**Discovery Park Online Apartment Rental Management System**

INFO 5707: Term Project Proposal

**Group: 14**

**Team Members:**

Sai Satish Pallapolu (Team Coordinator)

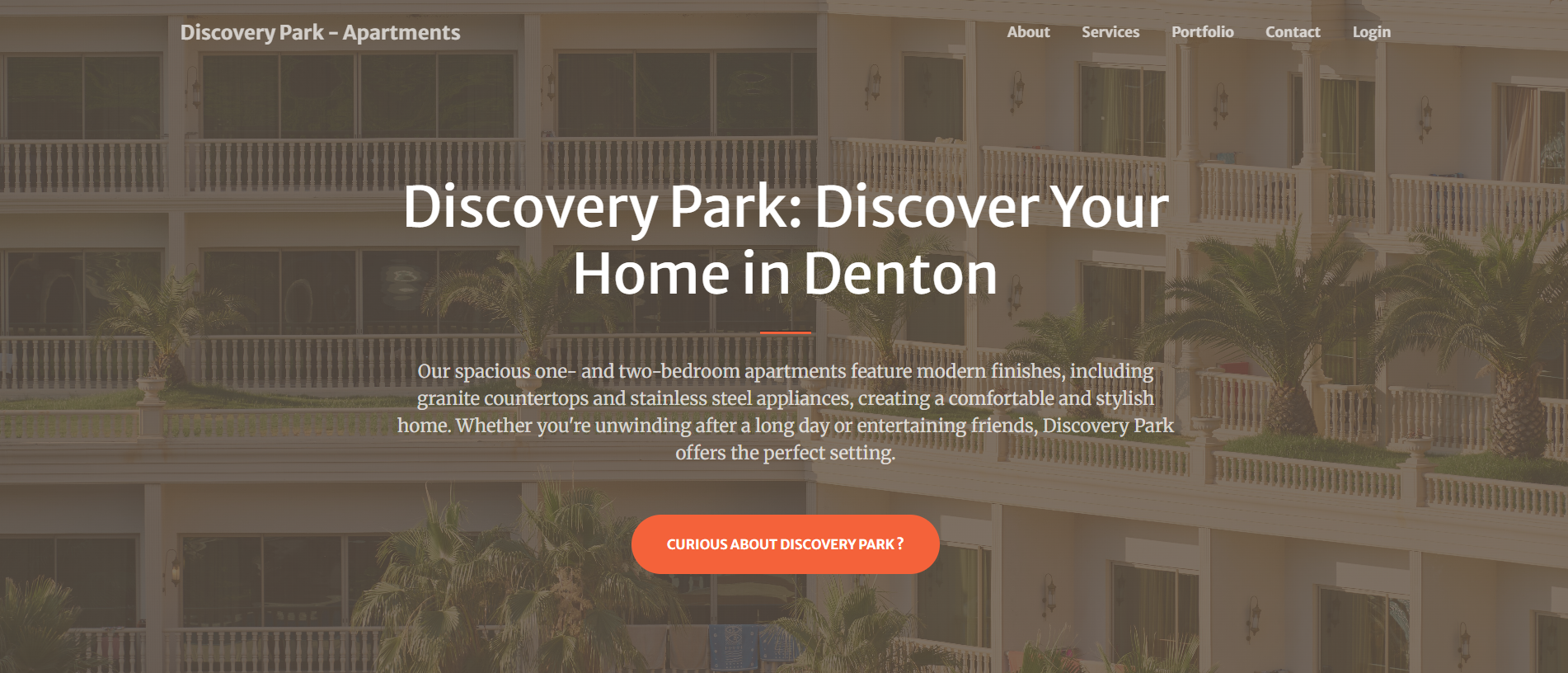
Viswa Sai Teja Gogineni

Venkata Rajyalakshmi Gudala

Karthik Goud Kasula

**Title: Discovery Park Online Apartment Rental Management System**

**Application Landing Page:**



**Objective:**

The Discovery Park Online Apartment Rental Management System provides a digital tool for apartment rental processes with improvements for efficiency along with better user experiences. The platform functions as a single database through which property owners, tenants and administrators can execute rental contract management tasks combined with property availability functions and facility request systems and financial process handling.

**Scope:**

* This application facilitates online property listings and rental management like payments and maintenance.
* This database performs complete lease management functions that include agreement end dates and rent amount.
* This database maintains rental payment records track and status.
* The database supports the handling of maintenance requests which also includes monitoring service completion statuses.
* This database is supported by multi-user access with role-based privileges which means multiple user can access and perform operations like public and administrators.
* This database system protects data security through PostgreSQL ACID compliance combined with role-based access control features which means features are shown or will be enabled as per the login user.
* This database system enables usage by users from different language backgrounds and those who require currency conversion capabilities.

**User requirements that the system will be able to address:**

1. This database system provides capabilities for landowners to publish their property listings along with images and descriptions of property.
2. This database platform grants tenants a system to find all the listings and displays the rental application documentation.
3. This system also provides payment processing functionality that enables tenants to pay rent online and owners can track of payments pending transactions status.
4. This allows tenants to file service requests that property managers can assign a technician.
5. This system implements the features for secure user logins and shows certain pages to users.
6. Tenants can know their neighbours present in the community by a feature know your community.
7. A super admin can onboard a tenant, owner or other managers to access database as per their roles and privileges.
8. This database system provides different roles for landlords and managers and tenants. This includes three different user groups to interact with the system.
9. Manager can onboard technicians to assign service request based on their availability.
10. Technicians can update their status availability as per their availability also it do not assign more than one ticket per technician till the ticket is closed.

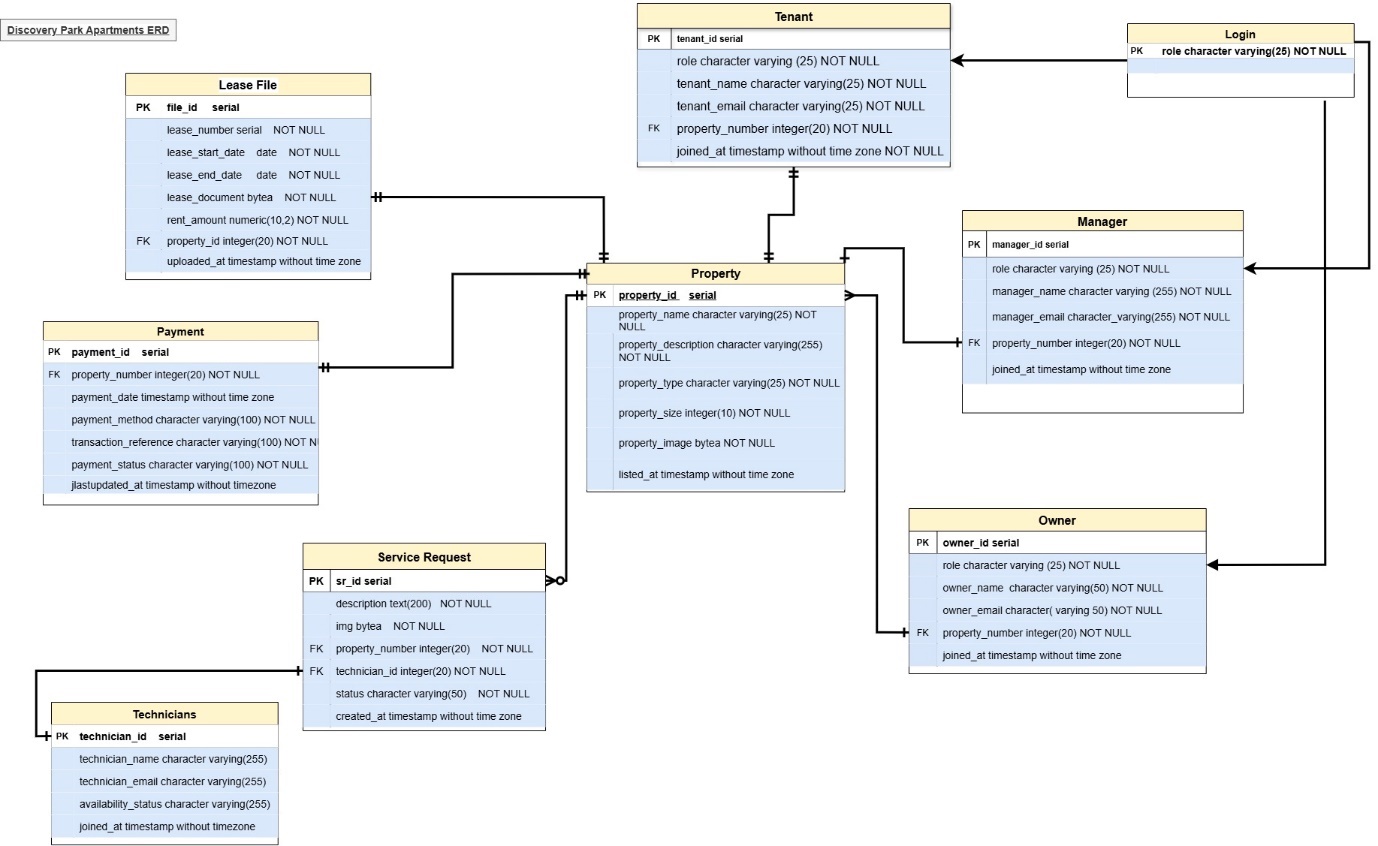
**Business rules:**

The following are the business rules enforced by the database system

1. For each property there will be only one property owner, and A owner can hold many properties.
2. For each property there will be only one tenant and one tenant can rent only one property at a time under his name.
3. For each property there can be one and only manager assigned.
4. Each tenant must have one and only one property owner.
5. Each property owner can have one or more tenants.
6. Each lease file will be associated with only one property and one lease file should dedicated to one and only property.
7. One Property can have only one listing at a time.
8. Each payment will refer to one and only one property and payments are made to a valid house number which is property\_id in property table.
9. There can be only one open service request raised for a property.
10. There will be only one technician associated with service request.
11. Status of a service request will be open by default when created unless it is accepted by a technician.
12. Each Property has a unique “property\_id” field, which is referred by payments, lease file and service request.
13. A tenant must have a lease agreement associated with the property to make a payment on rent. (Since rent amount is decided with lease).

**ERD Diagram:**

The below diagram shows the ERD representation of the tables given



**Relationships:**

**Each Owner can own many Properties (1:M)**

It is possible for a property owner to own more than one property. When people or investment groups own several rental properties, apartment buildings, or homes, this type of partnership is typical in the real estate industry. Every property has a distinct property\_id that is connected to the database's owner\_id. The system can effectively track ownership information thanks to this structure.

**One Property can have many Tenants (through Lease File) (1:M)**

Multiple tenants may eventually lease each building. Tenant and property lease agreements are documented in the Lease File table, which creates this relationship. Even though there might only be one renter living in each unit at any given time, the same property may have had several tenants over the years.

**One Tenant can have only one active Lease per Property (1:1)**

Tenants are only permitted to have one active lease per property at a time. This makes sure that a single rental property isn't rented to several people at once. By assigning each tenant a single lease per property, with a clearly specified lease\_start\_date and lease\_end\_date, the Lease File table upholds this requirement.

**One Tenant can submit multiple Service Requests for a Property (1:M)**

Many problems, such as plumbing leaks, HVAC failures, and electrical problems, might arise throughout a renter's lease. The system indicates that tenants can use their property\_id to submit multiple Service Requests (SRs). Tenants can record maintenance requirements in this way, and the management staff can monitor and address problems in a methodical manner.

**One Technician can handle multiple Service Requests (1:M)**

A single technician is in charge of managing several service requests from various properties. This guarantees effective task distribution and facilitates performance monitoring for technicians. In the Service Request database, this link is made by the technician\_id.

**One Property can have multiple Payments recorded (1:M)**

Monthly rent payments, security deposits, and maintenance fees are only a few of the numerous financial transactions that are necessary for each property. All financial records are connected to the appropriate property\_id thanks to the system's maintenance of a 1:M relationship between properties and payments.

**One Manager can manage multiple Owners and Tenants (1:M)**

Several property owners, renters, and service requests must be managed by a property manager. Tenant\_id and owner\_id is connected to manager\_id, guaranteeing that a single manager can oversee operations for several stakeholders.

**Data Dictionary:**

The below diagram shows the Data Dictionary representation of the tables given

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

**AI use disclosure statement:**

We acknowledge the use of Gemini (<https://gemini.google.com/app/c45d9c1c10508825> to refine the academic language for the description column of our data dictionary. The tool was used to provide description column of our data dictionary for the attribute names to improve academic language and for grammatical mistakes. The output from this tool was modified in our own words in a formal way.