Dokumentation Jacob Jäger

**Datenbank Klasse**

Die Datenbank Klasse dient zur Erstellung und Verwaltung von Daten innerhalb der Tabellen Patienten und ICD- Codes. Um die benötigten Bibliotheken verwenden zu können muss das Sql Modul in der CMakeLists- Datei mit den anderen verknüpft werden. Zudem wird dort die SQLite Datenbank von den Source Files in den build Ordner mithilfe des configure\_file Befehls kopiert. Dies hat den Vorteil, dass man so ungewollte Änderungen an der Datenbank einfach rückgängig machen kann, indem man dem build Ordner löscht und dann das Projekt neu erstellt.

Im Konstruktor der Klasse wird die Datenbank geöffnet. Das Datenbankobjekt wird dabei ganz am Anfang in der main Methode erstellt. Wenn die Datenbank nicht erfolgreich geöffnet wurde, wird dabei noch kein Fehler angezeigt, erst wenn man tatsächlich eine Datenbankanfrage durchführt. Im Destruktor wird die Datenbank wieder geschlossen.

insertPatient Methode:

In dieser Methode wird ein neuer Patient zu der Tabelle Patienten hinzugefügt. Dies geschieht mit prepared statements um SQL- Injections vorzubeugen. Wenn es bei der Datenbankanfrage einen Fehler gibt wird ein runtime error geworfen.

editPatient Methode:

Hier können die Daten eines bestehenden Patienten verändert werden. Die Methode bekommt als Parameter ein Objekt der Klasse io\_data mit den aktualisierten Daten des Patienten. Es werden wieder prepared statements verwendet und bei Fehlern ein runtime error geworfen.

getPatientbyColumn Methode:

Diese Methode findet alle Patienten in einer bestimmten Spalte, deren Daten mit dem anderen Parameter übereinstimmen. Es wird ein Vektor mit allen gefundenen Patienten zurückgegeben. Wenn die gesuchte Spalte die Patienten-ID ist werden keine substrings durchsucht, sonst schon.

getICD\_Code\_Information Methode:

Hier können die Daten zu einem dazugehörigen ICD- Code abgerufen werden. Die Methode verwendet wieder prepared statements und es wird ein Vektor mit den gefundenen Daten zurückgegeben.

**Backend für Datensatz anzeigen, Datensatz bearbeiten und Datensatz erstellen**

In der Klasse Datensatz anzeigen wird im Konstruktor das Datensatz anzeigen UI-File ausgefüllt. Hier werden neben den Informationen aus der Patienten Tabelle auch die ICD- Code Informationen und das Alter des Patienten angezeigt. Von hier aus kommt mach auch auf das Patient bearbeiten Fenster. Um die Berechtigung des Nutzers zu überprüfen, wird hier auch das mainwindow Attribut gesetzt.

Die Klasse Datensatz bearbeiten dient zur Bearbeitung eines Patienten, sowie der Erstellung. Zur Unterscheidung wird im Konstruktor entweder -1 für die Erstellung oder die tatsächliche ID, um einen Patienten zu bearbeiten, übergeben. Das UI-File wird dann einfach nicht ausgefüllt oder mit den Informationen des Patienten mit der jeweiligen ID. Wenn dann der Speicher- Button geklickt wird, werden die Informationen aus dem UI- File in das erstellte io\_data Objekt gespeichert. Dann wird die jeweilige Datenbank Anfrage durchgeführt. Alle Ausnahmen werden dabei in der mainwindow Klasse gefangen.

**Backend für Patientensuche und Filterfunktion**

Wenn man die Enter- Taste oder auf die Lupe drückt wird die getPatientbyColumn Methode der Datenbank Klasse aufgerufen und es wird nach der Spalte gesucht, die mit dem Filter zuvor ausgewählt wurde. Die Methode geht dann durch alle Patienten in dem zurückgegebenen Vektor und fügt sie zu der Tabelle hinzu. Wenn die ID ausgewählt wird, dann wird nicht nach Sub Strings gesucht, sonst schon. Die Filterfunktion bei keiner Auswahl die Spalte PatientID ausgewählt-

**Berechtigungssystem**

Es gibt 3 Berechtigungen: Lese-, Schreib- und Admin- Berechtigung. Der user wird beim Login dynamisch erstellt und beim Schließen des Main Window wieder gelöscht. Bevor man ein Ui-File öffnen kann wird die Berechtigung geprüft, die durch ein Attribut in der user Klasse dargestellt werden. Nur ein Admin kann neue user erstellen und mit Leseberechtigung kann man nur die Patienten als CSV- Datei exportieren und Patienten anzeigen.

**Darkmode Umstellung in allen offenen Fenstern**

Das Style- Sheet wird bei dem öffnen jeder Datei gesetzt. Wenn man den Modus ändert, während andere Fenster geöffnet sind, dann wird durch alle Fenster iteriert und alle werden automatisch geändert.

**Generelle Organisation**

Dazu gehören Überlegungen zur Struktur des Programms, Aufgabenverteilung, Zusammenführen von Zweigen und Behebung von Merge- Konflikten.