, @amenities, @parking, @capacity, @unit/-type, @others

Outputs: None

Pseudocode: Connect to the database

Query =

Begin

INSERT INTO Listing Values ((select Luser_id from Landlord where Email_id=@Email_id), @property_type, @Ad_title, @price, @availibilty, @Description, @street, @city, @postalcode, @legaldoc, @contract)

Select Last_Insert_id() INTO @Var

INSERT INTO Office values (@var, @size, @floor, @amenities, @parking, @capacity, @unit/-type, @others)

End;

Parse Query

Execute Query

Close connection to the database

ADD Images:

Inputs: @ List_id, @image

Outputs: None

Pseudocode: Connect to the database

Query = INSERT INTO Images Values(@List_id, image);

Parse Query

Execute Query

Close connection to the database

Search Residential Listing:

Inputs: @minprice, @maxprice, @availibilty, @street, @city, @unit_type, @bedroom

Outputs: @property_type, @Ad_title, @price, @availibilty, @Description, @street, @city, @postalcode, @legaldoc, @contract

@unit_type,@bedroom, @baths, @under_18, @pets, @utilities, @parking, @size

Pseudocode: Connect to the database

Query =

Select *

From Listing as L, Residential as R

Where L.availibilty=@availibility And

L.street=@street And

L.city=@city And

R.uniy_type= @unit_type And

R.bedroom=bedroom And

(L.price <= @maxprice And L.price >= @minprice));

Parse Query

Execute Query

Close connection to the database

Search Office Listing:

Inputs: @minprice, @maxprice, @availibilty, @street, @city, @unit_type,

Outputs: @property_type, @Ad_title, @price, @availibilty, @Description, @street, @city, @postalcode, @legaldoc, @contract

@size, @floor, @amenities, @parking, @capacity, @unit/-type, @others

Pseudocode: Connect to the database

Query =

Select *

From Listing as L, Office as O

Where L.availibilty=@availibility And

L.street=@street And

L.city=@city And

O.uniy_type= @unit_type And

(L.price <= @maxprice And L.price >= @minprice));

Parse Query

Execute Query

Close connection to the database

Save/fave Listing:

Inputs: @ List_id, @Ruser_id

Outputs: None

Pseudocode: Connect to the database

Query =

INSERT INTO Fave Values(@List_id, @Ruser_id);

Parse Query

Execute Query

Close connection to the database

Review/Rate:

Inputs: @ Luser_id, Ruser_id, Rate, Comment

Outputs: None

Pseudocode: Connect to the database

Query = INSERT INTO Review_Renters Values(@LUser_id, @Ruser_id, @Rate, @comment);

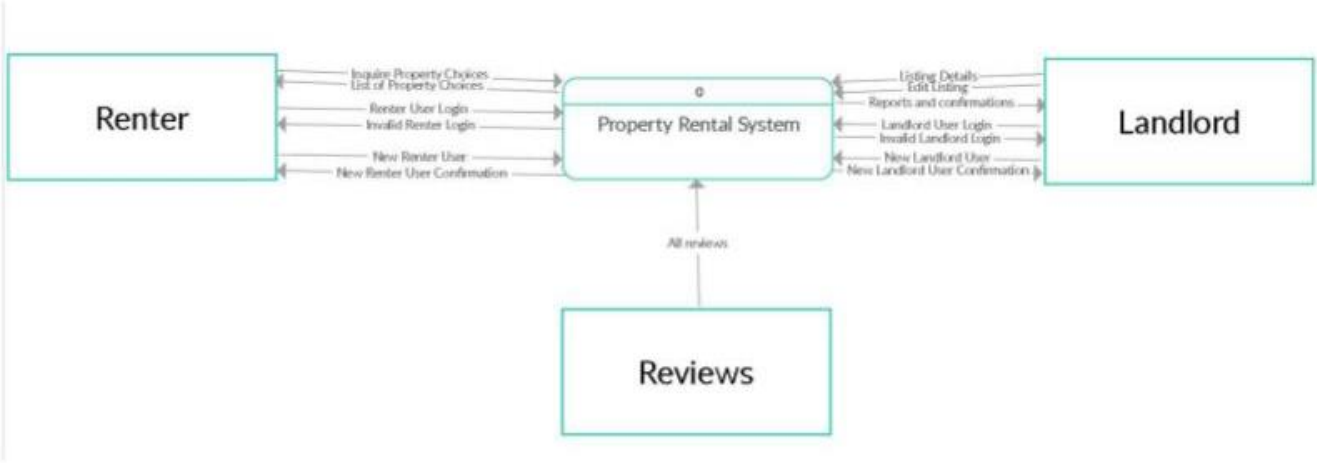
Parse Query

Execute Query

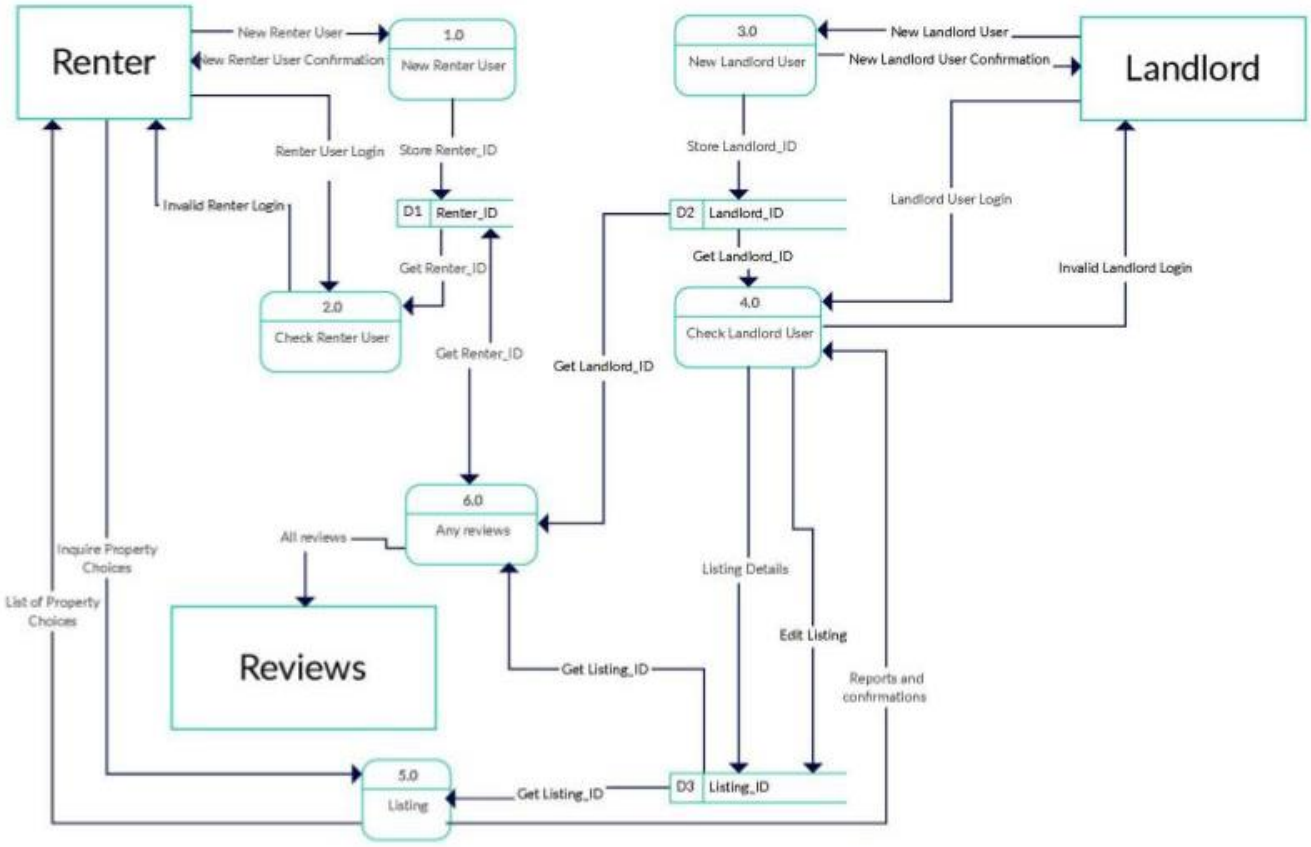
Close connection to the database

Data flow diagram

Context Diagram:

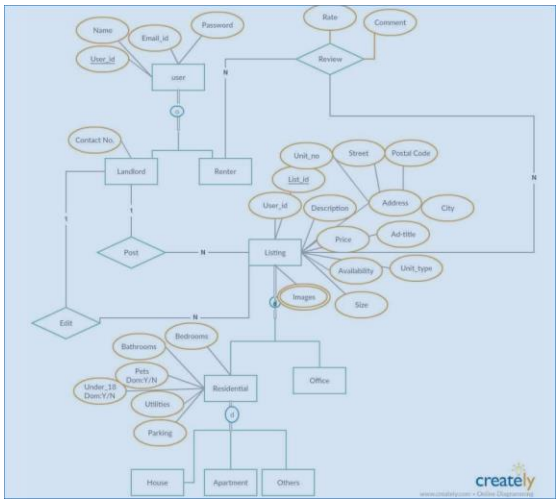


Level 0 Diagram:

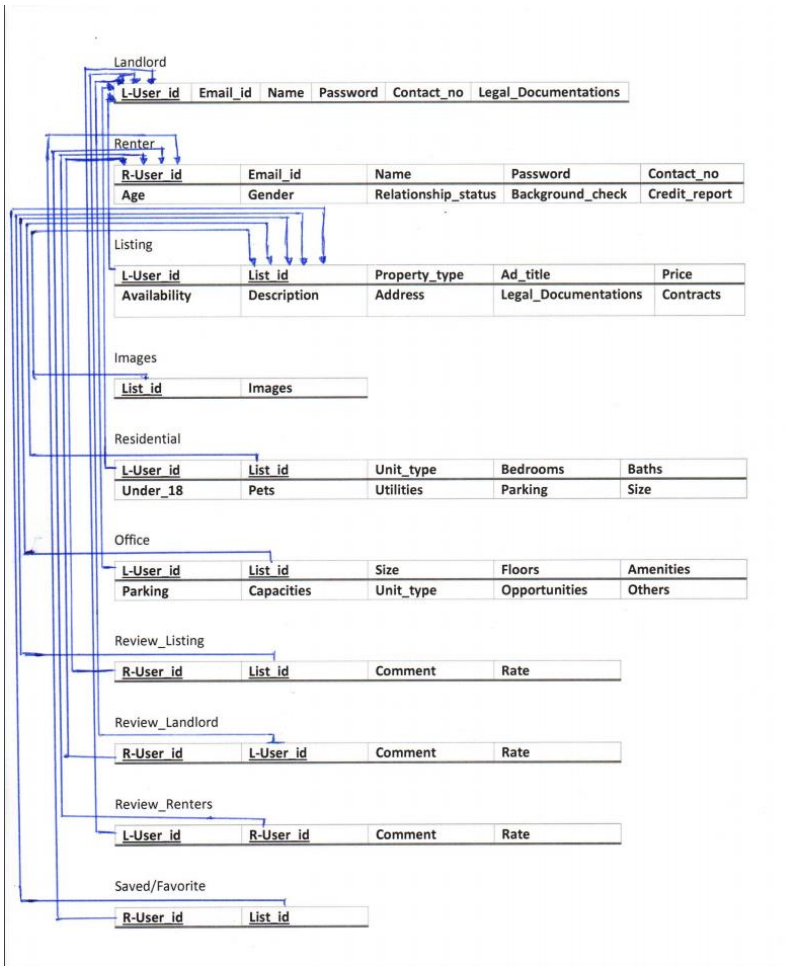


Previous Diagrams:

ERD:



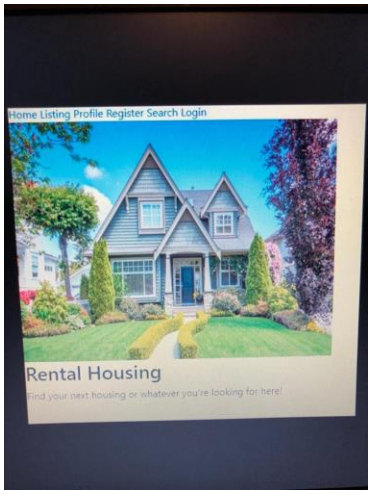
RM



3 User Manual (Draft to be completed)

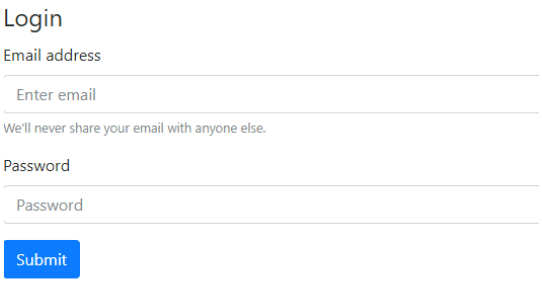
1. Main page

All users will be greeted by this main page. It has a simplistic design with 6 buttons on the top left corner. The buttons are Home, Listing, Profile, Register, Search and Login.

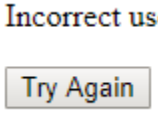


2. Signing into user account

By accessing the login page from the main, the user will be presented by this login page as shown below. This page contains 2 input fields for user email address and password respectively, along with a submit button. All landlords and renters have to use this login.

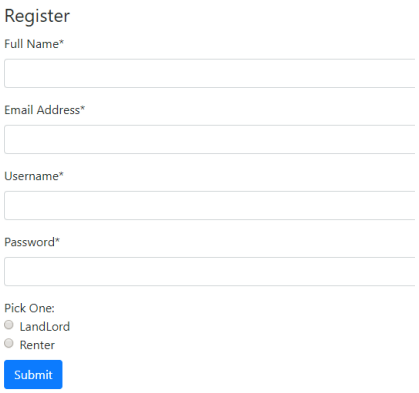


In the case that either the email address or the password, even both, is not properly entered, then an incorrect feedback will be displaying to be user.



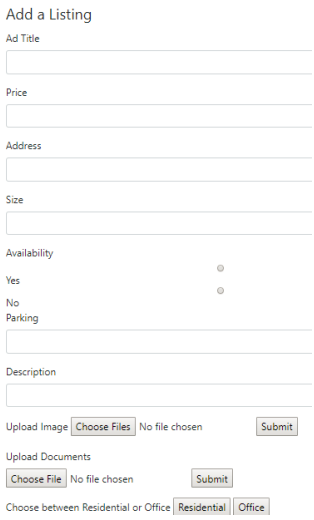
3. Registering an account

From the main page, any new landlord or renter user can create a new account by clicking on the Register button. This will open up a registration page as shown below, and will prompt all users to enter their respective Full Name, Email Address, Username and a potent Password. What will differentiate between a landlord or a user account, during registration, is the 2 Pick One radio button with one being "LandLord" and the other being "Renter". After selecting the proper button, the user can now click on the submit button, and their new account has been created.



4. Landlord adding a Listing

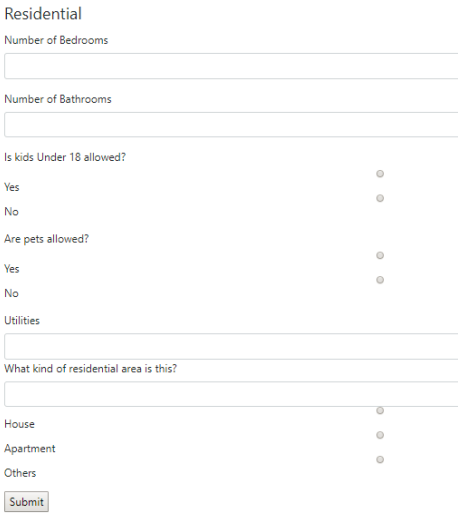
Landlord users can add a listing from their profile, which can be accessed by the Profile button on the main menu. Whenever a landlord is to add a listing, they would come across the Add a listing page, as shown below. The landlord user is to input an ad title, price, address, size, parking and description into separate written input fields. The two-availability button have a "Yes" or "No" as n option, so that renters will know if the property isn't for rent anymore. The landlord user is to upload pictures and legal documents to shown renters of their property being legally livable. At the end of the page, the landlord has to choose between 2 buttons named "Residential" and "Office", so that the resident and office listing can be filtered out. This act will create the listing page.



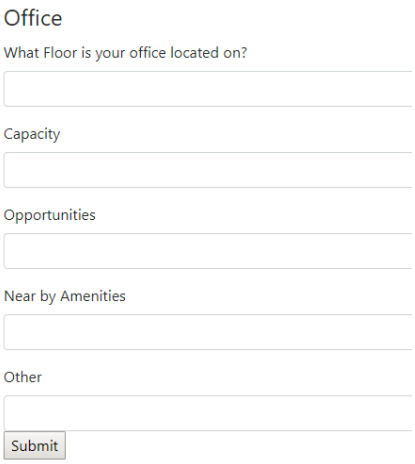
5. Searching Residential property

The Listing button from the main page, will guide a user to choose between 2 property types, either residential or office.

If the user chooses the Residential listing, then they will be loaded to the Residential page as shown below. From here, the user can input the number of beds, baths, utilities and what kind of residential area benefits the location provides. There's radio buttons to provide users with simple yes/no questions such as under 18 tenants allowed and pet friendly housing. The renter can also choose the property type of their liking such as house, apartment or others. Then the user is to submit and receive a list of residential listings.



If the user chooses the Office listing, then they will be loaded to the Office page as shown below. The user can input the wanted offices options listings such as office floor, capacity, opportunities, nearby amenities and others. The the user is to submit and receive a list of office listings.



6. Draft
7.

Appendix: List of relational instances

To be added later, populating the databases with sample relation instances.