

<u>Initially:</u>	yes?
Show me your tables in the database(Should have at least 6 tables => Doctors, Patients, DocPat, Appointments, Prescriptions, Notes)	
Your database should initiall have following entries in these 6 tables... (*) Doctor2 from Doctors table, Patient2 from Patients table, Entry links Doctor2 and Patient2 from DocPat table, Appointment2 from Appointments table, PrescriptionC from Prescriptions table, Note2 from Notes table	
<i>Test case 1 (Testing Doctors, Patients, and DocPat tables)</i>	
<u>From DoctorGUI:</u>	
1. Create a new doctor (Doctor1) to Doctors' table (Test INSERT in Doctors)	
2. Edit Doctor1's info from Doctors' table (Test EDIT in Doctors)	
3. Create a new patient (Patient1) into the database (Test INSERT in Patients)	
4. DocPat table will this entry which links both DocID and PatID (Test INSERT in DocPat)	
5. Search and display this patient1's info from the doctors' database (Test SELECT in Patients)	
6. Edit patient1's info from the doctors' databse (Test UPDATE in Patients)	
<i>Test case 2 (Testing Appointments table)</i>	
<u>From PatientGUI:</u>	
7. Have Patient1 to make appointment (Appointment1) with Doctor1 using Socket2 communication	
8. After appointment has been made, Appointments table now should have Appointment1 entry (Test INSERT in Appointments)	
<u>From DoctorGUI:</u>	
9. After seeing Patient1, Doctor1 assigned two prescriptions for this patient (PrescriptionA & PrescriptionB). Prescriptions table should have two more entries (Test INSERT in Prescriptions)	
10. Doctor1 then change PrescriptionA's name in Prescriptions table (Test UPDATE in Prescriptions)	
11. Doctor1 then delete prescription B from Prescriptions table (Test DELETE in Prescriptions)	
<i>Test case 3 (Testing Notes table)</i>	
<u>From DoctorGUI:</u>	
12. Doctor1 can add a note (Note1) which relates to Appointment1 (Test INSERT in Notes)	
13. Doctor1 can edit a Note1 (Test UPDATE in Notes)	
14. Doctor1 can now view Note1 in DoctorGUI	
<i>Test case 4 (Testing PatientsP, DoctorsP, and PrescriptionsP table) -> Extra credits</i>	
<u>From PharmacyGUI:</u>	
15. Pharmacy first create Doctor1's record into its DoctorsP table (Test INSERT in DoctorsP)	
16. Pharmacy then create this Patient1's record into its PatientsP table (Test INSERT in PatientsP)	
<u>From DoctorGUI:</u>	
17. Doctor1 then send prescriptionA to pharmcy using RMI	
<u>From PharmacyGUI:</u>	

18. Pharmacy received the prescriptionA, and insert PrescriptionA into PrescriptionsP table (Test INSERT in PrescriptionsP table)	
19. Pharmacy can view prescriptionA's info, along with Doctor1's info and Patient1's info in its GUI (Test SELECT in PrescriptionsP, PatientsP, DoctorsP tables)	
Test case 5 (Testing Doctors, Patients, DocPat, Appointments, Prescriptions, and Notes tables)	
<u>From DoctorGUI:</u>	
20. From Doctor1's GUI, delete Patient1. (Test DELETE in Patients table)	
21. Now Doctor1 shouldn't see Patient1's info, Appointment1, PrescriptionA, and Note1 from its GUI	
<u>From MySQL Browser (Checking database, wrapping up)</u>	
22. Show me that Patient1's info, Appointment1, PrescriptionA, Notes1 entries are all gone from Patients, DocPat, Appointments, Prescriptions, and Notes tables	
<u>Misc:</u>	
Be able to answer my question(s) during demo (45)	
Demo some fancy works for some extra points (5)	
Submit your rar/zip file on time (10)	
Demo your rar/zip file on time (10)	
Has a short report included in your zip/rar file (10)	
Has a readme.txt file explaining how to I can access your databse & tables (4)	
TA is able to compile/run your work from your submitted version (5)	