**Bil 372 Dersi Proje Ödevi Final Raporu**

Grup Üyeleri:

Selahattin Kaan Deniz - 181101057

Faruk Mert Akcan - 181101023

Mustafa Mert Ekin - 181101026

Projenin Amacı :

Bu projenin amacı bir havayolu şirketinde bulunan servis, admin, sefer alanlarında yetkili olan personellerin kullanabileceği veritabanına bağlı bir form uygulaması yapmaktır.

Proje Kapsamı:

Bu proje kapsamında servis yetkisine sahip kullanıcılar servis, genel bakım ve tamir entityleri üzerinde CRUD işlemleri yapabilir. Sefer yetkisine sahip kullanıcılar uçak oluşturup sefer ekledikten bu seferlere uçak atayabilir, tarihlerini güncelleyebilir ve yaklaşan seferleri görebilir. Admin yetkisine sahip kullanıcılar ise sistemdeki bütün özellikleri kullanma hakkına sahiptir. Personel ekleyip silebilir, personelleri mürettebata veya servis personeline atayabilmektedir.

Personel Table’ı

create table Personel

(

PersonelID varchar(11) not null,

Cinsiyet varchar(12) not null,

Maas varchar(9) not null,

DogumTarihi date not null,

Ad varchar(30) not null,

Soyad varchar(30) not null,

Egitim varchar(80) not null,

TCNO varchar(11) not null,

SGN varchar(11) not null,

IseAlimTarihi date not null,

TelefonNO varchar(13) not null,

Adres varchar(200) not null,

Eposta varchar(50) not null,

PersonelTipi varchar(20) not null,

primary key (PersonelID)

)

Personel tablosunda yapılan işlemler admin tarafından yapılır. Admin personel üzerinde CRUD işlemleri uygulayabilir.

Kullanıcı Table’ı

create table Kullanici

(

KullaniciAdi varchar(30) not null,

Sifre varchar(256) not null,

KullaniciTipi varchar(50) not null,

PersonelID varchar(11) not null,

primary key (KullaniciAdi),

foreign key (PersonelID) references Personel(PersonelID)

)

Kullanıcı tablosundaki işlemler admin tarafından yapılır. Admin kullanıcılar üzerinde CRUD işlemleri yapabilir. Şifreler databasede hashlenip encoded olarak tutulur.

Uçak Table’ı

create table Ucak

(

UcakID varchar(11) not null,

Model varchar(50) not null,

YolcuKapasitesi varchar(10) not null,

Agirlik varchar(10) not null,

UreticiFirma varchar(50) not null,

UretimYili varchar(10) not null,

primary key (UcakID)

)

Uçak tablosundaki işlemler hem admin hem de sefer yetkisine sahip olan kullanıcılar tarafından yapılabilir. Her iki kullanıcı da CRUD işlemleri uygulayabilir.

Mürettebat Table’i

create table Murettebat

(

MurettebatID varchar(11) not null,

UcakID varchar(11) not null,

PersonelID varchar(11) not null,

YabanciDil varchar(80) not null,

UcusSuresi varchar(20) not null,

MurettebatTipi varchar(20) not null,

primary key (MurettebatID),

foreign key (UcakID) references Ucak(UcakID),

foreign key (PersonelID) references Personel(PersonelID)

)

Mürettebat tablosundaki CRUD işlemleri admin yetkisine sahip olan kullanıcı tarafından yapılır.Bir mürettebat oluşturulduktan sonra UcakID ve PersonelID değiştirilemez çünkü bu değerler sırasıyla uçak ve personel tablolarına referans etmektedir.

Pilot Table’ı

create table Pilot

(

MurettebatID varchar(11) not null,

UcakID varchar(11) not null,

PersonelID varchar(11) not null,

LisansTuru varchar(20) not null,

PilotRolu varchar(20) not null,

foreign key (UcakID) references Ucak(UcakID),

foreign key (PersonelID) references Personel(PersonelID),

foreign key (MurettebatID) references Murettebat(MurettebatID)

)

Pilot tablosundaki CRUD işlemleri admin yetkisine sahip olan kullanıcı tarafından yapılır.

Kabin Memuru Table’ı

create table KabinMemuru

(

MurettebatID varchar(11) not null,

UcakID varchar(11) not null,

PersonelID varchar(11) not null,

MedeniDurumu varchar(20) not null,

foreign key (UcakID) references Ucak(UcakID),

foreign key (PersonelID) references Personel(PersonelID),

foreign key (MurettebatID) references Murettebat(MurettebatID)

)

Kabin memuru tablosundaki CRUD işlemleri admin yetkisine sahip olan kullanıcı tarafından yapılır.

Sefer Table’ı

create table Sefer

(

SeferID varchar(11) not null,

UcakID varchar(11) not null,

UcusSuresi varchar(20) not null,

SeferTarihi date not null,

VarisZamani varchar(10) not null,

KalkisZamani varchar(10) not null,

VarisYeri varchar(100) not null,

KalkisYeri varchar(100) not null,

primary key (SeferID),

foreign key (UcakID) references Ucak(UcakID)

)

Sefer tablosundaki CRUD işlemleri admin veya sefer yetkisine sahip olan kullanıcı tarafından yapılır. Seferler uçaklarla ilişkilendirilerek uçaklar atanır. Bir uçak bir sefere atıldıktan sonra o uçak silinemez. Kullanıcı aynı zamanda yaklaşan seferleri görebilir.

Servis Table’ı

create table Servis

(

ServisID varchar(11) not null ,

UcakID varchar(11) not null ,

BaslangicTarihi date not null,

BitisTarihi date not null,

ServisTipi varchar(20) not null,

primary key (ServisID),

foreign key (UcakID) references Ucak(UcakID)

)

Servis tablosundaki CRUD işlemleri admin veya servis yetkisine sahip olan kullanıcı tarafından yapılır. Servisler uçak ID’leri ile bağlanır. Böylece bir uçak servise yollanabilir tarihleri de belirlenerek.

Genel Bakım Table’ı

create table GenelBakim

(

PersonelID varchar(11) not null,

ServisID varchar(11) not null,

UcakID varchar(11) not null,

BakimTuru varchar(50) not null,

foreign key (UcakID) references Ucak(UcakID),

foreign key (ServisID) references Servis(ServisID),

foreign key (PersonelID) references Personel(PersonelID)

)

Genel Bakım tablosunda admin ve servis yetkisine sahip kullanıcılar işlem yapabilir. Bu işlemler genel bakıma uçak yollamak, genel bakımdaki bir uçağı veya bakım türünü güncellemek veya genel bakımdan uçak çıkartmak şeklinde olabilir.

Tamir Table’ı

create table Tamir

(

PersonelID varchar(11) not null,

ServisID varchar(11) not null,

UcakID varchar(11) not null,

KazaTarihi date not null,

KazaSebebi varchar(100) not null,

TamirSonucu varchar(100) not null,

foreign key (UcakID) references Ucak(UcakID),

foreign key (ServisID) references Servis(ServisID),

foreign key (PersonelID) references Personel(PersonelID)

)

Tamir tablosu da genel bakım tablosuna benzemektedir. Farkı bir bakım yerine arıza veya kaza olduğu zaman uçakların tamir tablosuyla bağlanmasıdır. Bu tabloyu da admin ve servis yetkisine sahip olan kullanıcılar kullanabilir.

Servis Personeli Table’ı

create table ServisPersoneli

(

ServisPersoneliID varchar(11) not null,

ServisPersonelTipi varchar(20) not null,

PersonelID varchar(11) not null,

CalistigiHavalimani varchar(50) not null,

foreign key (PersonelID) references Personel(PersonelID),

primary key (ServisPersoneliID),

)

Servis personeli tablosunu admin yetkisine sahip olan kullanıcı kullanabilir. Hazırda var olan fakat bir yere atanmamış ve personel tipi ‘servis’ olan personelleri servis personeline atayabilir.

Tamir Personeli Table’ı

create table TamirPersoneli

(

ServisPersoneliID varchar(11) not null,

PersonelID varchar(11) not null,

Uzmanlik varchar(20) not null,

foreign key (PersonelID) references Personel(PersonelID),

foreign key (ServisPersoneliID) references ServisPersoneli(ServisPersoneliID)

)

Tamir personeli tablosunu admin yetkisine sahip olan kullanıcı kullanabilir. Hazırda var olan fakat bir yere atanmamış ve personel tipi ‘tamir’ olan servis personellerini tamir personeli olarak atayabilir.

Genel Bakım Teknisyeni Table’ı

create table GenelBakimTeknisyeni

(

ServisPersoneliID varchar(11) not null,

PersonelID varchar(11) not null,

TeknisyenDerecesi varchar(11) not null,

foreign key (ServisPersoneliID) references ServisPersoneli(ServisPersoneliID),

foreign key (PersonelID) references Personel(PersonelID)

)

Genel Bakım Teknisyeni tablosunu admin yetkisine sahip olan kullanıcı kullanabilir. Hazırda var olan fakat bir yere atanmamış ve personel tipi ‘genel bakım’ olan servis personellerini genel bakım teknisyeni olarak atayabilir.