FACILITY MANAGEMENT SERVICES SYSTEM

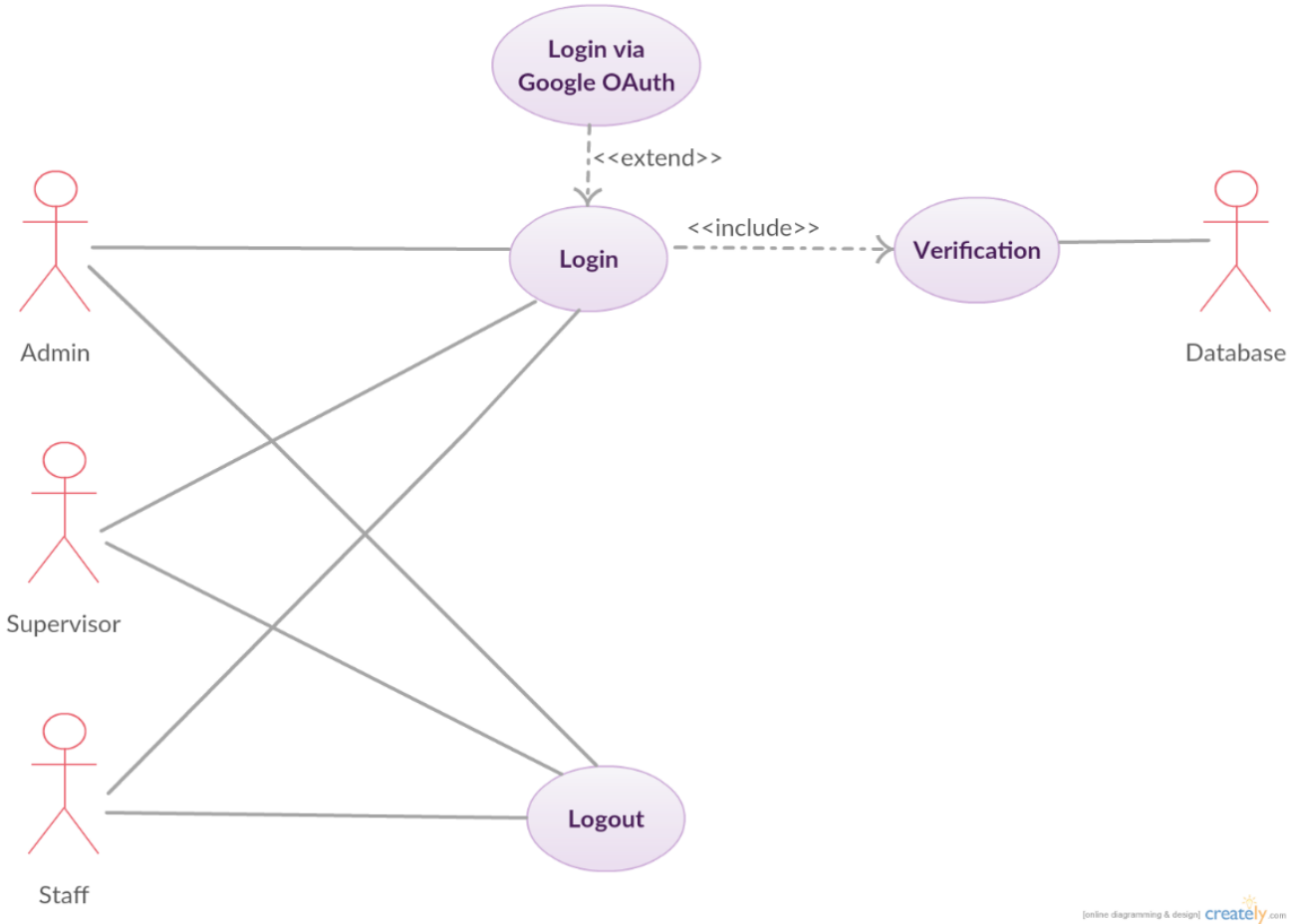
Anannya Uberoi (2015014)

Suril Mehta (2015104)

Design Analysis

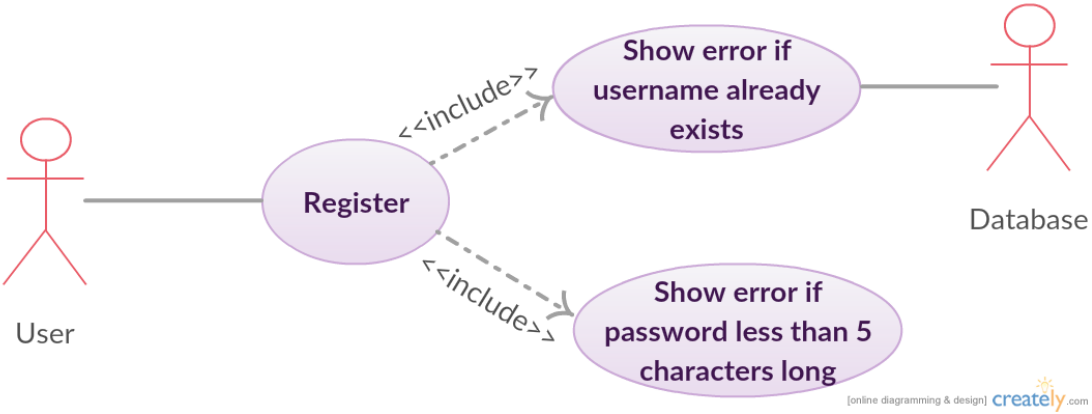
Use Case Diagrams:

1. Login Functionality



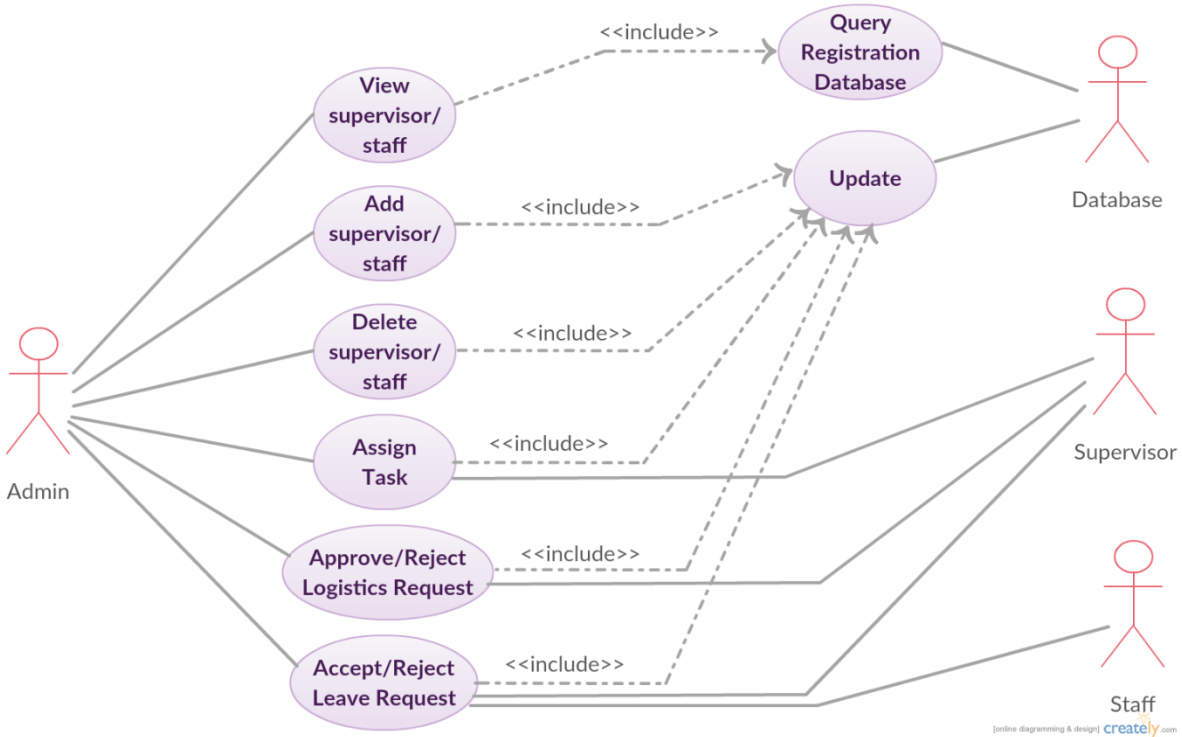
Login functionality allows admin, supervisor and staff to login into or logout from the system. Google OAuth is also a kind of login, hence the “extends” relationship. Login includes the functionality of verification by comparing with the database.

2. Registration Functionality



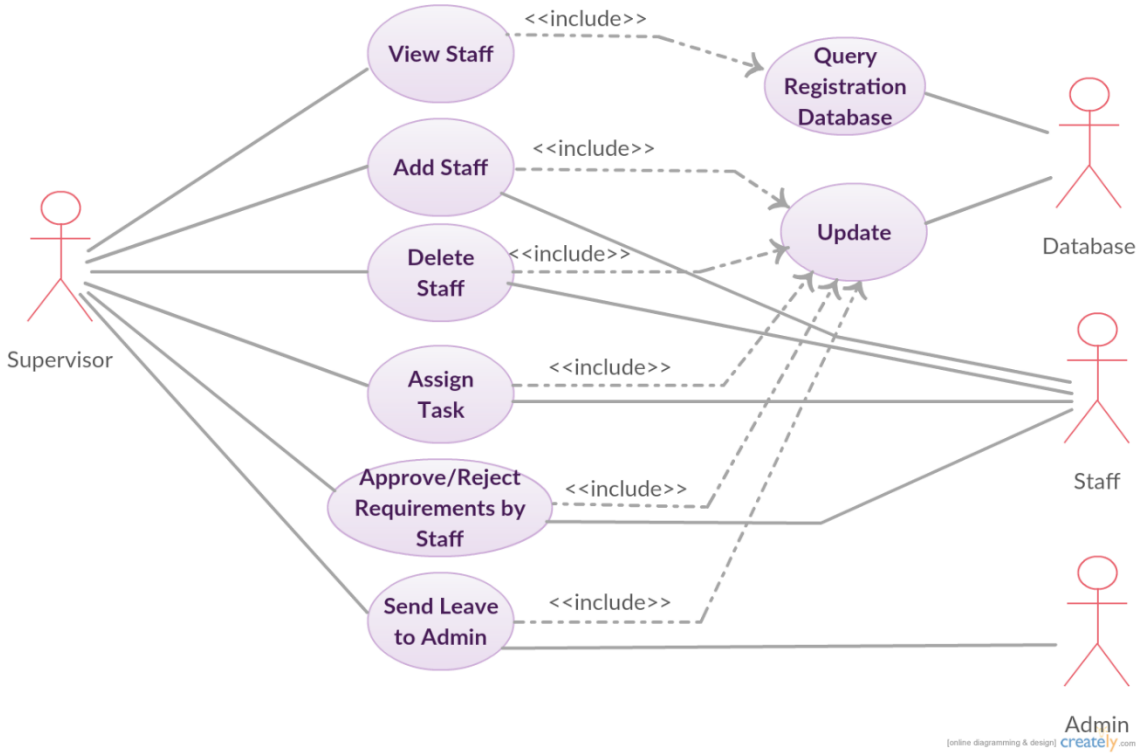
Every user gets a form to fill upon clicking “Register” in the UI. Registration includes two functionalities which should be fulfilled before making the user applicable to the database. These are that the username should be unique and the password should be at least characters long.

3. Functionalities of the Admin



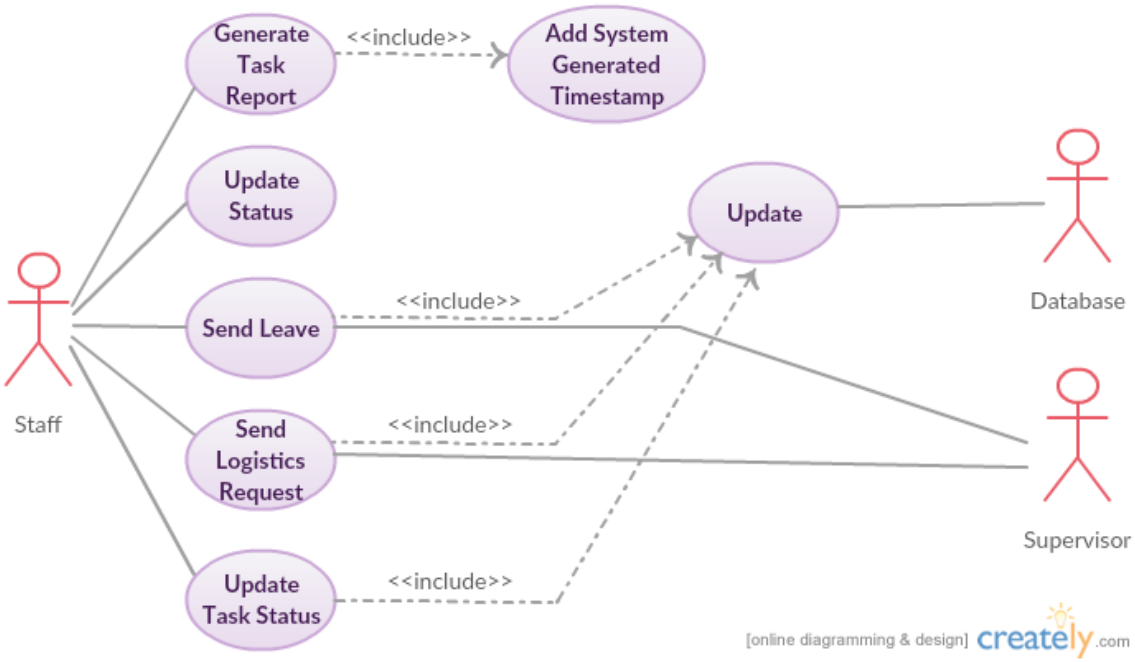
This diagram illustrates the functionalities of the administrator, which include viewing, adding, and deleting staff and supervisors (since everything visible to the staff and supervisors is also visible to the admin). Also, it can assign tasks to the supervisor, approve/reject logistics, and accept/reject leave requests of staff and supervisors. All these include updating of database.

4. Functionalities of the Supervisors



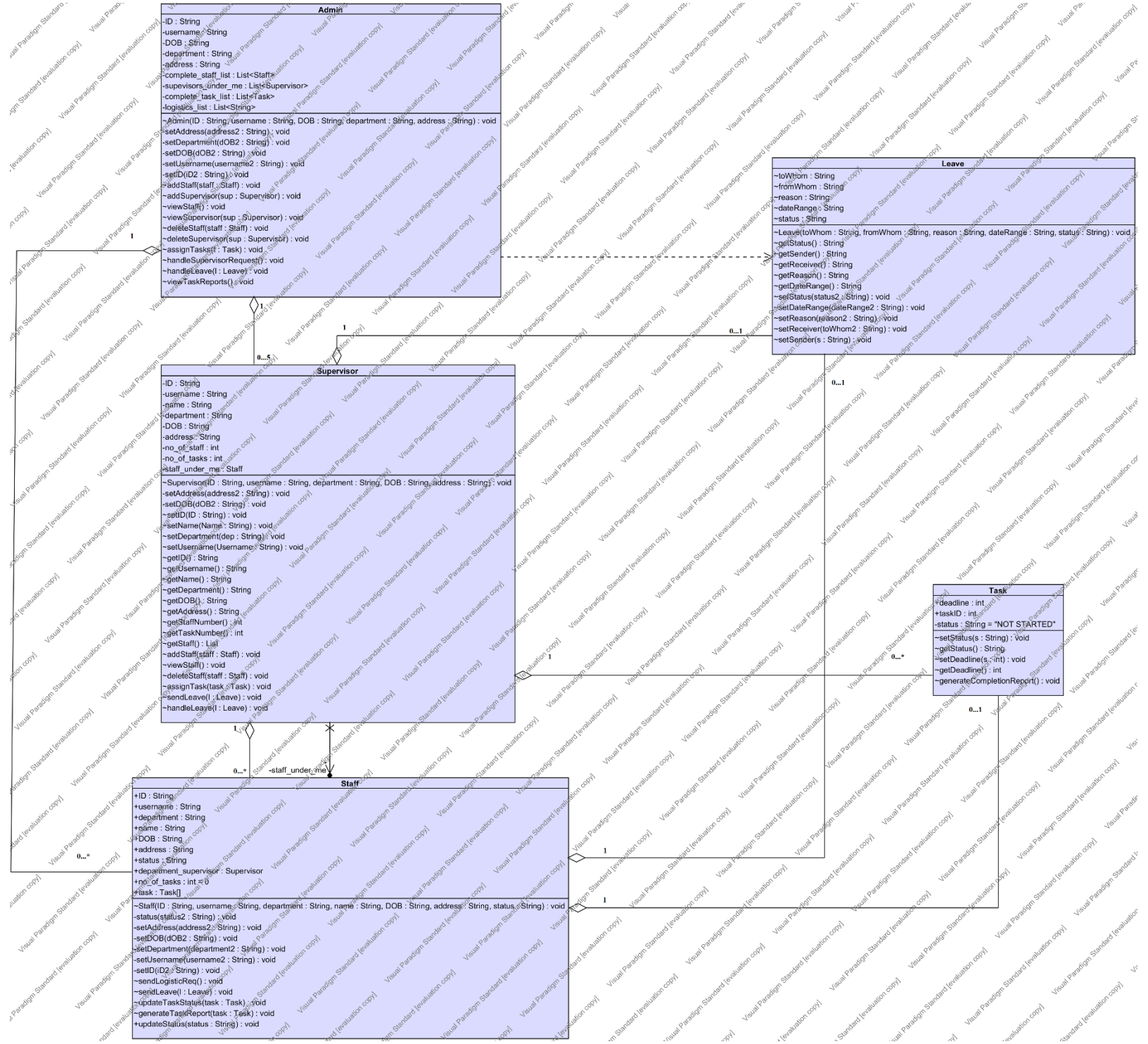
This diagram has the use cases for supervisor- viewing, adding and deleting staff, assigning tasks to staff and handling logistics requests and leave requests of staff. Database is also updated in these processes. It can also send leave to the admin.

5. Functionalities of Staff



The diagram shows the functionalities of the staff, such as generating task reports, updating status (available/busy), sending leave requests to supervisor, sending logistics request to supervisor and updating task status.

Class Diagram:


Admin
-ID : String
-username : String
-DOB : String
-department : String
-address : String
-complete_staff_list : List<Staff>
-supevisors_under_me : List<Supervisor>
-complete_task_list : List<Task>
-logistics_list : List<String>
~Admin(ID : String, username : String, DOB : String, department : String, address : String) : void
-setAddress(address2 : String) : void
-setDepartment(dOB2 : String) : void
-setDOB(dOB2 : String) : void
-setUsername(username2 : String) : void
-setID(iD2 : String) : void
~addStaff(staff : Staff) : void
~addSupervisor(sup : Supervisor) : void
~viewStaff() : void
~viewSupervisor(sup : Supervisor) : void
~deleteStaff(staff : Staff) : void
~deleteSupervisor(sup : Supervisor) : void
~assignTasks(t : Task) : void
~handleSupervisorRequest() : void
~handleLeave(l : Leave) : void
~viewTaskReports() : void
Leave
~toWhom : String
~fromWhom : String
~reason : String
~dateRange : String
~status : String
~Leave(toWhom : String, fromWhom : String, reason : String, dateRange : String, status : String) : void
~getStatus() : String
~getSender() : String
~getReceiver() : String
~getReason() : String
~getDateRange() : String
~setStatus(status2 : String) : void
~setDateRange(dateRange2 : String) : void
~setReason(reason2 : String) : void
~setReceiver(toWhom2 : String) : void
~setSender(s : String) : void
Supervisor
-ID : String
-username : String
-name : String
-department : String
-DOB : String
-address : String
-no_of_staff : int
-no_of_tasks : int
-staff_under_me : Staff
~Supervisor(ID : String, username : String, department : String, DOB : String, address : String) : void
-setAddress(address2 : String) : void
-setDOB(dOB2 : String) : void
-setID(ID : String) : void
~setName(Name : String) : void
~setDepartment(dep : String) : void
~setUsername(Username : String) : void
~getID() : String
~getUsername() : String
~getName() : String
~getDepartment() : String
~getDOB() : String
~getAddress() : String
~getStaffNumber() : int
~getTaskNumber() : int
~getStaff() : List
~addStaff(staff : Staff) : void
~viewStaff() : void
~deleteStaff(staff : Staff) : void
~assignTask(task : Task) : void
~sendLeave(l : Leave) : void
~handleLeave(l : Leave) : void
Task
+deadline : int
+taskID : int
-status : String = "NOT STARTED"
~setStatus(s : String) : void
~getStatus() : String
~setDeadline(s : int) : void
~getDeadline() : int
~generateCompletionReport() : void
Staff
+ID : String
+username : String
+department : String
+name : String
+DOB : String
+address : String
+status : String
+department_supervisor : Supervisor
+no_of_tasks : int = 0
+task : Task[]
~Staff(ID : String, username : String, department : String, name : String, DOB : String, address : String, status : String) : void
-status(status2 : String) : void
-setAddress(address2 : String) : void
-setDOB(dOB2 : String) : void
-setDepartment(department2 : String) : void
-setUsername(username2 : String) : void
-setID(iD2 : String) : void
~sendLogisticReq() : void
~sendLeave(l : Leave) : void
~updateTaskStatus(task : Task) : void
~generateTaskReport(task : Task) : void
+updateStatus(status : String) : void
-staff_under_me
1
0...5
0...1
0...*

Admin Class:

```
private String ID, username,DOB,department,address;
private List<Staff> complete_staff_list;
private List<Supervisor> supevisors_under_me;
private List<Task> complete_task_list;
private List<String> logistics_list;
```

- Setters for admin---initialises all the fields of the admin
- void addStaff(Staff staff)---add a new member in the staff
- void addSupervisor(Supervisor sup)---adds the supervisor in any of the five departments
- void viewStaff()---views the complete staff with information of each of them
- void viewSupervisor()---views 5 supervisors and their respective departments
- void deleteStaff(Staff staff)---can delete any staff member
- void deleteSupervisor(Supervisor sup)---can delete any supervisor
- void assignTasks(Task t)---Assign task to supervisor and staff
- void handleSupervisorRequest()---Accepts or rejects requests for inventory
- void viewTaskReports()---views tasks assigned to each person

Supervisor Class:

```
private String ID,username,name,department,DOB,address;
private List<Staff> staff_under_me;
private int no_of_staff,no_of_tasks;
```

- Getters and setters for supervisor---initialises all the fields and returns field data when required
- void addStaff()---add a new member in the staff
- void viewStaff()---views the complete staff with information of each of them
- void deleteStaff()---can delete any staff member
- void assignTasks()---Assign task to staff
- void sendleave()---can send leave to supervisor
- void handleleave()---can accept and reject leaves sent by staff
- void sendrequest()---sends requests to admin for inventory.

Staff Class:

```
public String ID, username,department,name,DOB,address,status;
public Supervisor department_supervisor;
public int no_of_tasks=0;
public Task task[];
```

- Getters and setters for all the fields of the staff---sets the value for all the fields and returns the data of the required field when needed
- void updatestatus(String status2)---updates the status of the staff member
- void sendLogisticReq()---send logistic requirement to supervisor/admin
- void sendLeave(Leave l)---requests for a leave to supervisor/admin
- void generateTaskReport(Task task)---generates a task report for each staff member with respect to their department

Leave Class:

```
String toWhom,fromWhom,reason,dateRange,status;
```

- Getters and setters for all the fields

Task Class:

```
public int deadline,taskID;
private String status="NOT STARTED";
```

- Getters and setters for different variables.
- void GenerateCompletionReport()---generates completion report when task status is complete