# Residence Hall Management System

**Project Idea :**

We aim to design a residence hall management system. The main entities included in our application will be **Employees, Departments, Residents, Applicants, Rooms, Requests**, **Finance_Requests, Task_Requests** and **Dining_Hall_Services**. The Student Residence Management System is developed to facilitate application for accommodation online and to help the staff to manage the different residence activities such as handling application, payments, room allocation and completing tasks involving service requests. We aim to create a Web Front-End based application with basic authentication to ensure proper views are attributed to residents and employees.

**There will be several relationships as well in our application such as:**

- **Employees** *work in* the **Departments**.
- **Residents** *live in* the **Rooms**.
- **Residents** *raise* **Requests.**
- **Residents** *raise various* **Task Requests** which *are managed by* the Facilities **Department** and *monitored by* the **Employees** of the Facilities Department.
- **Residents** *raise various* **Finance Requests** which *are managed by* the Finance **Department** and *monitored by* the **Employees** of the Finance Department.
- **Admission Applications** *raised by* new **Applicants** *will be approved/rejected* by the Admissions Department.
- **Residents** *order food from* **Dining Hall Services**.

**Attributes:**

- Employees (empid, ssn, name, address, date_of_birth, gender, salary, designation)
- Departments (deptid, dept_name, budget)
- Rooms (room_number, room_type, room_cost)
- Resident (residentid, name, citizenship, passport_number, date_of_birth, gender, dining_hall_credit)
- Applicants (applicationid, name, citizenship, passport_number, date_of_birth, gender, room_preference, start_date, end_date)
- Requests (requestid, request_description, request_priority)
- Finance Requests (amount)
- Task Requests (category)
- Dining Hall Services (billid, item_name, bill_amount, category)

# Residence Hall Management System

**Constraints :**

1. Every **Resident** *has to occupy exactly one* **Room** and every **Room** *can be occupied by at most one* **Resident**.
2. Each **Employee** *works in exactly one* **Department.**
3. Every **Applicant** *will have only one application* in the **Applicants** table and it will be *approved by* exactly one **Department**.
4. Each **Request** which is *managed by* the **Department** is *monitored by* exactly one **Employee.**
5. Every **Dining Hall Service Bill** will be *associated with* exactly one **Resident.**

**Data Plan :**

We will request the residents and fellow students of our residence hall to enter the data into our application and also use fake data entered by us if necessary.

**Description of user interaction plans** :

The users of this application will fall under 2 broad categories. They can either be residents of the residence hall or they can be employees in the various departments in the residence hall. The residents (or potential residents) can perform the following operations through the User Interface :

1. Applicants can raise an application for admission to the residence hall and they become Residents when approved by the Admissions Department.
2. Residents can raise various Financial Requests with the Finance Department.
3. Residents can raise various Task Requests with the Facilities Department.
4. Residents can look at their current pending bill amount and pay it.
5. Residents can order food from the Dining Hall Services and also request a recharge of their dining hall credits through the Finance Office.

The employees can perform the following tasks through the User Interface :

1. The Admission Department can look at pending admission applications and approve them or reject them. If they approve the requests, they will also allocate room numbers.
2. Employees of the Finance Department will look at Financial Requests and approve them.
3. Employees of the Facilities Department will look at various service requests like Carpentry, Electricity etc. and update the status of these Task Requests.

**Contingency Plan :**

The web application and database schema will be scaled down to the main entities of employee, department, residents, room inventory and Task Request and their respective relationship sets.

# Residence Hall Management System

## SQL Schema:

```
CREATE TABLE Employees (
        empid INTEGER,
        ssn CHAR (11),
        name CHAR (100),
        address CHAR (200),
        date_of_birth DATE,
        gender CHAR(20),
        salary REAL,
        designation CHAR (50)
        PRIMARY KEY (empid)
)

CREATE TABLE Departments (
        deptid INTEGER,
        dept_name CHAR (30),
        budget REAL,
        PRIMARY KEY (deptid)
)

CREATE TABLE Residents (
        residentid INTEGER
        name CHAR (100),
        citizenship CHAR (30),
        passport_number CHAR (15),
        date_of_birth DATE,
        gender CHAR (10),
        dining_hall_credit REAL,
        PRIMARY KEY (resident_id),
        UNIQUE(citizenship, passport_number)
)

CREATE TABLE Dining_Hall_Services (
       billid INTEGER,
       item_name CHAR (100),
       bill_amount REAL,
       category CHAR (20),
       PRIMARY KEY (billid)
)
```

# Residence Hall Management System

```
CREATE TABLE Rooms (
        room_number INTEGER,
        room_type CHAR(30),
        room_cost REAL,
        PRIMARY KEY (room_number)
)

CREATE TABLE Applicants (
        applicationid INTEGER,
        name CHAR(100),
        citizenship CHAR(50),
        passport_number CHAR(20),
        date_of_birth DATE,
        gender CHAR(10),
        room_preference CHAR(10),
        start_date DATE,
        end_date DATE,
        PRIMARY KEY(application_id)
       UNIQUE(citizenship, passport_number)
)

CREATE TABLE Requests (
       requestid INTEGER,
       request_description CHAR (200),
       request_priority INTEGER,
       PRIMARY KEY (requestid)
)

CREATE TABLE Finance_Requests (
       requestid INTEGER,
       amount REAL,
       PRIMARY KEY (requestid),
       FOREIGN KEY (requestid) REFERENCES Requests ON DELETE CASCADE
)

CREATE TABLE Task_Requests (
       requestid INTEGER,
       category CHAR (50),
       PRIMARY KEY (requestid),
       FOREIGN KEY (requestid) REFERENCES Requests ON DELETE CASCADE
)
```

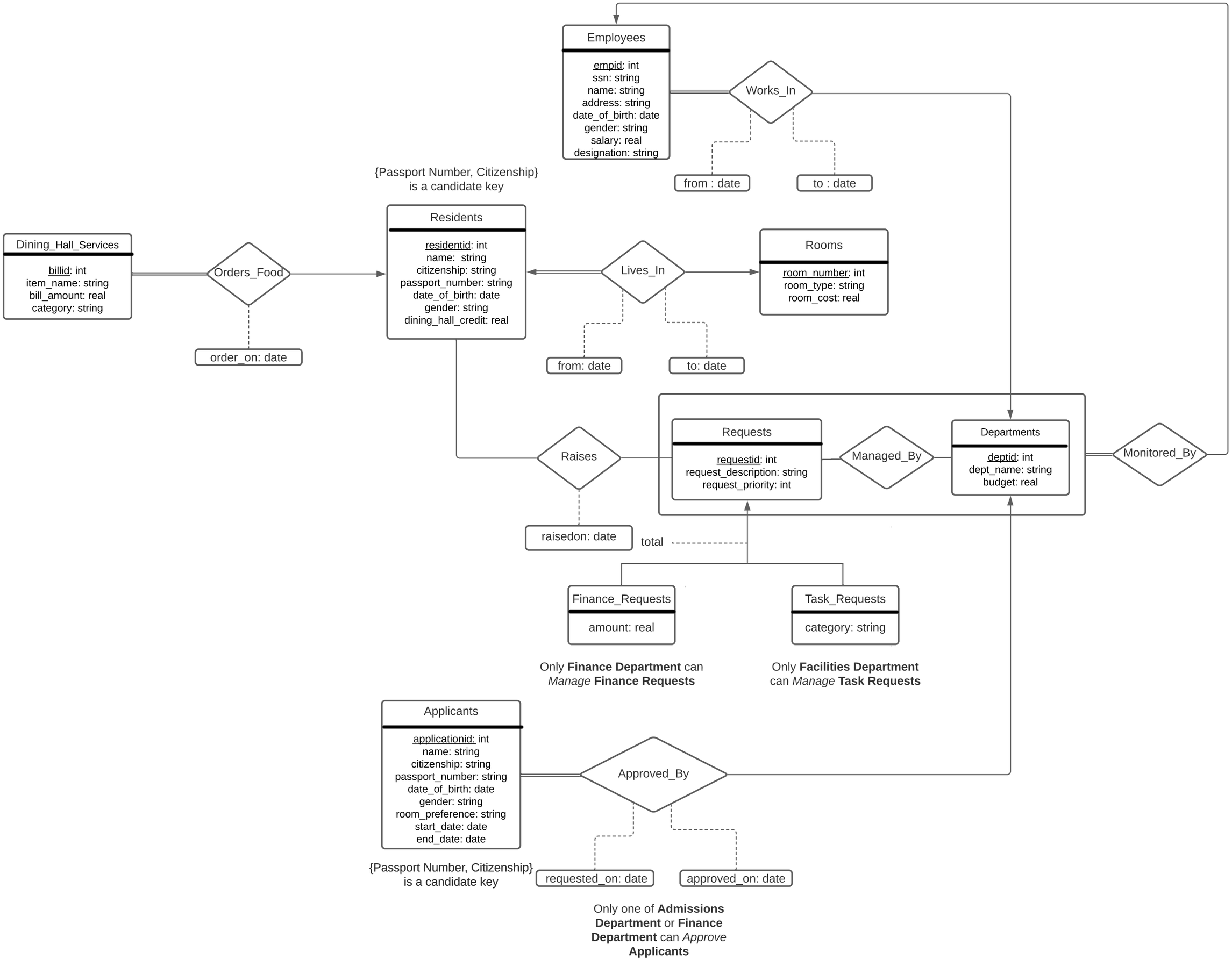
*Note: Referring to entities Requests, Finance_Requests and Task_Requests, every request will be either a finance request or task request.*

# Residence Hall Management System

```
CREATE TABLE Works_In (
      empid INTEGER,
      deptid INTEGER NOT NULL,
      from DATE,
      to DATE,
      PRIMARY KEY (empid),
      FOREIGN KEY (empid) REFERENCES Employees,
      FOREIGN KEY (deptid) REFERENCES Departments
)

CREATE  TABLE Lives_In (
      residentid INTEGER,
      room_number INTEGER NOT NULL,
      from DATE,
      to  DATE,
      PRIMARY KEY (residentid),
      UNIQUE (room_number),
      FOREIGN KEY (residentid) REFERENCES Residents,
      FOREIGN KEY (room_number) REFERENCES Rooms
)

CREATE TABLE Orders_Food (
      billid INTEGER,
      residentid INTEGER NOT NULL,
      order_on DATE,
      PRIMARY KEY (billid),
      FOREIGN KEY (billid) REFERENCES Dining_Hall_Services,
      FOREIGN KEY (residentid) REFERENCES Residents
)

CREATE TABLE Raises (
      residentid INTEGER,
      requestid INTEGER,
      raisedon DATE,
      PRIMARY KEY (residentid, requestid),
      FOREIGN KEY (residentid) REFERENCES Residents,
      FOREIGN KEY (requestid) REFERENCES Requests
)
```

# Residence Hall Management System

```
CREATE TABLE Approved_By (
      applicationid INTEGER,
      deptid INTEGER NOT NULL,
      approved_on DATE,
      requested_on DATE,
      PRIMARY KEY (applicationid),
      FOREIGN KEY (appicationid) REFERENCES Applicants,
      FOREIGN KEY (deptid) REFERENCES Departments
)

CREATE TABLE Managed_By (
      requestid INTEGER,
      deptid INTEGER,
      PRIMARY KEY (requestid, deptid),
      FOREIGN KEY (requestid) REFERENCES Requests,
      FOREIGN KEY (deptid) REFERENCES Departments,
)

CREATE TABLE Monitored_By (
     empid INTEGER NOT NULL,
     requestid INTEGER,
     deptid INTEGER,
     PRIMARY KEY (requestid, deptid),
     FOREIGN KEY (empid) REFERENCES Employees,
     FOREIGN KEY (requestid, deptid) REFERENCES Managed_By
)
```


Employees
empid: int
ssn: string
name: string
address: string
date_of_birth: date
gender: string
salary: real
designation: string
Works_In
from : date
to : date
{Passport Number, Citizenship}
is a candidate key
Residents
residentid: int
name: string
citizenship: string
passport_number: string
date_of_birth: date
gender: string
dining_hall_credit: real
Dining_Hall_Services
billid: int
item_name: string
bill_amount: real
category: string
Orders_Food
order_on: date
Lives_In
from: date
to: date
Rooms
room_number: int
room_type: string
room_cost: real
Raises
raisedon: date
Requests
requestid: int
request_description: string
request_priority: int
Managed_By
Departments
deptid: int
dept_name: string
budget: real
Monitored_By
total
Finance_Requests
amount: real
Task_Requests
category: string
Only Finance Department can Manage Finance Requests
Only Facilities Department can Manage Task Requests
Applicants
applicationid: int
name: string
citizenship: string
passport_number: string
date_of_birth: date
gender: string
room_preference: string
start_date: date
end_date: date
Approved_By
{Passport Number, Citizenship}
is a candidate key
requested_on: date
approved_on: date
Only one of Admissions Department or Finance Department can Approve Applicants