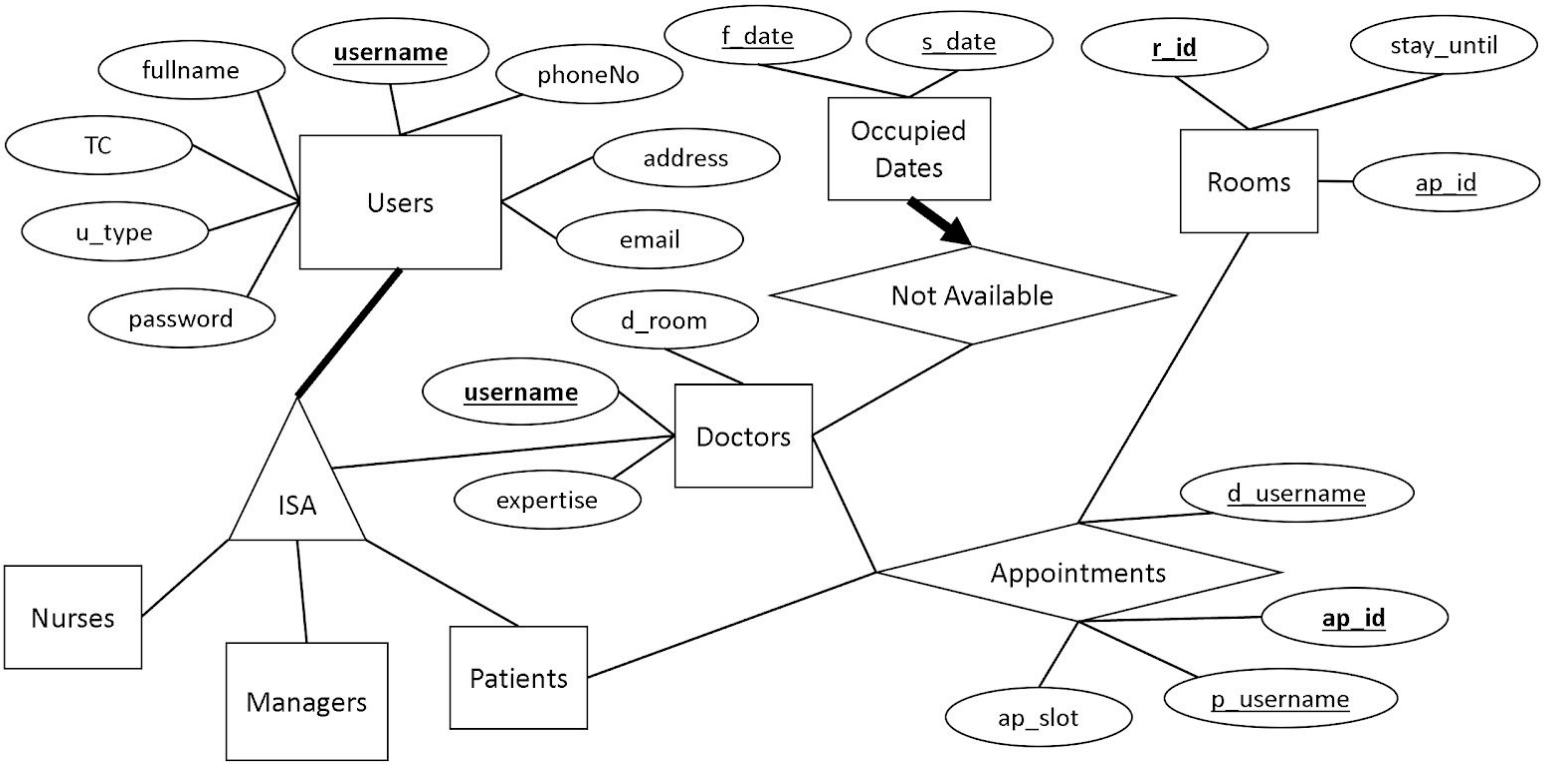
CS 202 Project Design Report

08.12.2019

Mehmet Batu Özmeteler
Kaan Şahin
Cengiz Emre Dedeağaç

Project Description

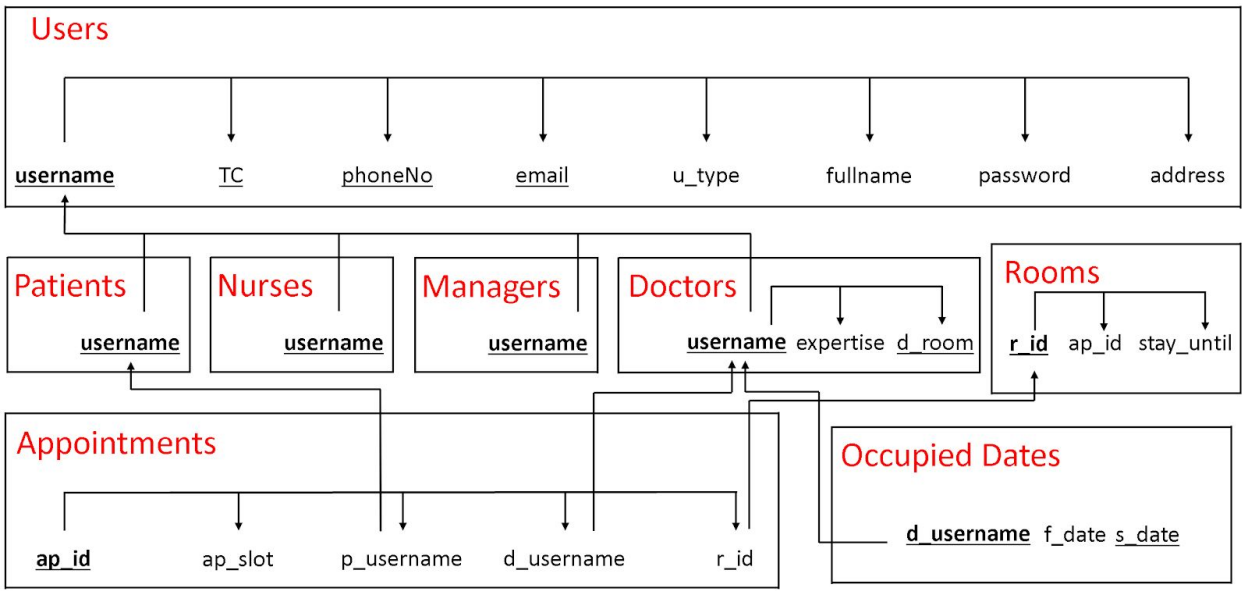
In this project, we are going to develop a hospital management system web application. To illustrate the database schema, here is the E-R diagram representation.



Critical Decision-Making

To clarify the reasoning behind, we should explain why some design decisions are taken. Since there are more than one type of user that shares multiple common attributes, an ISA relationship is used to minimize data redundancy. Secondly, it should also be mentioned that rooms have “ap_id” attribute to facilitate the natural join operation which occurs when availability is listed. Furthermore, to ease the implementation of the backend operation when the doctor takes the day off, “occupied dates” entity and “not available” relationship is utilized. In the room entity, a “stay_until” is used to keep date and time info about until when the doctor decides to have the patient settled in an available room.

Functional dependency diagram is depicted below.



Example Views

```
DROP VIEW IF EXISTS ListDoctors;
DROP VIEW IF EXISTS DoctorAppointmentList;
DROP VIEW IF EXISTS PatientAppointmentList;

CREATE VIEW ListDoctors AS
SELECT fullname, expertise
FROM Users NATURAL JOIN Doctors;

CREATE VIEW DoctorAppointmentList AS
SELECT d_username, fullname, expertise, d_room, ap_slot
FROM (Doctors JOIN Appointments ON username = d_username) JOIN Users ON
Users.username = d_username;

CREATE VIEW PatientAppointmentList AS
SELECT p_username, fullname, ap_slot
FROM (Patients JOIN Appointments ON username = p_username) JOIN Users ON
Users.username = p_username;
```

Example Queries

```
#(???) means whatever input comes from the user
#to check whether this username exists in database
SELECT COUNT(*) FROM Users WHERE username = (???);

#to print the detailed info for any ap_id
SELECT * FROM Appointments WHERE ap_id = (???);

#to print the name of the doctor for a specific expertise
SELECT fullname FROM Doctors natural join Users WHERE expertise = (???);

#to print detailed info about a doctors appointments
SELECT * FROM Appointments WHERE d_username = (???);

#when doctor checks for the availability of the room
SELECT COUNT(*) FROM Rooms WHERE r_id = (???) AND stay_until < now();

#to print detailed info about a patients appointments
SELECT * FROM Appointments WHERE p_username = (???);

#to print detailed info about appointments of a doctor of an expertise
SELECT * FROM Appointments NATURAL JOIN Doctors WHERE expertise = (???);

##doctors can list the availability of the rooms in the system and patient info

#doctors list the availability of the rooms
SELECT r_id, (stay_until < NOW()) FROM Rooms;

#when he clicks detailed info button for a specific room, patient info is printed
SELECT fullname FROM Appointments NATURAL JOIN (SELECT fullname,
p_username FROM (Users NATURAL JOIN Patients)) WHERE r_id = (???);

#to print patient info for the unavailable rooms
SELECT r_id, ap_id, fullname FROM (Rooms NATURAL JOIN Appointments) JOIN
(Patients NATURAL JOIN Users) ON p_username = username;
```

```
#creating a patient in the database
INSERT INTO Users VALUES ('eg00016','Emre Mirac
Gilim',10000000016,'password','emre.gilim@gmail.com','L-17',05323995417,'Patient');
INSERT INTO Patients VALUES('eg00016');

#to add an appointment
INSERT INTO Appointments (d_username,p_username,ap_slot)
VALUES('dk00007','my00028','2019-12-31 15:00:00');

#setting room info when a patient comes in
INSERT INTO Rooms (r_id) VALUES(2038);
UPDATE Rooms SET ap_id = 53, stay_until = ’2019-11-23 15:30:00’ WHERE r_id =
2038;
```