

Project Part 1

A rental company hires you to develop a new web application to accommodate their massive rental market. They want you to design and develop an application named *Roomio* that helps people find roommates and/or rooms in apartments.

In Part 1 of the project, you will design an ER model for Roomio's database, capturing the rental company's requirements, detailed below. In later parts of the project, we will provide a solution to Part 1 and you will use it to design a relational database schema, and then implement and test queries and application code for users to post data about available apartments and about users' interest in them.

ER requirements:

When a **user** registers to the Roomio system, they are required to have a unique username with additional information like first name, last name, date of birth, gender, one or more email addresses and phone numbers, and a password.

A user may have **pets**; they need to specify their pet's name, type, and size. A user may have pets of different types that have the same name, but their pets of the same type are required to have different names. (For example, a user could have a dog and a cat that are both named 'Spot', but cannot have two dogs that are both named 'Spot').

The database will store information about **ApartmentUnits**, where each unit has a unique UnitRentalID and other information like unit number, monthly rent price, approximate square footage, and an available date for move-in. Each unit may have different **Rooms** whose name (e.g. 'bedroom1', 'bedroom2', etc.), square footage and description will be stored. The names of rooms are unique within each Apartment unit, but the different Apartment Units may have the same room name(s). For example, both Apartment Unit 12345 and Unit 54321 might have a room named 'bedroom1'.

Each unit is in an **ApartmentBuilding**. Each apartment building will be identified by both the company name and the building name (for example 'Hudson View LLC', building 3). Apartment Buildings also have information like address (comprised of number, street, city, state, and zip code) and year built.

Buildings and Units may have **amenities**. The rental company will supply a fixed list of amenities, and the database should record which units have which amenities (such as dishwasher, air-conditioning, etc) and which buildings have which additional amenities available to all residents (such as shared laundry room, gym, roof-deck, etc). Extra comments about buildings' amenities, such as fees, waiting lists, etc., should also be noted.

Each apartment building may have a **Pet Policy** that indicates whether dogs and/or cats and/or rodents are allowed, the registration fee and the monthly fee for pets of different types and sizes.

Users can post Interest in an apartment unit and specify how many roommates and the expected move-in date for other users to see.

ADDITIONAL REQUIREMENTS:

Draw your ER diagram neatly, either by hand or using a drawing tool. You **must** use the notations studied in class. You may use either the double line and arrow notation for participation and cardinality constraints or the lower-bound .. upper-bound notation. Your diagram should fit easily onto a single page – if you have more than about 10 to 15 components (strong or weak entity sets or relationship sets) you should rethink your design.

On a separate page, indicate any assumptions you are making about any ambiguous requirements. Also indicate briefly what each (strong or weak) entity set and each relationship set in your design corresponds to from the textual description, above.

You may work alone or with one or two classmates. You may work alone on this part and then team up with others for the later parts of the project, but once you form a team, you should stay with them for the duration of the project (except in

very unusual circumstances, such as illness of a teammate or withdrawal from the class). Consequently, if you are forming a team for part 1, you should think ahead to what programming language(s) you'll use for the application programming part of the project, later. Material on how to implement web-based database applications in Python with Flask and MySQL will be presented in class soon. If you are working on Part 1 with others, you should send the names of everyone on your team to the TAs at least one week before Part 1 is due. (A form will be provided for doing this.)