Projekt: Typo

Software Architektur Spezifikation (Software Architecture Document)

[Dokumentstruktur basiert auf RUP „Software Architecture Document“]

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
14.05.	1.0	Erstellen des Dokuments	Nguyen, Frank
29.05.	2.0	Überarbeitung mit neuen Inhalten bei den Punkten: 5.1; %5.2.3; 5.2.11; 5.2.13;5.2.13.1.2;5.2.13.1.3	Nguyen, Frank

1.2 Inhalt

2 Einführung (Introduction)

2.1 Definitionen und Abkürzungen (Definitions, Acronyms, Abbreviations)

<Erklärung der Abkürzungen und Definitionen oder Verweis auf separates Glossar-Dokument> WPM = Wörter pro Minute
FPM = Fehler pro Minute

2.2 Referenzen (References)

<Liste aller verwendeten Dokumente, Bücher, etc…>
-

2.3 Übersicht (Overview)

<Übersicht über den restlichen Teil dieses Dokuments geben und dessen Aufbau erläutern>

3 Architektonische Darstellung (Architectural Representation)

<Verbale Beschreibung wichtiger Architekturentscheidungen. Ggf. auch andere Architekturideen kurz beschreiben, die diskutiert und (warum?) verworfen wurden.>

Logische Sicht: Schichtenarchitektur (Links -> Rechts == Oben -> Unten)
Benutzeroberfläche -> Logik des Programms -> Datenbank

4 Architektonische Ziele & Einschränkungen (Architectural Goals and Constraints)

<Begründung der Architekturentscheidungen: Welche Anforderungen / Einschränkungen haben zu dieser Architektur geführt, d.h.: warum so und nicht anders, welche Vorteile ergeben sich aus dieser Architektur?>

Wir haben uns für die Schichtenarchitektur entschieden, weil wir eine starke Abhängigkeit zwischen den Komponenten Benutzeroberfläche, Logik und Datenbank haben. Wir haben dadurch zwischen den Klassen eine geringe Kopplung und eine hohe Kohäsion und können ein besseres Verständnis bei der Wartung und weiteren Programmierung von zusätzlichen Funktionen vorweisen.

5 logische Architektur (Logical View)

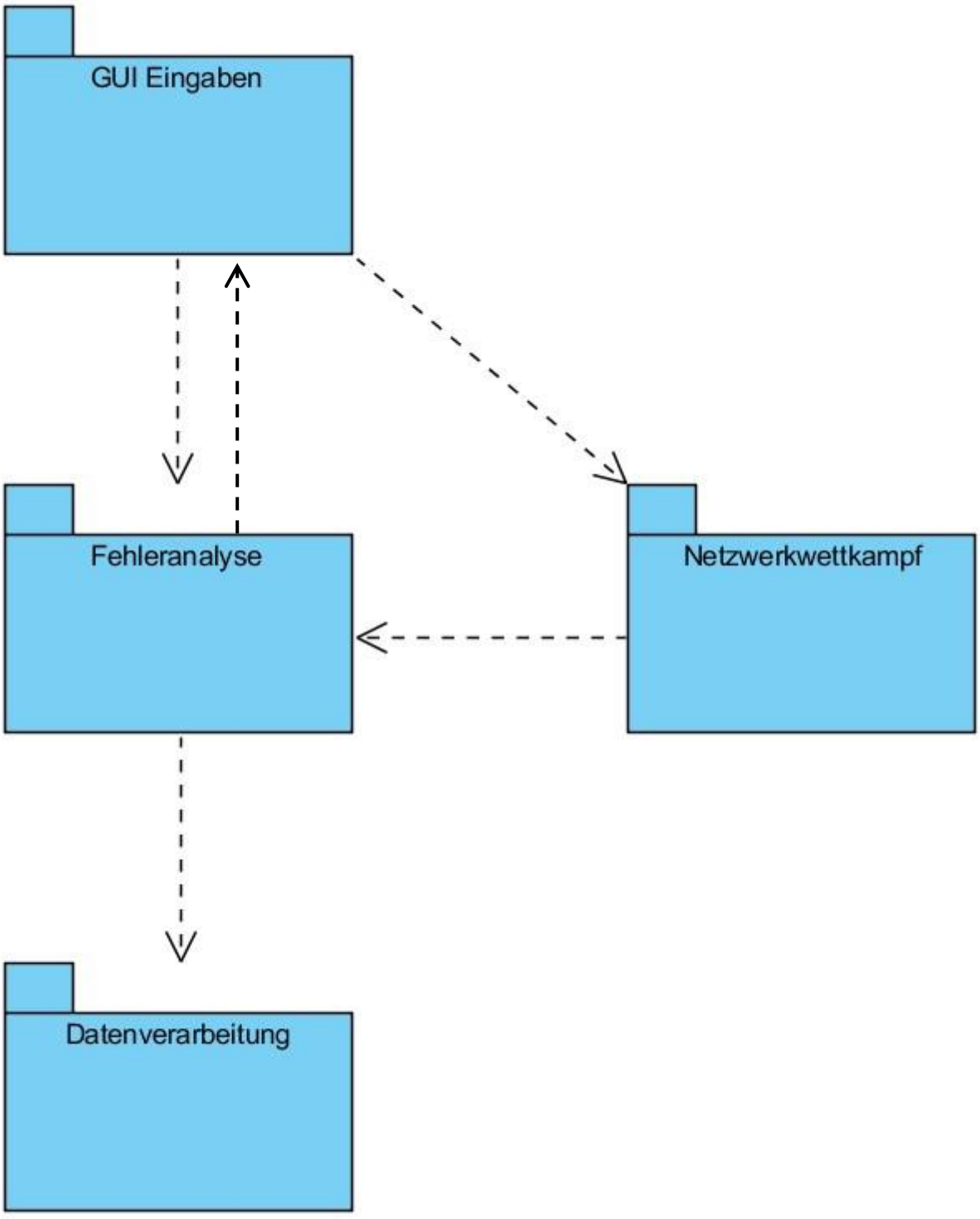
<Beschreibung der logischen Struktur des Projekts>

Der Grund für die Schichtenarchitektur ist die, dass alles voneinander in einer Reihenfolge abhängt. Die **Benutzeroberfläche** steht dabei ganz oben in der Schichtenarchitektur. Sie bildet die Bedienoberfläche für den Benutzer und repräsentiert das Programm in ihrem Design. Neben der Repräsentation kann der Nutzer ebenfalls auf der Benutzeroberfläche das Programm steuern und interaktiv lernen.
Um das zu schaffen muss unter der GUI die **Logik des Programms** sein. Dies beinhaltet beispielsweise die Fehlererkennung, das Berechnen der WPM und FPM oder das Erfassen von Nutzerdaten. Das Speichern der Nutzdaten kann nicht stattfinden, wenn wir keine **Datenbank** haben. Die Datenbank

speichert die WPM und FPM des Lernenden und aktualisiert diese Daten in einer Lernstatistik. Neben den sensiblen Daten gibt es noch offline Texte, die als Lernmittel für den Benutzer wichtig sind.

5.1 Übersicht (Overview)

<Aufteilung in Packages, Layer. Hier ein Übersichtsdiagramm, in den folgenden Abschnitten dann Detaildiagramme bis auf Klassenebene>


GUI Eingaben
Fehleranalyse
Netzwerkwettkampf
Datenverarbeitung

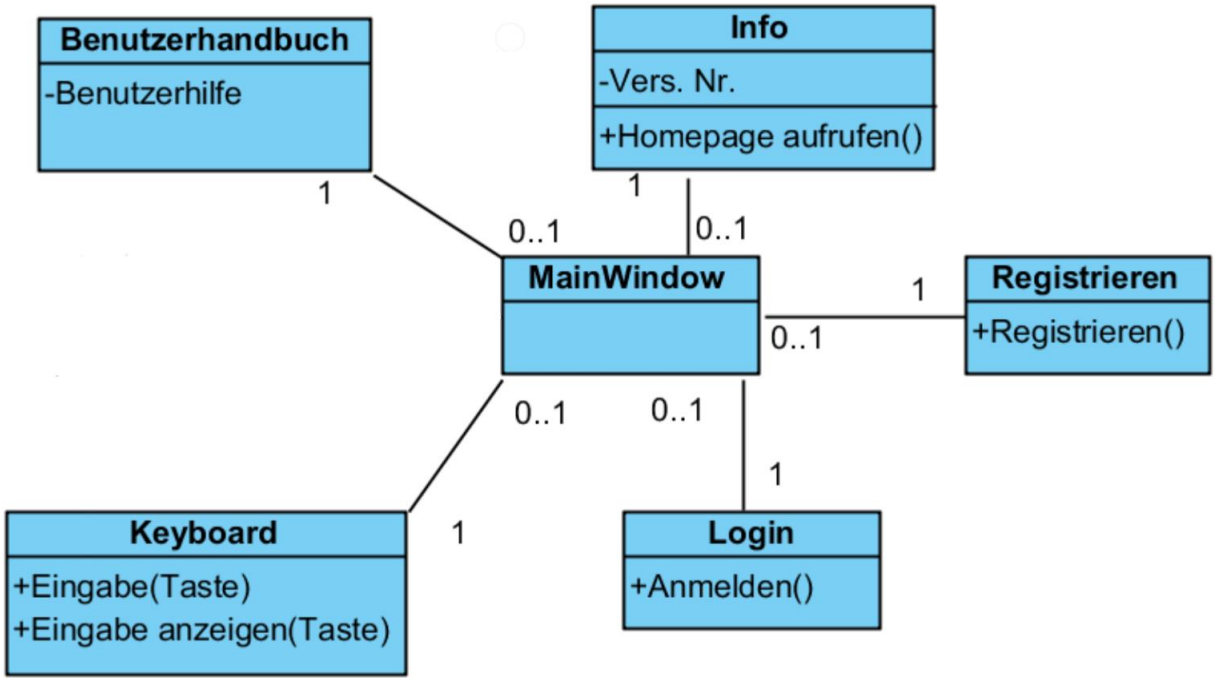
5.2 Design Pakete (Architecturally Significant Design Packages)

5.2.1.1 Package GUI Eingabe

5.2.2 Beschreibung des Package

Die GUI Eingaben werden im Mainwindow getätigt, es werden verschiedene Funktionen ausgeführt, die dem Benutzer bei dem Umgang mit Typo behilflich sind, wie das Benutzerhandbuch. Der Benutzer kann sich über eine Eingabe anmelden und registrieren, er kann sich auch auf die Homepage führen lassen. Die Hauptarbeit der GUI ist die Weitergabe der Nutzereingaben an das darunterliegende System.

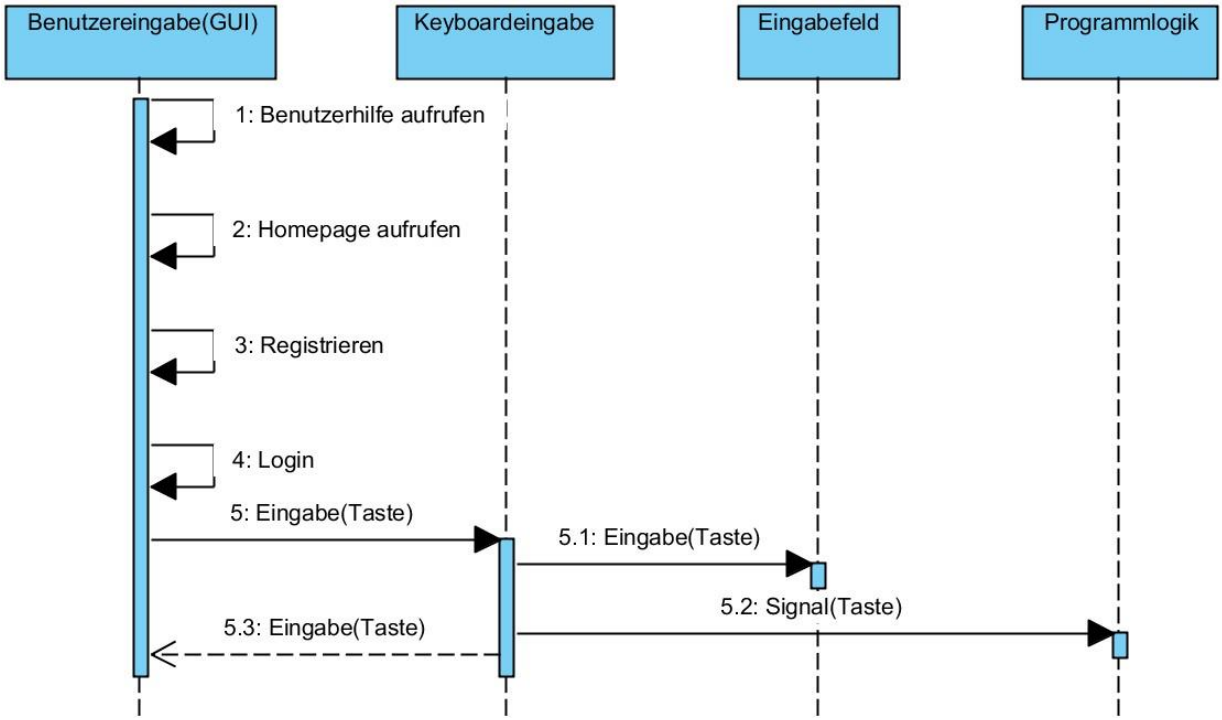
5.2.3 Diagramme



5.2.4 Schnittstellen

Schnittstelle über Signale von der GUI zu der Unterliegenden Programmlogik.

5.2.5 Operationen



5.2.5.1 Package Netzwerkwettkampf

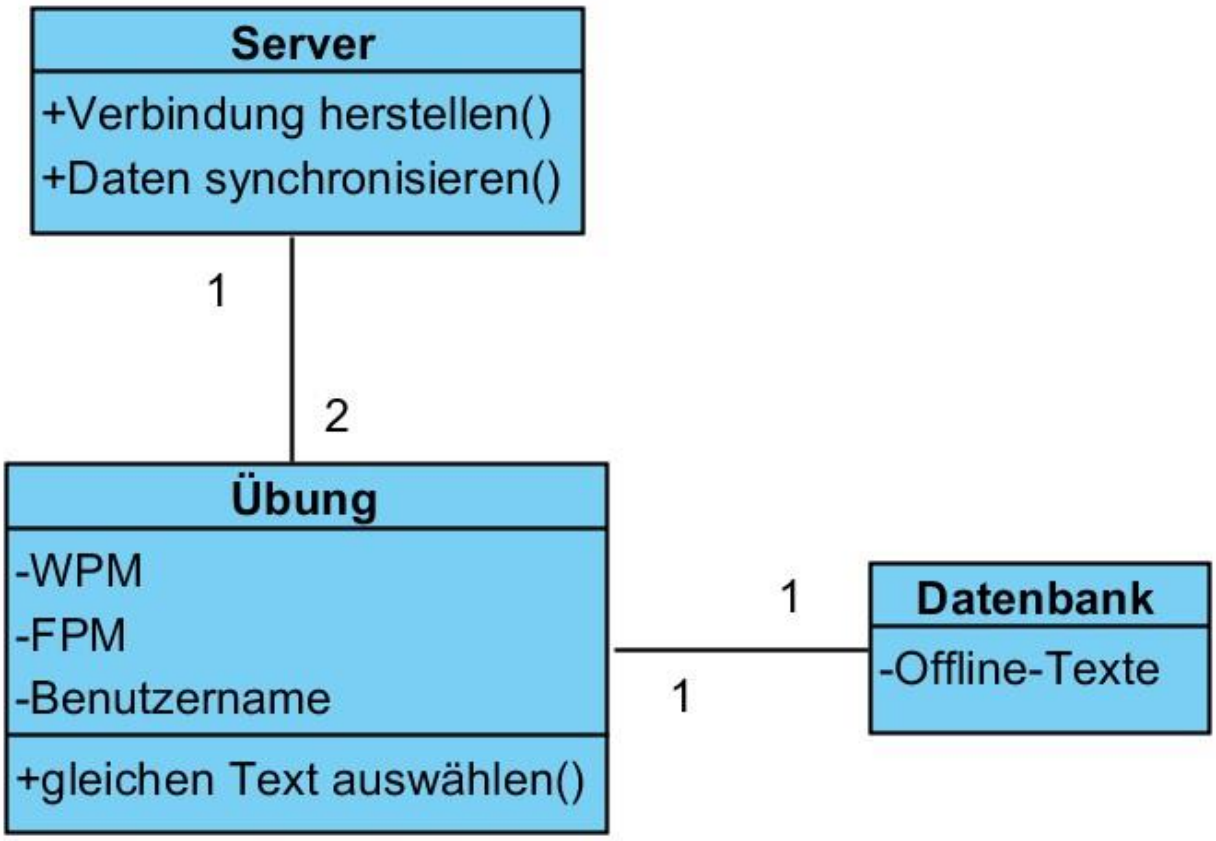
5.2.6 Beschreibung des Package

<Beschreibung des Package. Aufgabe, etc…>

Als Erstes versuchen zwei Clienten des Programms Typo eine Verbindung mit dem Server herzustellen, dann werden die zwei Clienten für einen Datenaustausch verbunden. Die Klasse Übung wird über die Datenbank anschließend einen Offline-Text nehmen und diesen beiden Clienten zur Verfügung stellen. Nach dem Wettkampf werden die Nutzerdaten (Attribute) wie Benutzername, WPM und FPM erfasst und in die Datenbank für die Lernstatistik gespeichert.

5.2.7 Diagramme

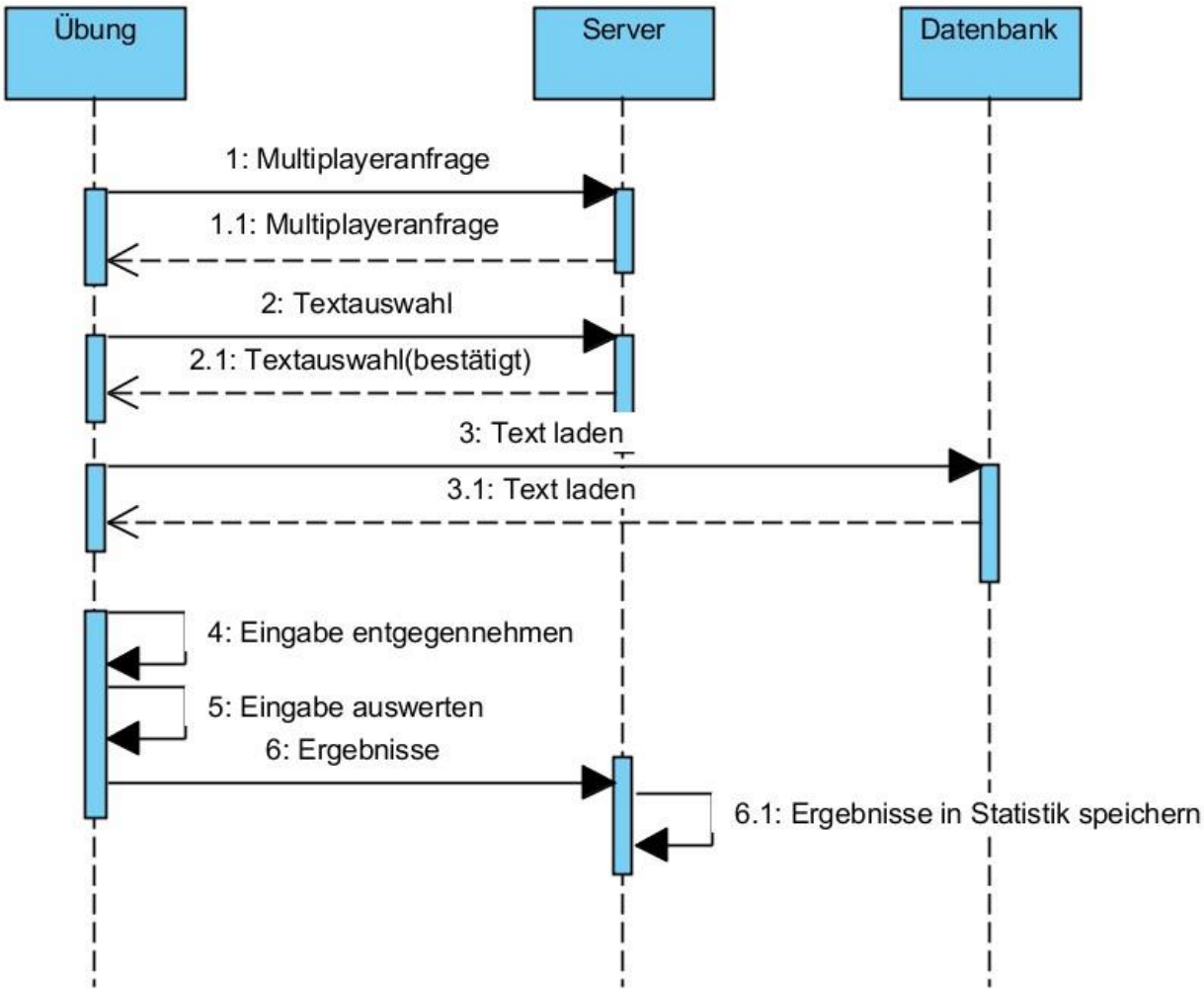
Vorläufiger Plan des Multiplayers



5.2.8 Schnittstellen

Schnittstelle Programm <-> Netzwerkverbindung zum Server, um eine Verbindung zwischen Zwei Benutzern zu erzeugen.

5.2.9 Operationen

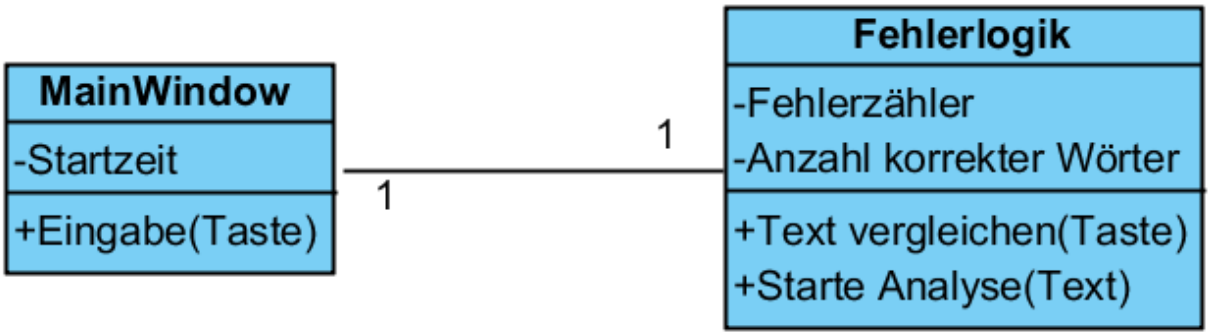


5.2.9.1 Package Fehleranalyse

5.2.10 Beschreibung des Package

<Beschreibung des Package. Aufgabe, etc…>
In der Übung werden Signale über Tasteneingaben verarbeitet. Diese Signale werden in der Fehlerlogik mit dem dazugehörigen Text verglichen. Bei Ungleichheiten wird der Fehlerzähler (Counter) für die FPM jeweils um 1 addiert.

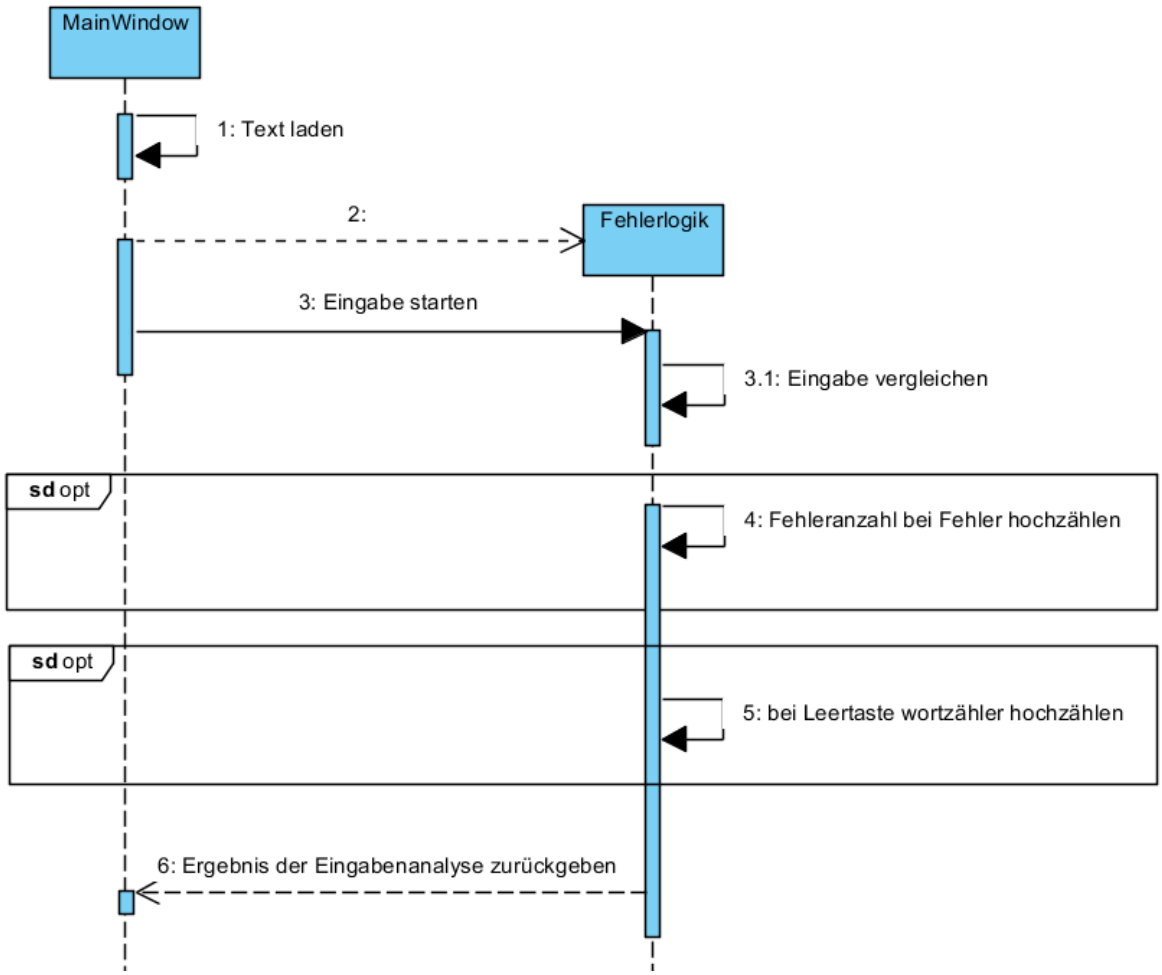
5.2.11 Diagramme



5.2.12 Schnittstellen

Schnittstelle GUI <-> Fehlerlogik, über die Eingaben und Ergebnisse ausgetauscht werden

5.2.13 Operationen

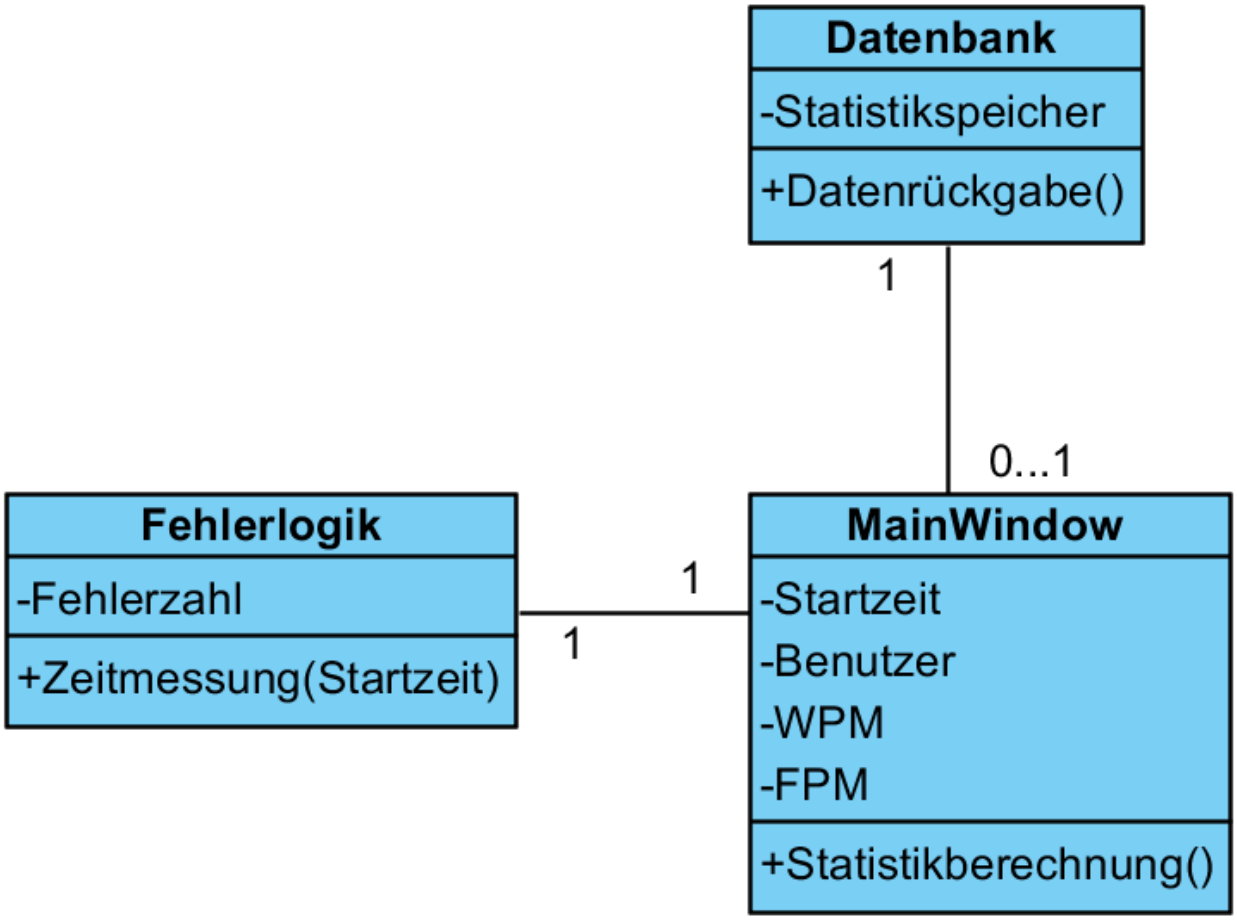


5.2.13.1 Package Datenverarbeitung

5.2.13.1.1 Beschreibung des Package

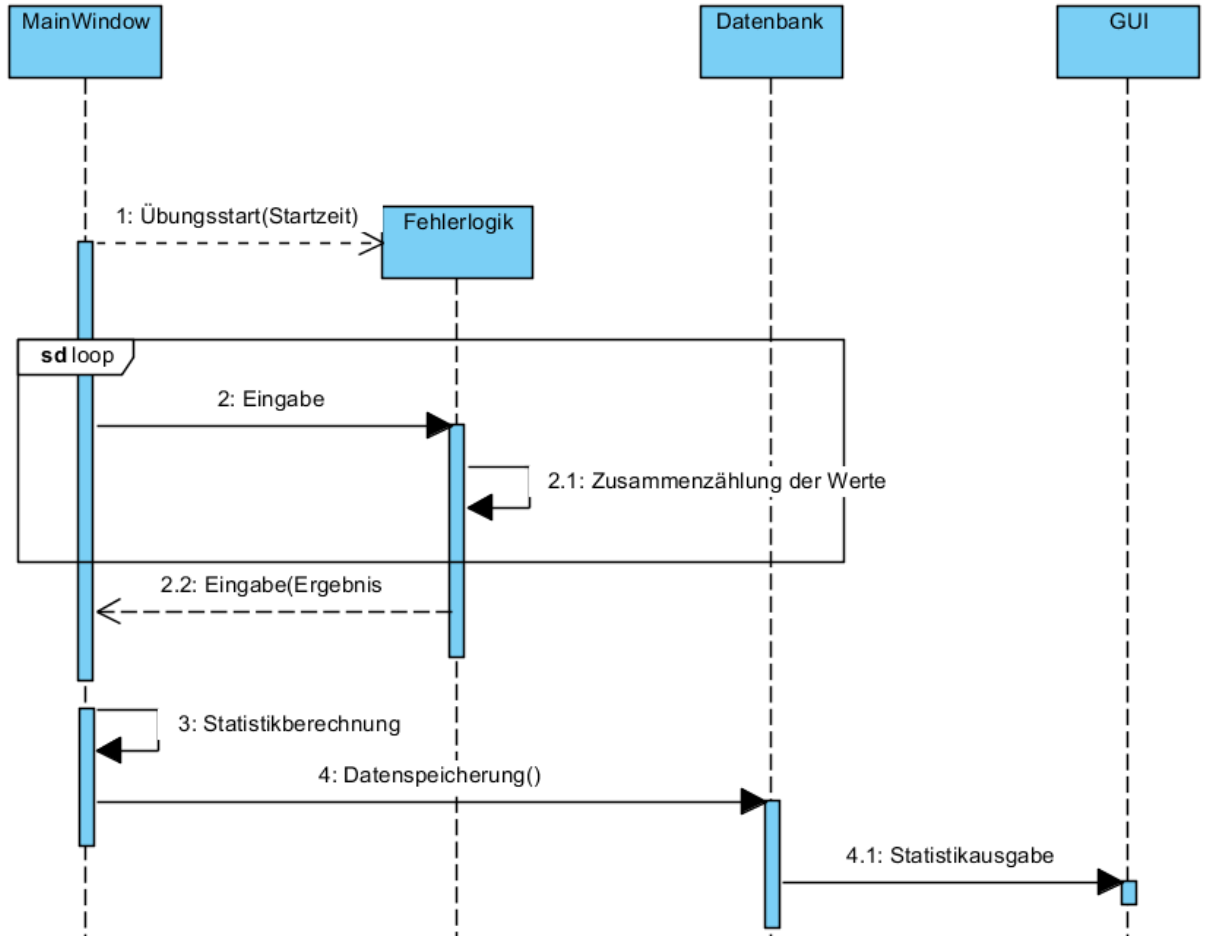
<Beschreibung des Package. Aufgabe, etc…>
In der Datenverarbeitung werden die Daten aus der Übung bzw. die darunterliegende Fehlerlogik in die Datenbank gespeichert. Dies dient für eine transparente Lernstatistik für den Nutzer, um seinen kontinuierlichen Lernfortschritt anzusehen.
Die Datenbank speichert die Startzeit, Benutzernamen und die individuellen WPM und FPM von der Klasse Übung. Wenn der Nutzer sein Lernfortschritt anschauen möchte, gibt die Datenbank alle Nutzerdaten an die GUI weiter.

5.2.13.1.2 Diagramme



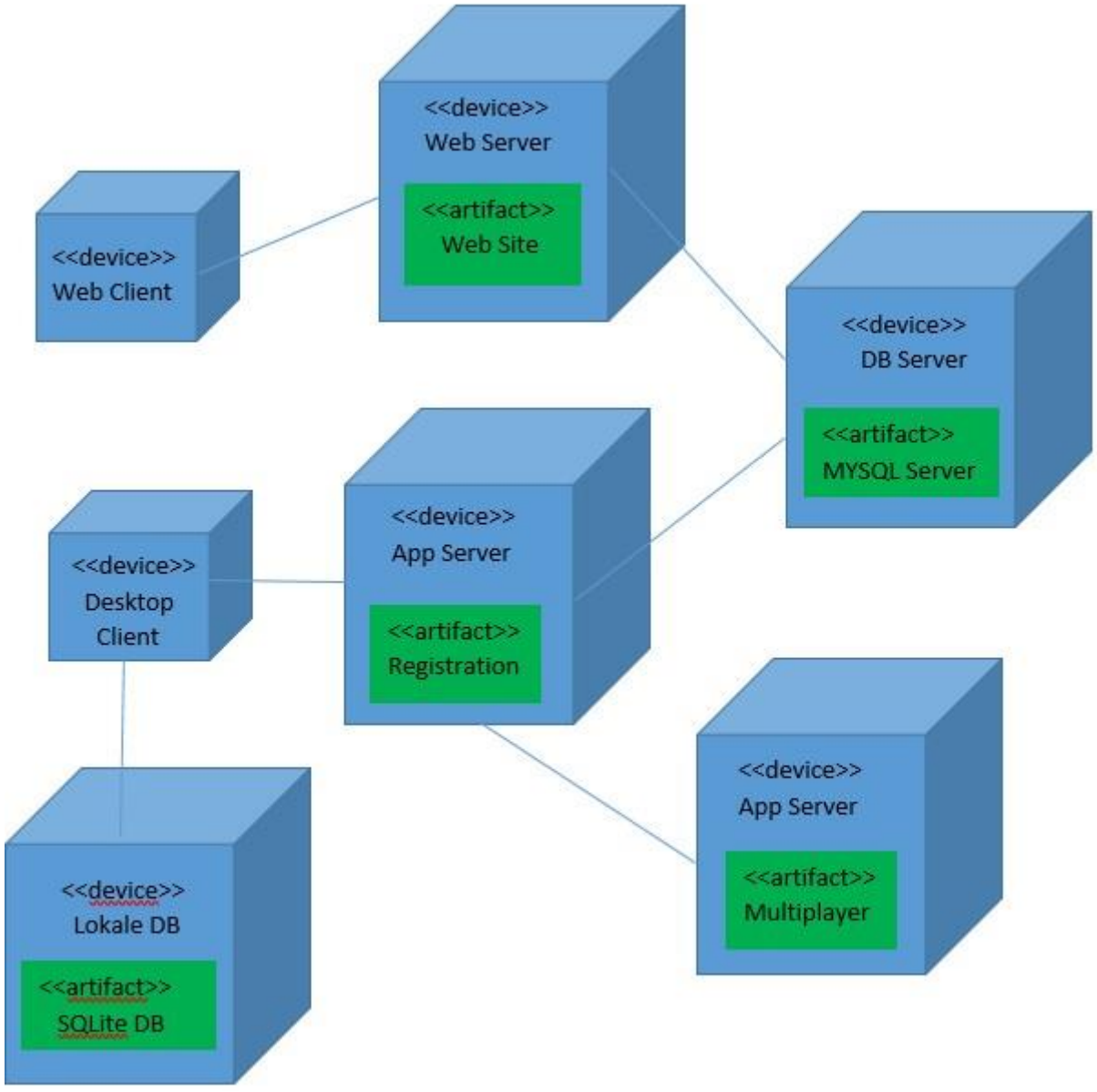
5.2.4.3 Schnittstellen

Schnittstelle zwischen Programm und Serverseitiger Datenbank.
5.2.13.1.3 Operationen



6 Physikalische Sicht (Physical View)

<Aufteilung in Tiers / Partitions, Zuordnung von Modulen oder Prozessen zu einzelnen Rechnern, wenn notwendig (z.B. bei Client-Server-Architektur.>



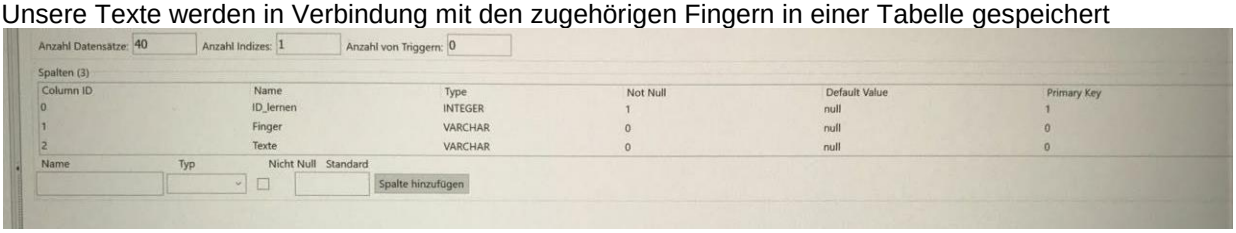
7 Prozesse und Threads (Process View)

<Wenn mehrere Prozesse oder Threads eingesetzt werden, wird hier beschrieben, wie diese ablaufen, miteinander funktionieren, Daten austauschen, sich synchronisieren, etc...>

In Typo laufen verschiedene Prozesse, die miteinander kommunizieren müssen. Dies ist zum einen die Eingabe der Zeichen des Benutzers in die GUI und die Fehleranalyse im Hintergrund des Programms. Zum anderen sind es die Auswertung der Fehleranalyse und die folgende Ausgabe der gefundenen Falscheingaben in der GUI. Die Weitergabe der Informationen wird von automatisch ablaufenden Signalen geregelt, die die Informationen von einem Prozess aus senden. Jeder Prozess hat gleichzeitig Funktionen, die die benötigten Signale akzeptiert, sodass jeder Prozess die richtigen Daten erhält. Die Signale werden bei dem Auftreten eines Ereignisses, z.B. eine benutzerseitige Eingabe in die GUI, ausgelöst.

8 Datenspeicherung (Data View)

<Beschreibung mit Diagramm der Datenspeicherung [Data Model]. (zum Beispiel: Datenbank)> Die Benutzernamen werden in Verbindung mit den Passwörtern und einem Schlüssel abgespeichert, dieser identifiziert die Benutzertexte.
Unsere Texte werden in Verbindung mit den zugehörigen Fingern in einer Tabelle gespeichert



9 Größen und Leistung (Size and Performance)

<Einschränkungen der Applikation bezüglich Speicher, Leistung, etc…. (zum Beispiel: Verwaltung unterstützt maximal 20'000 Einträge)>

Anmeldung/Registrierung:
Benutzername des Benutzers muss zwischen 5 und 15 Zeichen lang sein.
Passwort des Benutzers muss zwischen 4 und 20 Zeichen lang sein.

Offline Texte für den Modus „Lernen“:
Die Anzahl an Wörtern in einem Text muss zwischen 23 und 30.
Die Anzahl an Texten für einen Finger ist 5.
Daraus resultiert, dass wir insgesamt 40 Texte haben.
Daraus resultiert, dass wir zwischen 920 und 1.200 offline Wörter für den Modus „Lernen“ haben.

Offline Texte für die Modis „Üben“ und „Multiplayer“:
Die Anzahl an Wörtern in einem Text sind 200 und 1.500 Wörter.
Die Anzahl an verschiedenen Texten liegt bei 15-20.
Daraus resultiert, dass wir zwischen maximal 30.000 offline Wörter für den Modus „Üben“ und „Multiplayer“ haben.
Das Einfügen von eigenen Texten:
Die Anzahl an Wörtern in einen individuellen Text darf maximal 1.500 Wörter haben.