

Pflichtenheft

Projekt: Partiturverfahren

Auftraggeber: Benjamin Messner

+43 69915150062

Auftragnehmer: Dominik Senn, Felix Weissteiner, Elias Thaler

1. Zielbestimmung

Das Ziel des Systems ist es eine Verwaltung eines Notenarchives zu gewährleisten. Im Archiv befinden sich verschiedene Partituren (Originalausgabe eines Musikstücks), diese Partituren sollen digitalisiert werden um weiteres Arbeiten mit dessen erleichtert. Der User hat die Möglichkeit eine Partitur einzulagern, zu verschieben und zu löschen / auszulagern. Eine Partitur einzulagern werden folgende Informationen benötigt: Raum, Fach, Schublade, Komponist und Titel. Beim Verschieben wird das Musikstück von einem Regal ins andere verschoben oder von Raum zu Raum verschoben. Die Daten werden in einer lokalen SQL Datenbank hinterlegt bzw. gespeichert.

Das Programm wird speziell für Notenwarte von Musikkapellen entwickelt, damit diesen ein großer Arbeitsschritt beim Verwalten des Archivs abgenommen wird. Hauptsächlich bei der Suche und beim Einordnen wird die Arbeit vereinfacht.

2. Produkteinsatz

Das Produkt wird zur digitalen Archivierung des Notenarchivs angewandt. Durch das Programm weiß der Kunde immer wo die Partitur liegt, da er immer nachsehen kann wo die Partitur liegt. Das Programm bringt den Vorteil mit sich, dass sich der Verwaltungsaufwand um ein Vielfaches reduziert.

2.1 Beschreibung des Problembereichs

Begriffserklärungen:

Datenbank:

Eine Art Schließfach wo die eingegebenen Daten abgelegt werden und bei Bedarf wiedergeholt werden können.

System: Ein Programm, welches eine gewisse Tätigkeit durchführt.

Schublade:

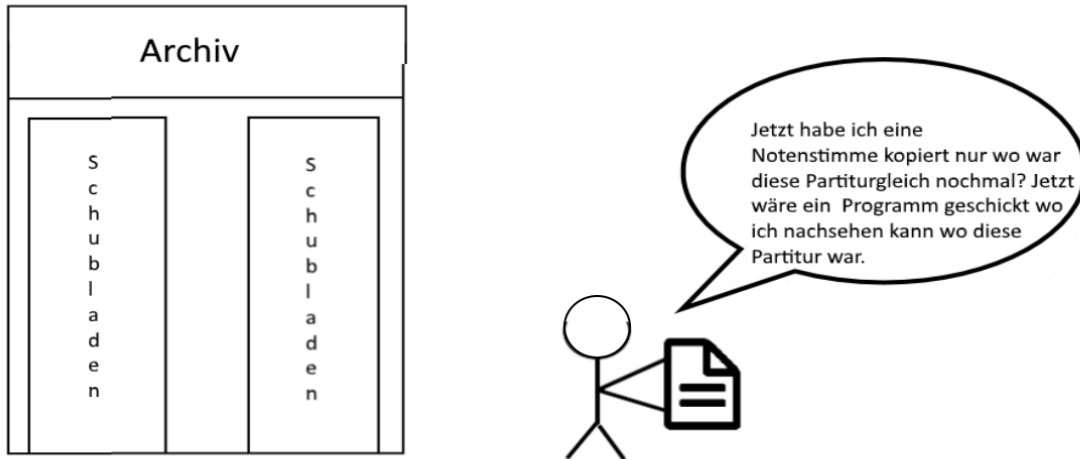
Die Schublade ist die Oberstruktur des Fachs. Eine Schublade enthält mehrere Fächer.

Fach:

Unterteilung der Schublade.

Partitur:

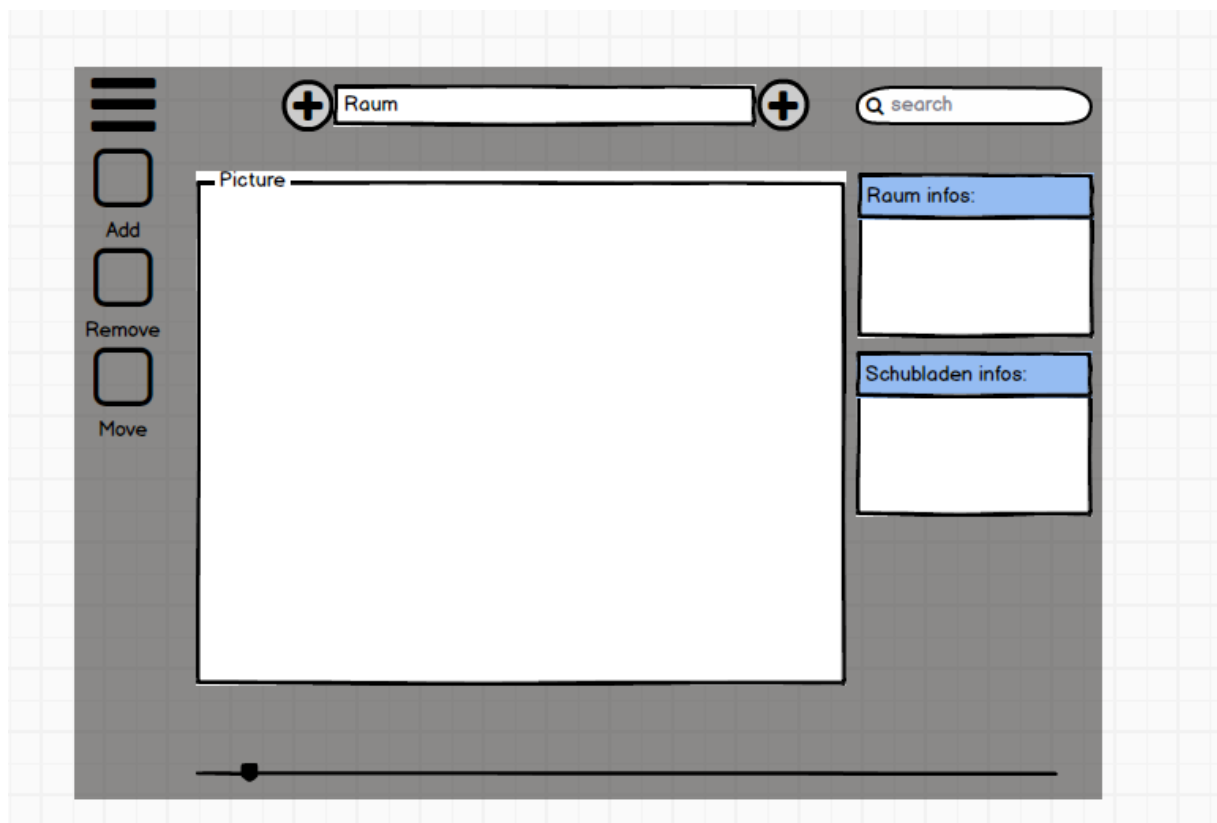
Eine Partitur ist eine untereinander angeordnete Zusammenstellung aller Einzelstimmen einer Komposition. Partituren werden dazu verwendet, um Musik reproduzierbar aufzubewahren.



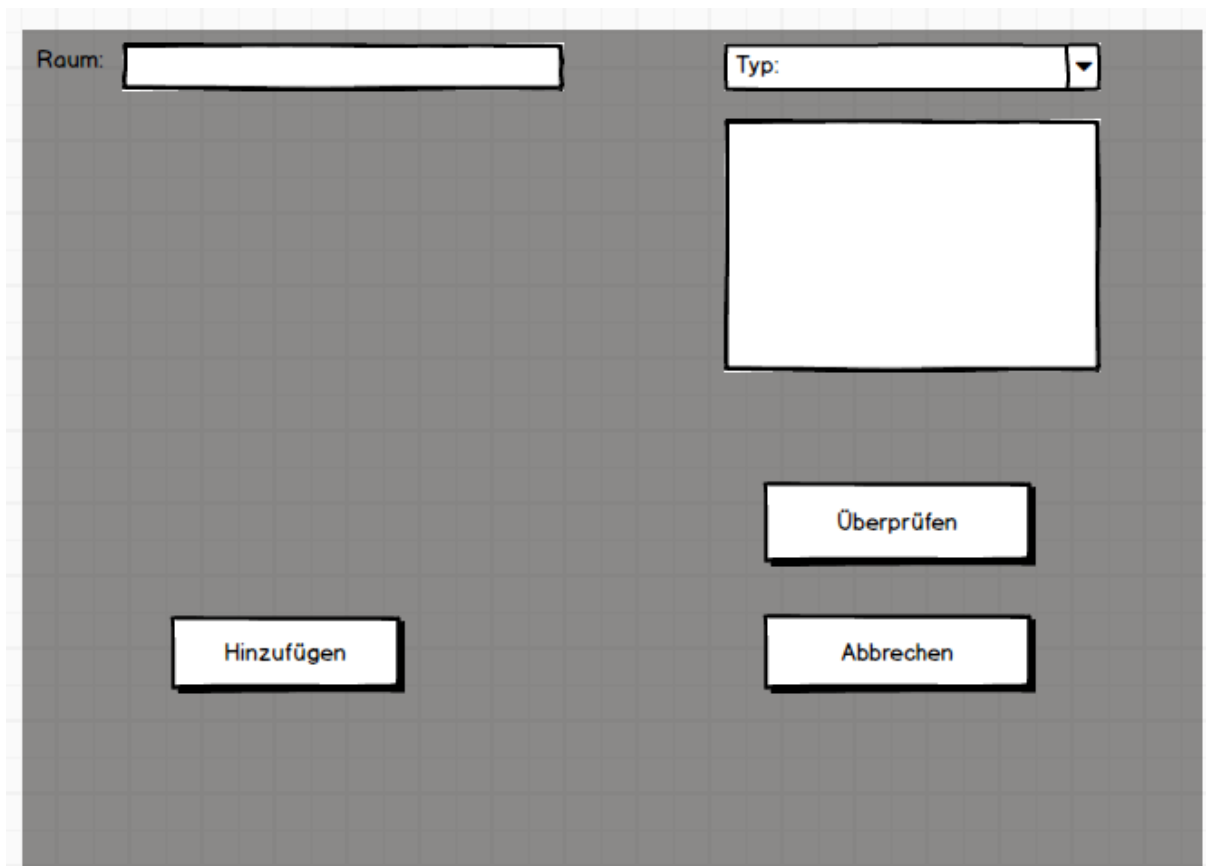
3. Produktfunktionen

- Partitur eintragen:
Das Programm hat die Möglichkeit eine Partitur eintragen zu lassen. Eine Partitur einzutragen muss man angeben in welchen Raum, welcher Schublade und Fach das Musikstück gelagert wird. Dieser Prozess wird immer durchgeführt, wenn man eine neue Partitur im Lager hinzufügt bzw. das bereits vorhandene Archiv digitalisiert.
- Partitur verschieben:
Das System unterstützt die Funktion eine Partitur von einem Raum in einen anderen zu verschieben bzw. von Schublade zu Schublade zu verschieben. Diese Methode wird angewandt, wenn das Archiv beispielsweise nach Genre geordnet wird oder ein Musikstück wegen Platzmangel oder anderen Gründen verschoben werden muss.
- Partitur löschen.
Eine essentielle Methode damit bei einem Fehleintrag das Musikstück wieder rausgenommen werden kann. Diese Methode kann auch sinnvoll sein, wenn eine Partitur verworfen wird, dann muss diese ausgebucht werden.

Oberflächen Prototyp: (Erstellt mit Balsamiq Mockups)



Start Fenster:



Hinzufügen Fenster:

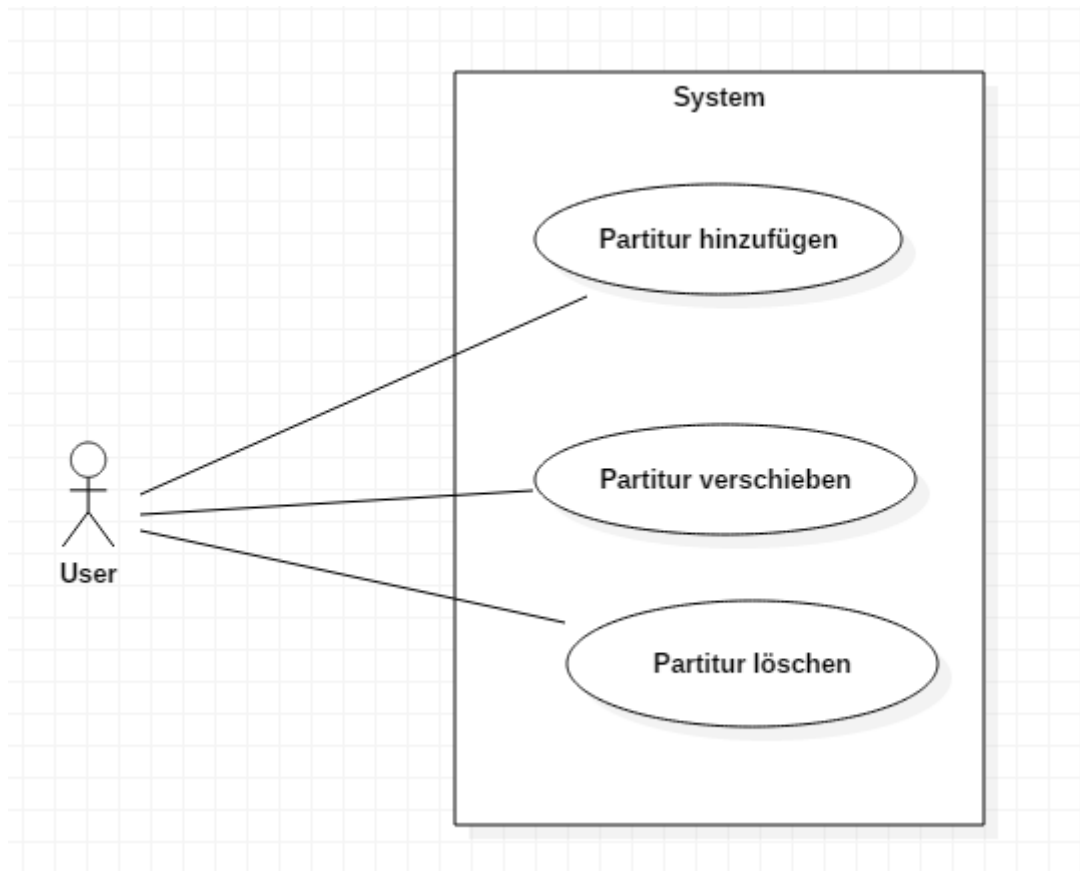
A screenshot of a 'Verschieben Fenster' (Move Window) dialog box. The dialog has a dark gray background with a light gray grid. At the top, there are three labels: 'Von:', 'Nach:', and 'Musikstück:'. Below 'Von:' is a text input field. Below 'Nach:' is a text input field with a small downward arrow on its right side. Below 'Musikstück:' is a text input field. In the center, there is a label 'Infos:' above a large, empty rectangular text area. At the bottom, there are two buttons: 'OK' on the left and 'Abbrechen' on the right. A mouse cursor is visible over the 'Von:' input field.

Verschieben Fenster:

A screenshot of a 'Löschen Fenster' (Delete Window) dialog box. The dialog has a dark gray background with a light gray grid. At the top left, there is a label 'Partiturnamen eingeben:' above a text input field. At the top right, there is a label 'Infos:' above a large, empty rectangular text area. At the bottom, there are two buttons: 'Löschen' on the left and 'Abbrechen' on the right.

Löschen Fenster:

3.1 Use-Case



Ziel des Use Cases:	<i>Im Use-Case Diagramm wird angezeigt welche Funktionen der User durchführen kann. Der User kann eine Partitur hinzufügen, eine Partitur löschen und eine Partitur von Raum zu Raum oder von Schrank zu Schrank verschieben.</i>
Umgebende Systemgrenze:	<i>Das System wird unterteilt in User Aktion (also Aktion wählen und benötigte Daten eingeben) und Verarbeitung des Systems. Der User wählt zum Beispiel hinzufügen aus und gibt die Benötigten Daten ein, das System verarbeitet dann die Daten und trägt diese in die Datenbank ein.</i>
Vorbedingung:	<i>Der User muss eine Aktion durchführen.</i>
Nachbedingung bei erfolgreicher Ausführung:	<i>Die erfordernten Daten müssen korrekt eingeben werden.</i>
Beteiligte Nutzer:	<i><User></i>

	<i>Der User interagiert mit dem System und kann eine Partitur hinzufügen, diese löschen oder verschieben.</i> <i><System></i> <i>Das System verarbeitet dann die Daten und füttert damit die Datenbank.</i>
Auslösendes Ereignis:	<i>User Interaktion mit dem Programm.</i>

4. Grobentwurf (Soll-Zustand)

Dieser Abschnitt beschreibt die Komponenten der zu entwickelnden Software. Grundsätzlich wurde die Software in Datenbankverwaltung und GUI aufgeteilt.

Der Entwicklungsprozess wurde in mehrere Unterstufen unterteilt.

Die Software:

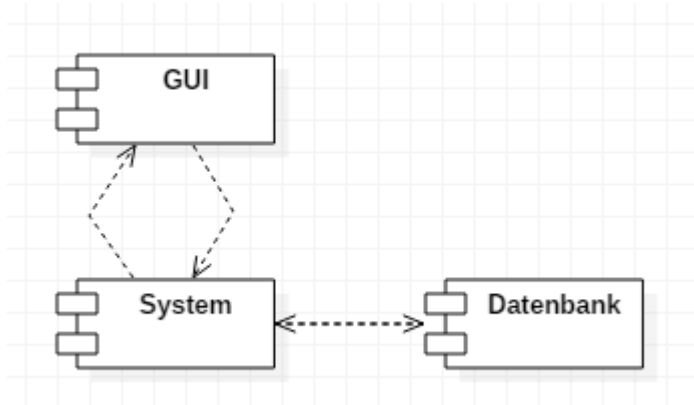
- Connection mit der Datenbank aufbauen.
- SQL Datenbank Eintrag vornehmen.
- Datenbank auslesen
- Suchen anhand Titel oder Komponist
- Suchen anhand Titel & Komponist
- Löschen anhand Titel oder Komponist
- Löschen anhand Titel & Komponist
- Räume der Datenbank auslesen

Die GUI

- Hauptfenster
- Hinzufügen
- Löschen
- Verschieben

Diese einzelnen Programmabschnitte werden jeweils mit dem dazugehörigen Teil der GUI entwickelt, getestet und implementiert. Die Software hat Zugriff auf eine Datenbankschnittstelle, wo die vom User eingegebenen Daten hinterlegt werden.

4.1 Komponentenstruktur

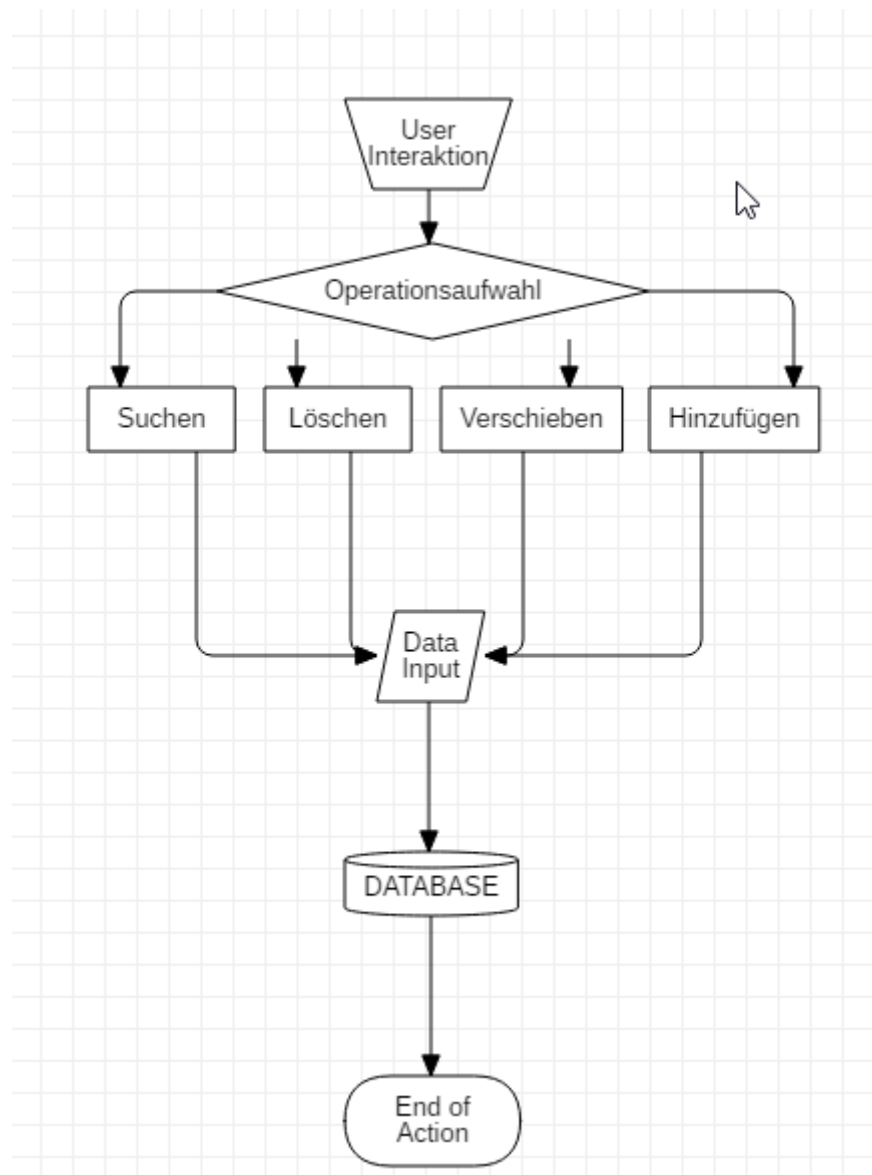


Die GUI ist vom Programm abhängig und umgekehrt. Das System ist abhängig von der Datenbank und die Datenbank vom System.

4.2 Komponenteninteraktion

Die Hauptkomponenten des Programms sind die Datenbank und das System mit GUI, welches mit der Datenbank interagiert. Der User nutzt das Programm indem er am Anfang eine Aktion wählt. Danach gibt der User die benötigten Daten ein und aufgrund dieser wird die Aktion durchgeführt und in der GUI aufgezeigt.

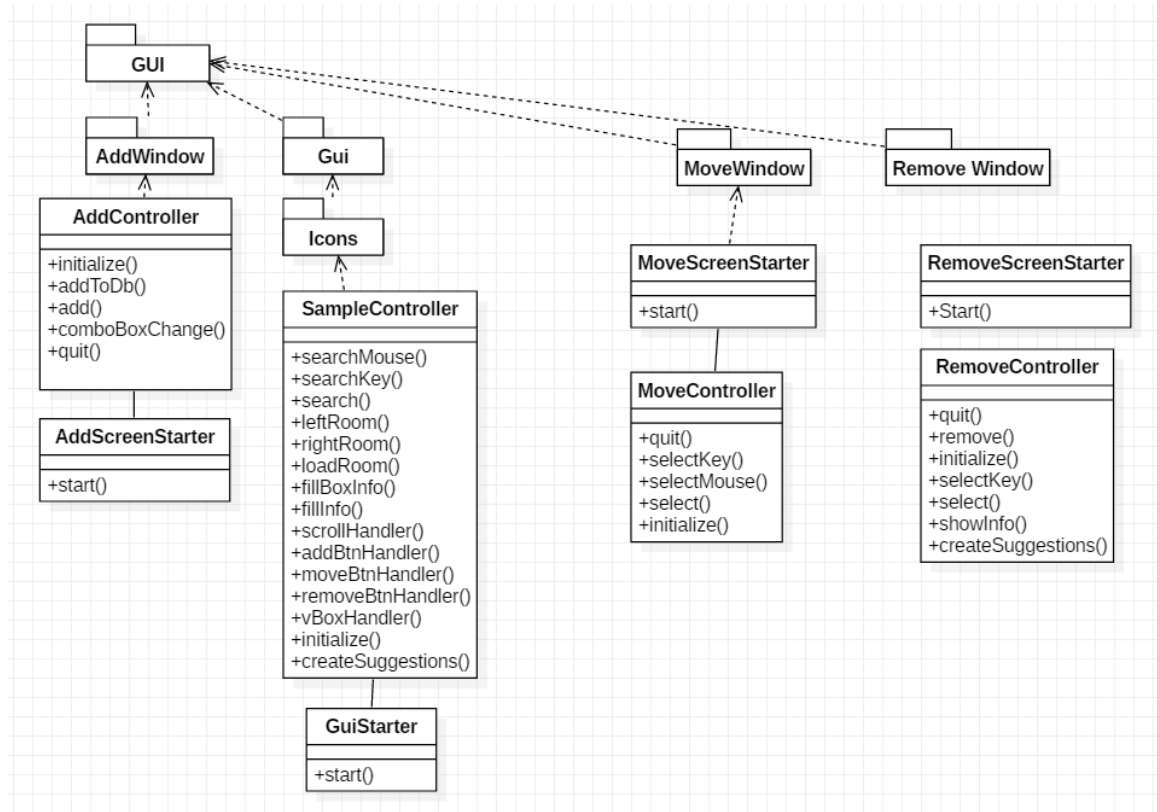
Komponenteninteraktion bei der Ausführung (Flowchart Diagramm)



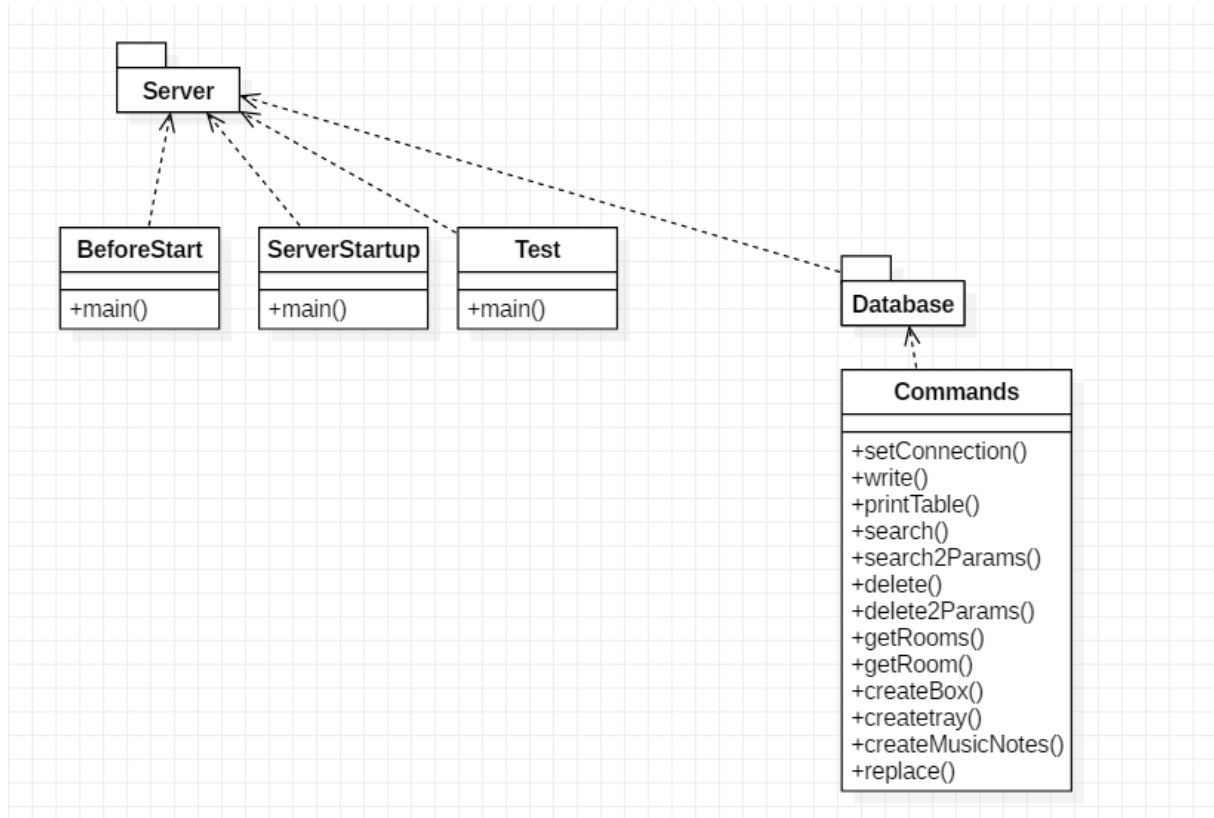
Autor: Thaler Elias

Klassendiagramm:

GUI:



Server:



5. Qualitätsmerkmale

- **Typ USE: Benutzbarkeitsanforderung**
Das Programm ist speziell für Notenwarte programmiert worden, welche die Notenlagerung und Verwaltung einer Musikkapelle überhaben. Ziel ist es die Software überschaubar und so einfach wie möglich zu gestalten, damit neue Notenwarte schnell eingelernt werden können und dass auch Laien die Software mit kurzer Erklärung nutzen können.
- **Typ EFFIZIENZ: Effizienzanforderung**
Das Programm wurde sehr Ressourcensparend programmiert und durch die Verwendung einer Datenbank konnte viel Aufwand zum Verwalten der Daten eingespart werden.
- **Typ PFLEGE: Wartbarkeits- und Portierbarkeitsanforderung**
Änderungen werden an diesem Produkt keine mehr vorgenommen, da das Produkt abgeschlossen wurde und das Programmierer Team zu keiner kommerziellen Firma gehört und dieses Projekt auch gratis umsetzte sind Änderungen ausgeschlossen.
- **Typ SICHER: Sicherheitsanforderung**
Die Datenbank wird vom Kunden selbst in periodische Zeitabständen exportiert und an einem anderen Ort aufbewahrt damit eventuelle Systemausfälle bzw. unvorhersehbare äußere Physische Einwirkungen ohne Probleme überstanden werden können.
- **Typ LEGAL: Gesetzliche Anforderung**
Es müssen keine spezifische Gesetze eingehalten werden, da das Produkt nur für die Privatnutzung konzipiert wurde.

6. Zeitaufwandschätzung

Zeitaufwand	Schätzung	Effektiver Aufwand	
GUI	12	10	
Database	17	15	
Logik	30	25	
Planung und	5	10	
Gesamt	64	60	Stunden