Project Report

FMS

Gurek Singh 2015033
Abhishek Chauhan 2015005

Design Analysis

We will be creating 3 types of view classes for GM Admin, Department Supervisor and Staff and 1 model class User.
LoggedInUser is a single object thus implying Singleton design pattern.

Admin controls the system and are the head of the other members of the system. It has the control of the whole functionality of the system. He can add, view, delete data from the system and any change in the system has to be approved by the admin.
Department Supervisors are the head of their respective Departments.
The staff member are allotted tasks which are issued by the supervisors/admin.

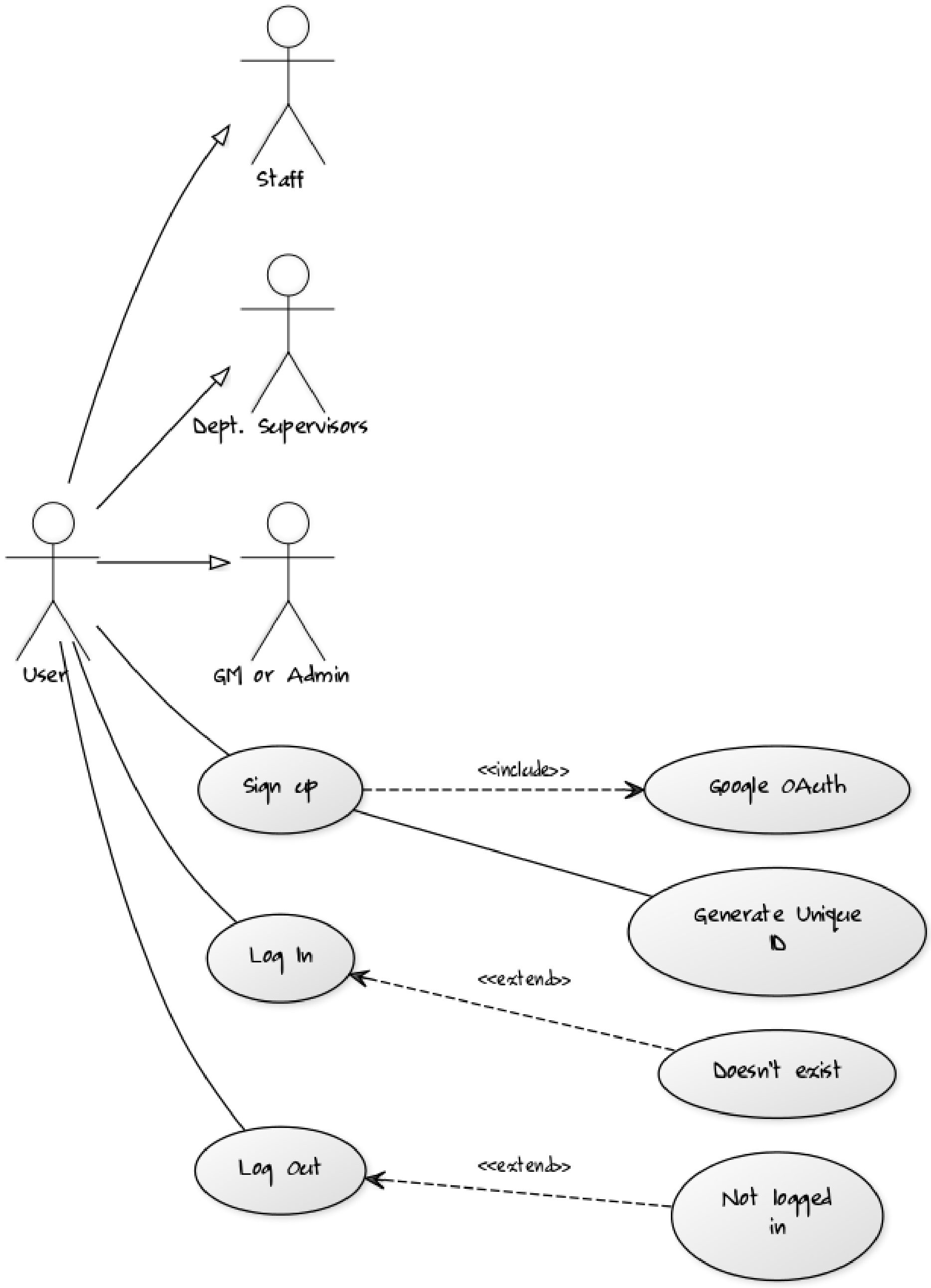
Individual Contributions

- **Gurek-** GUI, Controller functions, Users/Logisitics/Task Logic.

- **Abhishek-** Register, Model Structures, Project Report.

- **Both-** Database(Reading/Writing files), Other functionalities, Debugging.

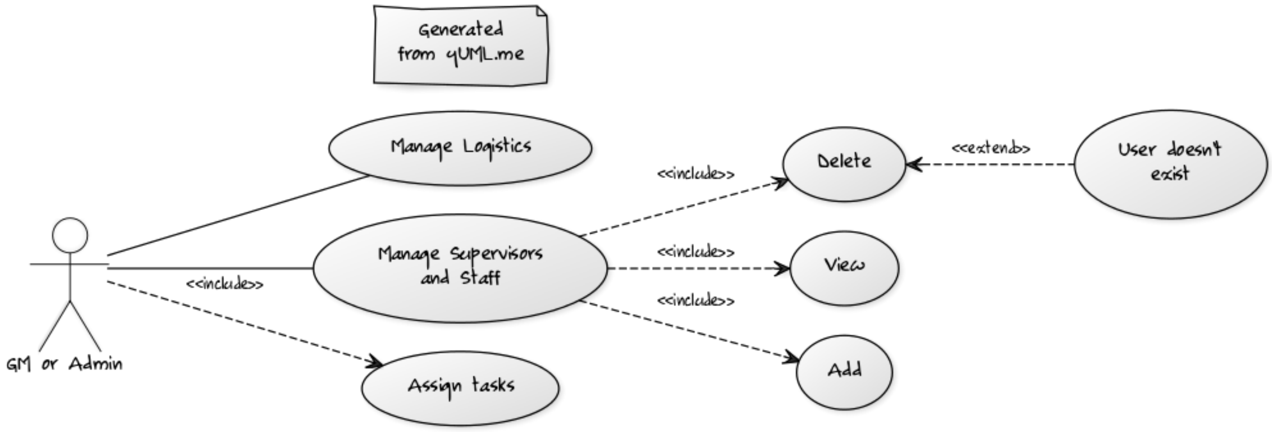
Use Case Diagrams:



- The User Actor will have child actors Staff, Dept. Supervisors and GM/Admin.

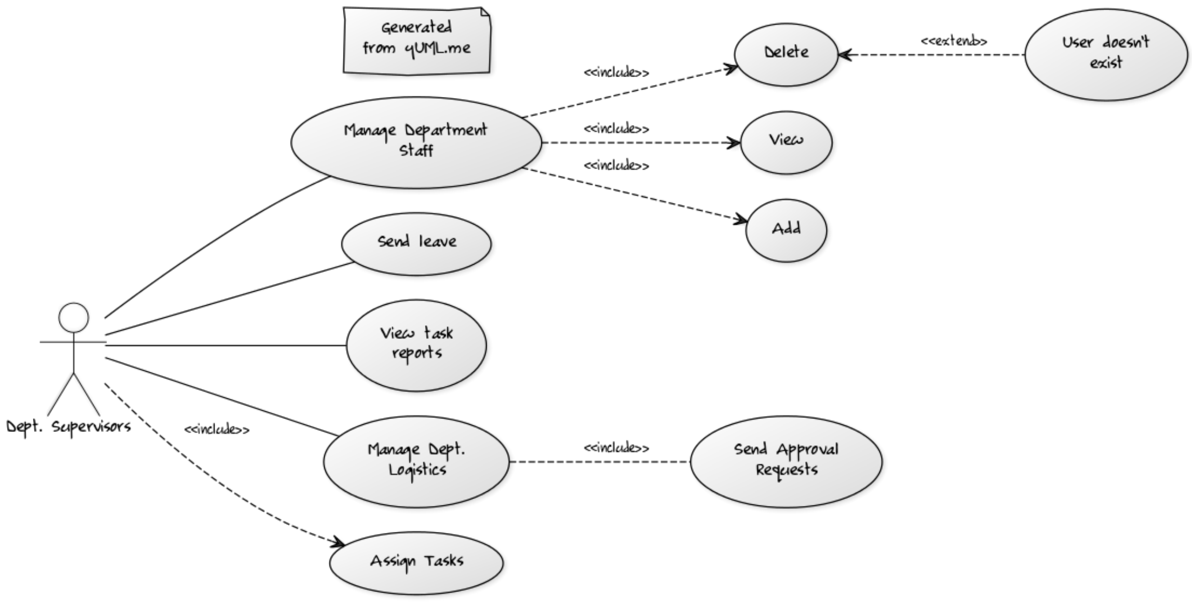
- The actor uses Sign Up for registering into the system for the first time. After Sign Up, a unique ID is given to every user.
- The user can Log In if it is registered in the system, else an exception will be thrown and Log Out if it is logged in, else an exception will be thrown.

GM or Admin:



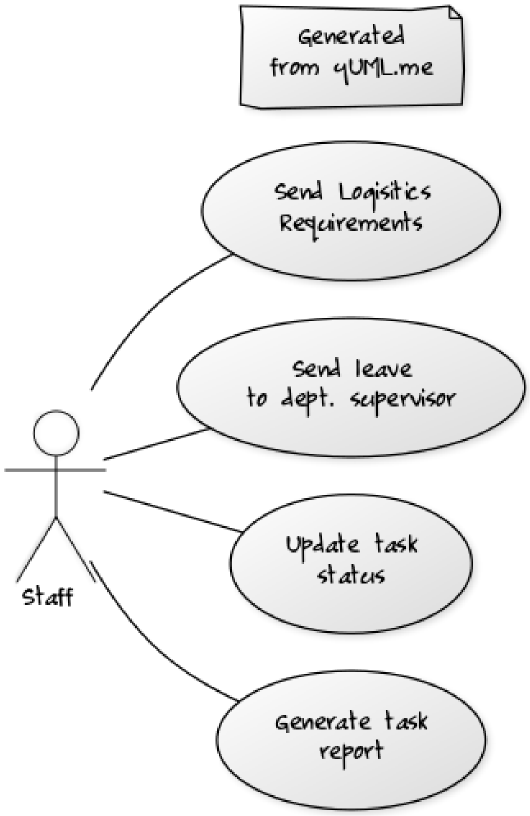
- The GM or Admin manages logistics and supervisors/staff.
- It assigns tasks to staff.
- While deleting a supervisor or staff, if it doesn't exist and exception is thrown.

Department Supervisors:



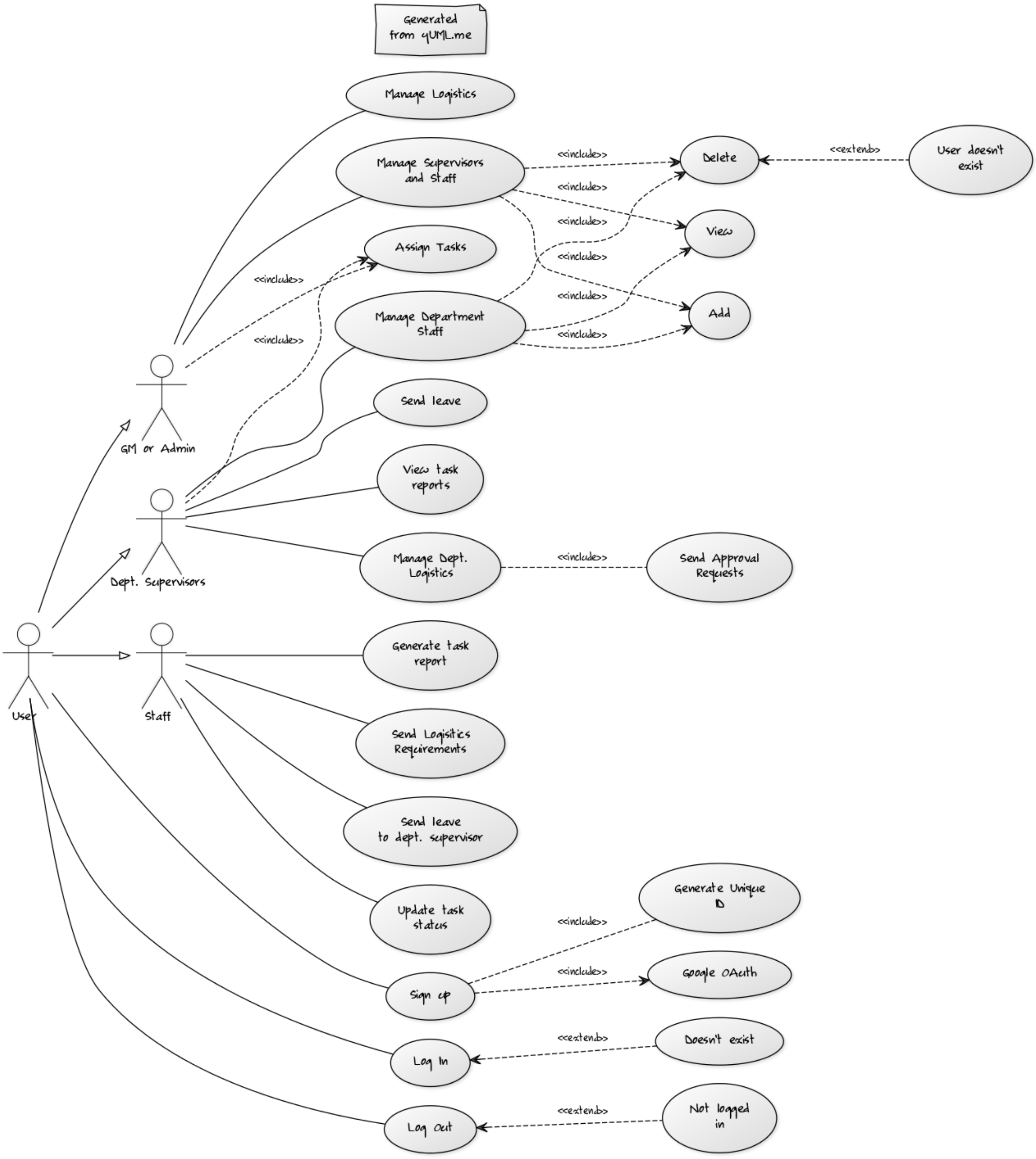
- Similar to GM/Admin, Supervisor is allowed to manage staff restricted to its department.
- It also assigns tasks to staff.
- Has the option to send a leave to GM/Admin.
- It also manages logistics restricted to its department and sends approval requests to GM when required.
- Views the task reports from staff.

Staff:



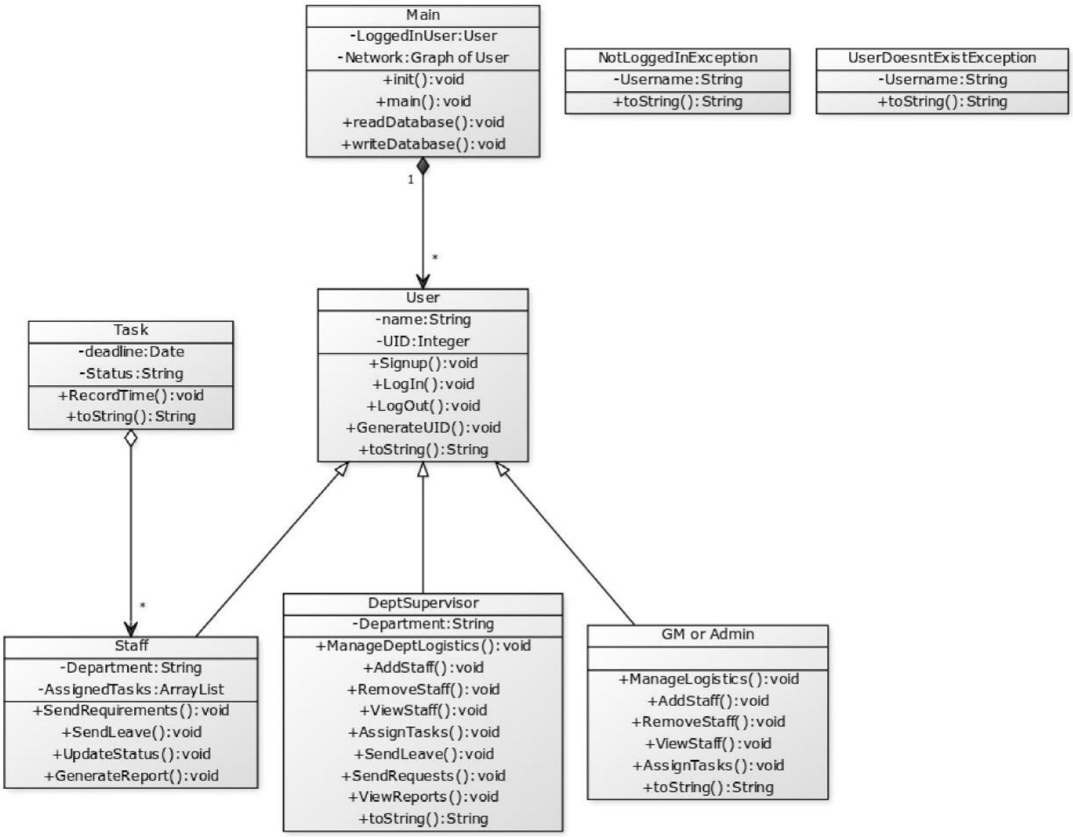
- Sends requirements and leave applications
- Updates status of the task if ONGOING or COMPLETED.
- Can generate task reports at any stage of the task.

Combined Use Case Diagram:



- The combined Use Case diagram for all the actors clearly shows the interdependence.

Class Diagram:



- The above diagram shows the ClassDiagram of the system we're going to create in the project.