APARTMENT MANAGER

Submitted by-
Leena Bhirud Madhura Jadhav Hrishikesh Tiwari Harsh Nanwani Divya Pai

INTRODUCTION

Our project is an in-house database system called Apartment Manager. In a local housing society with multiple homes, the administrative staff have to constantly communicate with the residents regarding various announcements or issues. This project aims at streamlining this interaction between resident and administrative staff.

Apartment Manager will be the go-to place for communications and information. It will enable residents to submit a maintenance related complaint, get announcements from administrative staff, get inspection schedule, and get a package and visitor notifications etc. Through community wall feature, residents can stay connected with fellow residents via two-way messaging.

Our objective is to make management of an apartment building more systematic and effective and to provide a comfortable and hassle-free living experience for residents.

HIGHLIGHTS

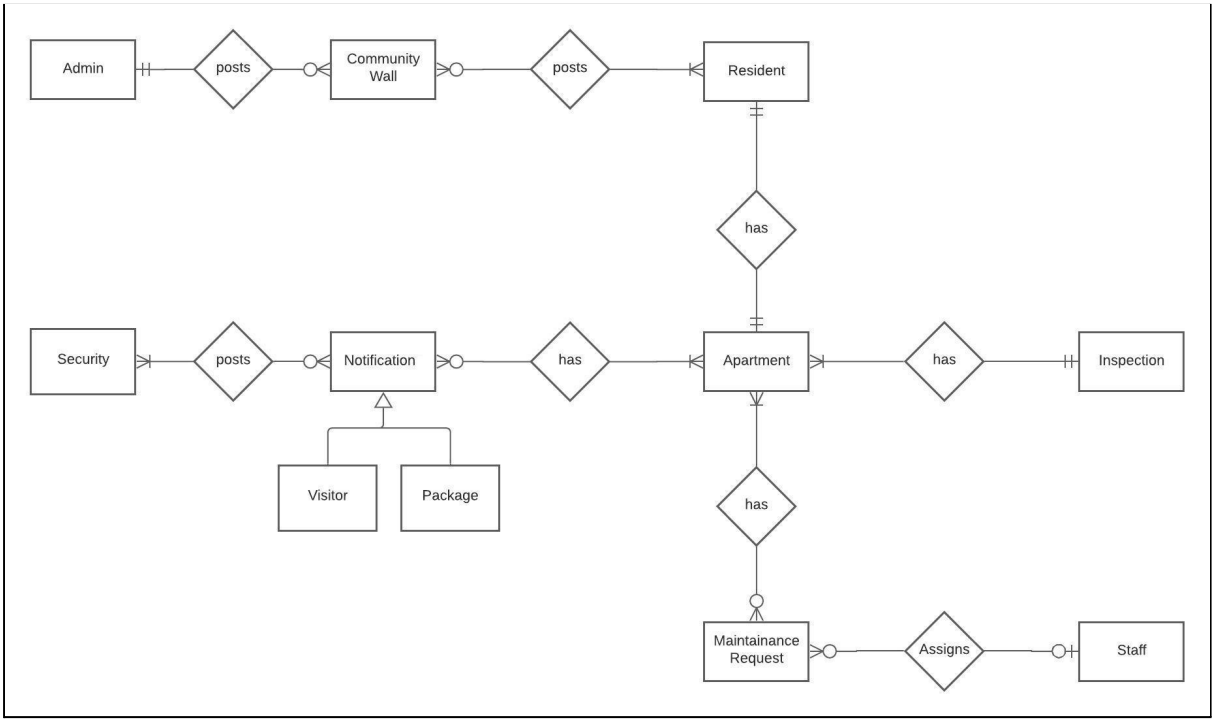
- **Maintenance Request**
  Residents can submit maintenance requests regarding leaks and water damages, appliance repairs, plumbing issues, pest control, etc. Maintenance staff will be assigned to work on these requests. Residents can also keep track of the status of the request with the portal.
- **Community Wall**
  Through the Community Wall, residents can post items for sale or rent, ask for recommendations, and organize events. Admin can also use this feature to send broadcast messages about events, specials or alerts to the entire community.
- **Notifications**
  Residents will be notified by message/email if their package has arrived or if they have a visitor at the society gate. Residents will get the notification when the package is picked up and can also approve/ reject the visitor entry through this portal.
- **Inspection Schedule**
  Admin can schedule half yearly or quarterly inspection for apartments. Residents will get a notification on the portal about details of the inspection visit like date and time, status and result.

ER Diagram

We have identified following entities for our database system:

Resident, Admin, Security, Staff, Apartment, Community Wall, Maintenance Request, Notification, Inspection. Out of these- Resident, Admin, Security, Staff are four types of users of the system. The ER diagram below describes all the entities in the system and their relationship with each other in terms of cardinality. The relationship can have one of the following cardinalities: zero or many, one or many, one and only one, zero or one.

ER Diagram for Apartment Manager



Database schema:

Database design consists of 7 tables. USER and APARTMENT tables are primary tables and NOTIFICATION, MAINTENANCE_REQUEST, MAINTENANCE_SERVICE, COMMUNITY_WALL and INSPECTION are related tables.

UID is a user key which represents one of the following types of users - Resident, Admin, Staff, Security. ApartmentId key represents apartment id – for example, A101 represents apartment number 101 of building A. NotificationId is an id representing notification for package or visitor. ReqId represents ids for the maintenance requests raised by residents.

PostId represents the id of a post by resident or admin. InspectionId represents the id of the scheduled/completed inspection for each apartment.

All primary keys are underlined. All Forign keys are in Italic

1. **USER** (UId VARCHAR(11), Name CHAR(40), Email VARCHAR(40), PhoneNumber INT(10), Type CHAR(20))
2. **APARTMENT** (ApartmentId VARCHAR(10), *UId* VARCHAR(11), BuildingName CHAR(20), Floor INT(3), Configuration VARCHAR(10), IsVacant BOOLEAN)
3. **NOTIFICATIONS** (NotificationId VARCHAR(10), *ApartmentId* VARCHAR(10), Timestamp DATETIME, NotificationText VARCHARr(100), Status VARCHAR(10), Type VARCHAR(10))
4. **MAINTENANCE_REQUEST** (ReqId VARCHAR(10), *UId* VARCHAR(11), *ApartmentId* VARCHAR(10), Date DATE ,Time TIME ,Type CHAR(10) , Description VARCHAR(60))
5. **MAINTENANCE_SERVICE** (ServiceId VARCHAR(10), *ReqId* VARCHAR(10), Status CHAR(10), Comment VARCHAR(50), Date DATE ,Time TIME, AssignedTo CHAR(30))
6. **COMMUNITY_WALL** (PostId VARCHAR(10), *UId* VARCHAR(11),Type CHAR(30), Message VARCHAR(100), Attachment BLOB, Date DATE, Time TIME)
7. **INSPECTION (**InspectionId VARCHAR(10), *ApartmentId* VARCHAR(10), InspectionDate DATE, InspectionResult CHAR(100), Status CHAR(20)**)**

Referential integrity constraints are:

1. UId in APARTMENT must exist in UId in USER
2. UId in MAINTENANCE_REQUEST must exist in UId in USER
3. UId in COMMUNITY_WALL must exist in UId in USER
4. ApartmentId in NOTIFICATIONS must exist in ApartmentId in APARTMENT
5. ApartmentId in MAINTENANCE_REQUEST must exist in ApartmentId in APARTMENT
6. ApartmentId in INSPECTION must exist in ApartmentId in APARTMENT
7. ReqId in MAINTENANCE_SERVICE must exist in ReqId in NOTIFICATIONS