**BABYLON STUDIO**

The aim of the database is to manage patient´s clinical histories in a hospital and allow them to ask for appointments being able to choose between the doctors belonging to the medical team of each speciality as well as to allow the doctors see their pending appointments.

The program will also generate a pdf of the appointments (either of patients or doctors) and the clinical records if needed. In order to generate a pdf of a clinical record, a password may be introduced, complying to our data protection policy.

Either doctors and patients will be able to manage their appointments being able to ask for them, modify them or cancel them. When asking for appointments, the patient will have a graphic available in order for him to know the hours which have more people affluency. In addition to this, there is a personalized calendar for each doctor and patient so that it is more comfortable for them to manage their appointments.

In case someone (either patients or doctors) forget his credentials, the program will send him and email so that he can recover them.

To fulfil all these functionalities, the database has an user interface to make the users (patients, doctors and hospital´s receptionists) easier to access to the advantages this program offers them.

In order to have an structured user interface, Babylon studio has three types of accounts: administrator, doctor and patient.

* The administrator account manage the creation of doctors accounts so that nobody but doctors can create a doctor account. This type of account also manages problems such as a lost of password by any doctor or patient and the elimination of any type of account.
* The doctor account is created for doctors to be able to see all their pending appointments, the patients they have associated and the clinical history of his patients as well as modifying this last ones.
* The patient account aim is to make the patients easy to get, modify or eliminate an appointment and to visualize all the appointments they have. Patients would never have access to their clinical histories nor modify them.

Both doctors and patients account have all their personal information and they can include a photo in order to have a visual look of them. This information and the photo can be changed by patients and doctors directly without having to do it through an administrator account.

Internally, the database is developed by using JDBC and JPA. In order to use both to build up the database, an interface with a connector method has been created. This way, the database stablishes connection with JDBC and JPA depending on the needs in each situation.

In order to build the database up, we created an ER diagram and an UML diagram so that we had an structured schema to start writing code.

The ER diagram has 11 entities with many attributes that describe the most important features of each entity.

In order to represent the entities and their attributes in java, we created what are called “the pojos”. They are basic java classes. The class is the entity itself and the attributes are the ER diagram attributes. In the case of the entities that are related, in each class of the relation there is an object of the other class. That way they are joined. For example, the entity doctor is related with patient, so the pojo doctor has an object that is patients and vice versa.

In order to put all the information about all the entities of the ER diagram, tables has been created. Each entity has a table and the many to many relationship has a mapping table too.

An UML diagram has been done too. This has been done so that everyone can see an overall schema of the database. These have allowed us, the programmers, to write structured and perfectly connected java code to develop the database.

Finally, the database can be exported and imported to XML. This is done for future improvements in which a webpage can be created so that the users do not need to have the program to work with it.

To sum up, a brief summary of the things each user can do is the following:

* Administrator:
* Create doctor accounts.
* Send emails when credentials have been lost.
* Patient:
* Create an account.
* Ask for appointments.
* Modify appointments.
* Delete appointments.
* Modify personal information.
* Browse a photo.
* Take a picture.
* See a personal calendar with all his/her appointments.
* Doctor:
* See pending appointments.
* See a personal calendar with all his/her appointments.
* Modify personal information.
* Browse a photo.
* Take a picture.
* See patients´ clinical histories
* Modify patients´ clinical histories
* Delete appointments.
* Additional features:
* Export database.
* Import database.
* Export appointments to pfd.
* Export clinical histories to pdf (introducing a password).
* Print appointments through the pdf.
* Print clinical histories through the pdf.