2nd Semester 2022-23

CSF 212 (Database Systems)

Project (Home Assignment); Total marks:20

Focus: In this activity the focus will be on ER/EER Modeling, Relational schema design and refinement (Normalization), implementing database (creating tables with constraints and inserting initial data necessary) and writing SQL/PL-SQL code for data entry, updates, generating reports, manipulation and enforcing certain complex constraints or rules.

Domain description:

We assume a Property Rental Agency (PRA) scenario where the PRA facilitates mediating between Owners of a properties and the Tenants, across multiple cities in India. Both Owner and Tenants are treated as the users of the system. There is a supper user (only one) who is the DBA. Some users are just Managers. DBA can add/delete/modify other users. Managers can add/modify/delete a property. Add details about which property is rented to which tenant etc. Some users can play both owner and tenant roles. That is, they may give their property for rent and may seek another property from others for rent. A property can be a residential property (independent-house/flat) or a commercial property (shop or warehouse). Each property has an ID which is unique. Each property has an owner. A user has adharID, name, age, address(includes door#, Street, city, state and PIN code) and multiple phone numbers. All users will have login credentials. A user as an owner can register a property he wants to give for rent. Once a property is given to a tenant, we capture that information. That means, we capture details like what is property is rented to which tenant, rent per month, start_date, end_date, yearly hike in rent(in %), agency commission, and other info as necessary. A property may not have a tenant if it is not rented. We also maintain the history of renting for a property. Every property is entered in to the system with the details like propertyID, owner, available from which date, available till what date, rent pm, %ge of annual hike in rent, total area(not null), plinth area(not null), number of bed-rooms(if residential), number of floors, year

of construction (not null), locality, address, other facilities etc. A user when he uploads a property becomes owner. He can upload zero to many properties. A user when he takes a property for rent becomes tenant. He can be a tenant for zero to many properties.

Functionality Expected:

1. Adding user to the database by DBA with necessary privileges.
2. Allowing DBA and Manager to add/delete/update any property record.
3. Allowing owner to add/delete/update his property record only.
4. Allowing Admin and Manager to add property-rent details after property is rented by a tenant.
5. Generate a report on rent history of a property.
6. To check list available properties for rent with in a given city/locality/price range.
7. To check the status of a property based on property ID by all users.
8. A tenant can't insert/delete/modify any data in any table. He can just look for available properties for rent.

Note: students can assume other necessary constraints, and attributes if missing.

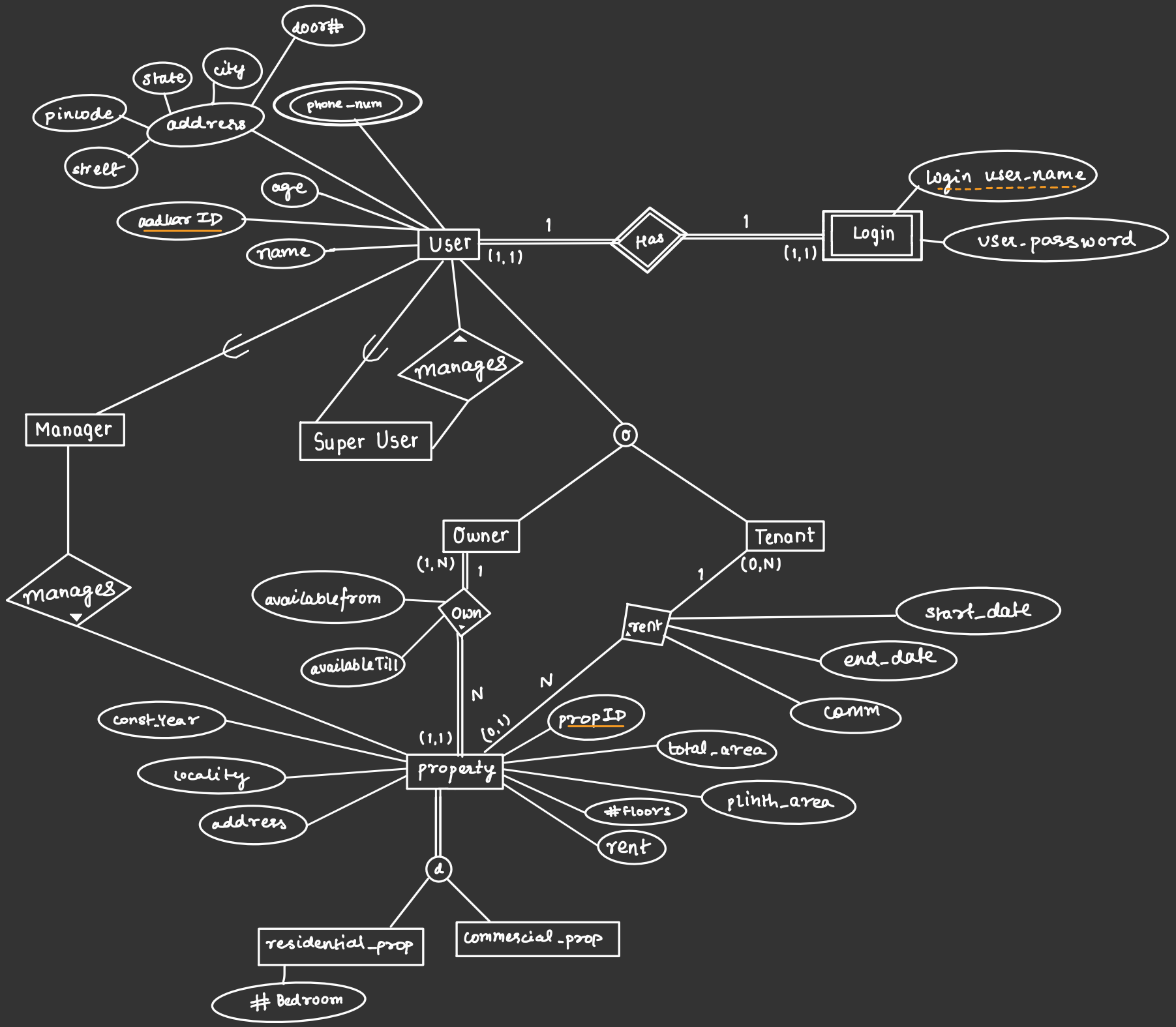
1. **Mid-semester evaluation**: Will be for 10 marks; timelines: between 20th and 26th March; Only ER/EER modeling and relational scheme design with constraints (shown on a sheet of paper or computer) will be evaluated. We will not look at the implementation. Viva will carry 4 marks. Clarity and cleanliness of your presentation on paper will carry some weightage please keep this in mind.
2. **End-semester Evaluation**: Will be for 10 marks; timelines: between 24th and 26th April; We look at implementation, data entry, report generation, enforcing the constraints,

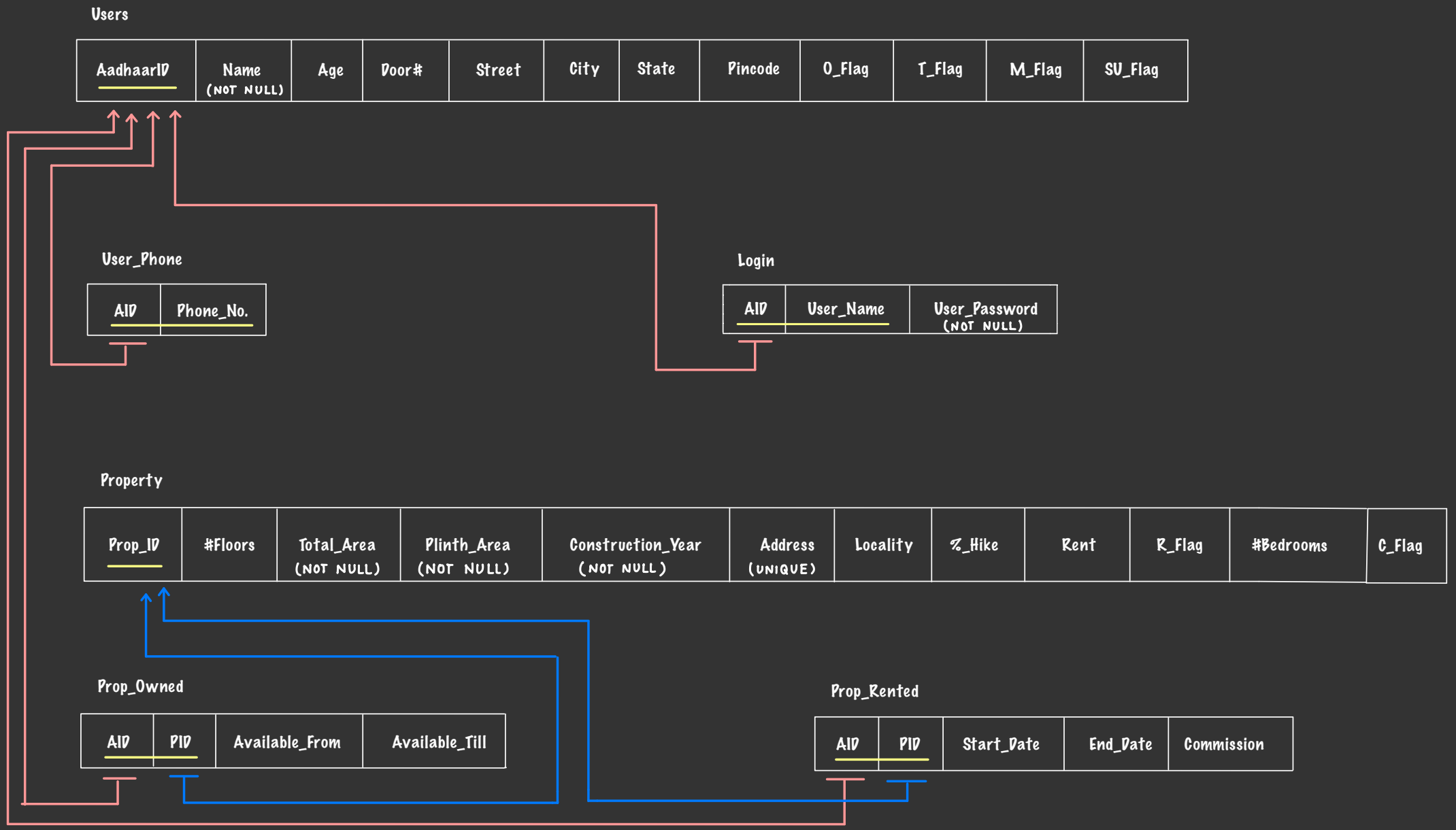
SQL/PL-SQL code etc. Viva will carry 4 marks. Groups will be demonstrating all these on their Laptop.

3. **GUI is not mandatory**. The entire work (data, functionality like data entry, report generation etc.) can be demonstrated at command prompt. Hence GUI does not carry any weightage. Still you can use to make your application more elegant.
4. Any act of **plagiarism** will be taken to the Disciplinary Committee.

Note: *Any additions/changes/corrections to the specifications or requirements will be announced in the class and on CMS as required.*

Prof R Gururaj,
IC CSF212;
Dt: 25-Feb-2023.


door#
state
city
pincode
address
phone_num
street
age
aadhar ID
name
User
1
(1,1)
Has
1
(1,1)
Login
login user_name
user_password
manages
Manager
Super User
O
Owner
Tenant
(1,N)
1
1
(0,N)
manages
availablefrom
Own
availableTill
rent
start_date
end_date
comm
N
N
(1,1)
(0,1)
propID
const_year
locality
address
property
total_area
plinth_area
#floors
rent
d
residential_prop
commercial_prop
Bedroom


Users
AadhaarID
Name (NOT NULL)
Age
Door#
Street
City
State
Pincode
O_Flag
T_Flag
M_Flag
SU_Flag
User_Phone
AID
Phone_No.
Login
AID
User_Name
User_Password (NOT NULL)
Property
Prop_ID
#Floors
Total_Area (NOT NULL)
Plinth_Area (NOT NULL)
Construction_Year (NOT NULL)
Address (UNIQUE)
Locality
%_Hike
Rent
R_Flag
#Bedrooms
C_Flag
Prop_Owned
AID
PID
Available_From
Available_Till
Prop_Rented
AID
PID
Start_Date
End_Date
Commission

Roles Permission

SU ⇒ Users, User_Phone, Login

M ⇒ Property, Prop-Owned, Prop_Rented (Full)

Users, User_Phone, Login (S, U)

O ⇒ Property, Prop-Owned, Prop-Rented (Full)

Users, User_Phone, Login (S, U)

(Full)

T ⇒ Property, Prop_Owned, Prop_Rented

(S, I) (S, I) (S, I, U, D)

Users, User_Phone, Login

(S, U) (S, I, U) (S, U)