**Project team 7 - Apartment Complex Management**

**Team Members:Tanya Peddi  
Shreyas Vaidyanath  
Umasankar Srinivas Varma Pusapati  
Abhilash Gandham**

PROJECT PROPOSAL

The goal is to create an application that allows for management of an apartment complex. The application allows for the storage of resident information which in turn allows for residents to gain basic and contact information about other residents. It also allows for the residents to register requests for services such as maintenance, plumbing etc. Residents can be either permanent residents or tenants each having different privileges. Property managers of any apartments have to register themselves with the management which again enables the users of the system to gain information about them. Residents will also be able to find out about upcoming events in the complex. The application will allow permanent owners and property managers to know about their due amount for maintenance as well as pay it. The application will alert the residents when their parking permits are going to expire.

PROJECT ENVIRONMENT

JSP: To create dynamic web content using HTML

HTML: For Page layout designing

MYSQL: Database

GIT,GITHUB: Version Control

Eclipse IDE: For coding in general

**HIGH LEVEL REQUIREMENTS**

Initial user roles

1. Admin- The admin manages the application and edits/ modifies data of office staff.
2. Office Staff- They manage the details of the residents, apartments and employees
3. Employee- They are the maintenance employees who cater service requests and events.
4. Property Manager- They represent the tenants of an apartment.
5. Homeowner- They are the owners of the apartment.
6. Tenant- They are temporary/rent based residents of the apartment.

Initial user story descriptions

**Admin:**

1. As an admin I want to login to the system
2. As an admin I want to add Office Staff
3. As an admin I want to update/modify Office Staff details
4. As an admin I want to delete Office Staff

**Office Staff**:

1. As an office staff I want to login to the system
2. As an office staff I want to add Apartments
3. As an office staff I want to update/modify Apartment details
4. As an office staff I want to delete Apartments
5. As an office staff I want to add Employees
6. As an office staff I want to update/modify Employee’s details
7. As an office staff I want to delete Employees
8. As an office staff I want to add Homeowners
9. As an office staff I want to update/modify Homeowner’s details
10. As an office staff I want to delete Homeowners
11. As an office staff I want to add Residents
12. As an office staff I want to update/modify Resident’s details
13. As an office staff I want to delete Residents
14. As an office staff I want to allocate an Apartment to a Resident
15. As an office staff I want to deallocate an Apartment to a Resident
16. As an office staff I want to add Events
17. As an office staff I want to update/modify Event details
18. As an office staff I want to delete Events
19. As an office staff I want to allocate Parking Space

**Employee:**

1. As an Employee I want to login to the system
2. As an Employee I want to accept Service Request
3. As an Employee I want to reject Service Request
4. As an Employee I want to raise a Complaint

**Property Manager**

1. As a Property Manager I want to login to the system
2. As a Property Manager I want to accept the Payments by tenant.
3. As a Property Manager I want to pay the maintenance fees
4. As a Property Manager I want to request Parking Permits
5. As a Property Manager I want to renew the Parking Permit
6. As a Property Manager I want to view the Service Request by Tenant
7. As a Property Manager I want to view all my Tenant details

**Homeowner**

1. As a homeowner I want to login to the system
2. As a homeowner I want to pay my Bills
3. As a homeowner I want to request Parking Permit
4. As a homeowner I want to renew Parking Permit
5. As a homeowner I want to raise a Service Request
6. As a homeowner I want to cancel Service Request raised by me
7. As a homeowner I want to view Events

**Tenant**

1. As a tenant I want to login to the system
2. As a tenant I want to pay my Bill to the Property Manager
3. As a tenant I want to request Parking Permit
4. As a tenant I want to renew Parking Permit
5. As a tenant I want to raise a Service Request
6. As a tenant I want to cancel Service Request raised by me.
7. As a tenant I want to view Events

**HIGH LEVEL CONCEPTUAL DESIGN**

Entities:

1. Admin
2. Apartment
3. Office Staff // check
4. Employee
5. Property Manager
6. Resident
7. Homeowner
8. Tenant
9. Bill Generation
10. Events
11. Parking
12. Service Request

Relationships:

1. Homeowner has aparment
2. Tenants raise service request
3. Homeowner raise service request
4. Apartment has a property manager
5. Homeowner pays maintenance bill.
6. Property Manager pays maintenance bill.
7. Homeowner has parking permit
8. Tenants have parking permit
9. Property Manager requests parking permit.
10. Office Staff creates events
11. Tenants view events
12. Homeowners view events
13. Employee accepts/rejects service request.

**Prioritized User Stories**

1. As an admin I want to login to the system

2. As an admin I want to add Office Staff

3. As an office staff I want to login to the system

4. As an office staff I want to add Apartments

5. As an office staff I want to add Employees

6. As an office staff I want to add Homeowners

7. As an office staff I want to add Residents

8. As an office staff I want to allocate an Apartment to a Resident

9. As an Employee I want to login to the system

10. As a homeowner I want to login to the system

11. As a tenant I want to login to the system

12. As a Property Manager I want to login to the system

Story refinement, with notes:

1. As an admin, I want to login to the system

Notes:

- Since login and logging out are closely related. Add user stories for admin logout as well.

Updated stories:

a. As an admin, I want to login to the system so that I can access features specific to my role.

b. As an admin, I want to logout to the system.

2. As an admin I want to add Office Staff

Notes:

- Adding the office staff details such as name, contact\_number, email\_id and address.

Updated stories:

a. As an admin, I want to add the details of the office staff.

3. As an office staff I want to login to the system.

Notes:

- Since login and logging out are closely related. Add user stories for office staff logout as well.

Updated stories:

a. As a office staff, I want to login to the system so that I can access features specific to my role.

b. As a office, I want to logout to the system.

4. As an office staff I want to add Apartments

Notes:

- Apartments must have number of rooms ,facilities, number of restrooms,floor level,tenant id,deposit amount,rent, owner id , property manager.

Updated stories:

a. As a office staff, I want to add apartment details to the system.

5. As an office staff I want to add Employees

Notes:

- Employees must have name, contact\_number, designation, email\_id and address.

Updated stories:

a. As an office staff , I want to add employee details to the system.

6. As an office staff I want to add Homeowners.

7. As an office staff I want to add Residents

Notes:

- Rename Homeowners to Permanent Residents to better reflect meaning

- Residents is the generalization of Permanent Residents and tenants.

- Residents may have attributes such as name, contact\_number, email\_id,apartment\_number.

Updated Stories:

a. As an office staff, I want to add resident details.

8.As an office staff I want to allocate an Apartment to a Resident.

Notes:

- No changes needed.

9. As an Employee I want to login to the system

Notes:

- Since login and logging out are closely related. Add user stories for employee logout as well.

Updated stories:

a. As an employee, I want to login to the system so that I can access features specific to my role.

b. As an employee, I want to logout to the system.

10. As a homeowner I want to login to the system

Notes:

- As per the previous refinement, homeowner has been renamed to Permanent Resident.

- Since login and logging out are closely related. Add user stories for logout as well.

Updated stories:

a. As a Permanent Resident, I want to login to the system so that I can access features specific to my role.

b. As a Permanent Resident, I want to logout to the system.

11. As a tenant I want to login to the system

Notes:

- Since login and logging out are closely related. Add user stories for logout as well.

Updated stories:

a. As a tenant, I want to login to the system so that I can access features specific to my role.

b. As a tenant, I want to logout to the system.

12. As a Property Manager I want to login to the system

Notes:

- Since login and logging out are closely related. Add user stories for logout as well.

Updated stories:

a. As a Property Manager, I want to login to the system so that I can access features specific to my role.

b. As a Property Manager, I want to logout to the system.

**REFINED STORIES FOR SPRINT 1:**

1. As an admin, I want to login to the system so that I can access features specific to my role.

2. As an admin, I want to logout to the system.

3. As an admin, I want to add the details of the office staff.

4. As a office staff, I want to login to the system so that I can access features specific to my role.

5. As a office, I want to logout to the system.

6. As a office staff, I want to add apartment details to the system.

7. As an office staff , I want to add employee details to the system.

8. As an office staff, I want to add resident details.

9. As an office staff I want to allocate an Apartment to a Resident.

10. As an employee, I want to login to the system so that I can access features specific to my role.

11. As an employee, I want to logout to the system.

12. As a Permanent Resident, I want to login to the system so that I can access features specific to my role.

13. As a Permanent Resident, I want to logout to the system.

14. As a tenant, I want to login to the system so that I can access features specific to my role.

15. As a tenant, I want to logout to the system.

16. As a Property Manager, I want to login to the system so that I can access features specific to my role.

17. As a Property Manager, I want to logout to the system

**Conceptual Design for Sprint 1**

Entity: **Admin**

Attributes:

username

password

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Office Staff**

Attributes:

staff\_id

password

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Property Manager**

Attributes:

property\_manager\_id

password

apartment\_number

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Apartment**  
Attributes:

apartment\_number

number\_of\_rooms

floor\_level

property\_mgr\_id

status

number\_of\_parkingpermit

Entity: **Resident**

Attributes:

Resident\_id

password

name [composite]

last\_name

first\_name

email

phone\_number

apt\_number

Parking\_permit\_id

Entity: **Permanent Resident**

Attributes:

resident\_id

Entity: **Tenant**

Attributes:

resident\_id

property\_manager

Entity: **Parking**  
Attributes:

Apartment number

parking\_spot\_number

start\_date

End\_date

Relationships

Relationship: **Apartment** has **PropertyManager**

Cardinality: One to one

Participation:

Apartment has partial participation

PropertyManager has total participation

Relationship: **Apartment** has **Parking**

Cardinality: One to many

Participation:

Apartment has total participation Resident has total participation

Relationship: **Apartment** has **Resident**

Cardinality: One to many

Participation:

Apartment has partial participation

Resident has total participation

**Logical design for Sprint 1**

Table: **user\_login**

Columns**:**

user\_id

uname

pass

role

Table: **Admin**

Columns:

admin\_id [foreign key; references user\_id of user\_login table]

first\_name

last\_name

email

phone\_number

Table:: **OfficeStaff**

Columns:

staff\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

Table: **Resident**

Attributes:

resident\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

apt\_number[foreign key; references apartment\_number of Apartment table]

Table: **PermanentResident**

Attributes:

pr\_id [foreign key; references resident\_id of Resident table]

Table: **Tenant**

Attributes:

tenant\_id [foreign key; references resident\_id of Resident table]

propertymanager\_id [foreign key; references propertymanager\_id of PropertyManager table]

Table:**PropertyManager**

Columns:

propertymanager\_id[foreign key; references user\_id of user\_login table]

apartment\_number [foreign key; references apartment\_number of Apartment table]

last\_name

first\_name

email

phone\_number

Table: **Apartment**  
Columns:

apart\_no

number\_of\_rooms

floor\_level

status

number\_of\_parking\_permits

Table: **Parking**  
Column:

parking\_permit\_id

parking\_lot\_number

apartment\_no[foreign key; references apart\_no of Apartment table]

resident\_id [foreign key; references resident\_id of Resident table]

start\_date

end\_date

Note : As requested we have included the interface work of Resident along with that of an Admin and Office Staff. The next sprint will include the work of the specializations, permanent resident and tenant.

**SPRINT 2**

**Prioritized Stories:**

1. As an office staff I want to add Homeowners
2. As an office staff I want to add tenants.
3. As an employee, I want to login to the system so that I can access features specific to my role.
4. As an employee, I want to logout to the system.
5. As a Permanent Resident, I want to login to the system so that I can access features specific to my role.
6. As a Permanent Resident, I want to logout to the system.
7. As a tenant, I want to login to the system so that I can access features specific to my role.
8. As a tenant, I want to logout to the system.
9. As a homeowner I want to raise a Service Request
10. As a homeowner I want to cancel Service Request raised by me.
11. As a tenant I want to raise a Service Request
12. As a tenant I want to cancel Service Request raised by me.
13. As an Employee I want to accept Service Request
14. As an Employee I want to reject Service Request.
15. As a Property Manager, I want to login to the system so that I can access features specific to my role.
16. As a Property Manager, I want to logout to the system
17. As a Property Manager I want to view the Service Request by Tenant

Story refinement, with notes:

1. As an office staff I want to add Homeowners

Notes:

According to the refinement in the previous sprint,

Rename Homeowners to Permanent Residents to better reflect meaning

* As an office staff I want to add Permanent residents.

1. As an office staff I want to add tenants.
2. As an employee, I want to login to the system so that I can access features specific to my role.
3. As an employee, I want to logout to the system.
4. As a Permanent Resident, I want to login to the system so that I can access features specific to my role
5. As a Permanent Resident, I want to logout to the system.
6. As a tenant, I want to login to the system so that I can access features specific to my role.
7. As a tenant, I want to logout to the system.

Notes:

The first 8 stories are refined user stories carried forward from the previous sprint.

1. As a homeowner I want to raise a Service Request
2. As a homeowner I want to cancel Service Request raised by me.

Notes:

According to the refinement in the previous sprint,

Rename Homeowners to Permanent Residents to better reflect meaning

* As a permanent resident I want to raise a Service Request.
* As a permanent resident I want to cancel Service Request raised by me.

11. As a tenant I want to raise a Service Request

12. As a tenant I want to cancel Service Request raised by me.

13. As an Employee I want to accept Service Request.

14. As an Employee I want to reject Service Request.

15. As a Property Manager, I want to login to the system so that I can access features specific to my role.

16. As a Property Manager, I want to logout to the system

17. As a Property Manager I want to view the Service Request by Tenant

REFINED STORIES FOR SPRINT 2:

1. As an office staff I want to add Permanent residents.
2. As an office staff I want to add tenants.
3. As an employee, I want to login to the system so that I can access features specific to my role.
4. As an employee, I want to logout to the system.
5. As a Permanent Resident, I want to login to the system so that I can access features specific to my role
6. As a Permanent Resident, I want to logout to the system.
7. As a tenant, I want to login to the system so that I can access features specific to my role.
8. As a tenant, I want to logout to the system.
9. As a permanent resident I want to raise a Service Request
10. As a permanent resident I want to cancel Service Request raised by me.
11. As a tenant I want to raise a Service Request
12. As a tenant I want to cancel Service Request raised by me.
13. As an Employee I want to accept Service Request.
14. As an Employee I want to reject Service Request.
15. As a Property Manager, I want to login to the system so that I can access features specific to my role.
16. As a Property Manager, I want to logout to the system
17. As a Property Manager I want to view the Service Request by Tenant

**Conceptual design for Sprint 2**

Entity: **Admin**

Attributes:

username

password

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Office Staff**

Attributes:

staff\_id

password

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Property Manager**

Attributes:

property\_manager\_id

password

apartment\_number

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Apartment**  
Attributes:

apartment\_number

number\_of\_rooms

floor\_level

property\_mgr\_id

status

Number\_of\_parkingpermit

Entity: **Resident**

Attributes:

resident\_id

password

name [composite]

last\_name

First\_name

email

phone\_number

apt\_number

parking\_permit\_id

Entity: **Permanent Resident**

Attributes:

resident\_id

Entity: **Tenant**

Attributes:

resident\_id

property\_manager

Entity: **Property Manager**

Attributes:

property\_manager\_id

password

apartment\_number

name [composite]

last\_name

first\_name

email

phone\_number

NOTE: Assumption is being made that a property manager can manage only one apartment, thus making the cardinality one-to-one.

Entity: **Employee**

Attributes:

employee\_id

password

name [composite]

last\_name

first\_name

job\_type

email

phone\_number

Entity: **Service Request**

Attributes:

service\_request\_id;

employee\_assigned

apartment\_number

status

date\_of\_complaint

complaint\_description

NOTE: Assumption is being made that a service request can be handled by only one employee, thus making the cardinality one-to-many.

Relationships

Relationship: **Apartment** has **PropertyManager**

Cardinality: One to one

Participation:

Apartment has partial participation

PropertyManager has total participation

Relationship: **Apartment** has **Resident**

Cardinality: One to many

Participation:

Apartment has partial participation

Resident has total participation

Relationship: **Tenant** raises **Service Request**

Cardinality: Many to many

Participation:

Tenant has partial participation

Service Request has total participation

Relationship: **Permanent Resident** raises **Service Request**

Cardinality: One to many

Participation:

Permanent Resident has partial participation

Service Request has total participation

Relationship: **Employee** updates **Service Request**

Cardinality: One to many

Participation:

Employee has partial participation

Service Request has total participation

**Logical design for Sprint 2:**

Table: **user\_login**

Columns**:**

user\_id

uname

pass

role

Highest normalization level: <4NF>

Table: **Admin**

Columns:

admin\_id [foreign key; references user\_id of user\_login table]

first\_name

last\_name

email

phone\_number

Highest normalization level: <4NF>

Table: **OfficeStaff**

Columns:

staff\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

Highest normalization level: <4NF>

Table: **Resident**

Columns:

resident\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

apt\_number[foreign key; references apartment\_number of Apartment table]

Highest normalization level: <4NF>

Table: **PermanentResident**

Columns:

pr\_id [foreign key; references resident\_id of Resident table]

Highest normalization level: <4NF>

Table: **Tenant**

Columns:

tenant\_id [foreign key; references resident\_id of Resident table]

propertymanager\_id [foreign key; references propertymanager\_id of PropertyManager table]

Highest normalization level: <4NF>

Table: **PropertyManager**

Columns:

propertymanager\_id[foreign key; references user\_id of user\_login table]

apartment\_number [foreign key; references apartment\_number of Apartment table]

last\_name

first\_name

email

phone\_number

Highest normalization level: <4NF>

Table: **Apartment**  
Columns:

apart\_no

number\_of\_rooms

floor\_level

status

number\_of\_parking\_permits

Highest normalization level: <4NF>

Table: **Employee**

Columns:

employee\_id[foreign key; references user\_id of user\_login table]

name [composite]

last\_name

first\_name

job\_type

email

phone\_number

Highest normalization level: <4NF>

Table: **Service\_Request**

Columns:

service\_request\_id

employee\_assigned\_id[foreign key; references user\_id of employee table]

apartment\_number[foreign key; references apartment\_number of Apartment table]

status

complaint\_description

date\_of\_complaint

Highest normalization level: <4NF>

**SPRINT 3**

**Prioritized Stories:**

1. As a homeowner I want to pay my Bills.
2. As a tenant I want to pay my Bill to the Property Manager.
3. As a Property Manager I want to accept the Payments by tenant.
4. As a Property Manager I want to pay the maintenance fees.
5. As an office staff I want to allocate Parking Space
6. As a homeowner I want to request Parking Permit
7. As a tenant I want to request Parking Permit
8. As a Property Manager I want to request Parking Permits
9. As a homeowner I want to renew Parking Permit
10. As a tenant I want to renew Parking Permit.
11. As a Property Manager I want to renew the Parking Permit.
12. As an office staff I want to add Events.
13. As a homeowner I want to view Events.
14. As a tenant I want to view Events.
15. As an office staff I want to update/modify Event details
16. As an office staff I want to delete Events

Story refinement, with notes:

1. As a homeowner I want to pay my Bills.

Notes:

* As per the previous refinement, homeowner has been renamed to Permanent Resident.
* Bills in this story stand for maintenance bills, we shall use Maintenance bill instead of Bills to better reflect meaning.

Updated stories:

As a Permanent Resident, I want to pay my Maintenance bill.

1. As a tenant I want to pay my Bill to the Property Manager.

Notes:

* The Bill stands for Maintenance Bill as well as Rent.

Updated stories:

* 1. As a tenant, I want to pay my Maintenance bill to the Property Manager.
  2. As a tenant, I want to pay my Rent to the Property Manager.

1. As a Property Manager I want to accept the Payments by tenant.

Notes:

* No refinement needed.

1. As a Property Manager I want to pay the maintenance fees.

Notes:

* We shall rename Maintenance fee to Maintenance Bill of my tenants to better reflect meaning.

Updated Stories:

As a Property Manager I want to pay the maintenance bill.

1. As an office staff I want to allocate Parking Space.

Notes:

* No refinement needed.

1. As a homeowner I want to request Parking Permit.

Notes:

As per the previous refinement, homeowner has been renamed to Permanent Resident.

Updated Stories:

As a Permanent Resident I want to request Parking Permit.

1. As a tenant I want to request Parking Permit

Notes:

* The tenant requests the parking permit to his respective property manager so we shall refine the story to better reflect meaning.

Updated Stories:

As a tenant, I want to request Parking Permit to the property manager.

1. As a Property Manager I want to request Parking Permits.

Notes:

-Here the property manager is requesting parking permits on behalf of the requesting tenants.

1. As a homeowner I want to renew Parking Permit.

Notes:

As per the previous refinement, homeowner has been renamed to Permanent Resident.

Updated Stories:

As a Permanent Resident I want to renew Parking Permit.

1. As a tenant I want to renew Parking Permit

Notes:

* The tenant requests to renew the parking permit to his respective property manager so we shall refine the story to better reflect meaning.

Updated Stories:

As a tenant, I want to request the property manager to renew Parking Permit.

1. As a Property Manager I want to renew Parking Permits.

Notes:

* Here the property manager is renewing parking permits on behalf of the requesting tenants.

1. As an office staff I want to add Events.

Notes:

* No refinement needed.

1. As a homeowner I want to view Events.

Notes:

As per the previous refinement, homeowner has been renamed to Permanent Resident.

Updated Stories:

As a Permanent Resident I want to view Events.

1. As a tenant I want to view Events.

Notes:

* No refinement needed.

1. As an office staff I want to update/modify Event details

Notes:

* No refinement needed.

1. As an office staff I want to delete Events.

Notes:

* No refinement needed.

REFINED STORIES FOR SPRINT 3:

1. As a Permanent Resident, I want to pay my Maintenance bill.
2. As a tenant, I want to pay my Maintenance bill to the Property Manager.
3. As a tenant, I want to pay my Rent to the Property Manager.
4. As a Property Manager I want to accept the Payments by tenant.
5. As a Property Manager I want to pay the maintenance bill.
6. As an office staff I want to allocate Parking Space.
7. As a Permanent Resident I want to request Parking Permit.
8. As a tenant, I want to request Parking Permit to the property manager.
9. As a Property Manager I want to request Parking Permits.
10. As a Permanent Resident I want to renew Parking Permit.
11. As a tenant, I want to request the property manager to renew Parking Permit.
12. As a Property Manager I want to renew Parking Permits.
13. As an office staff I want to add Events.
14. As a Permanent Resident I want to view Events.
15. As a tenant I want to view Events.
16. As an office staff I want to update/modify Event details.
17. As an office staff I want to delete Events.

**Conceptual design for Sprint 3**

Entity: **Admin**

Attributes:

username

password

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Office Staff**

Attributes:

staff\_id

password

name [composite]

last\_name

first\_name

email

Phone\_number

Entity: **Apartment**  
Attributes:

apartment\_number

number\_of\_rooms

floor\_level

status

Number\_of\_parkingpermit

Entity: **Resident**

Attributes:

resident\_id

password

name [composite]

last\_name

First\_name

email

phone\_number

apt\_number

Entity: **Permanent Resident**

Attributes:

resident\_id

Entity: **Tenant**

Attributes:

resident\_id

property\_manager

Entity: **Property Manager**

Attributes:

property\_manager\_id

password

apartment\_number[multivalued]

name [composite]

last\_name

first\_name

email

phone\_number

Entity: **Employee**

Attributes:

employee\_id

password

name [composite]

last\_name

first\_name

job\_id

email

phone\_number

Entity: **Service Request**

Attributes:

service\_request\_id;

job\_type\_id

status

date\_of\_complaint

complaint\_description

Entity: **Maintenance Bill**

Attributes :

due\_date

amount\_due

status\_of\_current\_month

Note: The system works in the manner that the propertymanager and the permanent resident pay the maintenance fee the value of which is regularly updated by the office staff.

Entity: **Rent**

Attributes :

rent\_id

rent\_amount

due\_date

payment\_status

comment

Note:Each tenant pays the rent to the property manger of their apartment. The rent will include their portion of the maintenance bill, the rent and any other extra charges.

Entity: **Parking\_permit**

Attributes:

parking\_permit\_id

expiry\_date

Entity: **Event**

Attributes:

event\_id

event\_title

event\_startdate

event\_location

event\_enddate

Relationships

Relationship: **Apartment** has **PropertyManager**

**Attributes: property\_manager\_id**

Cardinality: Many to one

Participation:

Apartment has partial participation

PropertyManager has total participation

Relationship: **Apartment** has **Resident**

Cardinality: One to many

Participation:

Apartment has partial participation

Resident has total participation

Relationship: **Tenant** raises **Service Request**

**Attribute : apartment\_no**

Cardinality: One to many

Participation:

Tenant has partial participation

Service Request has total participation

Relationship: **Permanent Resident** raises **Service Request**

**Attribute: apartment\_no**

Cardinality: One to many

Participation:

Permanent Resident has partial participation

Service Request has total participation

Relationship: **Employee** updates **Service Request**

Cardinality: One to many

Participation:

Employee has partial participation

Service Request has total participation

Relationship: **Apartment** has a **Parking\_permit**

**Attribute : parking\_permit\_id**

**: apartment\_no**

Cardinality: One to many

Participation:

Apartment has partial participation

Parking has total participation

Relationship: **Permanent Resident** pays **maintenance bill**

**Attribute : apartment\_no**

Cardinality : One to many

Participation:

Permanent Resident has total participation

Maintenance Bill has total participation

Relationship: **Property Manager** pays **maintenance bill.**

**Attribute : apartment\_no**

Cardinality: One to many

Participation:

Property Manager has total participation

Maintenance Bill has total participation

Relationship: **Permanent Resident** has **parking permit**

**Attribute : parking\_permit\_id**

Cardinality: One to Many

Participation:

Permanent Resident has partial participation

Parking Permit has total participation

Relationship: **Tenant** has **parking permit**

**Attribute : parking\_permit\_id**

Cardinality: One to many

Participation:

Tenant has partial participation

Parking permit has total participation

Relationship: **Property Manager** requests **parking permit**.

Attribute : parking\_ request\_id

apartment\_number

property\_manager\_id

Cardinality: One to many

Participation:

Property Manager has partial participation

Parking permit has total participation

Relationship : **Tenant** requests **parking permit**

Attribute: parking\_ request\_id

apartment\_number

tenant\_id

Cardinality: One to many

Participation : Tenant has partial participation

Parking Permit has total participation

Relationship : **Permanent Resident** requests **parking permit**

Attribute: parking\_ request\_id

apartment\_number

resident\_id

Cardinality: One to many

Participation : Permanent Resident has partial participation

Parking Permit has total participation

Relationship : **Tenant** pays **Rent**

Attribute : tenant\_id

Cardinality: One to many

Participation:

Tenant has partial participation

Rent has total participation

**Logical design for Sprint 3:**

Table: **user\_login**

Columns**:**

user\_id

uname

pass

role\_id[foreign key; references **role\_id** of **user\_roles** table]

Highest normalization level: <4NF>

Indexes:

user\_id [clustered index]

uname[non-clustered index]

Table: **user\_roles**

Columns**:**

role\_id

role\_title

Highest normalization level: <4NF>

Indexes:

role\_id[clustered index]

Table: **Admin**

Columns:

admin\_id [foreign key; references user\_id of user\_login table]

first\_name

last\_name

email

phone\_number

Highest normalization level: <4NF>

Indexes:

admin\_id[clustered index]

Table: **OfficeStaff**

Columns:

staff\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

Highest normalization level: <4NF>

Indexes:

staff\_id[clustered index]

Table: **Resident**

Columns:

resident\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

apt\_number[foreign key; references apartment\_number of Apartment table]

Highest normalization level: <4NF>

Indexes:

resident\_id[clustered index]

Table: **PermanentResident**

Columns:

pr\_id [foreign key; references resident\_id of Resident table]

Highest normalization level: <4NF>

Indexes:

pr\_id[clustered index]

Table: **Tenant**

Columns:

tenant\_id [foreign key; references resident\_id of Resident table]

propertymanager\_id [foreign key; references propertymanager\_id of PropertyManager table]

Highest normalization level: <4NF>

Indexes:

tenant\_id[clustered index]

Table: **PropertyManager**

Columns:

propertymanager\_id[foreign key; references user\_id of user\_login table]

last\_name

first\_name

email

phone\_number

Highest normalization level: <4NF>

Indexes:

propertymanager\_id[clustered index]

Table: **Apartment**  
Columns:

apart\_no

number\_of\_rooms

floor\_level

status

number\_of\_parking\_permits

Highest normalization level: <4NF>

Indexes:

apart\_no[clustered index]

Table: **Apartment\_PropManager**

Columns:

apartment\_number [foreign key; references apartment\_number of Apartment table]

property\_mgr\_id [foreign key; references propertymanager\_id of propertymanager table]

Highest normalization level:<4NF>

Indexes:

Apartment\_number and property\_mgr\_id[clustered index]

Table: **Employee**

Columns:

employee\_id[foreign key; references user\_id of user\_login table]

name [composite]

last\_name

first\_name

job\_id[foreign key; references **job\_id** of **job\_types** table]

email

phone\_number

Highest normalization level: <4NF>

Indexes:

employee\_id[clustered index]

job\_id[non-clustered index]

Table: **job\_types**

Columns**:**

job\_id

job\_title

Highest normalization level: <4NF>

Indexes:

job\_id[clustered index]

Table: **Service\_Request**

Columns:

service\_request\_id

job\_type\_id[foreign key; references job\_id of job\_types table]

apartment\_number[foreign key; references apartment\_number of Apartment table]

status

complaint\_description

date\_of\_complaint

Highest normalization level: <4NF>

Indexes:

service\_request\_id[clustered index]

status[non-clustered index]

apartment\_number[non-clustered index]

Table: **Service\_Request\_Assigned**

Columns:

service\_req\_id[foreign key; references service\_request\_id of service\_request table]

employee\_assigned\_id[foreign key; references employee\_id of employee table]

Highest normalization level: <4NF>

Indexes:

Service\_req\_id and employee\_assigned\_id[clustered index]

Table: **Maintenance\_Bill**

Columns :

bill\_number

apartment\_number[foreign key; references apartment\_number of Apartment table]

due\_date

amount\_due

status\_of\_current\_month

Highest normalization level: <4NF>

Indexes:

bill\_number[clustered index]

apartment\_number[non-clustered index]

Table: **Rent**

Columns :

rent\_id

tenant\_id[foreign key; references tenant\_id of tenant table]

apartment\_number[foreign key; references apartment\_number of Apartment table]

rent\_amount

due\_date

payment\_status

comment

Highest normalization level: <4NF>

Indexes:

rent\_id[clustered index]

payment\_status[non-clustered index]

apartment\_number[non-clustered index]

Table: **Parking\_Permit**

Columns:

parking\_permit\_id

apartment\_number[foreign key; references apartment\_number of Apartment table]

expiry\_date

Highest normalization level: <4NF>

Indexes:

parking \_permit\_id[clustered index]

appartment\_number[non-clustered index]

Table: **Parking \_Request\_tenant**

Columns:

parking\_\_req\_tenant\_id

apartment\_number[foreign key; references apartment\_number of Apartment table]

tenant\_id[foreign key; references tenant\_id of tenant table]

status

Highest normalization level: <4NF>

Indexes:

parking\_\_req\_tenant\_id[clustered index]

status[non-clustered index]

apartment\_number[non-clustered index]

Table: **Parking \_Request**

Columns:

parking\_request\_id

apartment\_number[foreign key; references apartment\_number of Apartment table]

requester\_id[foreign key; references user\_id of user\_login table]

status

Highest normalization level: <4NF>

Indexes:

parking\_request\_id[clustered index]

status[non-clustered index]

apartment\_number[non-clustered index]

Table: **Event**

Columns:

event\_id

event\_title

event\_startdate

event\_location

event\_enddate

Highest normalization level: <4NF>

Indexes:

event\_id[clustered index]

event\_startdate[non-clustered index]