

- Dürfen neue Querys nur abgesetzt werden wenn ein Editorfenster geschlossen wurde? (wie garantiere ich Konsistente Daten)

## Design Patterns

- Schichtenmodel: eine Schicht kann nur den Dienst ihres unteren Nachbarn nutzen, sie darf keine Schichten überspringen
- Abbildung der Ideen auf Pakete, Klassen & Schnittstellen
- Klassen sollten nicht zuviel über andere Klassen wissen
- InterClass Kommunikation über wohldefinierte Schnittstellen so dass Änderungen mgl

## Zuklärende Fragen:

- Threads? Damit z.B. nach absetzen eines Querys weiter im Programm navigiert/gearbeitet werden kann auch wenn die Ausführung des Querys lange dauert
- Eigene Datenstruktur für das MySQL ResultSet? Damit behalten wir Flexibilität der Datenbankversion / -firma gegenüber, da das jeweilige ResultSet seriell in unser Format übersetzt wird. Diese Übersetzungsschnittstelle müsste dann also nur ausgetauscht werden.

## Konzepte:

- Database Klasse sollte abstrakt sein und SingletonPattern verwenden (es gibt nur eine Instanz), die Spezialisierung trägt möglichen Änderungen des Db-Typs Rechnung

### GUI (Sepp, Jakob)

- Welche Daten werden pro Bereich benötigt? → müssen diese aufbereitet werden
- → UseCase als Grundlage
- Über MVC gelesen, werden Threads im GUI Controller gestartet
- Wie wird dem Nutzer symbolisiert, dass seine Aktion in Bearbeitung ist, das Programm darf nicht blockieren
- Gestaltung Ivo
- Escapetasten Funktionalität

### DB Grundlage (Rick)

- Schnittstellen definieren

### Suche (Rick)

- → Funktionsweise
- Autovervollständigung
- Filter ?

### Export / Druck (Ivo)

- Schnittstelle definieren
- Vorlagen?

### Mail (Jakob)

- Schnittstelle definieren
- Vorlagen?

### Import (Sepp)

- Zuordnung von .xls zu Db Struktur