**Database Systems Project Part 1**

**Team Information**

**Team Members**: Arya Kotibhaskar, Aruzhan Bolatova

**Project Title**: NYU Student Roommate and Rental Finder- Hive Buddy

**Entities and Their Relationships in Hive Buddy ERD**

The ERD for the Hive Buddy project visualizes these relationships, illustrating how key entities like Student, Group, and Rental Listing interact through intermediary tables such as Rent Request and Student Rental Match. The design ensures that students can match with compatible roommates and rental options, while rental owners can manage their listings and lease agreements efficiently. This structured approach supports the app's comprehensive features for simplifying the rental and roommate search process.

**Explanation of Entities and Their Relationships in Hive Buddy ERD**

**1. Student**

* **Attributes**:
  + Student\_ID: A unique identifier for each student.
  + First\_Name: The student's first name.
  + Last\_Name: The student's last name.
  + Gender: The gender of the student.
  + Age: The age of the student.
  + Phone\_number: The contact number of the student.
  + Email\_ID: The student's email address.
  + Course: The academic course the student is enrolled in.
* **Relationships**:
  + **Has Preferences**: Each student is associated with a set of roommate and rental preferences.
  + **Creates Groups**: Students can create or join groups for collective roommate searches.
  + **Matches with Listings**: Students are linked to rental listings through the Student Rental Match entity.

**2. Roommate and Rental Preferences**

* **Attributes**:
  + Student\_ID: Foreign key linking preferences to the corresponding student.
  + Budget\_Range: The financial range a student is comfortable with for rent.
  + Roommates Gender Preference: Preference regarding the gender of potential roommates.
  + Smoking Preference: Indicates if the student is okay with a roommate who smokes.
  + Pet Preference: States whether the student is willing to live with pets.
  + Food Habits (Veg/Non-veg): The dietary preferences of the student.
  + Lease-Term Preference: Preferred length of the lease term.
  + Neighborhood Preference: Areas or neighborhoods the student prefers to live in.
* **Relationships**:
  + **Belongs to Student**: Each set of preferences is uniquely tied to one student.

**3. Group**

* **Attributes**:
  + Group\_ID: Unique identifier for each group.
  + Group\_Name: Name of the group.
  + Number of Rooms: The number of rooms the group is looking for.
  + Number of Flatmates: The number of people in the group.
  + Availability Status: Indicates if the group is active and searching or settled.
* **Relationships**:
  + **Created by Student**: A student can form or be part of multiple groups, creating a one-to-many relationship.
  + **Matches Rental Listings**: Groups interact with rental listings through the Student Rental Match and Rent Request entities.
  + **Sends Rent Requests**: Groups can send requests to rental owners for specific properties.

**4. Rental Listing**

* **Attributes**:
  + Listing\_ID: Unique identifier for each rental property.
  + Number of Rooms: The total rooms available in the rental property.
  + Neighborhood: The neighborhood where the property is located.
  + Monthly Rent: The rental price per month.
  + Lease Term: The duration of the lease.
  + Distance from NYU: Distance from NYU campuses.
  + Amenities: A list of amenities available at the property (e.g., Wi-Fi, gym, laundry).
* **Relationships**:
  + **Owned by Rental Owner**: Each rental listing is managed by one rental owner.
  + **Matched with Groups**: The listing is linked to groups through rental matches and lease agreements.
  + **Leased via Lease Agreement**: Once a group’s request is accepted, the listing becomes part of a lease agreement.

**5. Rental Owner**

* **Attributes**:
  + Owner\_ID: Unique identifier for each rental owner.
  + Name: The name of the property owner.
  + Active Listings: A count or list of currently available properties.
* **Relationships**:
  + **Owns Listings**: A one-to-many relationship exists between rental owners and their properties.
  + **Receives Rent Requests**: Owners interact with incoming rental requests from groups.
  + **Accepts or Rejects Requests**: Rental owners have the authority to accept or reject requests from groups.

**6. Student Rental Match**

* **Attributes**:
  + Match\_ID: Unique identifier for each match instance.
  + Group\_ID: Links the match to a specific group.
  + Listing\_ID: Links the match to a specific rental listing.
  + Match Status: Indicates the status of the match (e.g., pending, accepted, rejected).
* **Relationships**:
  + **Matches Groups with Listings**: This acts as an intermediary table to create a many-to-many relationship between groups and rental listings.
  + **Links to Lease Agreement**: Accepted matches can progress to form a lease agreement.

**7. Lease Agreement**

* **Attributes**:
  + Lease Agreement Number: Unique number for each lease agreement.
  + Date: The date when the agreement is signed.
  + Listing\_ID: Links the lease agreement to a specific rental listing.
  + Group\_ID: Links the lease agreement to a specific group.
  + Lease Agreement Duration: Specifies the length of the lease period.
* **Relationships**:
  + **Signed by Group**: The group that has successfully been matched and accepted for a property signs the lease.
  + **Linked to Rental Owner**: The lease agreement involves the property owner who manages the rental listing.

**8. Rent Request**

* **Attributes**:
  + Request\_ID: Unique identifier for each rental request.
  + Group\_ID: The group making the rental request.
  + Owner\_ID: The rental owner receiving the request.
  + Listing\_ID: The specific rental listing involved in the request.
  + Request Status: Indicates the state of the request (e.g., submitted, under review, accepted, rejected).
  + Request Date: The date the request was made.
* **Relationships**:
  + **Sent by Group to Rental Owner**: Connects the group and rental owner for the rental process.
  + **Linked to Listing**: Establishes which listing the request pertains to.

**ERD Overview**

The Entity-Relationship Diagram (ERD) for Hive Buddy serves as a visual blueprint for the underlying database structure. It defines how data is stored and interrelated to support the functionalities of the platform, which include roommate matching, rental browsing, and lease management. The ERD showcases:

* **One-to-One Relationships**: Such as between Student and Roommate and Rental Preferences.
* **One-to-Many Relationships**: Such as between Rental Owner and Rental Listing, or Student and Group.
* **Many-to-Many Relationships**: Mediated by Student Rental Match and Rent Request, connecting Groups with Rental Listings and Rental Owners.
* **Associative Entities**: Student Rental Match, Lease Agreement, and Rent Request facilitate complex interactions between primary entities.

**Project Features**

Hive Buddy is designed to solve the common challenges faced by students, especially international ones, in finding affordable housing and compatible roommates. The ERD enables this by providing a clear map of how data flows between different functionalities. The integration of tables like Lease Agreement and Rent Request ensures the platform can handle end-to-end processes, from search and match to lease finalization, making it an efficient solution for both students and rental owners.

**Use Case: Finding a Roommate and Renting a Property**

**Title**: Roommate Matching and Rental Request Submission

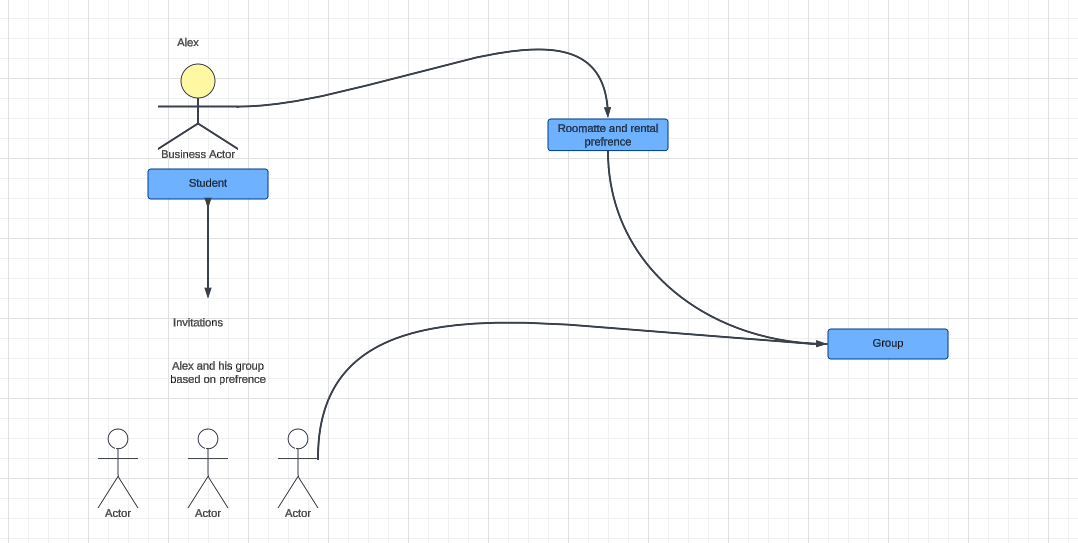
**Actor**: Student (NYU)

**Primary Goal**: To find a compatible roommate, browse rental listings, form a group, and send a rental request.

**Preconditions**:

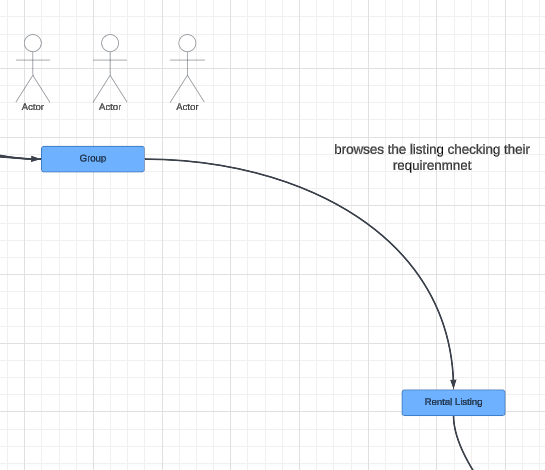
* The student must be registered on Hive Buddy.
* The student's profile, including preferences and budget, must be filled out.

**Main Scenario**:



1. **Suppose a student**, Alex, has just joined NYU and is looking for a roommate and a rental property.
2. Alex **logs into Hive Buddy** and navigates to the profile section.
3. Alex **completes the profile creation**, adding details such as budget range, neighborhood preferences, lifestyle choices (e.g., non-smoker, vegetarian), and roommate preferences.
4. The system **stores Alex’s preferences** in the Roommate and Rental Preferences table linked to Alex’s Student\_ID.
5. Alex then navigates to the **“Find a Roommate” section** and initiates the roommate search.
6. The app’s **matching algorithm** scans through the Student and Roommate and Rental Preferences tables to find compatible roommate matches.
7. Alex receives a list of potential roommates with details such as shared preferences and compatibility scores.
8. Alex reviews the profiles and sends a **group invitation** to two compatible students to form a group.
9. The system **creates a new entry** in the Group table, linking it to Alex and the invited students.
10. The group is formed, and its Group\_ID is assigned, showing as active in the system.

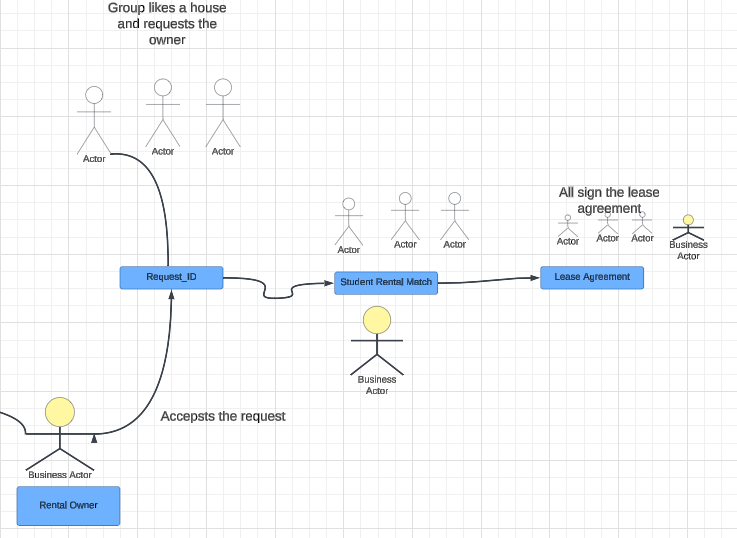
**Alternative Flow**: If Alex does not find compatible roommates, the system suggests updating preferences to widen search parameters or sets a notification for future matches.



**Continuation**: 11. Alex and the group members **browse rental listings** in areas of interest using the search feature.

12. The app queries the Rental Listing table for available properties that match the group’s combined budget and preferences.

13. Alex views detailed property listings, including Monthly Rent, Lease Term, Number of Rooms, Distance from NYU, and Amenities.



14. The group agrees on a listing and **sends a rental request**, which creates an entry in the Rent Request table with Request\_ID, Group\_ID, Owner\_ID, Listing\_ID, and Request Status as "Pending".

15. The property owner receives the rental request notification and reviews the group profile.

16. The owner **accepts the request**, updating the Request Status to "Accepted" and moving the listing to “under lease.”

17. The system prompts Alex and the group to proceed with signing the **lease agreement**.

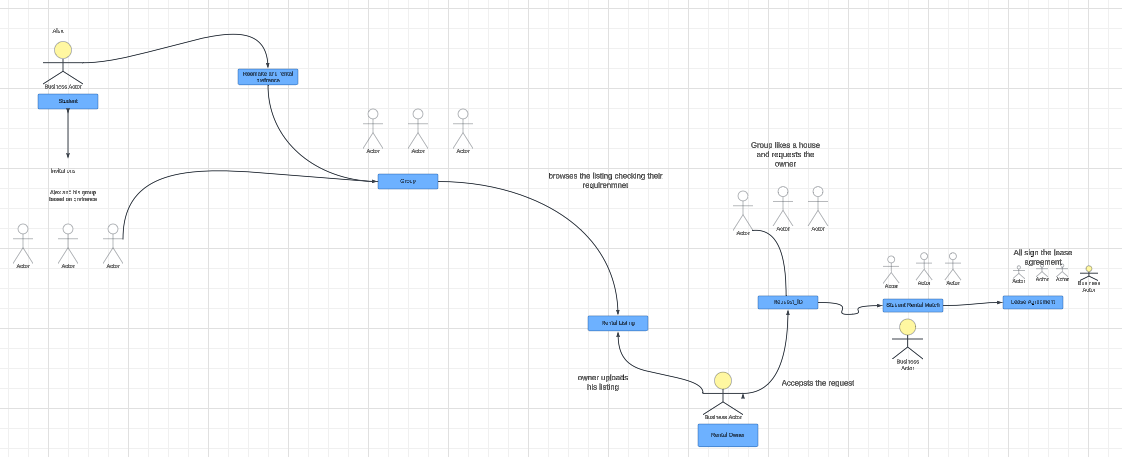
18. The Lease Agreement entry is created, linking Lease Agreement Number, Group\_ID, Listing\_ID, and Lease Agreement Duration to finalize the process. 19. The status is updated to indicate the **lease is active**, and the rental is no longer available in public listings.

**Postconditions**:

* Alex and the group have secured housing.
* The lease agreement is in effect, recorded in the database.

**Exceptions**:

* If the owner rejects the rental request, the system updates the Request Status to "Rejected" and notifies Alex’s group to look for alternative properties.
* If a group member leaves the group before the lease is signed, the Group entry updates the potential requests.



Refer this image from Use case.pdf in the zip file