Team HomeSick

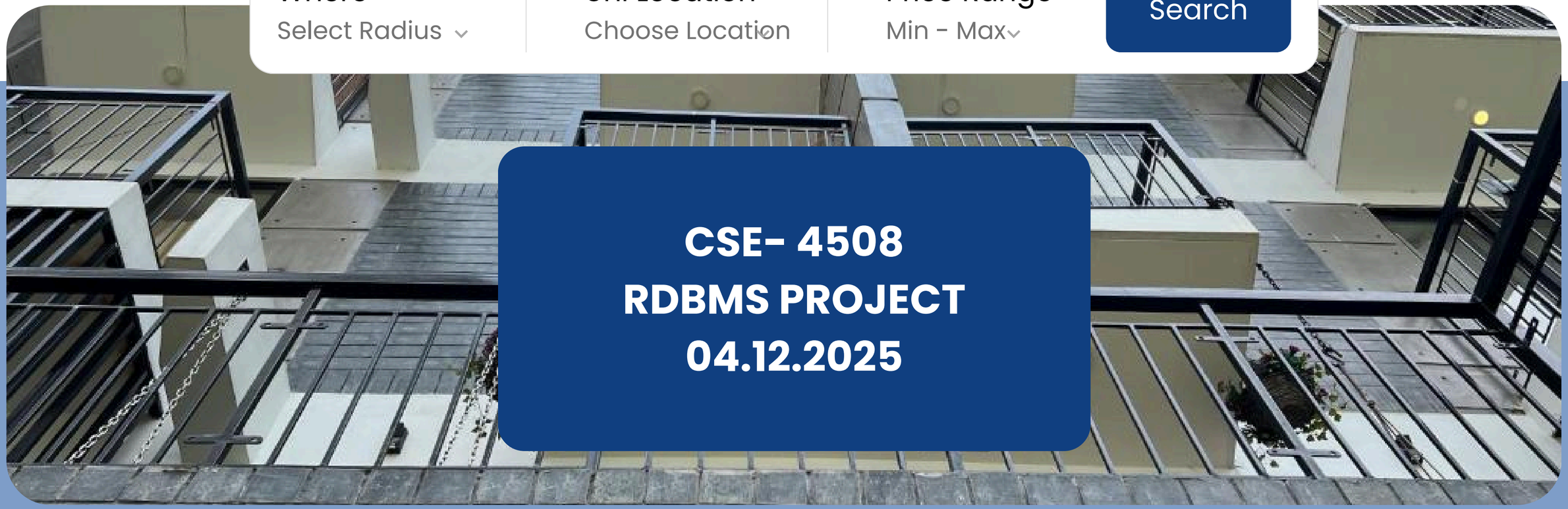
RoomLagbe

Where
Select Radius

Uni Location
Choose Location

Price Range
Min - Max

Search



CSE- 4508
RDBMS PROJECT
04.12.2025

OUR TEAM

Rahatut Tahrim Mounota

ID 220041122

Nayeemul Hasan Prince

ID 220041125

Alfi Shahrin

ID 220041153

PROBLEM DOMAIN

Students who want to live near university campuses face difficulties finding affordable housing and trustworthy roommates. Traditional rental platforms focus primarily on landlords and do not support student-to-student room sharing, leading to:

Lack of verified, trustworthy listings

Lack of saved search or wishlist features

Difficulty estimating commute to campus

No platform for student-to-student room sharing

Inconsistent rent information

OUR SOLUTION

RoomLagbe addresses these challenges by providing:

- **Landlord apartment listings**
- **Student-to-student room sharing**
- **Commute time estimation**
- **Verification badges and Rent fairness score**
- **Secure messaging and visit booking workflow**
- **Wishlist and saved searches**

Connecting students looking for rent, house owners offering rentals & students renting out rooms

ROLE OF THE RELATIONAL DATABASE (RDB)

Central data storage for users, listings, rooms etc

Manages relationships (1:M, M:N)

Ensures transactional safety using ACID properties

Optimized performance

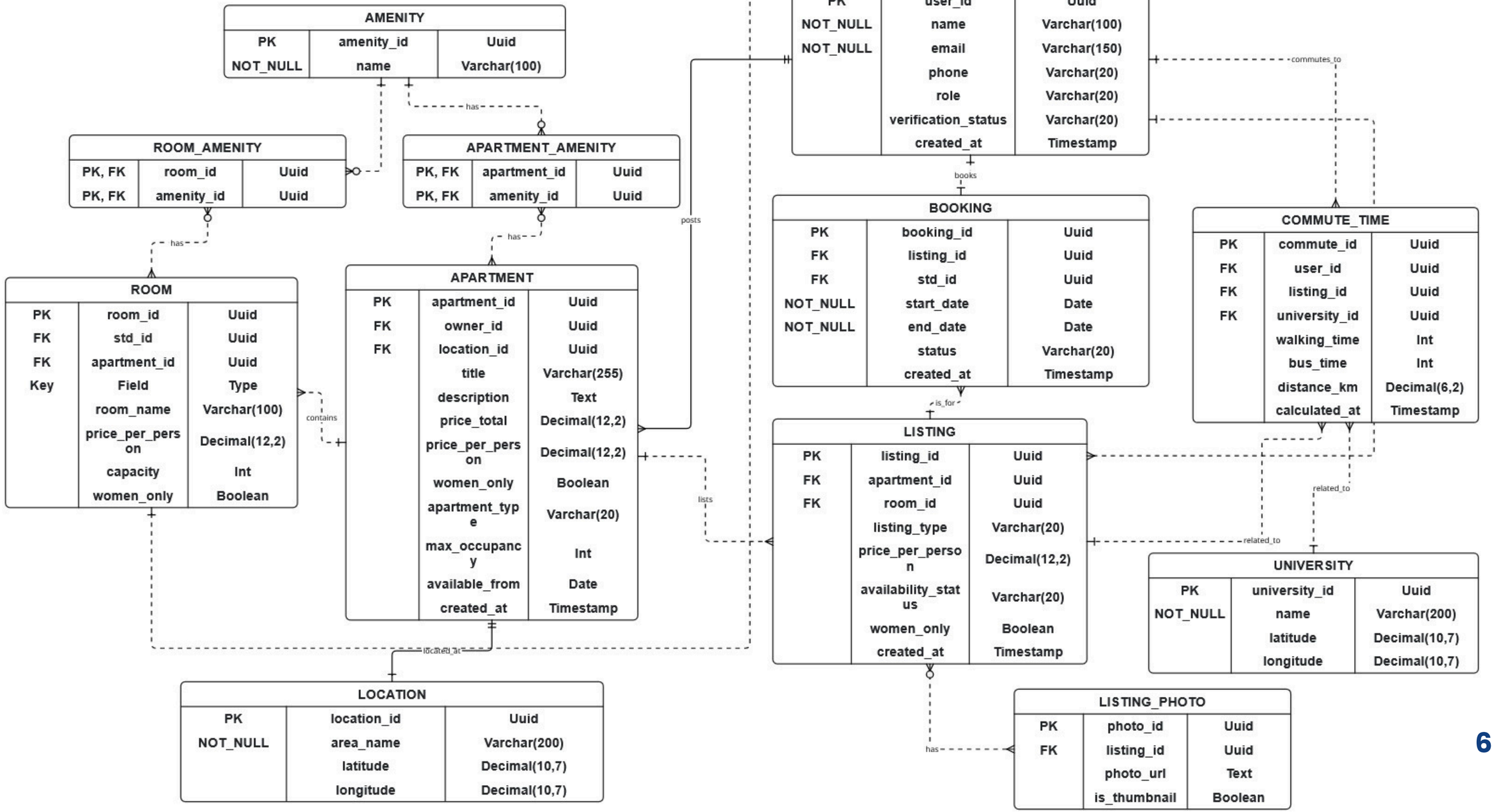
Role-based access & security for admin, owners & students

Implements business logic

Supports complex queries & analytics

Enforces integrity using PKs, FKs, constraints etc

DRAFT ER DIAGRAM



INITIAL SCHEMA OUTLINE

USER	user_id, name, email, phone, role, verification_status, created_at
APARTMENT	apartment_id, owner_id (FK → USER), location_id (FK → LOCATION), title, description, price_total, price_per_person, women_only, apartment_type, max_occupancy, available_from, created_at
ROOM	room_id, apartment_id (FK → APARTMENT), room_name, price_per_person, capacity, women_only
AMENITY	amenity_id, name
ROOM_AMENITY	room_id (FK → ROOM), amenity_id (FK → AMENITY)
APARTMENT_AMENITY	apartment_id (FK → APARTMENT), amenity_id (FK → AMENITY)

INITIAL SCHEMA OUTLINE

LOCATION	location_id, area_name, latitude, longitude
LISTING	listing_id, apartment_id (FK → APARTMENT), room_id (FK → ROOM, nullable), listing_type, price_per_person, availability_status, women_only, created_at
LISTING_PHOTO	photo_id, listing_id (FK → LISTING), photo_url, is_thumbnail
BOOKING	booking_id, listing_id (FK → LISTING), user_id (FK → USER), start_date, end_date, status, created_at
UNIVERSITY	university_id, name, latitude, longitude
COMMUTE_TIME	commute_id, user_id (FK → USER), listing_id (FK → LISTING), university_id (FK → UNIVERSITY), walking_time, bus_time, distance_km, calculated_at

EXPECTED STORED PROCEDURES

User & Authentication

- sp_RegisterUser - Registers a new user
- sp_LoginUser - Validates login credentials

Property Listing

- sp_AddProperty - Inserts a new rental property posted by an owner
- sp_UpdatePropertyStatus - Used to mark a listing as AVAILABLE or RENTED

Student Room-Sharing

- sp_AddRoomShareListing - Student posts a vacant room in their rented flat
- sp_UpdateRoomShareAvailability - Marks student room share listing as filled or available

EXPECTED STORED PROCEDURES

Search, Wishlist & Saved Searches	• sp_AddToWishlist - Adds a property or room-share listing to a user's wishlist • sp_SaveSearch - Saves user's preferred search filters (rent range, area, sharing type)
Analytics & Fair Rent	• sp_CalculateAverageRent - Computes average rent for a given area • sp_UpdateFairRentScore - Calculates a property's Fair Rent Score based on area average, property rent, size & amenities.

EXPECTED TRIGGERS

Trigger Name	Timing	Table	Purpose
trg_before_user_insert_validation	Before Insert	User	Validates new user registration by checking credentials
trg_before_property_insert_validation	Before Insert	Listing	Ensures that only users with “Owner” role can post rental listings
trg_before_roomshare_insert_validation	Before Insert	RoomShareListing	Allows only students to post roommate listing
trg_after_wishlist_insert_audit	After Insert	Wishlist	Logs every Wishlist insertion into an Audit_Log table for analytics
trg_prevent_duplicate_wishlist	Before Insert	Wishlist	Prevents the same user from adding the same listing multiple times

EXPECTED TRIGGERS

Trigger Name	Timing	Table	Purpose
trg_before_visit_booking_validation	Before Insert	Booking	Prevents time-slot conflicts & blocks booking on inactive listing
trg_after_visit_booking_status_update	After Update	Booking	When a visit is approved, the selected time slot is automatically blocked
trg_after_listing_insert_update_rent_stats	After Insert or Update	Listing	Updates area-wise rent statistics used in analytics & fair rent estimation
trg_after_any_listing_update_audit	After Update	Listing	Stores old and new values of modified listings in Audit_Log table for rent changes
trg_after_verification_approved_tag_update	After Update	Verification	Automatically updates the related user after getting approved.

MAJOR SQL QUERIES

Query Name	Type	Purpose
avg_rent_by_area	Aggregation	Calculates average, min, max rent per area for fairness analysis
top_wishlisted_listings	Join + Aggregation	Returns top most wishlisted listings
fairness_score_per_listing	Function + Join	Computes fairness score using area rent averages
search_listings_jsonb	JSONB Query	Retrieves listings matching JSONB-based saved search filters
room_availability_statistics	Aggregation	Count available rooms by area and month.

MAJOR SQL QUERIES

Query Name	Type	Purpose
market_rent_statistics	Rollup/Reporting	Show overall and area-wise rental trends across the campus zone.
property_visit_reports	Window Function	Show total and approved visit requests for each property.
listings_grouped_location	ROLLUP	Aggregates listings by location for admin reporting
active_users_week	Window Function	Tracks weekly active users
saved_search_notifications	Nested Subquery + EXISTS	Identifies new listings matching saved searches for notifications

DBMS CHOICE

PostgreSQL

Why choosing this?

JSONB support for saved searches

Strong support for window and analytical functions

PL/pgSQL for procedures and triggers

Indexing options (B-tree, GIN) suitable for complex queries

TECH STACK

- **Frontend:** React.js
- **Backend:** Node.js + Express.js
- **Database:** PostgreSQL (Using Supabase)
- **Authentication:** JWT
- **Version Control:** Git + GitHub
- **Styling:** Tailwind CSS
- **Real-Time Features:** Socket.IO
- **Mapping / Commute:** OpenStreetMap API
- **Testing:** Postman

TEAM ROLES

Member	Database	Backend	Frontend
Rahatut Tahrim Mounota	Schema design, ER diagram, normalization, indexing	Procedures, functions for listings and commute calculations	Listing creation and display pages
Nayeemul Hasan Prince	Triggers, audit log design, data validation rules	Booking and visit scheduling procedures	Booking interface, wishlist and saved search implementation
Alfi Shahrin	Views, analytics queries, JSONB filters	Fair rent calculation, saved search matching functions, notifications	Dashboard for analytics, verification badges, user role interfaces

THANK YOU