**Requirement 1-: Task Assignment to a department and not a single person.**

Solution -: The 'Assigned to' field in the task will now allow you to choose amongst the departments and not an individual. Also the administrator will have to mention the maximum number of employees required for the task. It also should allow choosing not only department but also just employee.

*For this there will be a selection field asking to choose whether Manager wants to choose according to the “Users” or “Departments”. If the user chooses the “Users” option then the manyone field pointing to “res.users” will be displayed else many2one field pointing to “hr.department”.*

*Now once a new task has been created a mail will be sent to all the employees of that particular department.For this in the “create” method of the object “project.task” the mail functionality will be used. The mail should be sent from the manager who created the task to the employees of tha department.*

*If the manager has assigned the employee while creating the task only then the mail will be dispatched to the employee to whom the task is assigned to and also the manager.So again the create() method functionality is going to be wroked with.*

*Q1. Do you want the mail wizard to pop up when the task is created or the mail should automatically be sent to you?*

**Requirement 2-: Task viewing restrictions**

Solution -: Once the task has been assigned by the administrator to a particular department the task will be viewable to all the employees of that particular department. Correct

Now the employees of that department will go to the task and assigne the task to himself, meaning that the task is taken over by a specific employee, he can also invite other employees in the social network. Invited employees can also track their time and view the task. The assigned to status will change the status of the task to 'In progress", the manager will be notified by e-mail that the task is accepted by specific employee.

*Once the task is assingned, it should be only viewable to the chosen depatment employees or the chosen user.*

*First of all add a new field “departmnet\_id” in res.users form and make it readonly.Now in the object “hr.emplyee” object's write method and create method synchronize the change of the department in the “hr.employee” to be mapped in the “res.user” form always.*

*For example columns = {*

*#Assign To field for a particular user is already there so need of making a new fields t o avoid any redudancy*

*# declaring a field for the department*

*assign\_dept = fields.many2one('hr.department”,”Assign To”),*

*}*

*Readonly – task object - ['|' , ('user\_id','=',uid),('assign\_dept','=','user.dept\_id')]*

*So now the project will be viewed to only those who are in the department or to whom the project has been directly assigned.But once the project has been taken up by the employee then the project should only be viewd to the manager ans the assigned employee.For this i have to make sure that*

*either only the “user\_id” assign to is fild or “dept\_id” is filled.Hence for this, each time either create method or write method of the object “project.task” is called i have to check that there is value for only one of them. Hence following are the conditions*

* *if manager enters the “user\_id” during the creation then in the create() make sure that “dept\_id” fields is totally empty.*
* *After that once the emplyee is assigning the task to himself then the make sure that in the write method of “project.task” i will pass values to both 'user\_id' and 'dept\_id' accodingly*
* *Also to ensure this i will create a \_contraint so that this is never ever allowed.*

*Now another thing is that this task should be viewable to only those who have been invited for this tasks. For this add a new field which is “many2many” in the “project.task” object and name it “invite\_ids”.The task should be viewable to these invites too.*

*Readonly – task object - ['|' ,'|', ('user\_id','=',uid),('assign\_dept','=','user.dept\_id'),(user.id','in','invite\_ids')]*

*If the “user\_id”,”dept\_id” field is changed and the vals are sent in the write method then again the mail generation will take place to the “user\_id” or the employess of the “dept\_id” for the availability if the task*

*Also two email emplates will created.One for the emplyee(“user\_id”) and the other for the (“dept\_id” )*

*Q. Ask again the change status problem mentioned above in the requirements!!!?*

**Reuiqrement 3-: For example we have three tasks A,B,C and task B,C should not start before the task A has been completed. Or the task C, should not start if the task B is not completed, or vice versa, it can only start when the A and B is completed.**

Solution -: For this we will have a new field in the task form called **chained task.** For example if we are in the form of task A we will enter B,C as the chained task and click on save.Now B,C task will remain readonly until the task A has been completed.Also the above volunteering process for B&C will start only once the task A has been completed.If the task A is not completed no employee will be able to volunteer for the task B,C .

*Create field “writable\_task” of boolean type in project.task. The task is writable if this field value is True else the task in not writable by the users.By default the value of this field is True.*

*To achieve this functionality we will have to add a many2many field in “project.task” object pointing to the “project.task”.This will allow the manager to chain the task B,C to task A.In the create method of the project.task .*

*In the create() method of the project.task then the writable\_task field will be switched to False.So now the user will not be able to make any changes in these task.Also these task will not be viewable to any user also.Also now they won't be able to view it also*

*For this edit the above record rule further - :*

*Readonly - ['|' ,'|', ('user\_id','=',uid),('assign\_dept','=','user.dept\_id'),(user.id','in','invite\_ids'),('writable\_task','=',True)]*

*Write [('writable\_task','=',True)]*

*Note : These record rules are only defined for the user hence this means that manager has absolute power to do anything even is the* ***writable***  *is True/False.*

*Also block the witting on the work summary lines of the task by creating the* ***write*** *record rule.*

***project.task.work*** *is the name of the object which is connected to* ***project.task*** *by the field* ***work\_ids***

*and the many2one field in* ***project.task.work*** *is* ***task\_id***

*Now put a rule on the project.task.work ---- create record rule ---*

*create [('task\_id.writable\_task','=',True)] for the users only*

*Also now when the task is completed the user will press the button* ***Done.****When this button is pressed in the method that will be called makes the chained tasks writable after that.In this case the* ***writable\_task*** *field of the* ***project.task*** *will be made True again.*

*There will also be custom button that is accessible only to the manager and will be able to make the task writable to user if they want to override the situation i.e write on task B when task A is not completed or another thing can be done. If the manager changes the “chained task” field of the task A and for example makes it only B then task C's writable\_task should automatically become True or the manager sustitiutes the task C with task D then the* ***writable\_task of D*** *becomes* ***False*** *and of C becomes* ***True.***

*For this when calling the* ***write()*** *of* ***project.task*** *find out which task has been added and which task been deleted and then accordingly manipultae their* ***writable*** *field.*

*For Multi Chaining -:*

*If A ----> B,C*

*L ----> B,D*

*The* ***project.task*** *will have a field that will have a list of all the task to which it will be chained to.So until this list is not empty this task will not be editable by the users i.e the* ***writable\_task***  *field will be False.So when the task is* ***created or written*** *this has to be checked always for the parent and the child tasks.In the end the* ***writable\_task*** *field will determine whether the task is writable or not.*

*Also this field has to be manipulated all the time when the task is deleted or created.*

Requirement - 5

When we create a project for specific customer, the customer should receive an e-mail, telling that the project for him has been created, and if he wants he can sign in and watch the progress of the project and the tasks. When the project is created customer gets an e-mail. When the task is created customer also gets an email. When the task is completed, customer gets an email, when the project is completed or marked as done also gets an email.

Requirement – 6

If the project budget is over the limit it should be in red colour. I mean if the project estimated time with the tasks was 100, and employees where working 110 hours the whole project should be in red. The same with the tasks. If it exceeds the allowed time limit it should get red.

*Just change the tree view of both the project.task and project.project*

Requirement - 7

Customers should be able to login to the openerp and submit trouble tickets (issues) to a particular project. When the issue is created, by default, project managaers departments should be notified by e-mail about created issue on the project.