



HACETTEPE  
UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BBM 473 FALL2018  
DATABASE MANAGEMENT SYSTEMS  
LABORATORY

ADVISOR:  
DR. TUĞBA GÜRGEN ERDOĞAN  
AYSUN KOÇAK

SUBJECT:  
PROJECT ADMINISTRATION SYSTEM

PROJECT MEMBERS:  
21527381 MELTEM TOKGÖZ  
21426807 BÜŞRA ÇİÇEK

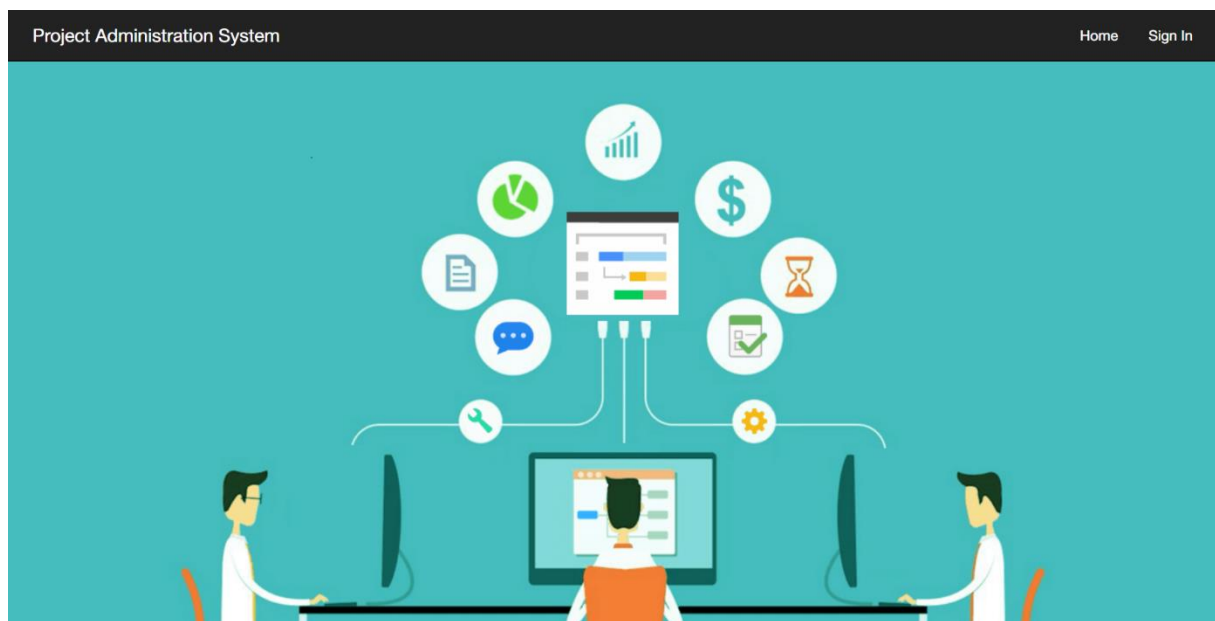
# PROJECT ADMINISTRATION SYSTEM

## 1. PROJECT DEFINITION

System is a Project administration system. This system is used to ensure coordination and increase efficiency. With this system, a company can create their own business schema and thanks to this, easy execution of projects is ensured.

Admin adds the companies and customers that will use the system to the system. Companies can create different projects according to the departments within them. Projects can be divided into various forms (issue and subtask) and assigned to employees with company. Restrictions by manager can be imposed on the projects that employees see and work with. The projects plan and the company's workflow are created. Projects can be sold to customers.

## 2. USER TYPE AND APPLICATION INFORMATION



We made our project using the django framework and postgresql database. Our project consists of 3 app (account, userprofile and home). All the elements we use in our project and their roles in the project are as follows;

\*Admin: Adds users (company or customer) to the project administration system and controls them.

\*Company: User of this Project.

- \*Customer: He/she requests a project from the company.
- \*Department: Area where projects belong according to subject.
- \*Employee: Worked people on a Project.
- \*Head: Person who administer to department.
- \*Other: People working in the department
- \*Project: Product requested by customer.
- \*Issue: A part of a Project.
- \*SubTask: A part of a Issue.
- \*Workflow: It shows workflow for company.
- \*Project Plan : It shows plan for project.

There are 3 different interfaces in our project for distinct user. First of all, you need to login for the user type you selected. Here user type can be admin, company or customer.

Project Administration System

Home Sign In

## Log In

User Name\*

Password\*

User Type\*

Log In

Now let's examine the things that can be done for each user type.

## 2.1 Admin

Project Administration System

Search on system Search

Home User Create User Settings Sign Out

Welcome to Admin Page, meltem  
 Last Log In Date = 13 Ocak 2019 15:13  
 Is user super user = True  
 Is user active = True

```

#####
WELCOME TO OUR ADMIN PAGE
#####
In this page you can all operations about our website,
1-You can see all Departments of Companies
2-You can see all Employees of Companies
3-You can see all Projects of Companies
4-You can see all subtasks of Issues
5-You can create a new user
#####
    
```

Admin adds users (company or customer) to the project administration system. And admin can see the information about the system. Admin can create a new company or customer user.

In this section, we also used the trigger. So when a new admin is added, a new company or customer can also be created same time by the help of the timer. But for the user you added we need to make some edits from the admin page. Create User button will direct you to this page.

Here you can view all the users in the system or you can see them as separately company or customer. You can also update and delete these users.

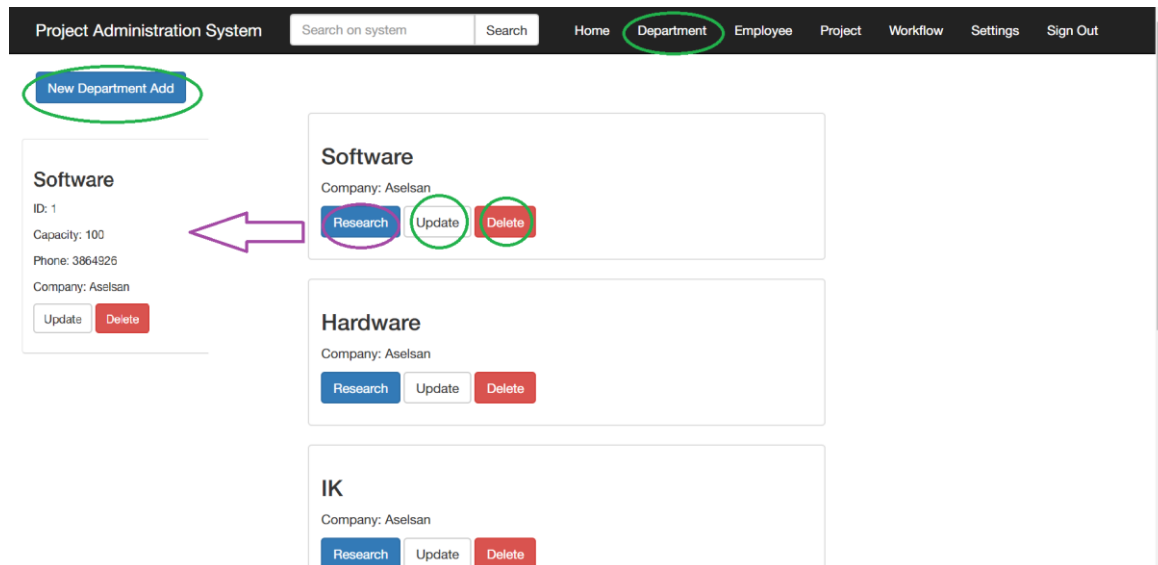
The screenshot shows the 'Project Administration System' interface. At the top, there is a navigation bar with 'Home', 'User' (circled in red), 'Create User', 'Settings', and 'Sign Out'. Below the navigation bar, there are two tabs: 'Companies' and 'Customers'. Two red arrows point to these tabs. The main content area displays a list of users: 'Büsa', 'Ali', and 'Sema'. Each user entry has a 'Research' button (blue), an 'Update' button (grey), and a 'Delete' button (red). Below this list, there is another section with 'Apple', 'Microsoft', and 'Aselsan', each with a 'Research' button (blue). The bottom navigation bar is identical to the top one.

You can change the password of the admin you entered in the change button on the Setting button. And you can check out. You can sign out with the Sign Out button in the admin page.

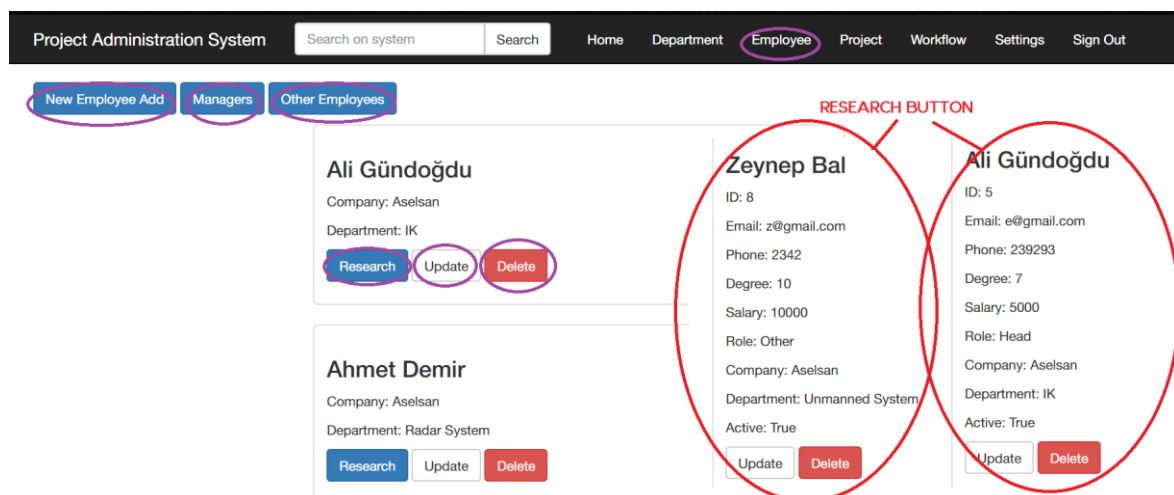
The screenshot shows two side-by-side forms. On the left is the 'User' form with fields for 'Username\*', 'Password\*', and 'Password Again\*', and a 'Create User' button. On the right is the 'Change Password' form with fields for 'Eski parola\*', 'Yeni parola\*', and 'Yeni parola onay\*', and a 'Save Changes' button. The top navigation bar is identical to the previous screenshot, with 'Create User' (circled in red) on the left and 'Settings' (circled in red) on the right.

## 2.2 Company

Companies are the most important users of the system. There is a separate user interface for companies. Companies, the following sections can access; department, employee (head or other), project (subtask or issue). They are divided into departments.(engineering department, human resources department and etc.) Companies can do the task of deleting and updating the new department.



Companies can add employees to their companies and determine their status as managers and other employees. Just like department, employees have the option of updating and deleting.



The companies can finally add projects for the topics studied in the company. They can assign these projects to any employee. They can divide projects into issue if they wish, and they can subdivide issues into subtask and assign them to individual employees.

The screenshot shows the 'Project Administration System' interface. The top navigation bar includes 'Home', 'Department', 'Employee', 'Project' (highlighted with a green circle), 'Workflow', 'Settings', and 'Sign Out'. Below the navigation bar, there are three tabs: 'New Project Add' (highlighted with a green circle), 'Issues of Projects' (highlighted with a green circle), and 'Projects Plan'. A black arrow points from 'Issues of Projects' to a 'New Issue Add' button (highlighted with a green circle) and a 'Subtasks of Issues' button (highlighted with a green circle). A green arrow points from the 'Research' button (highlighted with a green circle) of the 'Cryptographic Communication Design' project to a detailed view of the same project. The detailed view shows the project's ID, start and end dates, situation, company, department, and project manager, along with 'Update' and 'Delete' buttons.

Project Administration System

Search on system

Search

Home Department Employee **Project** Workflow Settings Sign Out

New Project Add Issues of Projects Projects Plan

**Cryptographic Communication Design**  
Company: Aselsan  
Research Update Delete

**Safety of critical facilities**  
Company: Aselsan  
Research Update Delete

**Border Security Systems**  
Company: Aselsan  
Research Update Delete

**1**  
Project: Cryptographic Communication Design  
Research Update Delete

**2**  
Project: Border Security Systems  
Research Update Delete

**Cryptographic Communication Design**  
ID: 3  
Start Date: 12 Aralık 2015 23:02  
End Date: 12 Aralık 2016 23:02  
Situation Finished  
Company: Aselsan  
Department: Crypto Systems  
Project Manager: Ahmet Demir  
Update Delete

Each company only sees its own employees and departments. You are not allowed to see the information of another company. Can also create workflow and project plan and see them. In short, companies have all the controls for the company.

The screenshot shows the 'Project Administration System' interface. The top navigation bar includes 'Home', 'Department', 'Employee', 'Project', 'Workflow' (highlighted with a green circle), 'Settings', and 'Sign Out'. Below the navigation bar, there is a 'New Workflow Add' button (highlighted with a green circle). A green arrow points from the 'Research' button (highlighted with a green circle) of the '1' workflow to a detailed view of the same workflow. The detailed view shows the workflow's ID, type, date, and content, along with 'Update' and 'Delete' buttons.

Project Administration System

Search on system

Search

Home Department Employee Project **Workflow** Settings Sign Out

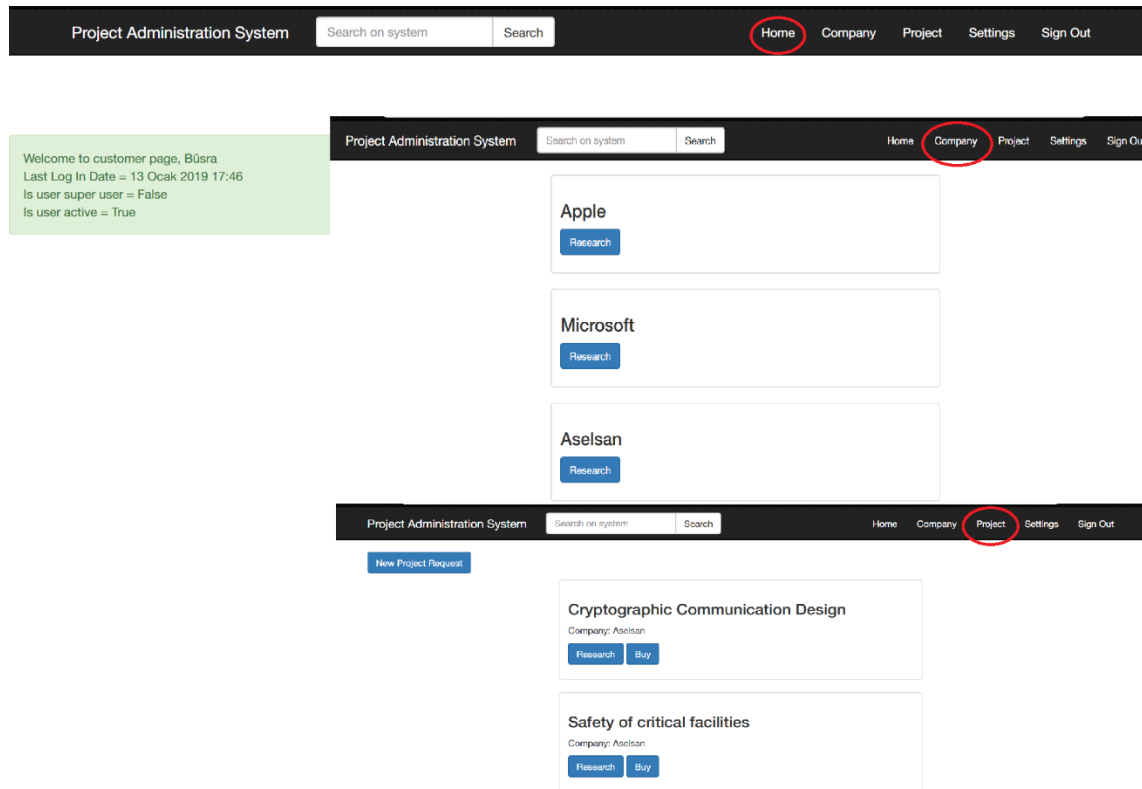
New Workflow Add

**1**  
Type: To Do List  
Research Update Delete

**1**  
Type: To Do List  
Date: 23 Eylül 2020 01:01  
Content: <p>1. workflow</p>  
Update Delete

## 2.3 Customer

Customers are second type users in the system. Customers cannot see their company's private and their internal business. They can view company, projects and buy the project they want. When a customer buys a project, the project is deleted from the list of projects.



## 4.DATABASE And WEB APPLICATION REQUIREMENT

Insert, delete, update, login and view operations are shown above. In this section, we will talk about the operations we have implemented in the code that we cannot express visually.

### 4.1 Trigger

```
def create_profile(sender, **kwargs):
    if kwargs['created']:
        user_profile = Company.objects.create(user=kwargs['instance'])

    post_save.connect(create_profile, sender=User)
#####
```

When we create a new user from the admin page, a company or customer object is created by triggers. The code that makes this function is within the companyprofile app / models.py / company class.

## 4.2 Transaction

```
179
180     @transaction.atomic
181     def department_create(request):
182         form = DepartmentForm(request.POST or None)
183         if form.is_valid():
184             try:
185                 with transaction.atomic():
186                     departments = form.save(commit=True)
187                     messages.success(request, "Department is created, successfully!")
188                     return HttpResponseRedirect(departments.get_department_url())
189             except IntegrityError:
190                 messages.ValidationError(request, "Department is not created!")
191
192
193     context = {
194         'form': form,
195     }
196     return render(request, 'department/form.html', context)
```

We have implemented transaction property for many classes when creating and updating. If all features are fully populated, the form is committed with by transaction. There is one instance of this above.

## 4.3 Store Passwords in an Encrypted Format

```
34 #####
35 def user_view(request):
36     form = UserForm(request.POST or None)
37     if form.is_valid():
38         user = form.save(commit=False)
39         password = form.cleaned_data.get('password1')
40         user.set_password(password)
41         user.save()
42         return redirect('http://127.0.0.1:8000/admin/auth/user/')
43     return render(request, 'accounts/form.html', {'form': form, 'title': 'Create User'})
44 #####
```

Django yönetimi

HOŞ GELDİNİZ, MELTEM. SİTEYİ

Giriş > Kimlik Doğrulama ve Yetkilendirme > Kullanıcılar > Aselsan

kullanıcı değiştir

Kullanıcı adı:

Aselsan

Zorunlu. 150 karakter ya da daha az olmalı. Sadece harfler, rakamlar ve @/./+/-/\_ karakterleri kullanılabilir.

Parola:

algoritma: pbkdf2\_sha256 yinelemeler: 30000 tuz: 7i9JXG\*\*\*\*\* adresleme: 2vExG6\*\*\*\*\*

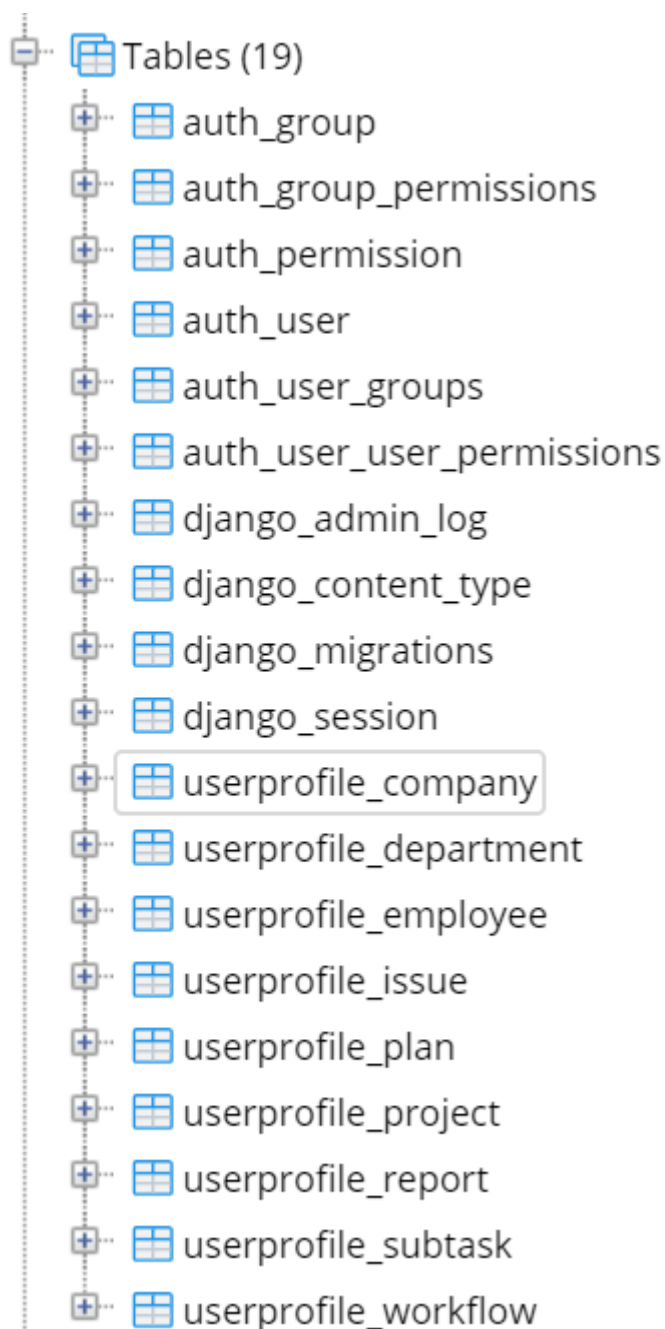
Ham parolalar saklanmazlar, bu yüzden bu kullanıcının parolasını görmenin yolu yoktur, fakat bu formu kullanarak parolayı değiştirebilirsiniz.



In this section, we have used the convenience provided by django because no method for password encryption is specified. As shown in the picture above, the code in the code that we write makes the passwords encrypted.

#### 4.4 Interfaces for data entry (Relation tables)

As previously mentioned, we used postgresql as a database. Our Relation Tables are like this. Also, these tables and their features are shown in the diagram below.



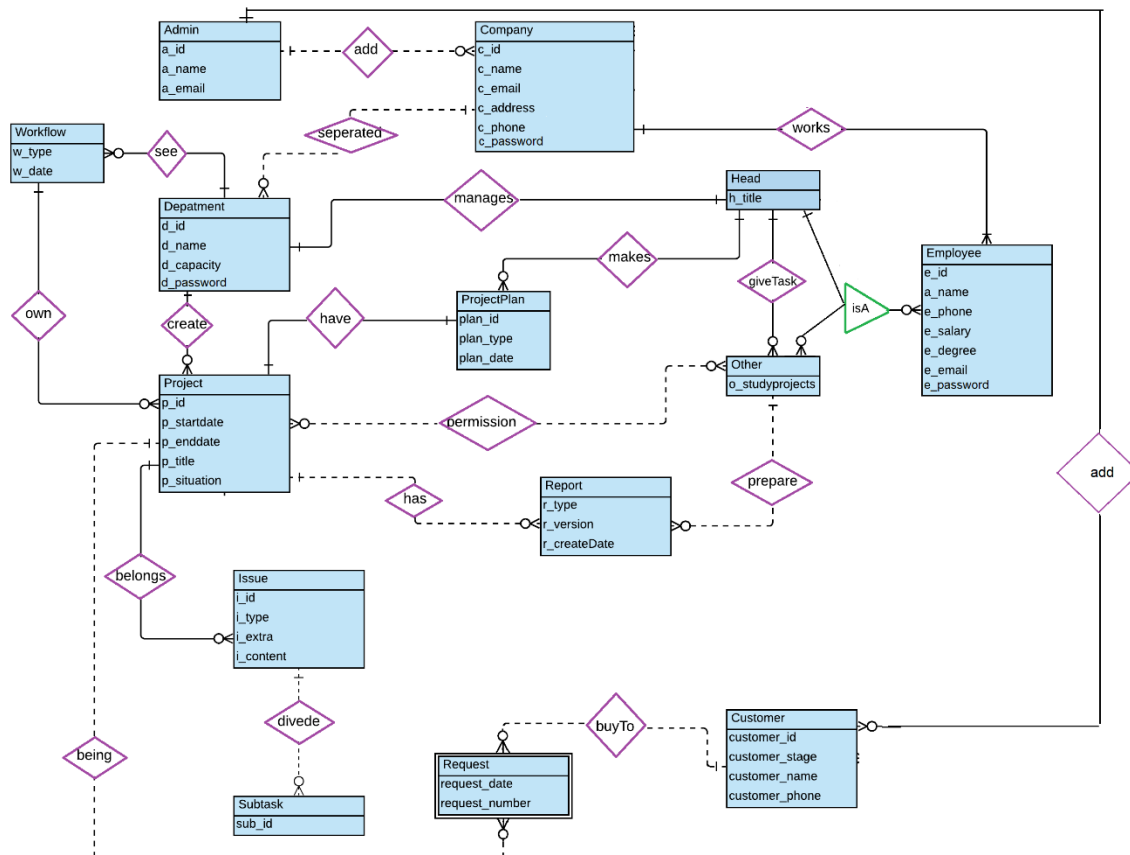
```

Properties SQL Statistics Dependencies Dependents
1 -- Table: public.userprofile_company
2
3 -- DROP TABLE public.userprofile_company;
4
5 CREATE TABLE public.userprofile_company
6 (
7     c_id integer NOT NULL DEFAULT nextval('userprofile_company_c_id_seq'::regclass),
8     c_name character varying(30) COLLATE pg_catalog."default" NOT NULL,
9     c_email character varying(254) COLLATE pg_catalog."default" NOT NULL,
10    c_address text COLLATE pg_catalog."default" NOT NULL,
11    c_phone character varying(20) COLLATE pg_catalog."default" NOT NULL,
12    c_password character varying(6) COLLATE pg_catalog."default" NOT NULL,
13    role character varying(30) COLLATE pg_catalog."default" NOT NULL,
14    user_id integer NOT NULL,
15    CONSTRAINT userprofile_company_pkey PRIMARY KEY (c_id),
16    CONSTRAINT userprofile_company_user_id_key UNIQUE (user_id)
17 ,
18    CONSTRAINT userprofile_company_user_id_6659832d_fk_auth_user_id FOREIGN KEY (user_id)
19        REFERENCES public.auth_user (id) MATCH SIMPLE
20        ON UPDATE NO ACTION
21        ON DELETE NO ACTION
22        DEFERRABLE INITIALLY DEFERRED
23 )
24 WITH (
25     OIDS = FALSE
26 )
27 TABLESPACE pg_default;
28
29 ALTER TABLE public.userprofile_company
30     OWNER to postgres;

```

One table example shown

## 5.RELATIONS



### **One to many relations:**

- Admin-company:We choose one to many relation for admin-company relation because of an admin can add many company into the program.
- Company-Employee:We choose one to many relation for company-employee relation because of a lot of employee can work in just one company
- Project-Workflow:We choose one to many relation for department-workflow relation because of a lot of project can be seen in just one workflow
- Department-Workflow:We choose one to many relation for department-workflow relation because of a department can has a lot of workflows
- Department-Project:We choose one to many relation for department-project relation because of a department can has a lot of project.
- Project-Issue:We choose one to many relation for project-issue relation because of a project can has a lot of issues.
- Issue-Subtask:We choose one to many relation for issue-subtask relation because of an issue can has a lot of subtasks
- Customer-Project:We choose one to many relation for customer-project relation because of a customer can has a lot of projects

### **Many to many relations:**

- Company-Department:We choose many to many relation for company-department relation because of a company can has a lot of department and a department can be in a lot of company
- Project-Employee:We choose many to many relation for project-employee relation because of a project can has a lot of employees and a employee can has a lot of project

### **One to one relations:**

- Department-Head:A department can has just a one head
- Project-Projectplan:A project can has just a one projectplan

**Subclass:** -Employee-Head-Other:Employee has Head and Other employees

## 7.REFERENCES

<https://docs.djangoproject.com/en/2.1/howto/custom-template-tags/>

<https://django-crispy-forms.readthedocs.io/en/latest/install.html>

<https://github.com/django-ckeditor/django-ckeditor>

<https://github.com/barissaslan/django-dersleri>

<http://www.pythondersleri.com/2014/03/veritaban-yaplandrmas.html>

<https://www.postgresql.org/docs/10/app-psql.html>

<https://stackoverflow.com/questions/21128899/how-to-make-an-auto-increment-integer-field-django>

<https://django-simple-history.readthedocs.io/en/stable/usage.html>

<https://micropyramid.com/blog/how-to-add-a-custom-managers-in-django/>

<https://www.youtube.com/watch?v=lxSZevvkcc4>