

**CS 202 Project Design Report**

**08.12.2019**

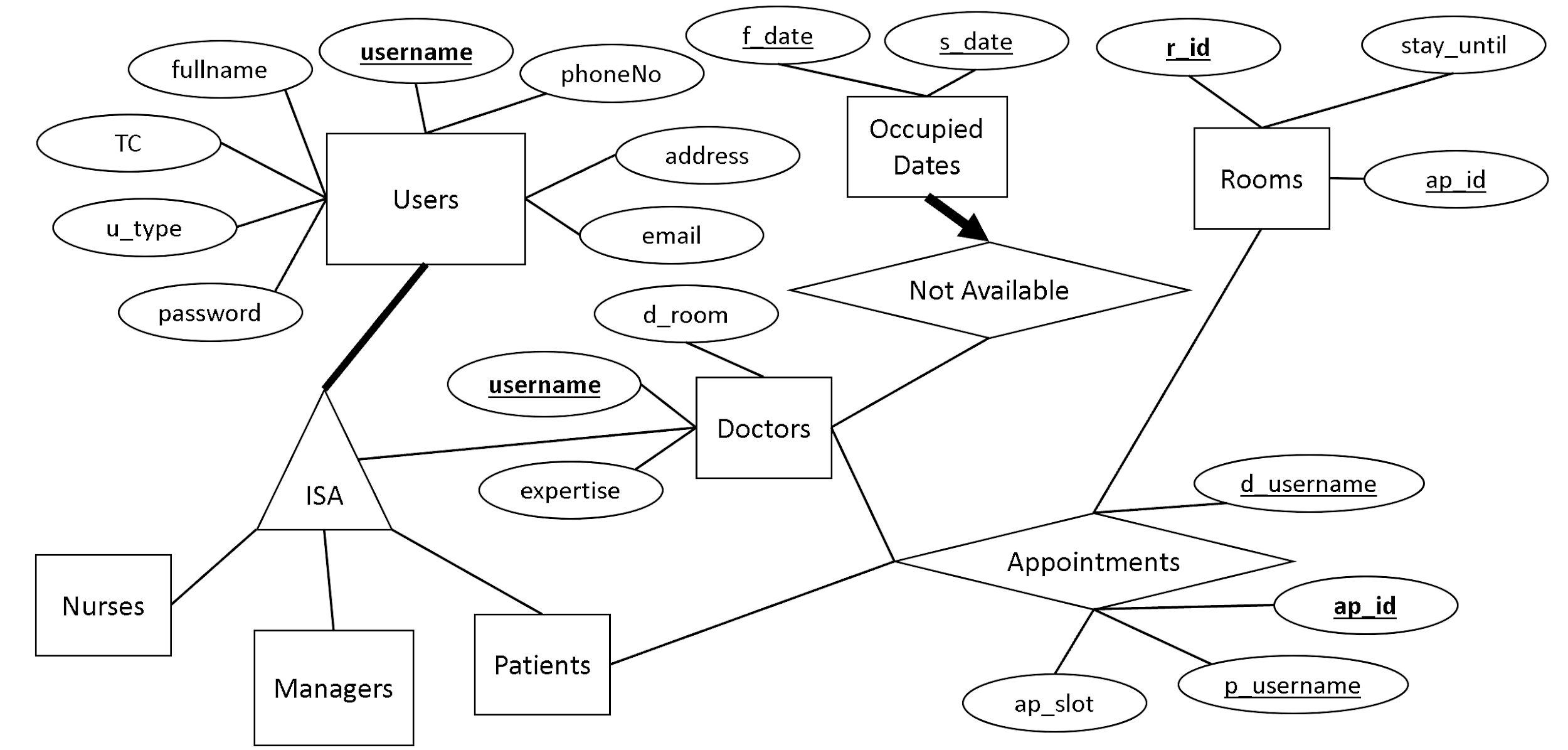
Mehmet Batu Özmeteler

Kaan Şahin

Cengiz Emre Dedeağaç

**Project Description**

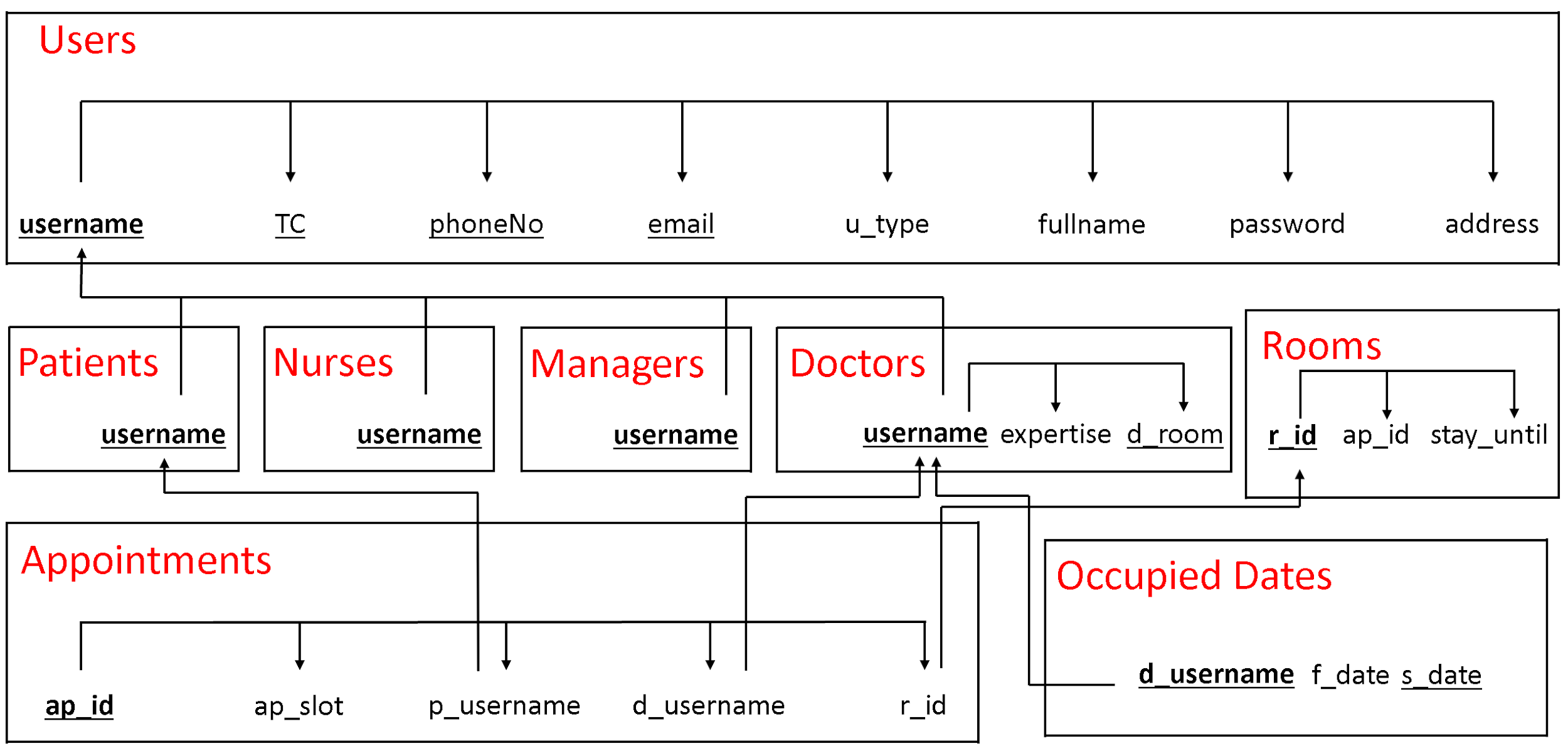
In this project, we are going to develop a hospital management system web application. To illustrate the database schema, here is the E-R diagram representation.



**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;