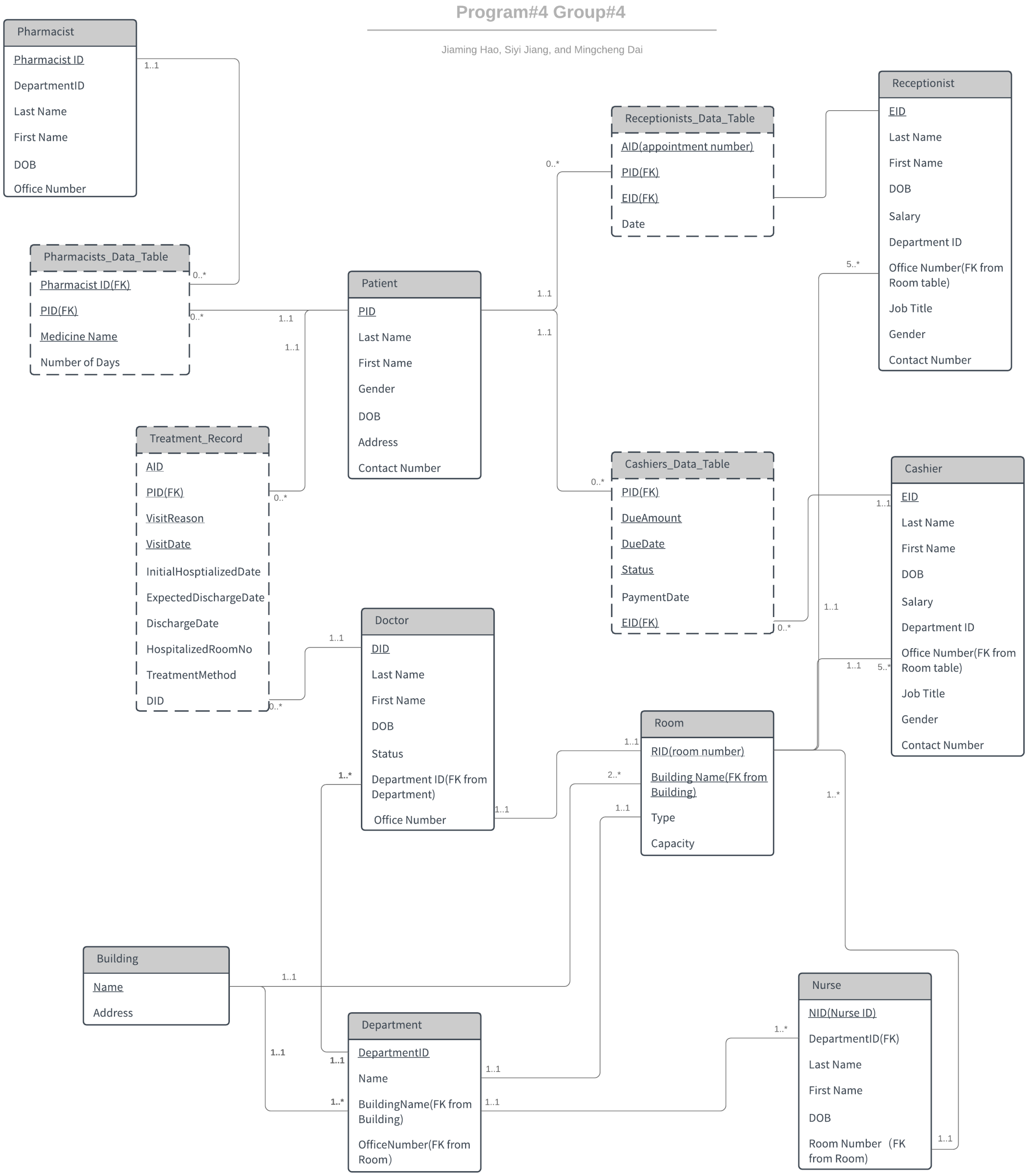
Group #4 (Jiaming Hao, Siyi Jiang, and Mingcheng Dai)

i.) Conceptual database design:

Final ER-Diagram



https://www.lucidchart.com/documents/edit/f9eebf86-3774-49d9-ac7e-9dd7f36e5b57/0?shared=true&

Other Description:

1. We implements aid(appointment id) as the primary key of Receptionists Data Table, Cashiers Data Table, and Treatment Record Table. One visit of a patient will generate one unique aid, which keeps track of all the records of the patient in the hospital.
2. About roomNo: If roomNo is 0, we regard it as no need of hospitalization.

Constraints：

doctor: did unique, officeNo unique, status(one of trainee, tenured, or visiting)
patient: pid unique
nurse: nid unique
pharmacist: pharmacist_id unique, officeNo unique
cashier: eid unique
receptionist: eid unique
treatmentRecord: aid unique, hospitalizedRoomNo(0-300)(0 means no need of hospitalization)
cashiersData: aid unique (This one is not in the spec originally, we added during implementation as PK)
receptionistsData: aid unique
pharmacistsData: pharmacistId unique
room: roomNo unique, roomNo(1-300), capacity(one of single, double, or five), type(one of doctor's offices, pharmacy, medical departments, emergency room, registration office, finance office, or hospitalization room)
department: departmentId unique
building: name unique

ii.) Logical database design:

dmcccccc.doctor(did, lastName, firstName, date_of_birth, status, departmentId, officeNo);
dmcccccc.patient(pid, lastName, firstName, gender, date_of_birth, address, contactNumber);
dmcccccc.nurse(nid, lastName, firstName, date_of_birth, departmentId, roomNo);
dmcccccc.pharmacist(pharmacist_id, departmentId, lastName, firstName, date_of_birth, officeNo);
dmcccccc.cashier(eid, lastName, firstName, date_of_birth, departmentId, officeNo, salary, jobTitle, gender, contactNumber);
dmcccccc.receptionist(eid, lastName, firstName, date_of_birth, departmentId, officeNo, salary int, jobTitle, gender, contactNumber);

dmcccccc.treatmentRecord(aid, pid, visitReason, visitDate, initialHospitalizedDate, expectedDischargeDate, dischargeDate, hospitalizedRoomNo, treatmentMethod, did);
dmcccccc.cashiersData(aid, pid, dueAmount number, dueDate, status, paymentDate date, eid);
dmcccccc.receptionistsData(aid, pid, eid, date);
dmcccccc.pharmacistsData(pharmacistId, pid, medicineName, numberOfDays);
dmcccccc.room(roomNo, buildingName, type, capacity);
dmcccccc.department(departmentId, name, buildingName, officeNo);
dmcccccc.building(name, address);

iii.) Normalization analysis:

Functional Dependency

doctor: {did}→{lastName}, {did}→{firstName}, {did}→{date_of_birth}, {did}→{status}, {did}→{departmentId}, {did}→{officeNo}
patient: {pid}→{lastName}, {pid}→{firstName}, {pid}→{gender}, {pid}→{date_of_birth}, {pid}→{address}, {pid}→{contactNumber}
nurse: {nid}→{lastName}, {nid}→{firstName}, {nid}→{date_of_birth}, {nid}→{departmentId}, {nid}→{roomNo}
pharmacist: {pharmacist_id}→{departmentId},{pharmacist_id}→{lastName}, {pharmacist_id}→{firstName}, {pharmacist_id}→{date_of_birth}, {pharmacist_id}→{officeNo}
cashier: {eid}→{lastName}, {eid}→{firstName}, {eid}→{date_of_birth}, {eid}→{departmentId}, {eid}→{officeNo}, {eid}→{salary}, {eid}→{jobTitle}, {eid}→{gender}, {eid}→{contactNumber}
receptionist: {eid}→{lastName}, {eid}→{firstName},{eid}→{date_of_birth}, {eid}→{departmentId}, {eid}→{officeNo}, {eid}→{salary},{eid}→{jobTitle}, {eid}→{gender}, {eid}→{contactNumber}
treatmentRecord: {aid}→{pid}, {aid}→{visitReason}, {aid}→{visitDate}, {aid}→{initialHospitalizedDate}, {aid}→{expectedDischargeDate}, {aid}→{dischargeDate}, {aid}→{hospitalizedRoomNo}, {aid}→{treatmentMethod}, {aid}→{did}
cashiersData: {aid}→{pid}, {aid}→{dueAmount}, {aid}→{dueDate}, {aid}→{status}, {aid}→{paymentDate}, {aid}→{eid}
receptionistsData: {aid}→{pid}, {aid}→{eid}, {aid}→{date}
pharmacistsData: {pharmacistId, pid, medicineName}→{numberOfDays}
room: {roomNo, buildingName}→{type}, {roomNo, buildingName}→{capacity}
department: {departmentId}→{name}, {departmentId}→{buildingName}
building: {name}→{address}

Justify Why Your Design Adheres To 3NF

First, there are no relations have set-valued attributes, so the design satisfies 1st normal form.

Second, in every relation above, every non-prime attribute in that relation is FFD upon on every CK of that relation, so the design satisfies 2nd normal form.
Third, for every FD listed above, the left hand sides is a super key of the relation, so the design satisfies 3rd normal form.

iv.) Query description:

Give the name and birth date of a receptionist, search for the most frequently patient he accessed, display the pid, patient full name, and the medicine name that the patient most frequently bought.
The most frequently visited patient should be regard as important patient. That kind of patient should be took care of carefully. And this query help staffs to find those kind of patients out. In the meanwhile, it shows the medicine that the patient frequently use, which helps doctor to analyze for patients' illness.