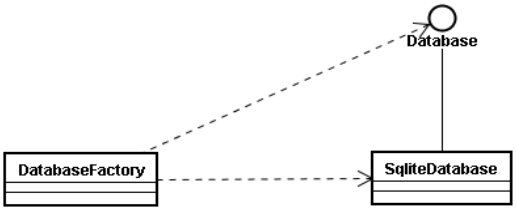
Developer's report

Task: Implementation

Name: Le Xuan Hoan

1. Refactor the Database Wrapper

- I applied **Factory** pattern for the database wrapper. In my opinion, at first this refactoring may not look not really helpful, but if we plan to support more Database systems later, it will be more convenient for user of the wrapper, they don't need to know the concrete implementation wrapper of the DB they intend to use

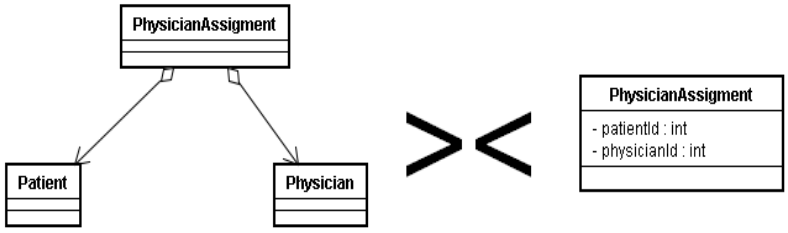


2. Update database model

- I refactored the database schema, removed unnecessary fields, created demo data for the program, you can see the SQL script to create the database model and demo data here https://github.com/pufm2/hms/blob/master/HMS.sql

3. Implementation of class model

- I implemented most of necessary business behaviors for hms model (puf.m2.hms.model), each of entities (Physician, Patient ...) has methods such as save, update, load ... from/to database.
- Abstraction is used intensively, relationships amongst DB schemas are reflected fully and at a higher level in our object model. I mean, for example: each instance of **PhysicianAssigment** loaded from database will refer correspondingly to an instance of the Physician and an instance of the Patient (you can see the difference at the figure below)



- Actually I thought about Hibernate, but seems it's too late to apply such a big change, Hibernate is effective for the object-relational mapping problem but it is complicated and it does not support SQLite officially)

- I also used a simple caching technique (in an effort to mimic Hibernate), each time when an object is loaded from database, it will be cached, and returned for the same requests later. Personally I think the caching is reasonable, it can help to make sure synchronization of data. For example: if we have two instances (of class Patient) of the same patient tuple from database. We change one (fix the name for the patient), and of course this change is not conveyed to the other instance, many conflicts can happen later.

4. Pair programming

- As a practice of agile methodology, I work collaboratively with the remaining programmer who is responsible for UI, and help to review his source code, give some advices about refactoring.

5. Next iteration

- For next iteration, the followings will be taken into account:
  i. Completion of business logics of hms model (puf.m2.hms.model). After that, it's only changed by discussion of all team
  ii. Play another role (it will be unified with team later)