Meeting 3 - 21/03/2021

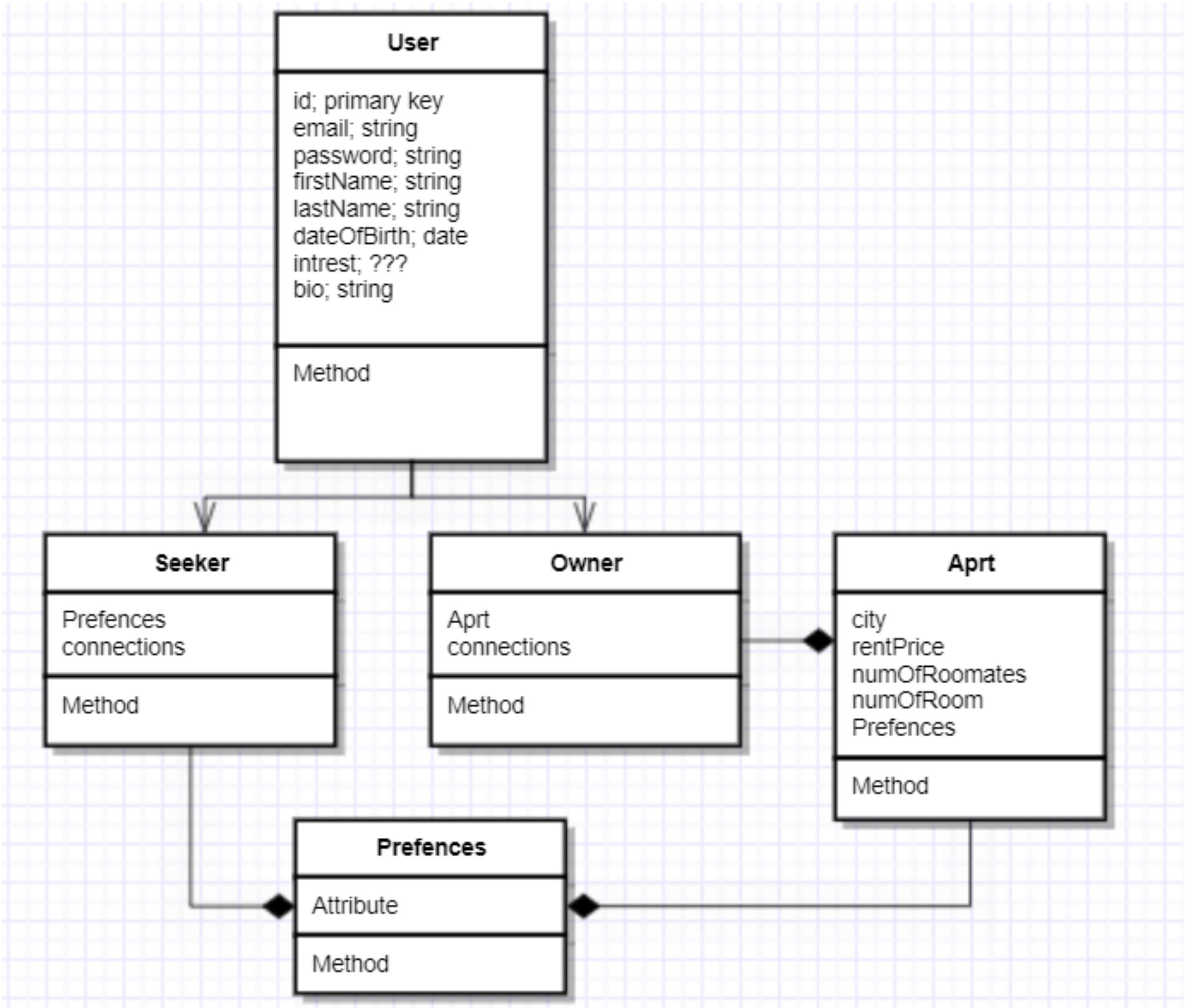
A short summary of the things we did today:

- Divided work on the model section (work division table is below)
- Came up with a basic idea of how the data model should look.
  - A full data model (data classes) to be completed by Tamir and Micha
- Came up with a basic idea of how the logic in the model should look.
  - General functions, what each function does, etc.

Work Division

Teammate	Work	Comment
Tamir	Data model	Finish UML, send to group so we can begin working on backend.
Amit	`search` + `orderApartments`	
Daniel	`like`	Finish UML, send to group so we can begin working on backend.
Micha	Data model	
Nadav	`getMatchingScore`	

Model - Data



Note: this is a basic idea, final version to be completed by Tamir and Micha

Model - Logic

The following are the logic classes that we will work on.

Engine

```
def like(seeker, owner):
    # pull seeker and owner from the database
    # adds owner to seeker's liked list
    # adds seeker to owner's liked list
    # updates the owner that someone liked him
    # write to database?

def orderApartments(seeker):
    # pulls seeker from the database
    # pull all apartments from the database
    # filter through all the necessary data (ApartmentPerferences + price)
    # for each apartment.owner:
        # getMatchingScore(seeker, owner)
    # order by matching score
    # return list

def getMatcingScore(seeker, owner):
    # give score based on tags through the use of weights maybe

def search(apartment_preferences):
    # get apartment preferences from controller
    # create dummy apartment_preferences object
    # use it to search in database based on preferences.
    # return list
```

Note: we may divide the engine into two - MatchingEngine and SearchEngine