Phase 1 : Rentals.com Database Project

Presented to
Professor Animesh, Animesh

Contributors
Delisle, Audrey
Mehri, Meriem
Luo, Xingchen
Majidi, Sheida
Yu, Xinran

INSY 661

McGill University - Desautels Faculty of Management

Table of Contents

1. Overview of the business scenario

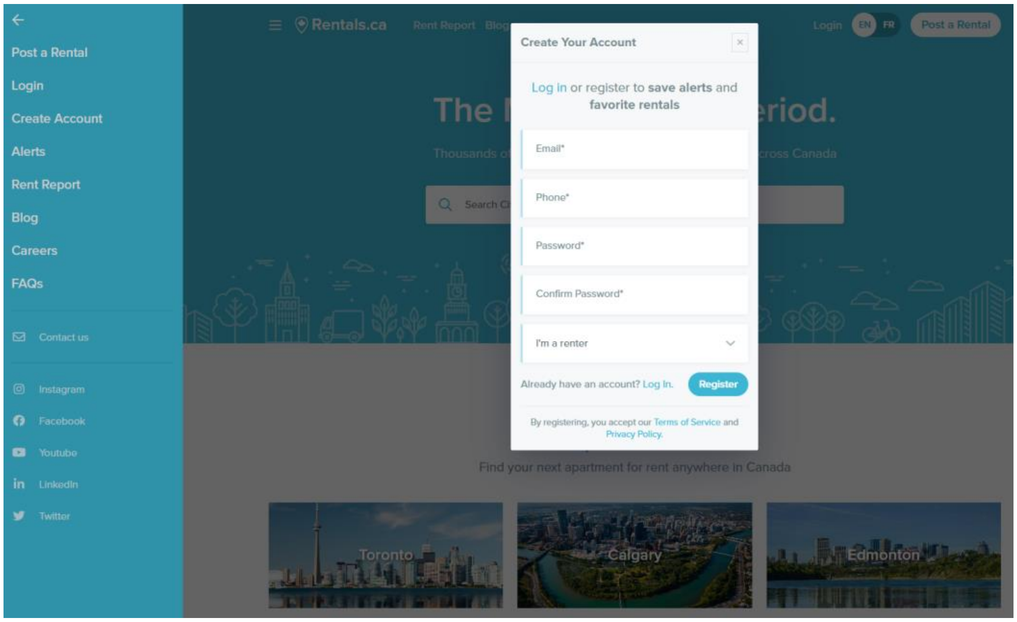
Rentals.ca is a website that provides rental accommodation listings across Canada. The business aims to help tenants find their ideal place to live, and landlords to advertise their properties to potential renters. The business processes include:

- Posting a rental: Landlords can create an account and post their rental properties on the website, with details such as location, price, size, amenities, and photos. They can also manage their listings, email preferences, and contact support through their account.
- Searching for a rental: Tenants can browse through thousands of apartments, houses, and condos for rent across Canada, using filters such as city, neighborhood, address, or ad number. They can also create alerts to receive notifications when new listings match their criteria.
- Rent report: Rentals.ca publishes a monthly rent report that analyzes the average asking rents in Canada and various provinces and cities, based on data from purpose-built and condominium apartments. The report provides insights into the trends and factors affecting the rental market in Canada.
- Blog: Rentals.ca also maintains a blog that features articles on topics such as renting tips, market news, lifestyle, and design. The blog aims to provide useful information and advice for both tenants and landlords.

2. Screenshots of Website Features & Business Functionalities for Database Scope

All screenshots are derived from *Rentals.ca* ©

User Profile Creation



Create Your Account

Log in or register to **save alerts** and **favorite rentals**

Email*

Phone*

Password*

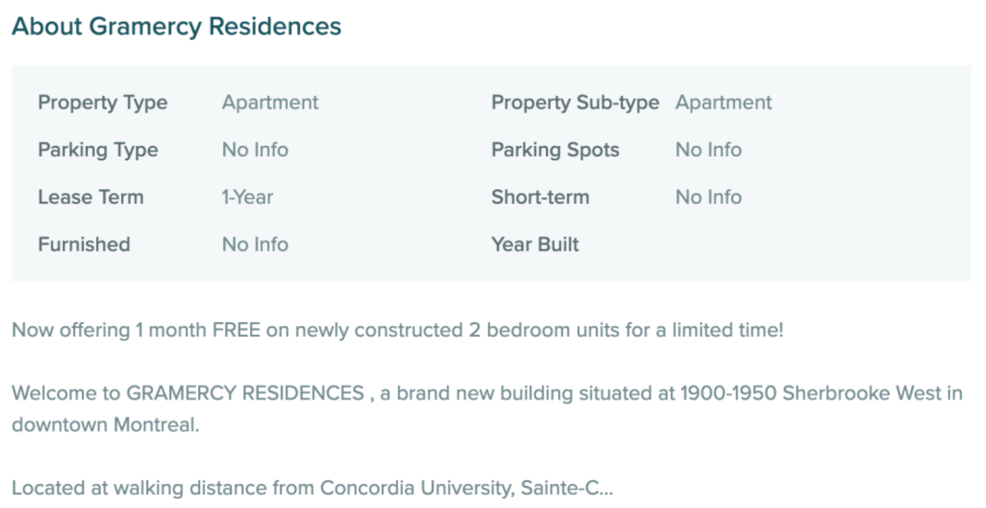
Confirm Password*

I'm a renter

Already have an account? Log In.

By registering, you accept our Terms of Service and Privacy Policy.

Unit Features & Amenities



About Gramercy Residences

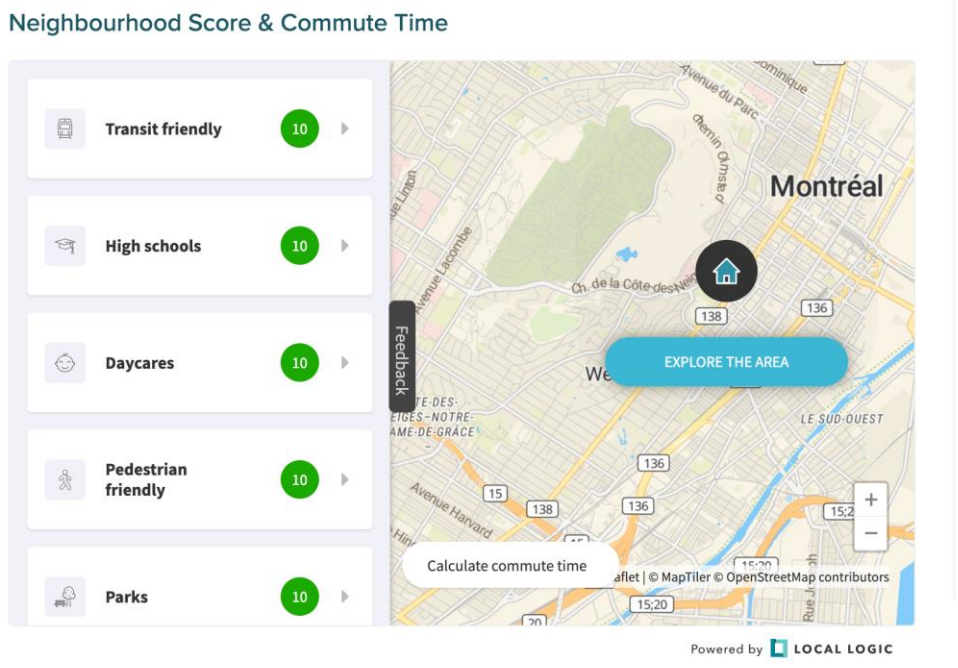
Property Type	Apartment	Property Sub-type	Apartment
Parking Type	No Info	Parking Spots	No Info
Lease Term	1-Year	Short-term	No Info
Furnished	No Info	Year Built	

Now offering 1 month FREE on newly constructed 2 bedroom units for a limited time!

Welcome to GRAMERCY RESIDENCES , a brand new building situated at 1900-1950 Sherbrooke West in downtown Montreal.

Located at walking distance from Concordia University, Sainte-C...

Building Rating (Score)

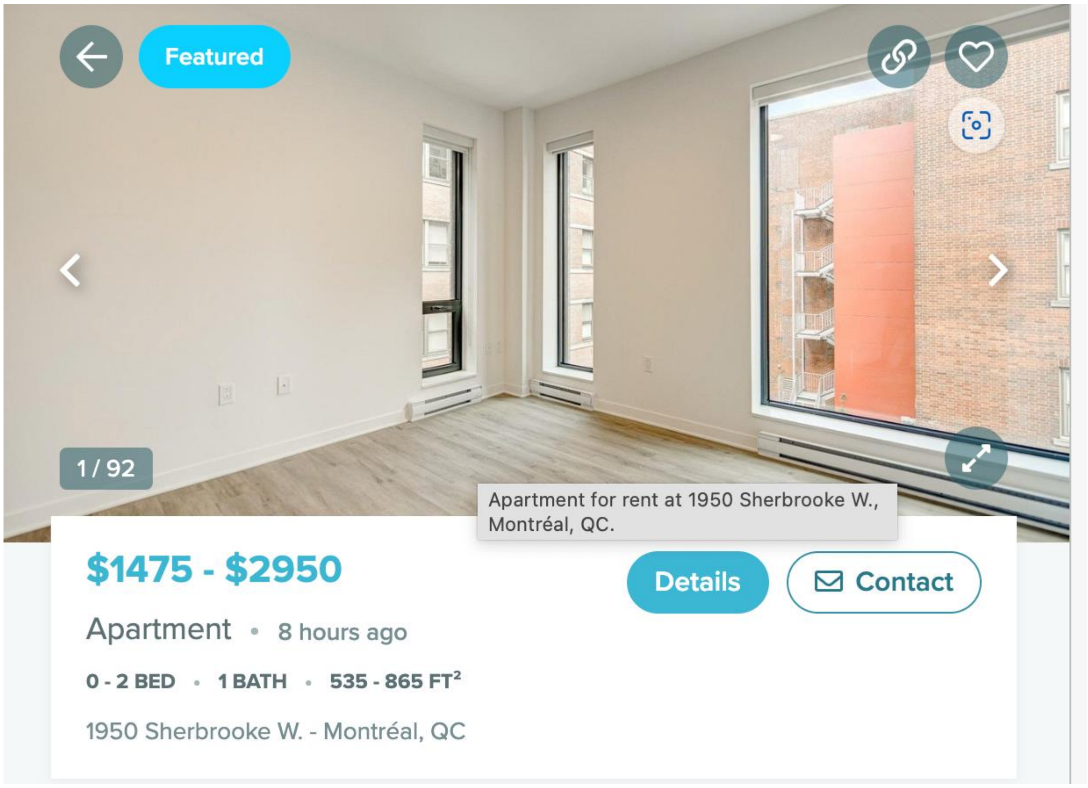


Neighbourhood Score & Commute Time

Transit friendly	10
High schools	10
Daycares	10
Pedestrian friendly	10
Parks	10

Powered by LOCAL LOGIC

Property Details/Unit Info



Apartment for rent at 1950 Sherbrooke W., Montréal, QC.

$1475 - $2950

Apartment • 8 hours ago

0 - 2 BED • 1 BATH • 535 - 865 FT²

1950 Sherbrooke W. - Montréal, QC

3. Mission Statement & Objectives

Mission Statement:

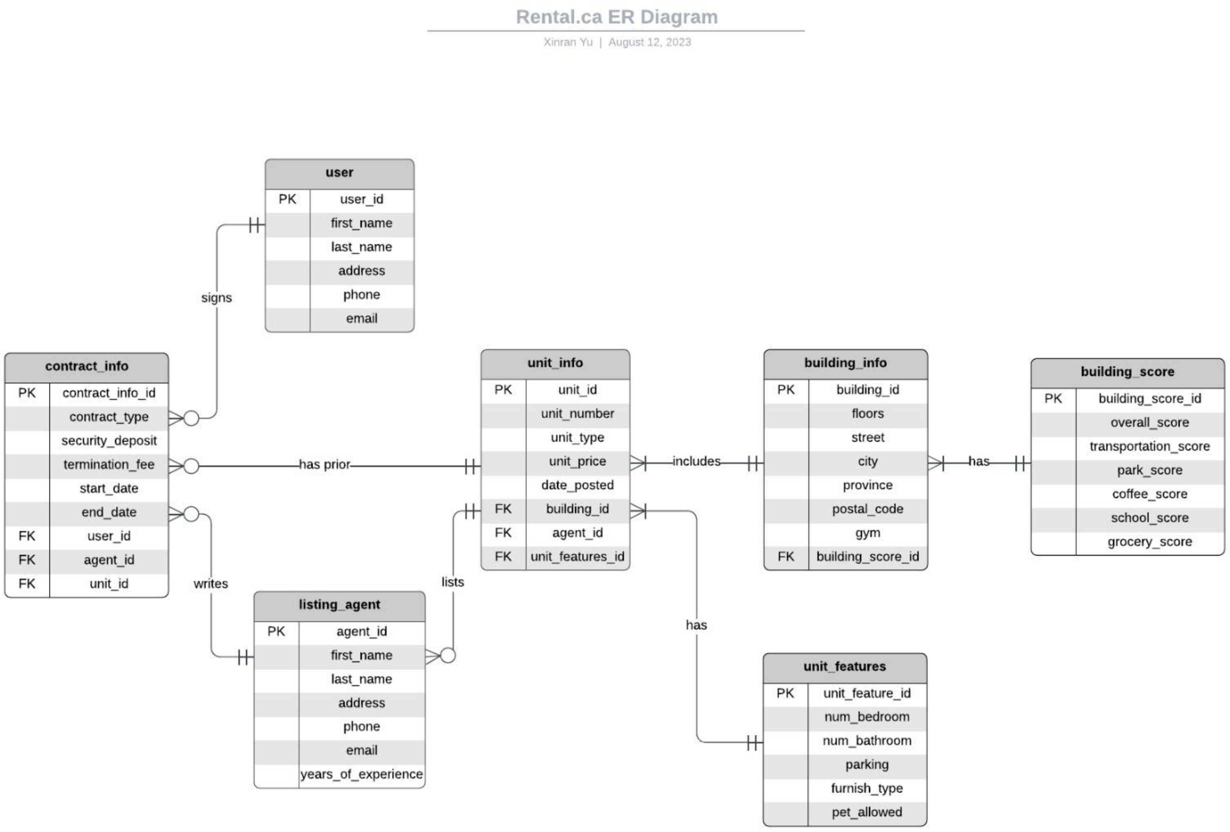
The Rentals.ca database system serves as the backbone of a dynamic, secure, and scalable national apartment rental platform. Its key mission is to provide instant listing updates, precise property details, and tailored user suggestions. Emphasizing data integrity and privacy, its integration enhances user engagement, refines search capabilities, and streamlines the rental process. By delivering a high-performing and reliable solution, Rentals.ca connects individuals with their ideal living spaces, fostering trust and convenience among tenants, property owners, and lenders across the country.

Database Objectives:

The Rentals.ca database encompasses multiple objectives, each pinpointing a specific task along with the necessary resources for its execution, including the required activities and mechanisms to be deployed.

Database Objective	**Required Resources/Activities**
To support instantaneous updates and availability of apartment listings.	*Synchronization between listing agents, property owners and the website to ensure an efficient listing management.*
To store and maintain precise information regarding the properties' details, features, amenities, location and lease terms.	*Adequate storage and data validation mechanisms to preserve accuracy.*
To develop workflows simplifying the apartment renting process, from listing selection to lease agreement.	*Adequate budget will be allocated for process optimization, user testing, and continuous improvement efforts.*
To ensure uninterrupted service availability thanks to a high-performance server infrastructure.	*Adequate financial resources will be dedicated to hardware upgrades and regular server performance testing.*
To safeguard sensitive information and establish encryption mechanisms & secured authentication processes.	*Adequate investment will be allocated to invest in robust cybersecurity tools.*
To maintain a user-friendly interface for intuitive navigation, and efficient search functionalities.	*Appropriate frontend and backend development resources will be allocated for continuous UI/UX enhancements on Rentals.ca website.*
To offer customized recommendations tailored to the users' needs and preferences in terms of budget, location, history, interactions, etc.	*Development of an algorithmic framework using machine learning tools for personalized recommendations based on user preferences and interactions.*
To foster user trust and convenience by facilitating transparent communication between tenants, property owners, and lenders.	*Sufficient budget for customer service enhancements and ongoing support resources.*
To ensure scalability of the database to accommodate a growing volume of listings and users across the country.	*Adequate financing for server scaling, database optimization, and network expansion.*

4. Entity Relationship Diagram (ERD)



5. Data Dictionary

a) Description of Entities

Entity Name	Description	Aliases	Occurrence
user	Contains all information relative to all individual users of the system including their unique identifiers, name, contact information, etc.	User	One user can sign zero or many contract_info.
unit_info	Contains information about the unit type, price, listed date, etc.	Unit	One unit_info matches with none or many contract_info. One unit_info corresponds to only one listing_agent.

unit_features	Conatains relative information regarding the unit's characteristics and included/excluded amenities.	Unit Features	One unit_feature fits with one or many info_unit.
building_info	Contains necessary information for a building, such as address and equipments.	Building Information	One building_info can has only one building score. One building_info can includes one or many unit_info.
building_score	Contains the scores for the rated buildings, based on several criteria such as proximity of stores, public transportation, etc.	Building/ Neighborhood rate	One building score can be owned by one or many building_info.
listing_agent	Contains the information of agents who are listing the units on the website.	Listing Agent	One listing_agent can write zero or many contract_info. One listing_agent can list zero or many unit_info
contract_info	Conatains the detailed information of contracts approved/signed on the website.	Lease Information	One contract_info can be listed by only one listing_agent. One contract_info can be signed by only one user. One contract_info can have by only one unit_info.

b) Description of Attributes

Entity Name	Attributes	Description	Data Type	Nulls	Muli-valued	Derived	Default
user	user_id	Unique ID for each user (*current or potential tenant*)	10 variable char	No	No	No	None
	first_name	The first name of the user	255 variable chars	No	No	No	None
	last_name	The last name of the user	255 variable chars	No	No	No	None
	address	The user's current address	text	No	No	No	None
	phone	The user's phone number	15 variable chars	No	No	No	None
	email	The user's email address	255 variable chars	No	No	No	None
unit_info	unit_id	Unique ID for unit	10 variable char	No	No	No	None
	unit_number	Number of units	50 variable chars	No	No	No	None
	unit_type	Type of unit	50 variable chars	No	No	No	None

	unit_price	Price of unit	10 digits with 2 decimals	No	No	No	None
	area	Unit area	int	No	No	No	None
	date_posted	Unit's posting date on the website	date	No	No	No	None[1]
	agent_id	The agent ID the contract was written by	10 variable char	No	No	listing_agent	None
	unit_features_id	Unique ID for the unit's features	10 variable char	No	No	unit_features	None
	building_id	Unique ID for each building	10 variable char	No	No	building_info	None
unit_features	unit_feature_id	Unique ID for the unit's features	10 variable char	No	No	No	None
	num_bedroom [not null]	Number of bedrooms	int	No	No	No	None
	num_bathroom [not null]	Number of bathrooms	int	No	No	No	None
	furnish_type	If the unit is furnished or not	255 variable chars	No	No	No	None
	pet_allowed	If pets are allowed or not	boolean	No	No	No	FALSE[2]
	parking	If there's a partking space or not	255 variable chars	No	No	No	None
building_info	building_id	Unique ID for each building	10 variable char	No	No	No	None
	floors	The number of floors in the building	int	No	No	No	None
	street	The street the building is on	255 variable chars	No	No	No	None
	city	The city the building is in	255 variable chars	No	No	No	None
	province	The province the building is in	50 variable chars	No	No	No	None

[1] If no calendar date is entered, we assume that the default date is set to the current calendar date. To do so, the functions GETDATE() or CURRENT_TIMESTAMP() or SYSDATETIME() can be used.

[2] If no record is entered in the field corressponding to pet_allowed we can suppose that they are not included in the units. The corresponding 'default' value would be FALSE.

	postal_code	The postal code the building has	10 variable chars	No	No	No	None
	gym	If the building has a gym	boolean	No	No	No	None
	building_score_id	The id of building score the building has	10 variable char	No	No	building_score	None
building_score	building_score_id	Unique ID for each building score	10 variable char	No	No	No	None
	overall_score	Average score the building score has	int	No	No	No	None
	transportation_score	The score for public transportation around the building	int	No	No	No	None
	park_score	The score for parks around the building	int	No	No	No	None
	coffee_score	The score for cafes around the building	int	No	No	No	None
	school_score	The score for schools around the building	int	No	No	No	None
	grocery_score	The score for grocerty stores around the building	int	No	No	No	None
listing_agent	agent_id	Unique ID for each agent	10 variable char	No	No	No	None
	first_name	The first name of the agent	255 variable chars	No	No	No	None
	last_name	The last name of the agent	255 variable chars	No	No	No	None
	address	The address of the agent	text	No	No	No	None
	phone	The phone number of the agent	15 variable chars	No	No	No	None
	email	The email of the agent	255 variable chars	No	No	No	None
	years_of_experience	The number of years the agent works as a agent	int	No	No	No	None
contract_info	contract_info_id	Unique ID for each contract	10 variable char	No	No	No	None
	unit_id	The unit ID the contract signed for	10 variable char	No	No	unit_info	None

	contract_type	The contract type: long_term or short_term	50 variable chars	No	No	No	None
	security_deposit	The amount of the security deposit	10 digits with 2 decimal	No	No	No	None
	termination_fee	The amount of the termination fee	10 digits with 2 decimal	No	No	No	None
	start_date	The start date of the contract	date	No	No	No	None
	end_date	The end date of the contract	date	No	No	No	None
	user_id	The user ID the contract was signed by	10 variable char	No	No	user_info	None
	agent_id	The agent ID the contract was written by	10 variable char	No	No	listing_agent	None

6. Relational Schema

User (user_id, first_name, last_name, address, phone, email)
Primary Key: user_id

Unit_Info (unit_id, unit_number, unit_type, unit_price, area, date_posted, agent_id, unit_features_id, building_id)
Primary Key: unit_id
Foreign Key: agent_id References Listing_Agent (agent_id)
Foreign Key: unit_features_id References Unit_Features (unit_features_id)
Foreign Key: building_id References Building_Info (building_id)

Unit_Features (unit_feature_id, num_bedroom, num_bathroom, furnish_type, pet_allowed, parking)
Primary Key: unit_feature_id

Building_Info (building_id, floors, street, city, province, postal_code, gym, building_score_id)
Primary Key: building_id
Foreign Key: building_score_id References Building_Score (building_score_id)

Building_Score (building_score_id, overall_score, transportation_score, park_score, coffee_score, school_score, grocery_score)
Primary Key: bulding_score_id

Listing_Agent (agent_id, first_name, last_name, address, phone, email, years_of_experience)

Primary Key: agent_id

Contract_Info (contract_info_id, unit_id, contract_type, security_deposit, termination_fee, start_date, end_date, user_id, agent_id)
Primary Key: contract_info_id
Foreign Key: unit_id References Unit_Info (unit_id)
Foreign Key: user_id References User (user_id)
Foreign Key: agent_id References Listing_Agent (agent_id)