

Topic:

An application that lets users explore a list of student housing. It will support:

- CRUD dorm/apartment listings
- Sort alphabetically and filter by location/building, cost, or by user input
- Owner/renter (user) username and password
- Users can rate dorm/apartment listings
- Filter available and unavailable units (Redis)
- Rank listings by ratings (Redis)

Business Requirements:

Narrative: **Nouns & Verbs**

Let's say there is a student William Peterson who is enrolled at Northeastern University Computer Science Graduate Program. William is looking for housing for the coming fall semester this year but is on a budget. He ends up signing up to a student housing site to explore student dorms/apartments listings that are available*. The listings include information regarding the date the unit opens on the market, the size by square foot of the unit, type of unit (either a dormitory room or an apartment), the location or building it is in depending on the type of unit, and the name of the owner. Once registered he starts by filtering criteria of apartment based on desired price, and location. Then selects a few that he is interested in and leaves excellent/good ratings on each apartment. The apartments that were not up to his standards he left bad ratings. Finally, William has found an apartment that he likes and sends the price he is willing to pay for the apartment.

The owner Xavier Joiner who posted the listing later that day receives a request from William Peterson sharing that he is interested in one of his properties and would like to make an offer. Xavier decides to accept his offer. William receives that his offer has been accepted. The apartment is now booked, and William can delete the listing.

Nouns:

- Student/Owner (User)
- Northeastern University (School - will probably want to extract this as an entity later on so it can be edited for multiple students/renters)
- Housing (dorm or apartment) (using unitType instead)
- semester
- year
- budget
- *availability
- Listing
- information
- date
- size (sqft.)
- unitType (dorm or apartment)
- location
- ownerName
- price

- ratings
- offer

Verbs:

- Enrolled → Create user (student/owner)
- Looking up
- Signing up
- Filter
- Selects
- Leaves → Rates
- Sends → Messages
- Posted
- Receives
- Accept
- Booked
- Delete

Redis Data Structures and Definitions:

- Implement a ranking list for listings based on highest rating ranking with Redis.
 - Hash with key: listing:\${listingID}:{authorID} and value: listing information
 - Sorted set with key: rankedListings and value: listing:\${listingID}:{authorID} scored by average rating
- Implement a filtering system for rendering available and unavailable listings separately to the browser when called for both owners and students with Redis.
 - List of available units with key “availableListings” and value “listingID”
 - List of unavailable units with key “unavailableListings” and value “listingID”
 - Hashes with key “listing:\${listingID}” and value of an array with all relevant listing information (e.g., “listingID”: `\${listingID}`, “title”: `\${title}`, etc...)