Property Management Project

General Requirements:

Java Version Compatibility: The project should be compatible with a specified version of Java (e.g., Java 8, Java 11).
Database: Utilize a relational database system (e.g., MS SQL SERVER) to store and manage data.
Connection Pooling: Implement connection pooling for efficient database connections.
Exception Handling: Handle exceptions gracefully and provide meaningful error messages.
Security: Implement security measures, such as parameterized queries, to prevent SQL injection attacks.

Functionality:

BaseModel:

- Provide a base model class (BaseModel) for common functionalities like database connection handling.
- Ensure that subclasses (e.g., Employee, Owner, OwnerShip, Property, SalesOffice) extend this base model.

Employee:

- CRUD Operations:
    - Create new employee records.
    - Update employee information.
    - Delete an employee record.
    - Retrieve a list of employees.

- *isManager* Functionality:
    - Determine if an employee is a manager based on the managerId field.

Owner:

- CRUD Operations:
    - Create new owner records.
    - Update owner information.
    - Delete an owner record.

    - Retrieve a list of owners.

OwnerShip:

- CRUD Operations:
    - Create new ownership records.
    - Update ownership information.
    - Delete an ownership record.
    - Retrieve a list of ownership records.

Property:

- CRUD Operations:
    - Create new property records.
    - Update property information.
    - Delete a property record.
    - Retrieve a list of properties.

SalesOffice:

- CRUD Operations:
    - Create new sales office records.
    - Update sales office information.
    - Delete a sales office record.
    - Retrieve a list of sales offices.

Data Retrieval:

- Implement methods to retrieve data from the database, including lists of entities and specific records.
- Ensure proper handling of result sets.

Column Metadata:

- Implement methods to retrieve column names for each entity table.
- Use this information dynamically in operations such as data retrieval and presentation.

Connection Management:

- Establish and manage database connections efficiently.
- Implement proper connection pooling to optimize resource usage.

Logging:

- Implement logging for crucial operations and exceptions.

- Log information that aids in debugging and monitoring.

Unit Testing:

- Develop unit tests to verify the correctness of key functionalities.
- Include tests for each entity's CRUD operations.

Documentation:

- Provide clear and comprehensive code documentation.
- Include a README file explaining how to set up and run the project.

Code Structure:

- Follow a modular and maintainable code structure.
- Consider using design patterns where applicable.

Error Handling:

- Implement appropriate error handling and reporting mechanisms.
- Provide user-friendly error messages.

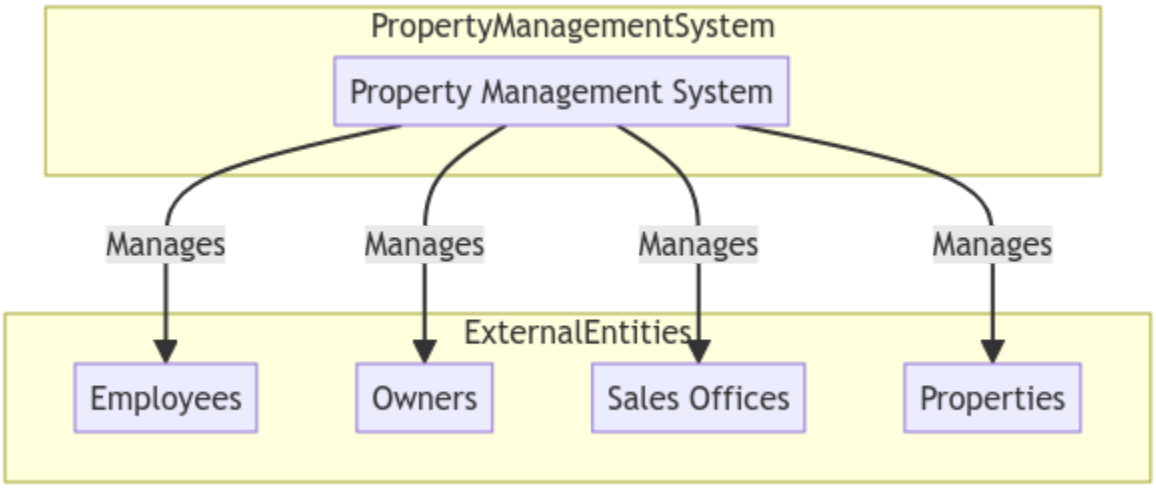
Security Measures:

- Ensure secure handling of sensitive information, such as passwords.
- Implement secure database access methods.

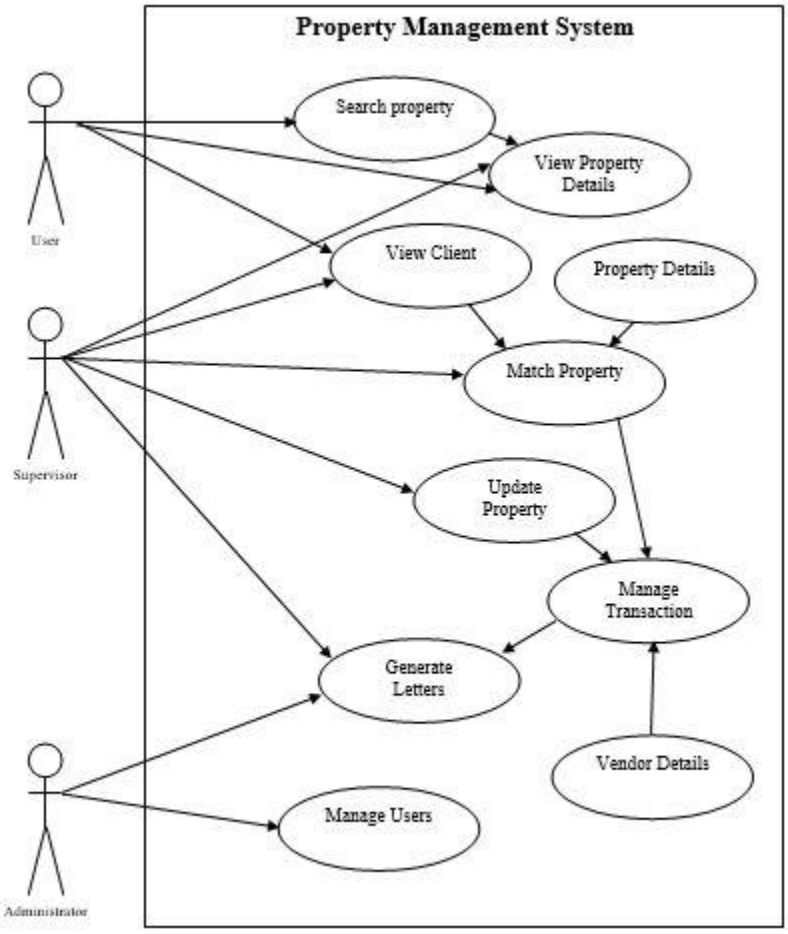
User Interface:

- Integrate the backend with a user interface for ease of interaction.
- Implement functionalities for user input validation.

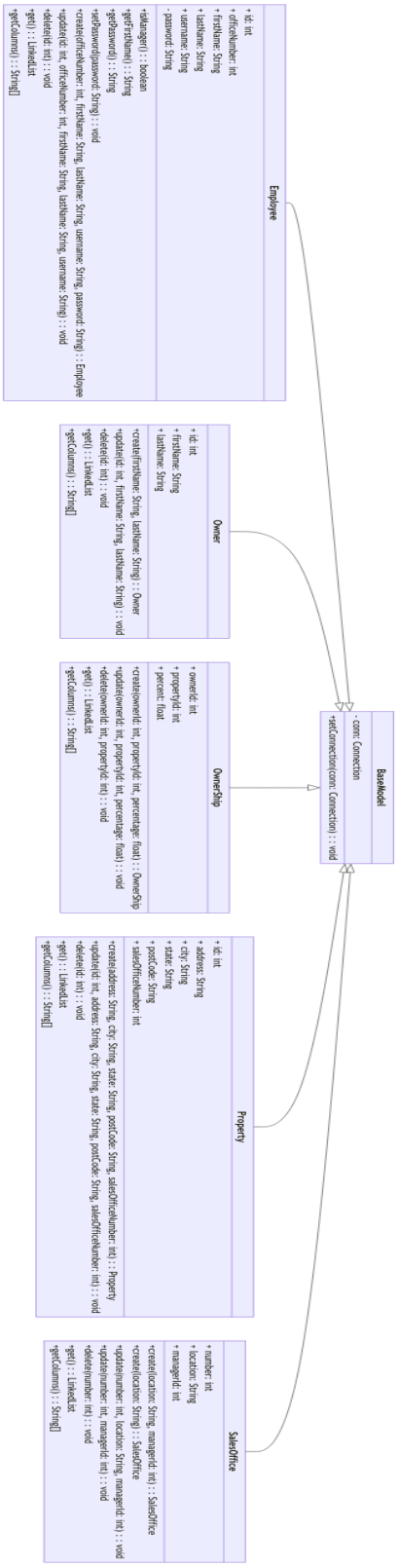
Context Diagram



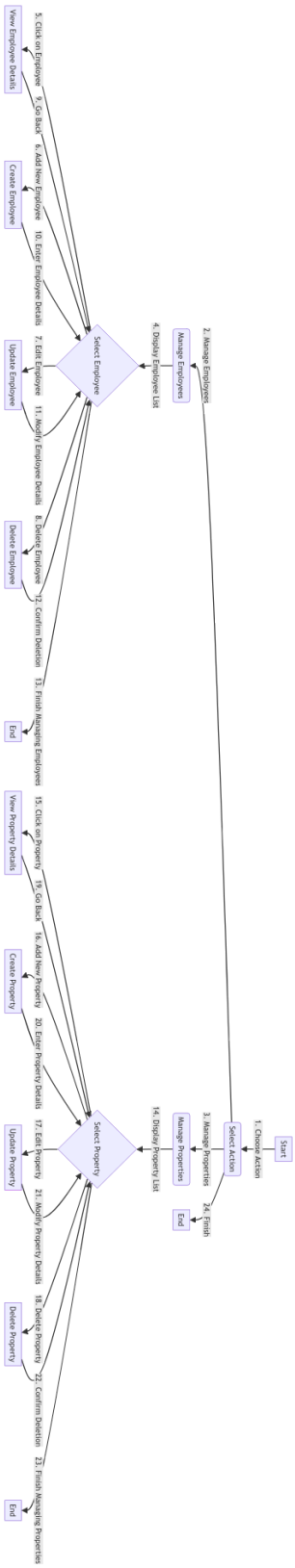
Use Case Diagram



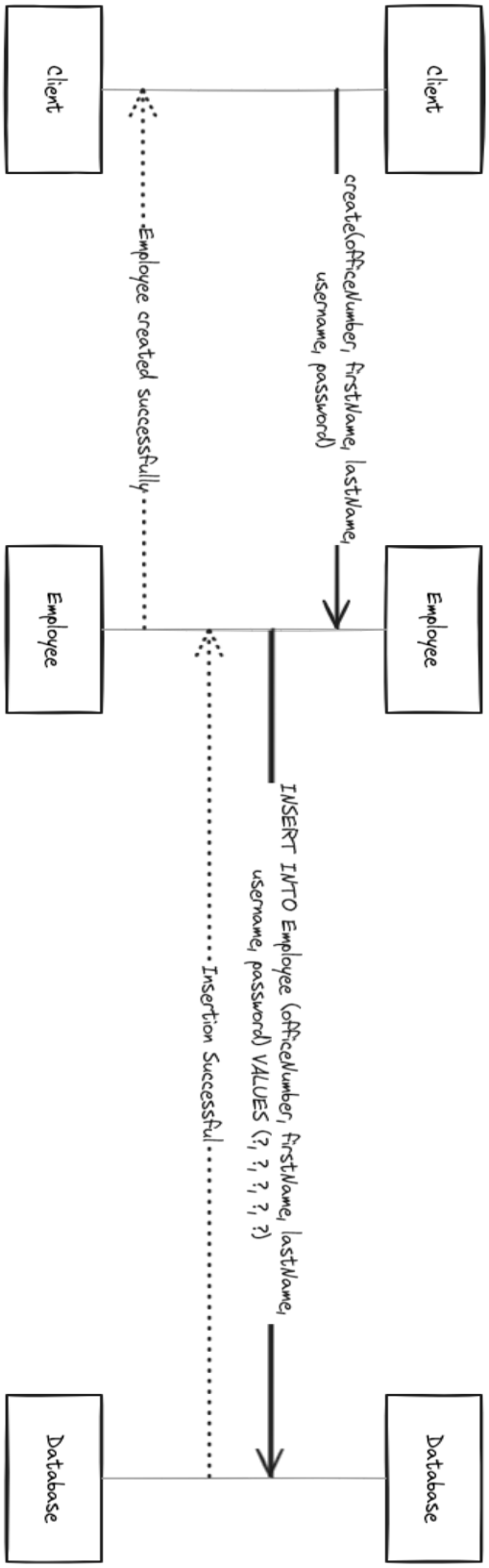
Class Diagram



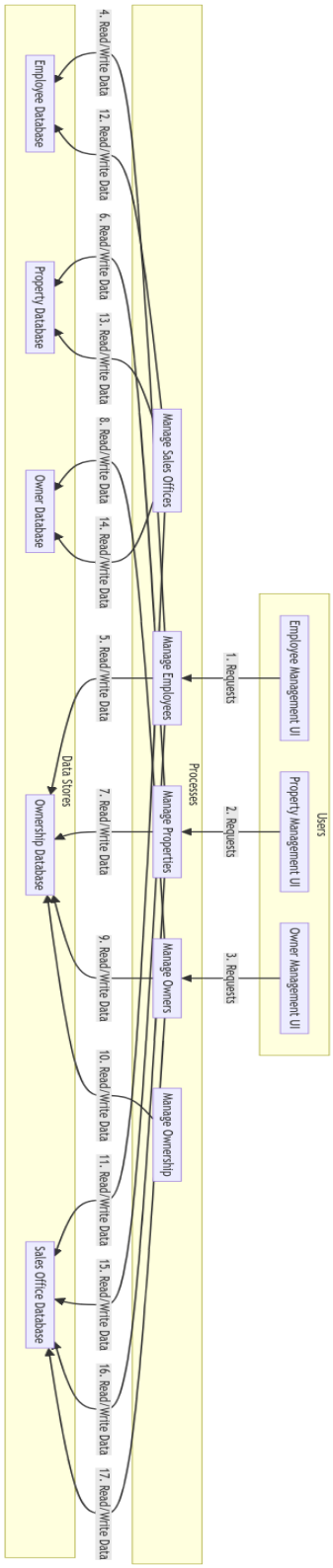
Activity Diagram



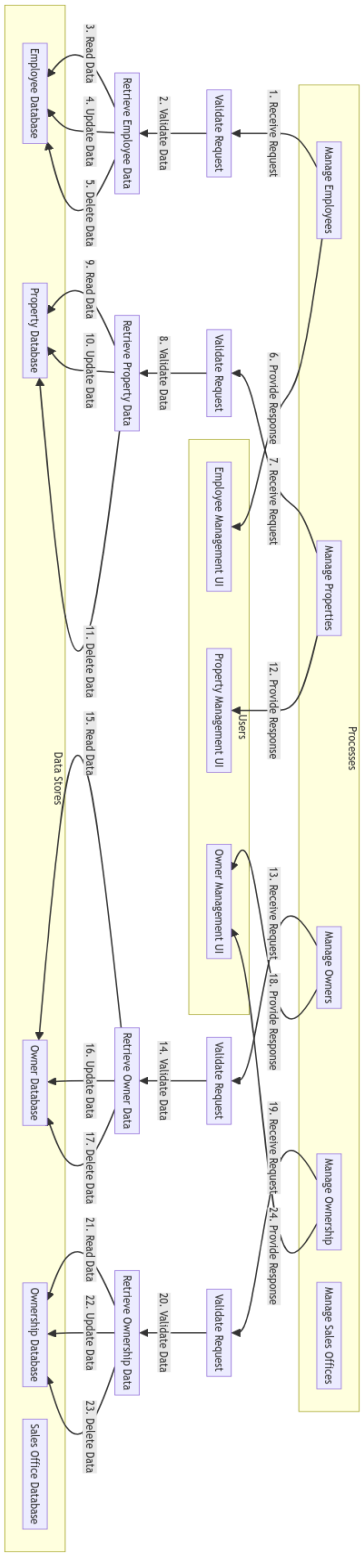
Sequence Diagram



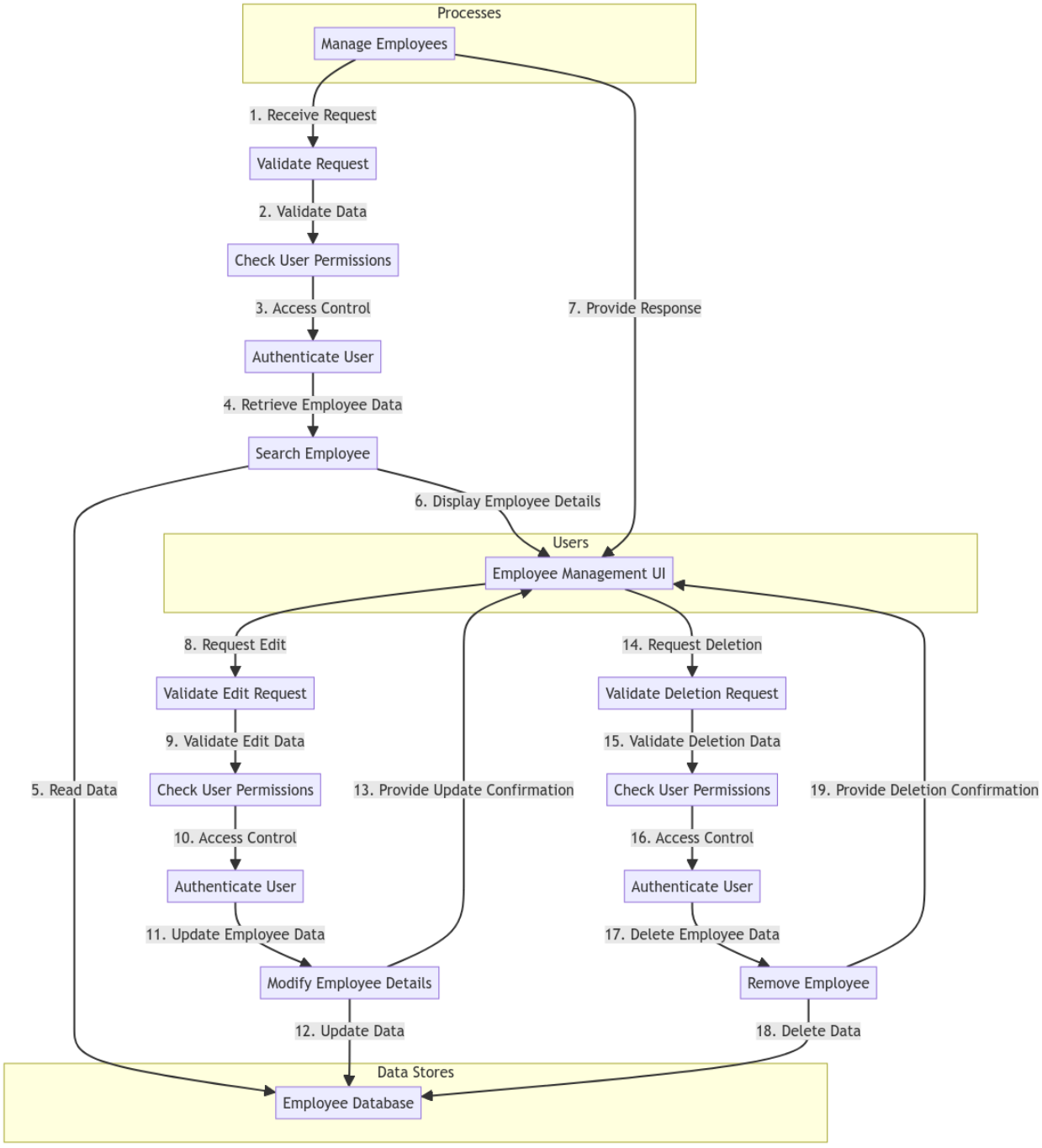
DFD Level 1


Users
Employee Management UI
Property Management UI
Owner Management UI
1. Requests
2. Requests
3. Requests
Processes
Manage Employees
Manage Properties
Manage Owners
Manage Sales Offices
Manage Ownership
Data Stores
Employee Database
Property Database
Owner Database
Ownership Database
Sales Office Database
4. Read/Write Data
5. Read/Write Data
6. Read/Write Data
7. Read/Write Data
8. Read/Write Data
9. Read/Write Data
10. Read/Write Data
11. Read/Write Data
12. Read/Write Data
13. Read/Write Data
14. Read/Write Data
15. Read/Write Data
16. Read/Write Data
17. Read/Write Data

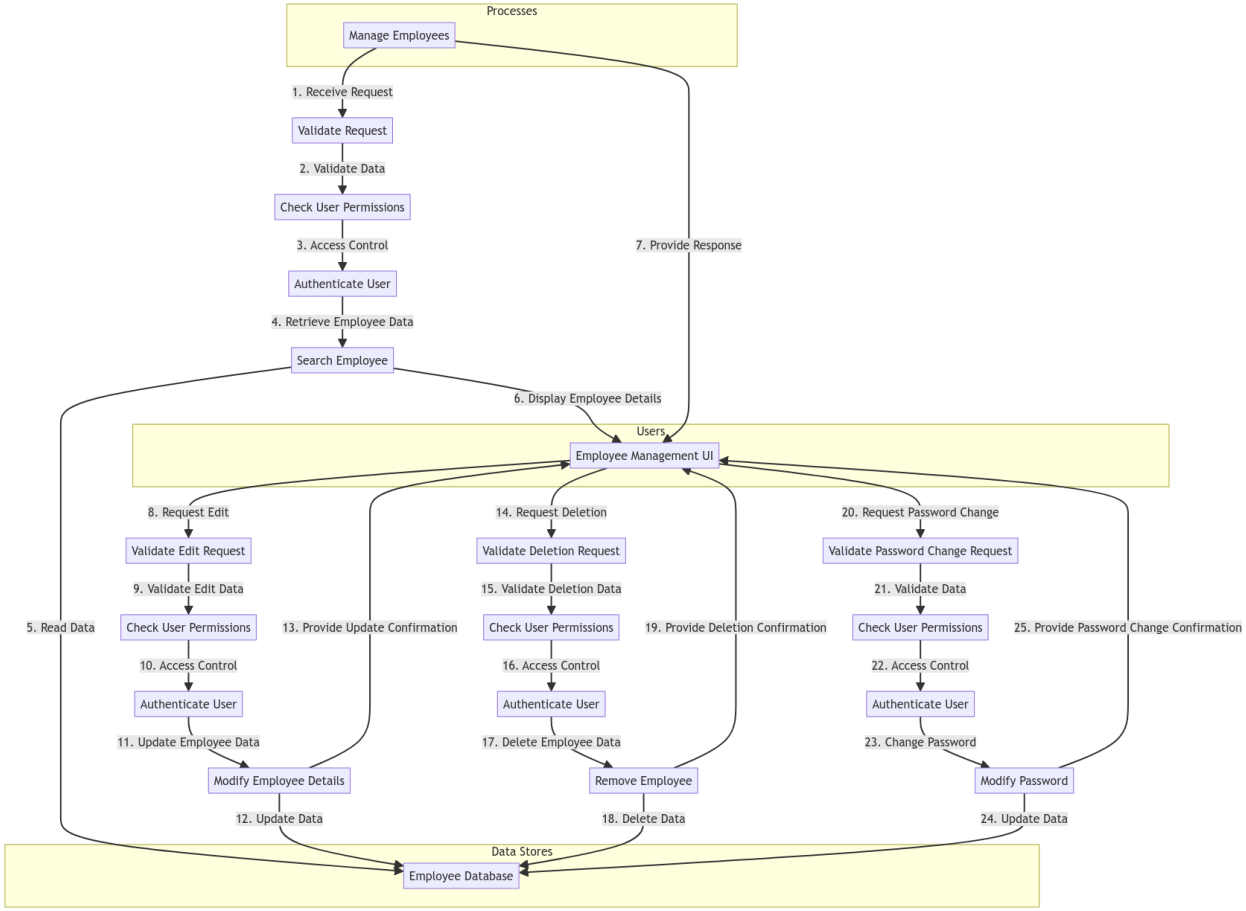
DFD Level 2


Processes
Manage Employees
Manage Properties
Manage Owners
Manage Ownership
Manage Sales Offices
Users
Employee Management UI
Property Management UI
Owner Management UI
Data Stores
Employee Database
Property Database
Owner Database
Ownership Database
Sales Office Database
Validate Request
Retrieve Employee Data
Retrieve Property Data
Retrieve Owner Data
Retrieve Ownership Data
1. Receive Request
2. Validate Data
3. Read Data
4. Update Data
5. Delete Data
6. Provide Response
7. Receive Request
8. Validate Data
9. Read Data
10. Update Data
11. Delete Data
12. Provide Response
13. Receive Request
14. Validate Data
15. Read Data
16. Update Data
17. Delete Data
18. Provide Response
19. Receive Request
20. Validate Data
21. Read Data
22. Update Data
23. Delete Data
24. Provide Response

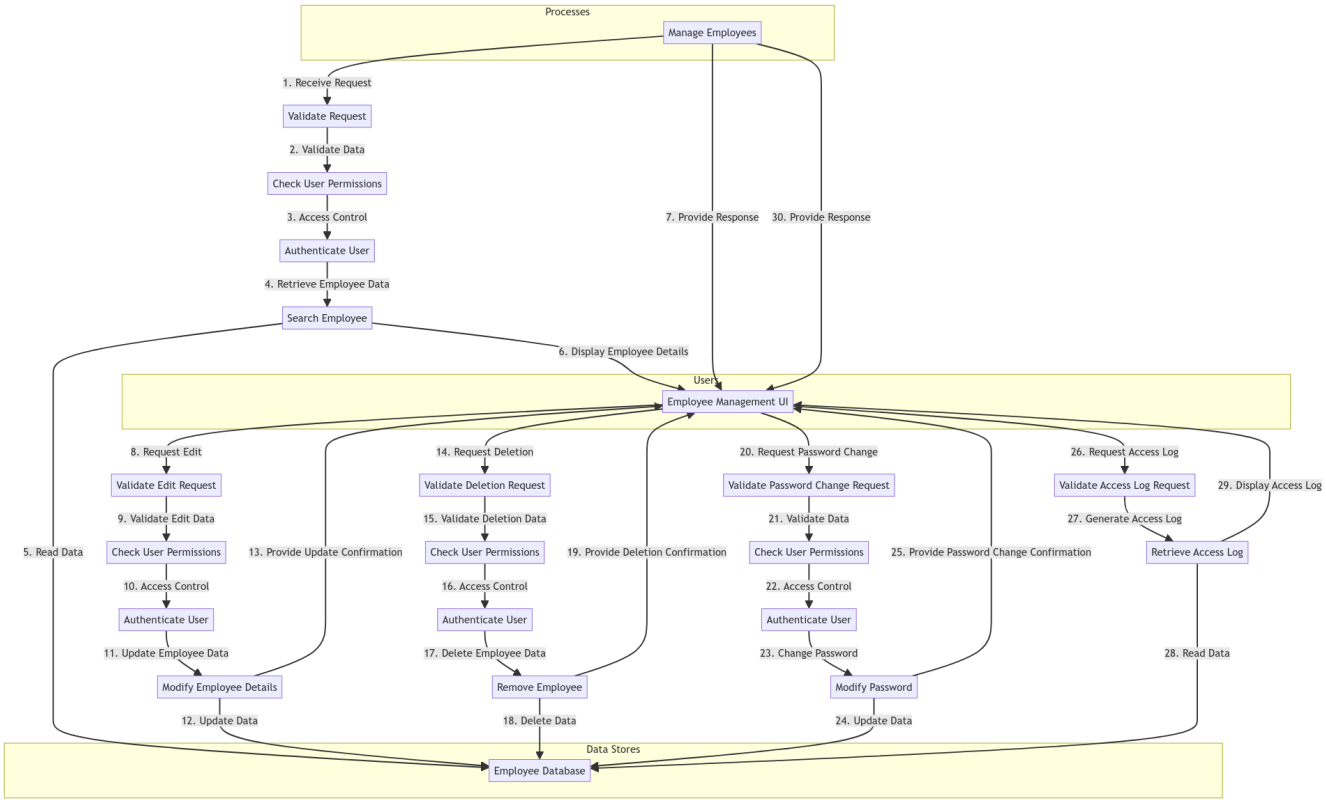
DFD Level 3


Processes
Manage Employees
1. Receive Request
Validate Request
2. Validate Data
Check User Permissions
3. Access Control
Authenticate User
4. Retrieve Employee Data
Search Employee
5. Read Data
6. Display Employee Details
7. Provide Response
Users
Employee Management UI
8. Request Edit
Validate Edit Request
9. Validate Edit Data
Check User Permissions
10. Access Control
Authenticate User
11. Update Employee Data
Modify Employee Details
12. Update Data
13. Provide Update Confirmation
14. Request Deletion
Validate Deletion Request
15. Validate Deletion Data
Check User Permissions
16. Access Control
Authenticate User
17. Delete Employee Data
Remove Employee
18. Delete Data
19. Provide Deletion Confirmation
Data Stores
Employee Database

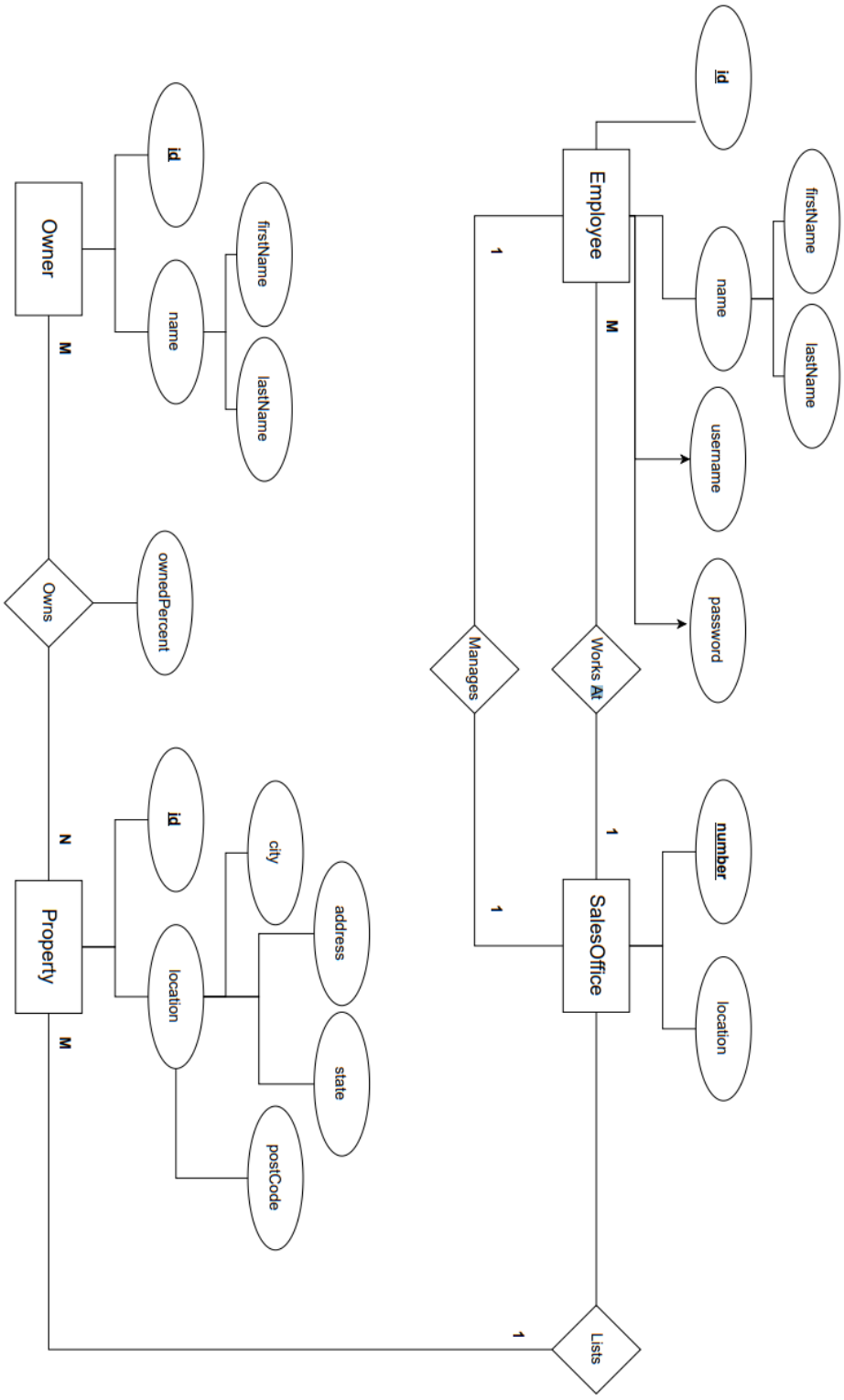
DFD Level 4


Processes
Manage Employees
1. Receive Request
Validate Request
2. Validate Data
Check User Permissions
3. Access Control
Authenticate User
4. Retrieve Employee Data
Search Employee
7. Provide Response
6. Display Employee Details
Users
Employee Management UI
8. Request Edit
Validate Edit Request
9. Validate Edit Data
Check User Permissions
10. Access Control
Authenticate User
11. Update Employee Data
Modify Employee Details
12. Update Data
5. Read Data
13. Provide Update Confirmation
14. Request Deletion
Validate Deletion Request
15. Validate Deletion Data
Check User Permissions
16. Access Control
Authenticate User
17. Delete Employee Data
Remove Employee
18. Delete Data
19. Provide Deletion Confirmation
20. Request Password Change
Validate Password Change Request
21. Validate Data
Check User Permissions
22. Access Control
Authenticate User
23. Change Password
Modify Password
24. Update Data
25. Provide Password Change Confirmation
Data Stores
Employee Database

DFD Level 5


Processes
Manage Employees
1. Receive Request
Validate Request
2. Validate Data
Check User Permissions
3. Access Control
Authenticate User
4. Retrieve Employee Data
Search Employee
7. Provide Response
30. Provide Response
6. Display Employee Details
Users
Employee Management UI
8. Request Edit
Validate Edit Request
9. Validate Edit Data
Check User Permissions
10. Access Control
Authenticate User
11. Update Employee Data
Modify Employee Details
12. Update Data
5. Read Data
13. Provide Update Confirmation
14. Request Deletion
Validate Deletion Request
15. Validate Deletion Data
Check User Permissions
16. Access Control
Authenticate User
17. Delete Employee Data
Remove Employee
18. Delete Data
19. Provide Deletion Confirmation
20. Request Password Change
Validate Password Change Request
21. Validate Data
Check User Permissions
22. Access Control
Authenticate User
23. Change Password
Modify Password
24. Update Data
25. Provide Password Change Confirmation
26. Request Access Log
Validate Access Log Request
27. Generate Access Log
Retrieve Access Log
28. Read Data
29. Display Access Log
Data Stores
Employee Database

ERD


id
firstName
name
lastName
Employee
username
password
1
M
Manages
Works At
1
1
number
SalesOffice
location
Lists
1
M
Owner
id
name
firstName
lastName
Owns
ownedPercent
N
Property
M
id
location
city
address
state
postCode