**IT2002 Project Report**

**Project members**

Ng Tze Kean A0269333Y

Chen Yao Quan A0238895H

Lee Wen Yang A0233784X

Lian Kah Seng A0233238J

What is our project about?

Our project aims to build a web application that acts as an online marketplace for users to apply for a short homestay, as well as to place their house up on listings for other users to apply.

We used Flask framework to create the web application. We kept track of the sessions that our users are in session manager.

Entity-Relationship Diagram

Our Entity Relationship Diagram consists of the entities Users, Houses, Credit Cards and Rental.

Our Users entity has the attributes email, first name, last name, age and password. Email is the primary key in this case.

Our Houses entity has the attributes ID, email, price, number of rooms, owner and location. ID is the primary key and email references the Users entity’s email.

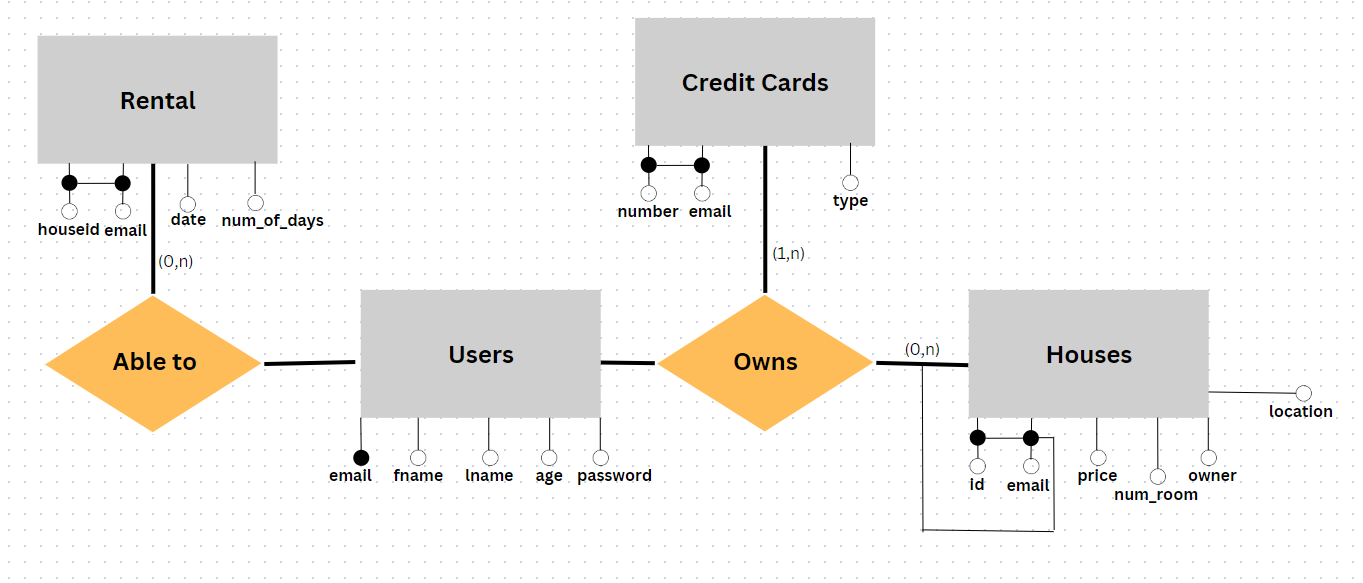
Our Credit Cards entity has attributes number, email and type. The primary key is the number, with email referencing the Users entity’s email.

Our Rental entity has attributes house ID, email, date and number of days. The house ID and email references Houses entity’s ID and Users entity’s email respectively.

Users can own from 0 to many houses with a cardinality of (0,n), as users may simply be on the web application to rent a house and not to put their house on listings.

Users can own from 0 to as many credit cards as he or she wants, with a participation of (0,n), as no credit card is needed for viewing the apartments. A User only has to update his or her credit card only when the user wants to rent an apartment

Users are able to rent from 0 to as many houses with a cardinality of (0,n). It is not mandatory as not all users register to rent an apartment.



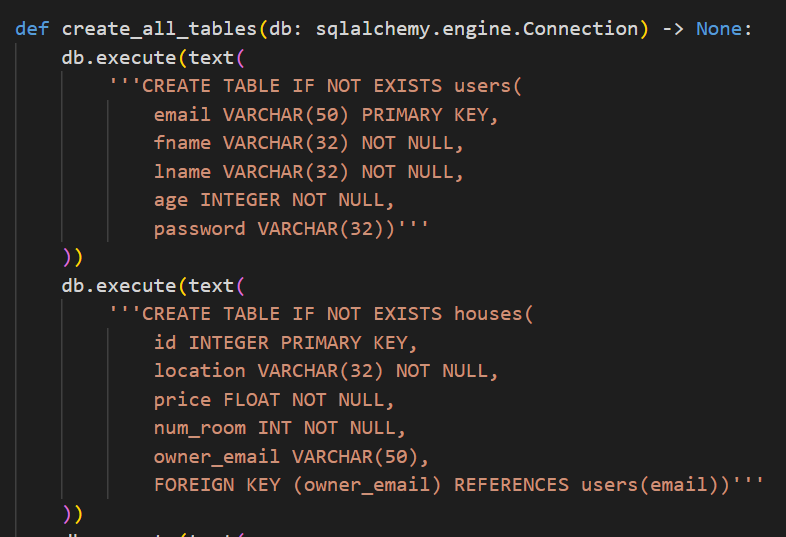
SQL Data Definition Language Code

SCHEMA

Our Schema consists of users, houses, credit cards, rental and house ratings.

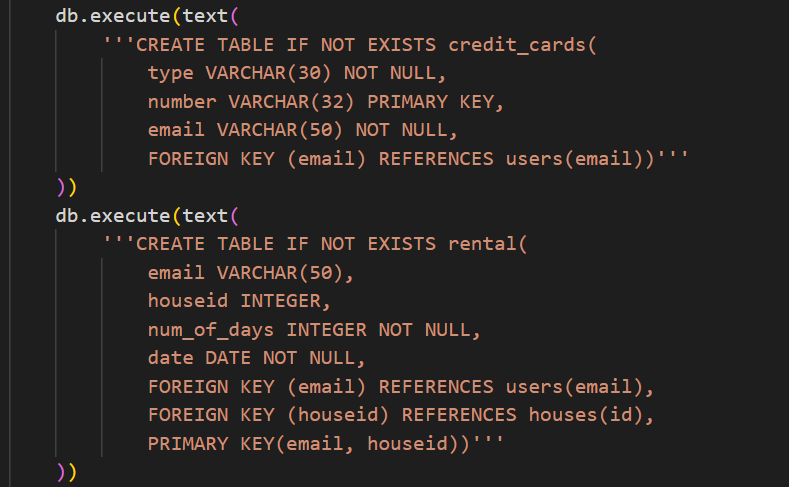
For our users table, we set the primary key as email as everyone’s email is unique. We set the first name, last name and age to be not null as these are information that is crucial and should not be blank.

For our houses table, we set the id of the user to be the primary key as everyone’s id is unique. We set the location, price, number of rooms as not null as these are essential information for people who want to rent an apartment. It references the table users, where table.email == users.email.



For our credit card table, we set the credit card number as the primary key as every credit card number is unique. The type and email are not null as these are important information of the user and card. It references users where credit\_card.email == users.email.

For our rental table, we set the primary keys to be email and houseid as both are unique. They are the foreign keys that references users and houses where rental.email == users.email and house.id = rental.houseid. The start date and number of days intended for rental is also set to be not null as they are important information to be filled in for the owner to know the time period of the rental. This ensures that the system does not allow a user to rent an apartment that has been booked.



SQL Queries

