

URL to front-end pages: <http://flip1.engr.oregonstate.edu:5276/>

Front and Backend Code: <https://github.com/JackoBright/Nest-Quest/>

Reviews:

Review #10:

- *Do the implemented CRUD steps function as the team expects?* If the team stated that a CRUD step worked but you found an error, please let them know.

Yes, I was able to perform CRUD operations such as adding users, deleting, and searching.

- *Would a user easily be able to use the UI to complete the step?* If not, or you have suggestions for how the UI can be improved, then please detail the suggestions.

Yes, the UI seems to offer all the options necessary to complete the operations such as deleting and adding users.

- *What suggestions do you have for the team in any areas where they are blocked or are having difficulty?* Detailed and helpful feedback will receive full credit.

I would say the UI could be improved. There is a lot of text and not much color which could make it harder for users to navigate and find what they are looking for. Some CSS styling could help a lot with this.

Review #11:

All of the CRUD steps seem to work in the way the team outlined.

The UI seems fine to me. There are little aesthetic aspects about it that I think would be good to change. For example, the delete and update buttons aren't really lined up with each other. I'm also not sure why the 'error message for search' and insert is there, I'd probably remove that until it becomes relevant in terms of functionality.

The way you guys have the update feature set up seems fine to me. One thing that I don't like about it is that you can hit the Update multiple times and have multiple queries show up to edit a specific user. I would somehow restrict to show one query at a time. Also (if time allotted), if it's simple to do, maybe you could find a way where the page immediately moves down when the Update query shows up so the user knows where to fill in immediately. That being said, I'm not sure what the expectation is in terms of aesthetics so that might be unnecessary.

Review #12:

- *Do the implemented CRUD steps function as the team expects?* If the team stated that a CRUD step worked but you found an error, please let them know.

All the CRUD functionality works as expected.

- *Would a user easily be able to use the UI to complete the step?* If not, or you have suggestions for how the UI can be improved, then please detail the suggestions.

The UI is intuitive and should allow for easy navigation and performing any CRUD actions without any issues.

- *What suggestions do you have for the team in any areas where they are blocked or are having difficulty?* Detailed and helpful feedback will receive full credit.

Similar to the update panel, I'd make the add user panel dynamic, meaning it only appears when an 'add user' button is pressed, and is hidden otherwise. Other than that, I think the design looks good and I don't have any other suggestions. Maybe also adding a cancel functionality would be nice, so a user can cancel out of adding or updating a user.

Review #13:

- ***Do the implemented CRUD steps function as the team expects?***
- *All of the CRUB steps appear to work for the Users entity.*
- ***Would a user easily be able to use the UI to complete the step?***
- *The UI is fairly easy to understand, could maybe be improved by compacting some of the forms or maybe making the UI less vertical.*
- ***What suggestions do you have for the team in any areas where they are blocked or are having difficulty?***
- *I think it would be nice to make the menu horizontal instead of vertical to separate it visually and save vertical space for page content. I also like the idea of having the table and search bar (if applicable) at the top of the page and forms below, but this is all just preference. For the update, maybe you could use something like the <dialog> HTML tag to make it a pop up form?*

Review #14

Do the implemented CRUD steps function as the team expects? If the team stated that a CRUD step worked but you found an error, please let them know.

- *CRUD steps all look good, adding and removing, searching and the like*
- *Would a user easily be able to use the UI to complete the step? If not, or you have suggestions for how the UI can be improved, then please detail the suggestions.*
 - *you killed it, UI is easy to use and intuitive to learn*
- *What suggestions do you have for the team in any areas where they are blocked or are having difficulty?*
 - *I think overall youve done a great job so far, you could probably figure out how to have the page scroll down when you click update but also as mentioned above I'm sure how important aesthetics are, keep it up!*

Revision Plan:

1. Implemented a modal for the update functionality
2. Add styling to the website

Upgrades to the Draft version

1. Added an update modal
2. Added styling

Revised Project Design:

Title: NestQuest

Team Members: Sankalp Patil and Michael Molineaux

Overview:

Our project is to develop a web application that allows students at Oregon State University to find roommates and places to live. It also allows landlords to post information about their properties to attract tenants. Our database will be a relational database tasked with storing and listing information about vacant rooms in Corvallis, Oregon. This application is intended for students at Oregon State University. There are approximately 40,000 students, most of which live in apartments or townhomes off campus. There are over 400 apartments, townhomes, houses, and other properties where students reside off campus. We estimate that the average size of apartments will be about 1200, with the majority of apartments being good quality. As such, our database needs to have the storage capacity and flexibility to accommodate a large variety of living situations. Users need to be able to insert, delete, and modify certain information from this database given their current living situation and goals. This database will contain information such as names, addresses, ratings, and preferences - such as non-smoking, pets, and noise levels. Additionally, people will be classified into two categories. Owners are people who currently reside in a Property and are looking for roommates. Seekers are people who are looking for a place to live.

Database Outline:

Primary keys have a star*

Foreign keys have a carat^

❖ **Users:** Holds the data of anyone who uses the application

- user_id*^: int, auto_increment
- fname: varchar
- lname: varchar
- email: varchar
- phone: varchar
- smoking: varchar
- pets: varchar
- gender: varchar
- age: varchar

➤ Relationships:

- A 1: 0 or M relationship between Users and Reviews (A User can write many Reviews).
- A 0 or 1 relationship between Users and Tenants (If a User exists, they can either be a Tenant or not a Tenant.)
- A 0 or 1 relationship between Users and Seeker (If a User exists, they can either be a Seeker or not a Seeker.)

❖ **Tenants:** Holds the data of people residing in a Property who are looking for roommates

- user_id*^: int
- role: varchar

➤ Relationships:

- A 1: 1 or M relationship between Tenants and Tenant Directory (Many Tenants can be living in many Properties).
- A 1:1 relationship between Tenants and Users (If a Tenant exists, they must be a User)

❖ **Seekers:** Holds data of people currently looking for roommates and a place to live.

- user_id*^: int, auto_increment
- price_upper: int
- price_lower: int

➤ Relationships:

- A 1: 0 or M relationship between Seekers and Information request (A Seeker can send out multiple Information requests).

- A 1:1 relationship between Seekers and Users (If a Seeker exists, they must be a User)

❖ **Information Requests:** Hold data about requests for Information about Properties sent by Seekers

- user_id^: int, not NULL, unique
- date_contacted: DATE, not NULL
- address*: varchar, not NULL
- Relationships:
 - A 1:1 relationship to Seeker(One Information request must have one Seeker)
 - A 1:1 relationship to Property(One Information request can be about only one Property)

❖ **Properties:** The Properties that Tenants are living at where openings are available

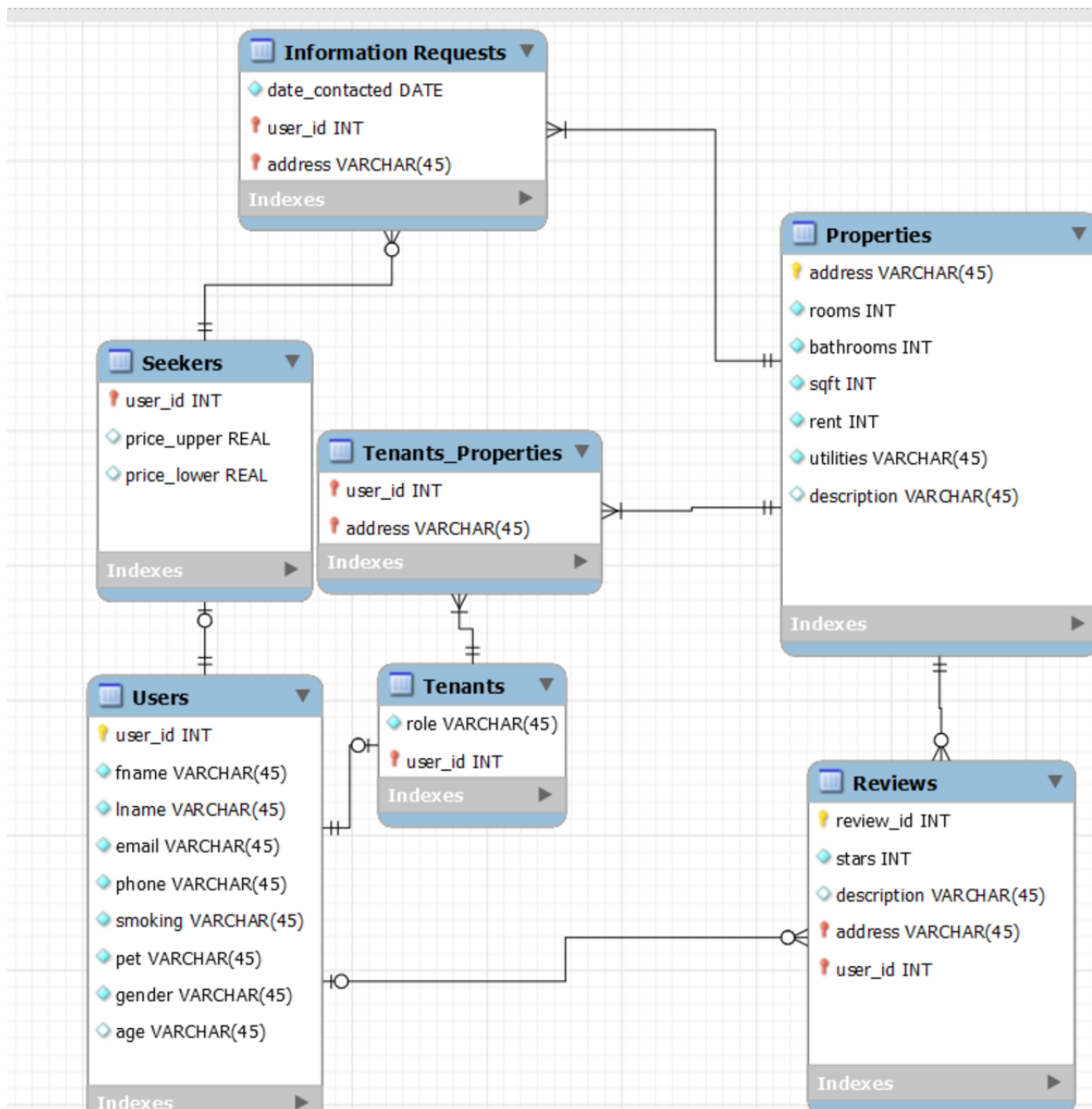
- address*: varchar, not NULL, unique
- rooms: int, not NULL
- bathrooms: int, not NULL
- sq_ft: varchar, not NULL
- rent: varchar, not NULL
- utilities: varchar, not NULL
- description: varchar
- Relationships:
 - A 1: 0 or M relationship to Information Request (One Property can have multiple Information requests)
 - A 1: 0 or M relationship to Tenant Directory (One Property can house multiple Tenants)
 - A 1: 0 or M relationship to Reviews (One Property can have none or multiple Reviews)

❖ **Reviews:** These are Reviews written by Users of the app about Properties

- review_id*: int, auto_increment, not NULL
- user_id^: int, not NULL
- address^: varchar, not NULL
- stars: int, not NULL
- description: varchar
- Relationships:
 - A 1:1 relationship with Users (Any given review can only be written by one User)
 - A 1:1 relationship to Properties (A review can only be related to one Property)

- ❖ **Tenants_Properties:** This is an intersection table between Tenants and Properties
 - user_id*^ int, not NULL (Note, the pk is the combination of the two)
 - address*^ varchar
 - Relationships:
 - A 1:1 relationship to Tenants
 - A 1:1 relationship to Properties

Entity Relationship Diagram:



Sample Data

Users								
user_id	fname	lname	email	phone	smoking	pet	gender	age
1	Rob	Michaels	rob@gmail.com	1234567890	yes	no	male	22
2	Amy	Stevenson	amy@outlook.com	987654321	no	no	female	20
3	Jim	Davis	jim@oregonstate.edu	5413214567	no	yes	male	25
4	Jordan	Brooks	jordan@gmail.com	3219876540	yes	yes	female	20
5	Bill	Bobert	bill@oregonstste.edu	4436758766	no	no	male	24
6	Jace	Smith	jace@yahoo.com	4336751369	no	yes	male	21

Tenants	
user_id	role
4	owner
5	tenant
6	tenant

Seekers		
user_id	price_upper	price_lower
1	750	0
2	950	300
3	600	0

Properties						
address	rooms	bathrooms	sqft	rent	utilities	description
12345 Berry st	3	3	1550	750	water, sewage, garbage, washer/dryer	Townhome 2.2 miles from campus
23465 west ave	2	2	1450	850	water, sewage, garbage, washer/dryer	Apartment room located 5 minutes from campus
876 robin blvd	4	3	1650	600	electricity, water, sewage, garbage, washer/dryer	nice place
23819 bald mtn rd	4	4	1800	650	electricity, water, sewage, garbage, washer/dryer	House from the 80s but in great condition.

Information Request		
date_contacted	user_id	address
12-11-2022	1	12345 Berry st
09-28-2022	2	12345 Berry st
02-24-2023	3	23819 bald mtn rd

Tenant_Properties
user_id address
4 12345 Berry st
4 23465 west ave
5 876 robin blvd
6 23819 bald mtn rd

Reviews				
review_id	user_id	address	stars	description
1	1	23819 bald mtn rd	4	great place! The location is nice
2	2	876 robin blvd	4	Included electricity is a huge plus.
3	2	23465 west ave	3	The rooms have a weird smell and the carpets are discolored