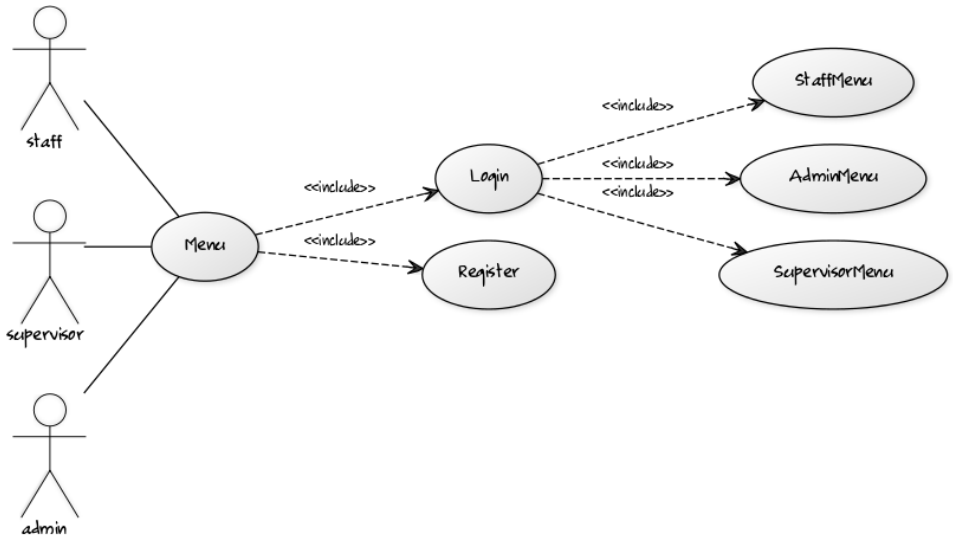
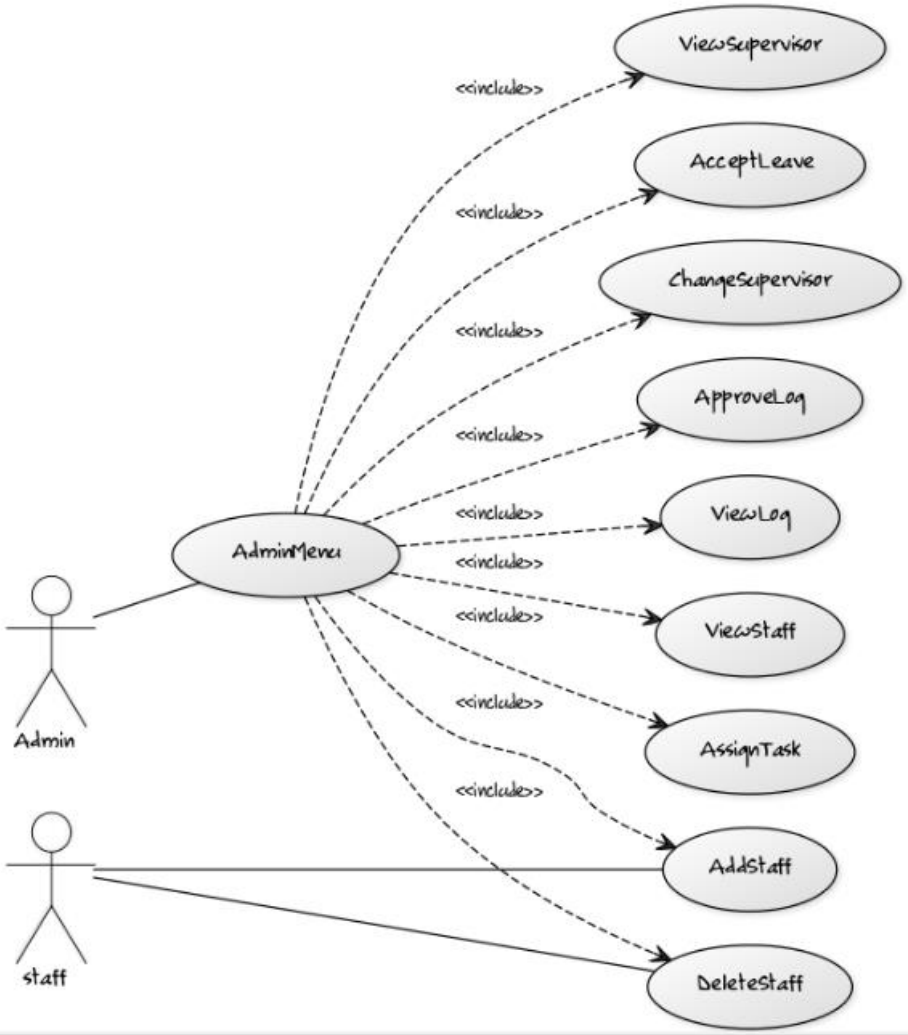
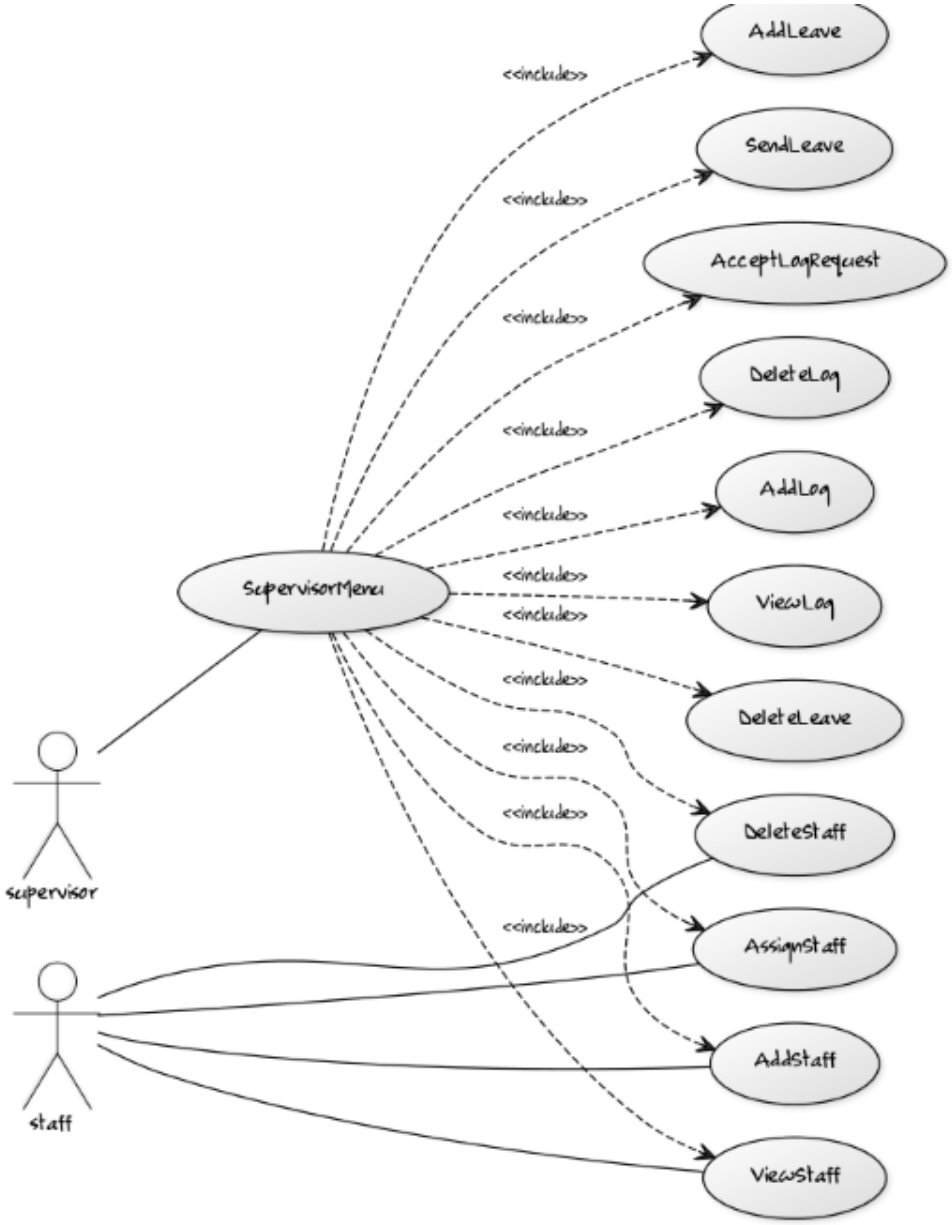
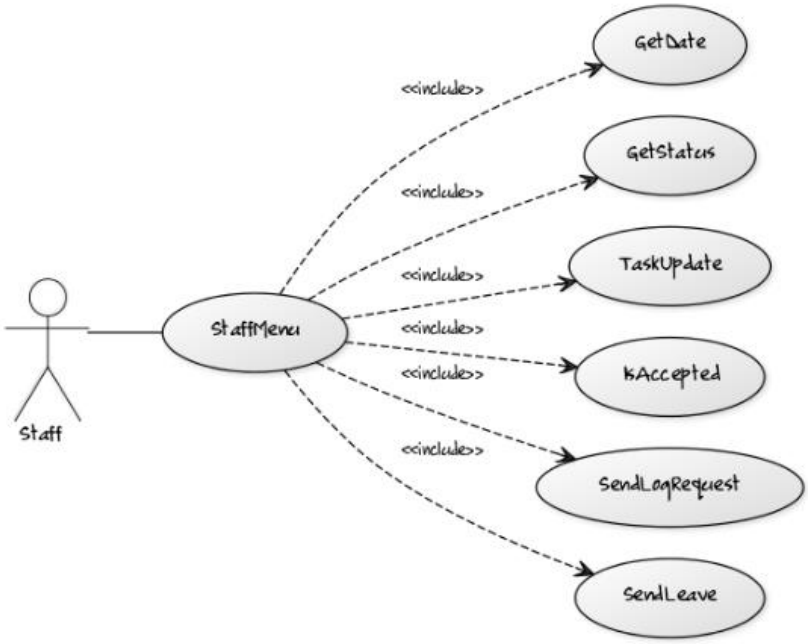
Use Case Diagrams

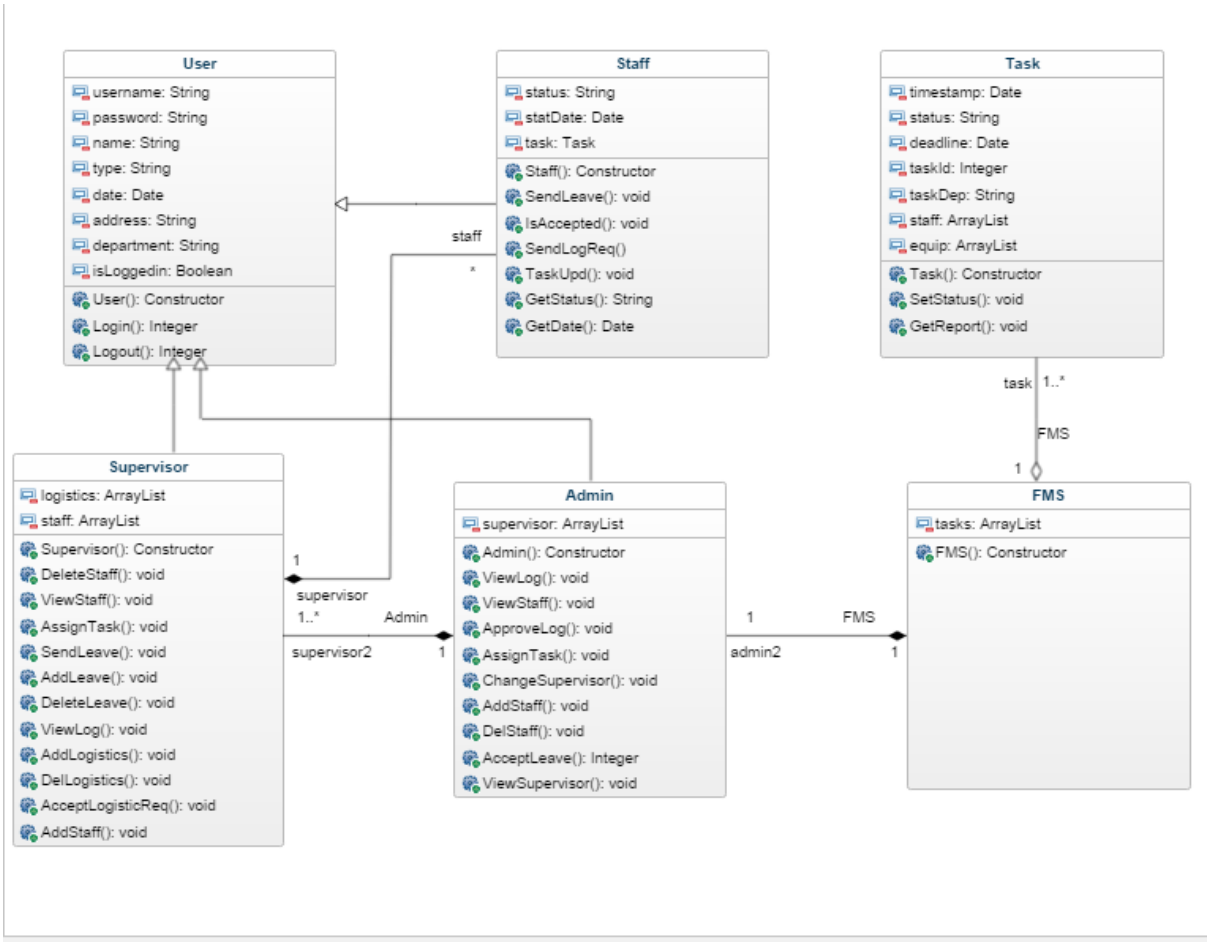





AddLeave
SendLeave
AcceptLogRequest
DeleteLog
AddLog
SupervisorMenu
ViewLog
DeleteLeave
DeleteStaff
AssignStaff
AddStaff
ViewStaff
<<include>>
supervisor
staff


GetDate
<<include>>
GetStatus
<<include>>
TaskUpdate
<<include>>
StaffMenu
<<include>>
IsAccepted
<<include>>
SendLogRequest
<<include>>
SendLeave
Staff

Class Diagrams



Use Case

Participating Actor : Staff,Admin,Supervisor
Entry Condition :
User selects the query : Login or Register

Login :User logs in using user-id and password

Register: User makes registration giving various details as per the User class drawn. It sends the information in the constructor.

Event Flow :
a)Reading the csv file to have the array list of staff, supervisor and also to set the admin.

b) **After login**:
Another menu displayed according to the type of the user logged in.
If admin has logged in :-
Will read the csv file of tasks and assign them as listed.
If supervisor/Staff logged in :-
Will deal with the tasks that are assigned to them.

Task Event Flow:-
First will be read by the admin and will be assigned to the supervisor of the department. Supervisor can then assign the task to someone from his staff who will then do the job and update the status accordingly. Meanwhile staff can ask for the logistics/equipment required and supervisor can approve them or if not available can forward it to the admin of the system.

The different functionalities that will be available to the users can be found in the report below or can be seen from the class diagrams.

c)**Exit Condition** : -
When user will logout. Before shutting the system off we will write back all the changes to our csv files.

Report

Classes:-

1.) FMS
2.) Admin
3.) Supervisor
4.) Staff
5.) User
6.) Task

FMS Class :-

It will act as main class of our function and all the activities and interaction will happen in this class. We will have task under this and also we will have our admin of FMS. The interactions of task with the other users will be initiated here only.

Imp. Functions:-

Main menu – Return type void. Will then do register or login as per the choice.

After login the execution will display the menu according to which type of user has logged in.

Admin Class :-

Here we will have our supervisors of different departments. The supervisors will also allow admin to access their department's staff and logistics.

Imp. Functions:-

All have return type void and their name gives their exp.
ChangeSupervisor
AddStaff
DelStaff
ApproveLog – Approve logistic requests.
AcceptLeave
AssignTask

Supervisor Class :-

The staff will be under them. The supervisors will have the rights to add/del staff and many more like :-

Imp. Functions:-

All have return type void and their name gives their exp.
DelStaff
AddStaff
SendLeave
AddLeave
DelLeave
DeleteLogistics
AcceptLogistics
AssignTask

Staff Class :-

Here we will have the status of the staff(busy etc.) and task they are assigned.

Imp. Functions:-

All have return type void and their name gives their exp.
TaskUpd – To update the status of the task they are performing.
SetStatus – To change their status.
Send Leave - To send application for leave

User Class :-

This class will have all the general information about the users like their id , address etc. i.e the information common to all supervisor, admin and staff.

Imp. Functions:-

User – Constructor to set the credentials.
Login – To login into the system.
Logout – To logout of the system.

Task Class :-

This will contain all the information about a task.

Imp. Functions:-

Task – Constructor to Initialize or sets diff. variables of the task.
SetStatus – Used to change the status of the task.
GetReport – To generate the report of the task after it's completed.