Project Proposal - Team 0503-08

**Brand Name:** *Studomes* - Homes for Students

**Mission Statement :**

* We provide analytical solutions to help rental housing websites to serve their customers by providing them real-time customised rental options.
* To create a user friendly analytical interface which provides rental options to customers that enable them to make better housing decisions.
* Enables owners to list their properties and helps client to manage the data.

**Mission Objectives:**

* Establish a relational database model that helps gather, store and manage the data from owners and customers.
* Provide insights to our client, from the organized data by using interactive analytics.
* Support property agents, owners and potential home buyers or tenants with superior market standard tools to help provide a better estimate of various quantifiable measures such as pricing, for the owners’ listings as well as customers’ preferences.
* Enable the client to keep track of each customer’s account, including lease agreement, application details.
* Enable the customers to go through a hassle free application process for the lease and managing their accounts.

**Business processes/transactions:**

* The owners of a property can list their property. Each property has at least one owner and all property owners in the database own at least one property. Each Property is identified by a unique identifier, a name, address of the property. Each Property should mention the square feet of carpet area, the number of bedrooms and the number of bathrooms.
* Each property is of exactly one type of house and there may be property types for which there are no listings, one listing or many listings.
* Each property can include 0 or more amenities and there can be amenities in the database that none of the listings have. Each amenity is
* Each interested customer submits an application for each property of interest
* One lease contract is signed between each customer and the owner for each property
* An agent facilitates the owner in managing the transactions related to one or more properties of an owner. Each agent has a unique identifier, their contact number and email.

**ER Schema:**

**Entities:**

* Property(**pID**, pName, pAddress, pCity, pState, pPostalCode, pRent, pBedroom, pBathroom, pAreaSqFt)
* Customer(**cID**, cName, cContact, cEmail)
* Type(**tID**, tCategory)
* Owner(**oID**, oName, oOrganization, oContact, oEmail, oWebsite)
* Agent(**agID**, agName, agContact, agEmail)
* Amenity(**amyID**, amyName)

**Relationships:**

* Apply (appStartDate, appEndDate): binary relationship

1 or more customer can apply for 1 or more properties

1 property can be applied by 1 or more customers

* Belong : binary relationship

1 type can have under it 0 or many properties

1 property can belong to only 1 type

* Own: binary relationship

1 owner can own 1 or many properties

1 property can be owned by only 1 owner

* Include: binary relationship

1 property can have 0 or many amenities

1 amenity can be present in 0 or many properties

* Facilitate: binary relationship

1 property to 0 or 1 agent

1 agent to 0 or more properties

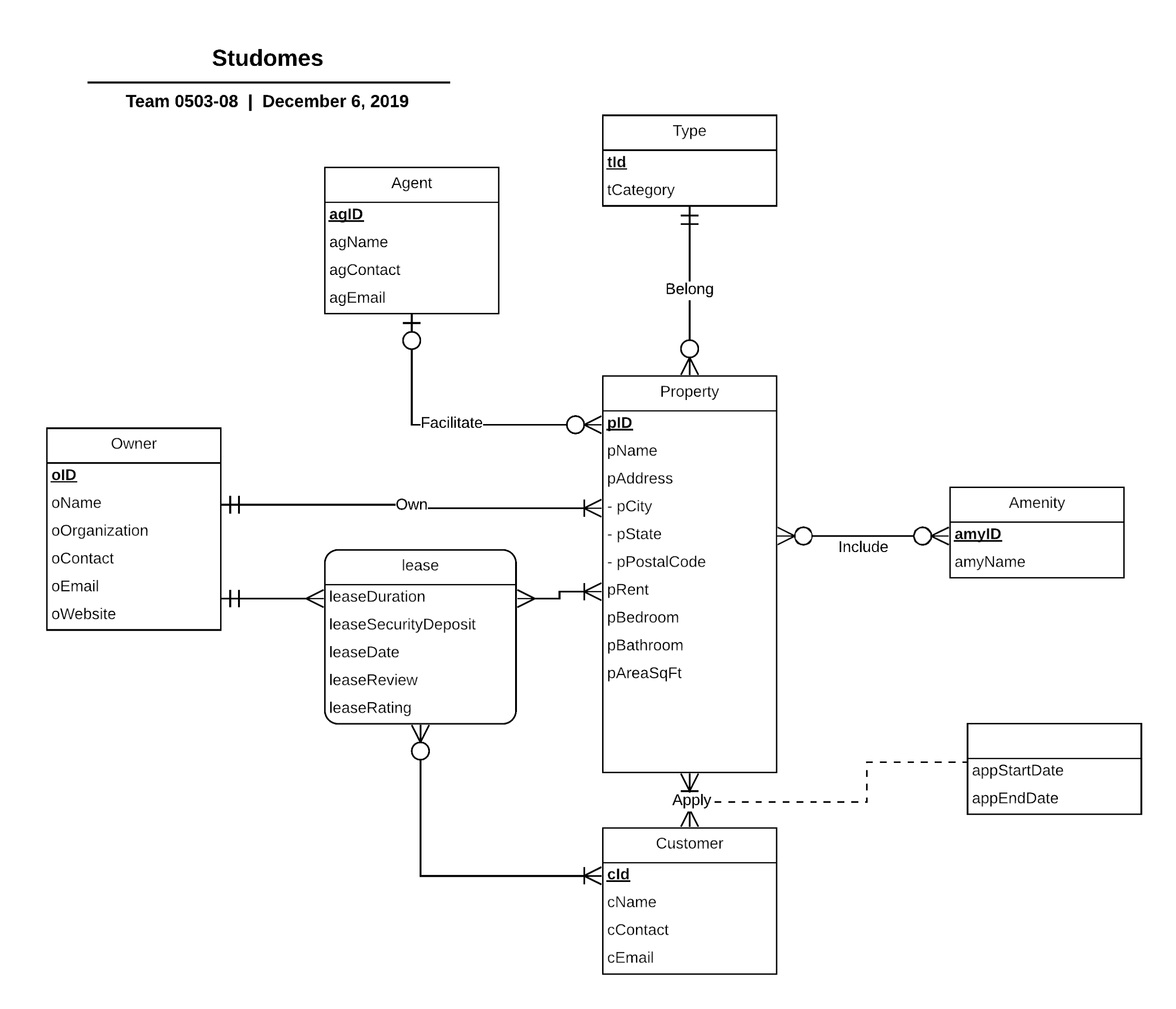
* Lease (leaseDuration, leaseSecurityDeposit, leaseDate, leaseReview): ternary relationship

1 property can be leased to 1 or more customers by 1 owner

1 owner can lease 1 or more properties to 1 or more customers

1 customer can get the lease by 1 owner for 1 or properties

**ER Diagram:**



**Relations:**

* Property(**pID**, pName, pAddress, pCity, pState, pPostalCode, pRent, pNumberOfBedroom,

pNumberOfBathroom, pAreaSqFt, *oID, tID, agID*)

* Customer(**cID**, cName, cContact, cEmail)
* Type(**tID**, tCategory)
* Owner(**oID**, oName, oOrganization, oContact, oEmail, oWebsite)
* Agent(**agID**, agName, agContact, agEmail)
* Amenity(**amyID**, amyName)
* Lease(***pID, cID,*** *oID*, leaseDuration, leaseSecurityDeposit, leaseDate, leaseReview, leaseRating)
* Include (***pID, amyID***)
* Apply (***pID, cID,*** appStartDate, appEndDate)

**Functional Dependency:**

* pID —> pName, pAddress, pCity, pState, pRent, pPostal Code, pNumberOfBedroom, pNumberOfBathroom, pAreaSqFt
* cID —> cName, cContact, cEmail
* tID —> tCategory
* oID —> oName, oOrganization, oContact, oEmail, oWebsite
* agID —> agName, agContact, agEmail
* amyID—> amyName
* pID, cID, oID —> leaseDuration, leaseSecurityDeposit, leaseDate, leaseReview, leaseRating
* pID, amyID —>
* pID, cID —> appStartDate, appEndDate

**Normalization**:

* Property(**pID**, pName, pAddress, pCity, pState, pPostalCode, pRent, pNumberOfBedroom,

pNumberOfBathroom, pAreaSqFt, *oID, tID, agID*) = 3NF

* Customer(**cID**, cName, cContact, cEmail) = 3NF
* Type(**tID**, tCategory) = 3NF
* Owner(**oID**, oName, oOrganization, oContact, oEmail, oWebsite) = 3NF
* Agent(**agID**, agName, agContact, agEmail) = 3NF
* Amenity(**amyID**, amyName) = 3NF
* Lease(***pID, cID,*** *oID*, leaseDuration, leaseSecurityDeposit, leaseDate, leaseReview, leaseRating) = 3NF
* Include (***pID, amyID***) = 3NF
* Apply (***pID, cID,*** appStartDate, appEndDate) = 3NF

**Business Rules:**

1. When a property owner is removed from the database, all properties listed by the particular owner should be removed from the database.
2. When the property owner’s information is updated, the property’s information must be modified.
3. When an agent is facilitating a property, and the agent is removed from managing the property, then the property’s agent information should be removed and set to null.
4. When an agent is facilitating a property, and there is a change in the agent information, then the property’s agent information should be updated.
5. When an apartment belongs to a particular type, if the information on that type is modified, the property’s type information must not be modified in the database.
6. When an apartment belongs to a particular type, if the type’s information is deleted, the property’s type information must not be deleted from the database.
7. When an amenity is deleted from a property, the amenity inclusion information should also be deleted from the property’s information in the database.
8. When an amenity is updated in a property, the corresponding inclusion information should also be changed accordingly.
9. When a property is updated or deleted from the database, all the applications for that property are either deleted or updated accordingly.
10. When the customer data is updated or deleted in the property application, the information is modified or deleted in the property’s information accordingly, in the database.
11. A customer cannot be deleted from the database when the customer has signed a lease for a property with the property owner. The same principle applies when a property or an owner is being deleted.

**Referential Integrities:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **FK Relation** | **Foreign Key** | **PK Relation** | **Primary Key** | **Business Rule** | **Constraint:**  **ON DELETE** | **Business Rule** | **Constraint:**  **ON UPDATE** |
| Property | oID | Owner | oID | R1 | Cascade | R2 | Cascade |
| Property | agID | Agent | agID | R3 | Set Null | R4 | Cascade |
| Property | tID | Type | tID | R5 | No Action | R6 | No Action |
| Include | pID | Property | pID | R7 | Cascade | R8 | Cascade |
| Include | amyID | Amenity | amyID | R7 | Cascade | R8 | Cascade |
| Apply | pID | Property | pID | R9 | Cascade | R9 | Cascade |
| Apply | cID | Customer | cID | R10 | Cascade | R10 | Cascade |
| Lease | pID | Property | pID | R11 | No Action | R11 | No Action |
| Lease | oID | Owner | oID | R11 | No Action | R11 | No Action |
| Lease | cID | Customer | cID | R11 | No Action | R11 | No Action |

**Sample Data:**

Property ('1000553','Parkside Apartments','8125 48th Avenue APT 111','College Park','Maryland','20740',2000,2,1,850,410002,11200002,1120003);

Customer('2133430', 'Jennifer Golbeck', '6463431543', 'jennifer.golbeck@gmail.com');

Type(11200002,'Condo');

Owner('410002','Simon Radcliffe', 'barkan management','5462075021', 'simon.radcliffe@gmail.com', 'www.spectrumam.com');

Agent('1120003','Robert Samuelson', '8932076067', 'robert.samuelson@gmail.com');

Amenity('102','Personal Garden');

Lease('1000553','2133430','410002',12,500,'01/03/2019','Good',4);

Include(‘1000553’, ‘102’);

Apply('1000553','2133430', '04/02/2019','05/03/2019');

Property: Parkside Apartments is the property name whose unique ID, address, price and characteristics are entered in the property relation.

Customer: Rohan is a customer whose unique ID, contact and email details are entered in the Customer relation.

Type: The type of the Parkside Apartments, which is a condominium, is entered along with a unique ID and the furnish type, in the Type relation.

Owner: The owner of the Apartment No.315 at Parkside Apartments, is Ron and his details along with a unique ID is entered in the Owner relation.

Agent: ABC Real Estates are the agency which is facilitating Ron to find customers such as Rohan. The unique ID, name and contact details of the agency is entered in the Agency relation.

Amenity: Parkside Apartments have a recreation room which is an amenity. These are entered in the Amenity relation which also has a unique ID.

Lease: The details of the lease contract between Ron and Rohan for the Apartment No.315 at Parkside Apartments are entered in the Lease relation which essentially contains the three unique IDs for Ron, Rohan and the Apartment No. 315 along with the lease duration, date and security deposit.

Own: The Own relation is between the Parkside Apartments No. 315 and Ron which contains their unique IDs. These are entered so that we have records of the owners and their houses.

Facilitate: The Facilitate relation is used to have records of the agency “ABC Real Estates”, the owner “Ron” who has hired the agency and the customer “Rohan”. It contains their respective unique IDs.

Belong: The Belong relation captures which house is of what type. Here, Parkside Apartments is a condominium. Thus, their unique IDs are entered in this relation.

Include: This relation tells us about the amenities offered with the property. Here, Parkside Apartments which has a recreation room is recorded using their unique IDs.

Apply: The apply relation records the application submitted by the customer “Rohan” for the property at Parkside Apartments using their respective unique IDs along with the start date and end date of Rohan’s application.