**Cosas a corregir:**

* Los métodos getController tienen que ser estáticos y no van definidos en la interfaz.
* Los search:
* El search by id tiene esta plantilla:
* SELECT FROM tableName WHERE id= *id que nos han pasado*
* El return debe ser lo último que se haga, antes hay que cerrar los statements
* En Rooms:
* Necesitamos un select con group by room type
* Y un update del coste de la habitación: al método habrá que pasarle el nuevo coste y el tipo de habitación en la que se modifica el coste:

UPDATE bill

SET cost\_per\_day = *coste que nos han pasado*

WHERE room\_type = *room type que nos han pasado*

* Esto es un poco tiquismiquis pero los search en los que se devuelva todo el contenido de la tabla, yo prefiero un getAll*Nombredelaclase.*
* En los getAll (search) he visto que en el de doctor no se añaden a la lista los doctores y la lista se devuelve antes de cerrar el statement.
* Para el search by id fijaros en el método de search by id de la clase nurse: solo hay que cambiar una cosa
* Si buscamos by id, al ser este único, solo devolverá un elemento en el result set con lo cual sobra el bucle. Como otra opción devuelve null, en cualquier caso sigue sobrando el bucle: EL OBJETO SE DEVUELVE LO ULTIMO, primero hay que cerrar los statements.
* La otra opción es hacer una lista en la que solo va a haber un elemento, por si se da el caso de que se devuelvan más de uno, y DEVOLVER LA LISTA AL FINAL, LOS STATEMENTS HAY QUE CERRARLOS ANTES.
* Los update:

No entiendo que hacen, EL ASTERISCO SIGNIFICA ABSOLUTAMENTE TODO LO DE LA TABLA.

Tiene que haber un SET en el update donde se especifique el/los campo/s a modificar y para seleccionar una fila concreta de las tablas es necesario poner una condición (WHERE)

* Los insert:
* En aquellas clases que tienen foto, no hay que hacer el file, en el objeto nurse ya está el array de bits que representan la foto! Asi que directamente hay que hacer setBytes (3, objeto.getPhoto() ).
* IRAN SALIENDO NUEVOS SELECTS QUE HAYA QUE IMPLEMENTAR CONFORME VAYA VIENDO LA FUNCIONALIDAD DE LA INTERFAZ
* POR FAVOR, PROBAD CADA FUNCION QUE HAGAIS ANTES DE SEGUIR PICANDO CODIGO (MIRAD EN EL SQLITE BROWSER SI LAS TABLAS SE MODIFICAN)
* Y CUIDADO CON LOS NOMBRES DE LAS TABLAS, QUE SIEMPRE SEAN LOS MISMOS. En plan, que no sea en uno INSERT INTO Bills, y en otro lado UPDATE bill (es un ejemplo, no me he fijado si se da el caso, pero es algo en lo que es muy fácil equivocarse). A ser posible, poned en un Word los nombres usados para las tablas y las columnas, para que el createTables sea rápido luego.
* Change in the inserts and selects and updates the parsing to LocalDate: setDate(Date.valueof(patient.getDob())) and select patient.setDob(date.toLocalDate());
* Poner en los ids autoincrement (en las tables)

**To do:**

-Finish the UML diagram (the controller one)

-Check insert, delete, update… every method from patients, bills, treatments, nurse and room.

- Poner autoincrement en todos los id in the tables. Change “speciality” to “specialty”.

- Tables: set on delete and on update of the foreign keys (we need to discuss this beforehand)

- We are not going to insert the ids: the database will automatically do this for us

- We wont update the ids: take a look at DoctorUpdate

- Username can’t be the primary key since it is not an integer: we need to add a column in the tables that is the id and then add unique to the column username in the tables definition.

- Pojo Bills will have patient as attribute: so we need to have a method that is getBillsFromPatient instead of searching bills by id: we will pass a patient and look where patient\_id in bills = patient.getId();

- Delete the “deletedoctorWithoutId” and “deleteNurseWithoudId” method

- We don’t know if the searchById is going to be useful

- We need to add to Doctor: searchBySpeciality and searchByschedule, searchByName

- Add to Nurse searchBySchedule, searchByName

- Add “addNurseToPatient” method and “addPatientToNurse” method

-Add “deleteNurseFromPatient” method and “deletePatientFromNurse” method.

- Add method getNurseWithoutPatient(everything but the list of patients), getPatientsFromNurse

- add getPtientWithoutTreatmentsAndBills method.

- Add method getBillsFromPtient

- Add method getTreatmentsFromPateints.

- Add method

- add to Room a list of patients. (pojo).

-add addPatientToRoom method (room controller and room interface)

- add deletepatientfromRoom method (room controller and room interface)

- add to treatment an attribute “patient” (pojo).

- add a list of treatments to doctor (in the pojo).

-in nurse:add setPatient, setListPatients(a list of patients) and getPatient and getListPatients methods.

- add listOfPatients to room pojo.

In room (pojo): setPatient, setMultiplePatients, getMultiplePatients

In treatment (pojo): attribute patient , setters and getters.