**Hospital Database Management System - Design Document**

**By Chanel Cheng, Michael Cote, and Shu-jun Pierre Lin**

**General Overview:**

The purpose of this hospital database system is to hold the hospital enterprise data in a database which provides data to users, namely, doctors, nurses, and administrative staff. The database allows for easy access between all staff members and files. Additionally, it implements proper security permissions by use of encryption, that is, it prevents changes from unauthorized staff.

**User Guide:**

The user experience for this application is dependent on one of the three aforementioned staff users. Specific staff are restricted in the actions they can perform. For example, doctors cannot create charts if they already have a new patient - only nurses can. If your hospital’s workflow conflicts with the coded implementation, please contact us for software customization and customer needs.

**General:**

* **Login:** After starting the application, the user will be greeted with a login screen where they must enter their username and password. There is no limit on the amount of times you may login. However, if you are consistently unable to login after many, please contact us.
* **Post-Login:** Depending on the staff user that your database administrator created, your profile will decide contain a different screen. This is built such that only you are permitted to view your profile and be able to perform your operations.

**Admin:**

* **Create Prescribed Drug Report:** A report that lists for each doctor the name and the total amount of each drug that the doctor prescribed in a specified period of time.Additionally, drugs that he or she did not prescribe in that period of time will not be listed.
* **List of Total Drug Prescribed:** For each category of drugs, a list will include the total amount prescribed for each drug in that category in a specified period of time. The report also contains a total for each category.
* **List of Given Diagnosis of Medications:** A list with a given diagnosis all possible medications that have been prescribed over time after that diagnosis (over all charts). The list is ordered by the frequency of the medication for the given diagnosis.
* **List of All Diagnoses for Given Drug:** A list for a given drug all the diagnoses that have been made before prescribing the drug (over all charts) will be created. The list is ordered by the average amount of the drug prescribed for the diagnoses.

**Nurse:**

* **Create new chart:** Nurses are able to create new charts for each patient entering the hospital. If the patient has not stayed at the hospital before, the nurse is able to enter new patient data for the incoming patient. If the patient has stayed at the hospital before, the system will import their previous patient data. A new chart can only be opened for a patient it any existing charts they have have been closed. Date and time of admission are entered automatically and a unique, randomized chart id number is assigned.
* **Closing a chart:** For any currently open chart for a patient currently staying at the hospital, the nurse is able to close a chart when a patient is discharged from the hospital with the discharge time being entered automatically.
* **Patient Selection:** Nurses are able to view patient information in the same as doctors. See **Patient Selection** in the Doctors section for further details.
* **Entry of a Symptom:** Nurses are able to enter symptoms the same way a doctor can. See **Entry of a Symptom** in the Doctors section for further details.

**Doctor:**

* **Patient Selection:** Doctors will use this database to access information on their patients, and therefore before being able to append changes, the doctor must specify which patient info they wish to access. This needs to be inputted as the patient's health care number (H.C.N.O).
* **Chart Selection:** After a doctor has selected the patient they wish to view the history of, they will be presented with all charts ever created for the patient (representative of individual visits). Now the doctor must decide whether they want to view the patient’s history, or append a change to a new chart. To select a chart the doctor will just have to input the index of the chart presented on the left side of each chart.
* **Entry of a symptom or diagnosis:** After both a patient and a chart has been selected the doctor can now make changes to the patient's open chart. He can add a symptom or a diagnosis by simply selecting to input a symptom or diagnosis. The doctor just has to input the symptom or diagnosis name. The database will automatically input the date and the doctor who inputted the data.
* **Entry of a medication:** To enter a medication prescription it is the same process as a symptom or diagnosis but the doctor will have to specify exact details of the prescription. The doctor will be prompted for these details.

**Testing Strategy:**

* **Bugs:**
  + (FIXED) Printing results out in unicode, on terminal
  + (FIXED) Missing top result of sqlite3 query
  + (FIXED) Logic error in queries
  + (FIXED) Syntactical and semantic errors with Python 2.7
  + (FIXED) Changes to the database were not being committed after alteration
* **Testing Strategy:**
  + UI functionality was tested through the terminal
  + Functionality of queries tested with sample data, with the simple approach of temporarily inserting data that meet a certain condition, then using the system to retrieve the data.
* The testing strategy discusses your general strategy for testing, with the scenarios being tested, the coverage of your test cases and (if applicable) some statistics on the number of bugs found and the nature of those bugs.

**Group Work Strategy:**

* **Michael Cote:**
  + **Login**
  + **Doctors**
  + **Design Document**
  + **Total Time Spent: 18 hours**
* **Shu-jun Pierre Lin:**
  + **Nurses**
  + **Design Document**
  + **Total Time Spent: 18 hours**
* **Chanel Cheng:**
  + **Administrative User**
  + **Design Document**
  + **Total Time Spent: 23 hours**

Files needed for the project were kept on a shared drive for easy collaboration. Changes to database system files were initially done on each collaborators separate machines and tested before being uploaded to the shared drive when functional to ensure good version control and that code created by each collaborator would work with that of the other group members.