# Project Part 1

## Team Member:

### Yumeng Chen …

### Chuanqi Xiong cx657

### (a)

The relational schema is as follows:

create table administrators  
(  
 username varchar(255) not null,  
 password varchar(255) not null,  
 email varchar(255) not null,  
 constraint administrators\_username\_uindex  
 unique (username),

primary key (username)  
);

create table appointment  
(  
 appointmentid int not null,  
 providerUsername varchar(255) not null,  
 appointmentDate datetime not null,

slotid int not null,  
 availableNumber int not null,  
 constraint uploadappointment\_providers\_username\_fk  
 foreign key (ProviderUsername) references providers (username),

constraint uploadappointment\_weeklytimeslot\_slotid\_fk  
 foreign key (slotid) references weeklytimeslot (slotid),

primary key (appointmentid)

);

create table definepriority  
(  
 adminUsername varchar(255) not null,  
 patientUsername varchar(255) not null,  
 priorityGroupNumber int not null,  
 primary key (adminUsername, patientUsername),  
 constraint definepriority\_administrators\_username\_fk  
 foreign key (AdminUsername) references administrators (username),  
 constraint definepriority\_patients\_username\_fk  
 foreign key (PatientUsername) references patients (username),  
 constraint definepriority\_prioritygroup\_groupNumber\_fk  
 foreign key (priorityGroupNumber) references prioritygroup (groupNumber)  
);

create table offerappointment  
(  
 appointmentid int not null,  
 adminUsername varchar(255) not null,  
 patientUsername varchar(255) not null,  
 status varchar(255) not null,  
 expireTime datetime not null,  
 primary key (appointmentid, adminUsername, patientUsername),  
 constraint offerappointment\_administrators\_username\_fk  
 foreign key (adminUsername) references administrators (username),  
 constraint offerappointment\_patients\_username\_fk  
 foreign key (PatientUsername) references patients (username),  
 constraint offerappointment\_uploadappointment\_appointmentid\_fk  
 foreign key (appointmentid) references appointment (appointmentid)  
);

create table patients  
(  
 username varchar(255) not null,  
 password varchar(255) not null,  
 firstName varchar(255) not null,  
 lastName varchar(255) not null,  
 SSN int not null,  
 dob date not null,  
 phone varchar(255) not null,  
 email varchar(255) not null,  
 addressLine1 varchar(255) not null,  
 addressLine2 varchar(255) null,  
 city varchar(255) not null,  
 state varchar(255) not null,  
 country varchar(255) not null,  
 zipcode varchar(255) not null,  
 maxDistancePreference float not null,  
 longitude decimal(11, 8) not null,  
 latitude decimal(10, 8) not null,  
 constraint patients\_SSN\_uindex  
 unique (SSN),  
 constraint patients\_username\_uindex  
 unique (username),

primary key (username)  
);

create table prioritygroup  
(  
 groupNumber int not null,  
 eligibleDate date not null,  
 constraint prioritygroup\_groupNumber\_uindex  
 unique (groupNumber),

primary key (groupNumber)  
);

create table providers  
(  
 username varchar(255) not null,  
 password varchar(255) not null,  
 name varchar(255) not null,  
 phone varchar(255) not null,  
 email varchar(255) not null,  
 providerType varchar(255) not null,  
 addressLine1 varchar(255) not null,  
 addressLine2 varchar(255) not null,  
 city varchar(255) not null,  
 state varchar(255) not null,  
 country varchar(255) not null,  
 zipcode varchar(255) not null,  
 longitude decimal(11, 8) not null,  
 latitude decimal(10, 8) not null,  
 constraint providers\_username\_uindex  
 unique (username),

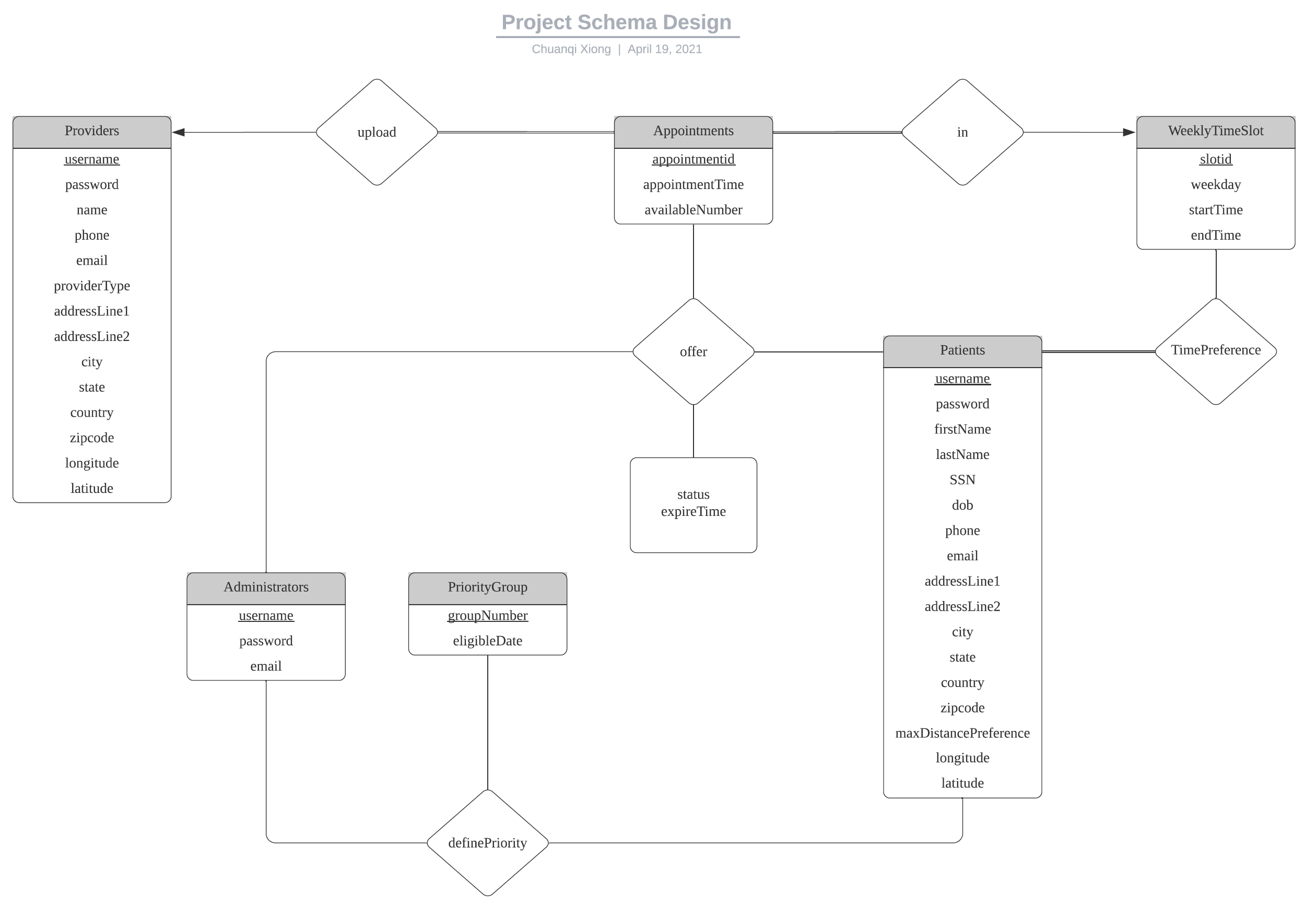
primary key (username)  
);

create table timepreference  
(  
 patientUsername varchar(255) not null,  
 slotid int not null,  
 primary key (patientUsername, slotid),  
 constraint timepreference\_patients\_username\_fk  
 foreign key (patientUsername) references patients (username),  
 constraint timepreference\_weeklytimeslot\_slotid\_fk  
 foreign key (slotid) references weeklytimeslot (slotid)  
);

create table weeklytimeslot  
(  
 slotid int not null,  
 weekday varchar(255) not null,  
 startTime time not null,  
 endTime time not null,

primary key (slotid)  
);

The ER Diagram is as follows:



Further Explanation:

1. Providers, Appointments, WeeklyTimeSlot, Patients, Administrators, PriorityGroup are all Strong Entity Set.
2. Appointments to upload is double-lined for total participation.
3. Appointments to in is double-lined for total participation.
4. Primary Key of each entity set is marked as underlined text.

### (b)

As shown in (a)

### (c)

I will assume all suitable data has already been inserted into the database for this question except some data modification questions such as (1) and (2). For detailed data about how each table looks like, please refer to part (d).

(1)

INSERT INTO patients VALUES ('sampleP1','samplePassword1','p1FirstName','p1LastName',  
 1234123412,'2000-01-12','732-111-1111',  
 'sampleP1@gmail.com','151 Centre St','apt1314','Bayonne',  
 'NJ','US','07002',30.0,40.677300,-74.100640);

(2)

INSERT INTO providers VALUES ('provider1','providerPassword1','CVS','733-111-1111',  
 'provider1@gmail.com','Pharmacy','18-40 Goldsborough Dr',  
 '', 'Bayonne', 'NJ','US','07002',40.672090,-74.105480);  
INSERT INTO weeklytimeslot VALUES(1,1,'08:00:00','12:00:00');  
INSERT INTO appointment VALUES(1,'provider1','2021-04-20 11:00:00' ,1,3);

(3)

WITH satisfiedAppointmentAddr AS(  
 SELECT a.appointmentid, p.longitude, p.latitude  
 FROM appointment a LEFT JOIN providers p ON a.ProviderUsername = p.username  
 WHERE a.slotid IN (  
 SELECT slotid  
 FROM timepreference  
 WHERE patientUsername = 'sampleP1'  
 )  
),  
 appointmentDistance AS(  
 SELECT saa.appointmentid, ROUND(((  
 (acos(sin((saa.latitude \* pi()/180)) \* sin((p.latitude \* pi()/180)) +  
 cos((saa.latitude \* pi()/180)) \* cos((p.latitude \* pi()/180)) \*  
 cos(((saa.longitude- p.longitude) \* pi()/180)))) \* 180/pi()) \* 60  
 \* 1.1515),2) AS distance  
 FROM satisfiedAppointmentAddr saa, patients p  
 WHERE p.username = 'sampleP1'  
 )  
SELECT appointmentid, distance  
FROM appointmentDistance  
ORDER BY distance ASC;

(4)

WITH allPatientStatus AS(  
 SELECT p.username, IFNULL(o.status, 'waiting') AS status  
 FROM patients p LEFT JOIN offerappointment o ON p.username = o.PatientUsername  
)  
SELECT pg.groupNumber,SUM(aps.status='finished') AS alreadyReceived,  
 SUM(aps.status='accepted') AS scheduled,SUM(aps.status='waiting') AS waiting  
FROM prioritygroup pg  
 LEFT JOIN definepriority d ON pg.groupNumber = d.priorityGroupNumber  
 LEFT JOIN allPatientStatus aps ON d.patientUsername = aps.username  
GROUP BY pg.groupNumber;

(5)

SELECT p.firstName, p.lastname, pg.eligibleDate  
FROM patients p  
 LEFT JOIN definepriority d on p.username = d.PatientUsername  
 JOIN prioritygroup pg on d.priorityGroupNumber = pg.groupNumber;

(6)

WITH patientCancelTimes AS(  
 SELECT o.patientUsername, COUNT(o.status) AS cnt  
 FROM offerappointment o  
 WHERE o.status = 'canceled'  
 GROUP BY o.patientUsername  
),  
 patientMissTimes AS(  
 SELECT o.patientUsername, COUNT(o.status) AS cnt  
 FROM offerappointment o  
 WHERE o.status = 'miss'  
 GROUP BY o.patientUsername  
 )  
SELECT pct.patientUsername  
FROM patientCancelTimes pct  
WHERE pct.cnt >= 3  
UNION  
SELECT pmt.patientUsername  
FROM patientMissTimes pmt  
WHERE pmt.cnt >= 2;

(7)

WITH uploadCount AS (  
 SELECT a.providerUsername, COUNT(\*) AS cnt  
 FROM appointment a  
 GROUP BY a.providerUsername  
)  
SELECT p.username, p.name  
FROM uploadCount uc JOIN providers p ON uc.providerUsername = p.username  
WHERE uc.cnt = (SELECT MAX(cnt) FROM uploadCount);

(d)

Data Populating: