

Projekt: Typo

Domainanalyse

[Dokumentstruktur basiert auf RUP „Design Model“ und Studentenvorlage]

1. Dokumentinformationen

1.1. Änderungsgeschichte

<i>Datum</i>	<i>Version</i>	<i>Änderung</i>	<i>Autor</i>
13.04.17	1.0	Erstellen des Dokuments	Lennart Frank
27.04.17	1.1	Einfügen der UML	Lennart Frank

1.2. Inhalt

1. Dokumentinformationen	2
1.1. Änderungsgeschichte.....	2
1.2. Inhalt.....	3
2. Einführung (Introduction).....	5
2.1. Definitionen und Abkürzungen	5
2.2. Referenzen.....	5
2.3. Übersicht	5
3. Domain Modell	5
3.1. Strukturdiagramm.....	5
3.2. Beschreibung der Domänenklassen (Konzeptionellen Klassen)	5
3.2.1. Benutzer.....	5
3.2.1.1. Beschreibung.....	5
3.2.1.2. Attribute	5
3.2.1.3. Beziehungen.....	6
3.2.2. Profil	6
3.2.2.1. Beschreibung.....	6
3.2.2.2. Attribute	6
3.2.2.3. Beziehungen.....	6
3.2.3. Übung	6
3.2.3.1. Beschreibung.....	6
3.2.3.2. Attribute	6
3.2.3.3. Beziehungen.....	7
3.2.4. Zeitwettkampf.....	7
3.2.4.1. Beschreibung.....	7
3.2.4.2. Attribute	7
3.2.4.3. Beziehungen.....	7
3.2.5. Multiplayerwettkampf	7
3.2.5.1. Beschreibung.....	7
3.2.5.2. Attribute	7
3.2.5.3. Beziehungen.....	7
3.2.6. Eingabefeld	7
3.2.6.1. Beschreibung.....	7
3.2.6.2. Attribute	7
3.2.6.3. Beziehungen.....	8
3.2.7. On Screen Keyboard	8
3.2.7.1. Beschreibung.....	8
3.2.7.2. Attribute	8
3.2.7.3. Beziehungen.....	8
3.2.8. Benutzerhandbuch.....	8
3.2.8.1. Beschreibung.....	8
3.2.8.2. Attribute	8
3.2.8.3. Beziehungen.....	8
4. Systemsequenzdiagramme.....	8
4.1. Text abtippen	9
4.2. Lernstatistik abfragen.....	10

4.3.	Fehler anzeigen	11
4.4.	Multiplayermodus	12
5.	Systemoperationen	12
5.1.1.	Vertrag Bestimmten Finger lernen	12
5.1.2.	Vertrag persönlichen Text einspeichern	12
5.1.3.	Vertrag Fehler anzeigen	13

2. Einführung (Introduction)

2.1. Definitionen und Abkürzungen

WPM = Wörter pro Minute

FPM = Fehler pro Minute

<Erklärung der Abkürzungen und Definitionen oder Verweis auf separates Glossar-Dokument>

2.2. Referenzen

<Liste aller verwendeten Dokumente, Bücher, etc...>

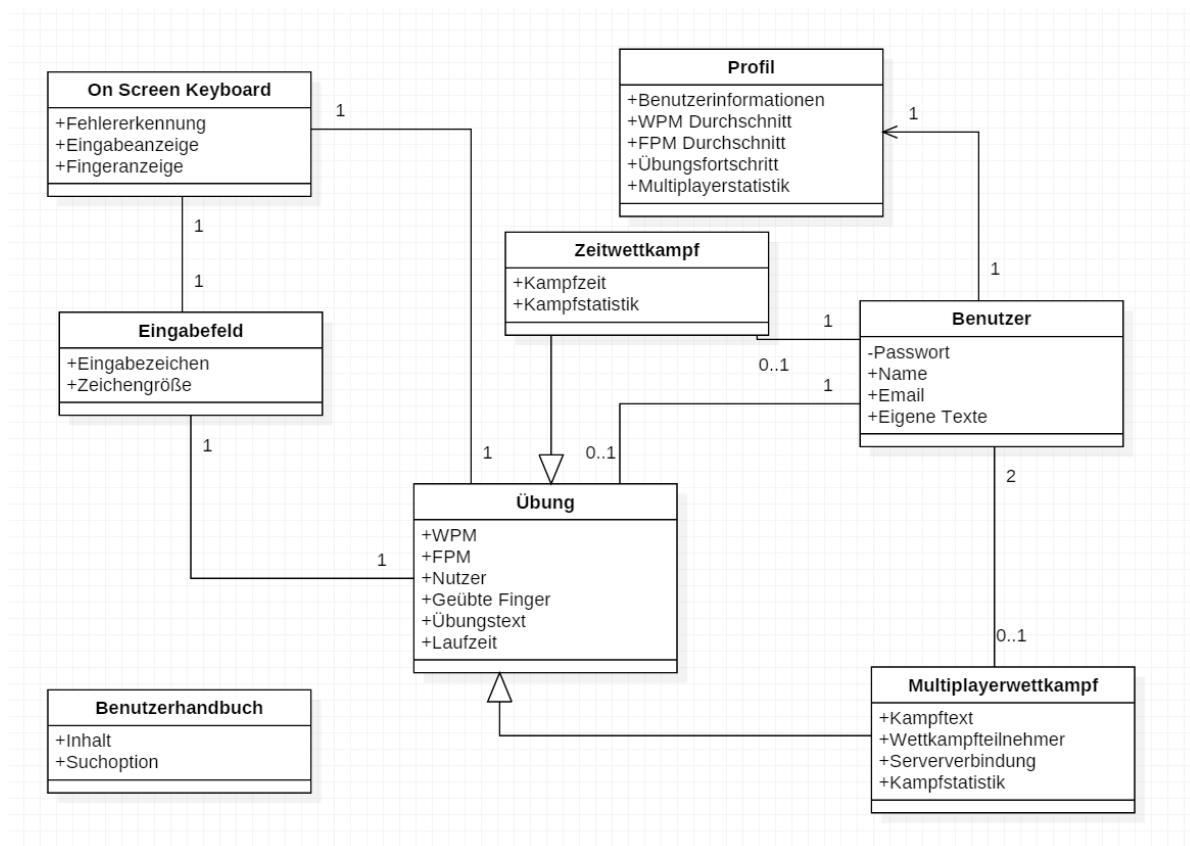
2.3. Übersicht

In diesem Dokument wird Ihnen unser Domainmodell gezeigt, die beinhalteten Domäinklassen, deren Attribute und Beziehungen. Danach werden zu den verschiedenen Systemabläufen uns Use-Cases Sequenzdiagramme dargestellt. Dann werden die Systemoperationen erläutert.

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

3. Domain Modell

3.1. Strukturdiagramm



3.2. Beschreibung der Domänenklassen (Konzeptionellen Klassen)

<Beschreibung der einzelnen verwendeten Konzepte des Domaindiagramms>

3.2.1. Benutzer

3.2.1.1. Beschreibung

Diese Klasse ist der Benutzer der das Programm bedient, Eingaben betätigt, die Lernstatistik nutzt, Übungen ausführt und sein Profil bearbeitet.

3.2.1.2. Attribute

Private: **Passwort**, Der Benutzer besitzt ein Passwort, welches ihm das Einloggen zu seinem persönlichen Account ermöglicht. Zwischen 4 und 20 Zeichen, Varchar.

Public: **Name**, Der Name des Benutzers, mit dem er sich einloggt. Zwischen 5 und 15 Zeichen, Varchar.

Public: **E-Mail**, Die E-Mail-Adresse des Benutzers, über die z.B. eine Passwortneuzuweisung stattfinden kann. 50 Zeichen, Varchar.

Public: **Eigene Texte**, Die von dem Benutzer eigens zur Übung hochgeladenen Texte. Zwischen 50 und 1500 Zeichen, Varchar.

3.2.1.3. Beziehungen

1 Benutzer hat 1 Profil.

1 Benutzer kann 0...1 Übung bestreiten.

1 Benutzer kann 0...1 Zeitwettkampf bestreiten.

2 Benutzer werden gebraucht um 1 Multiplayerwettkampf zu bestreiten.

3.2.2. Profil

3.2.2.1. Beschreibung

Das Profil des Benutzers ist die Sammelstelle für die Daten, die während den Übungen und Wettkämpfen gesammelt werden. Es werden die Normalen Benutzerinformationen, die WPM, FPM, der Übungsfortschritt und die Lernstatistiken im Profil gesammelt sein.

3.2.2.2. Attribute

Public: **Benutzerinformationen**, Die Infos von der Klasse Benutzer werden angezeigt.

Public: **WPM** Durchschnitt: Ein Wert, der den Fortschritt des Nutzers misst, bezogen auf verschiedene Zeitintervalle (z.B. WPM der letzten 10 Übungen). Integer.

Public: **FPM** Durchschnitt: Ein Wert, der den Fortschritt des Nutzers misst, bezogen auf verschiedene Zeitintervalle (z.B. FPM der letzten 10 Übungen). Integer.

Public: **Übungsfortschritt**: Eine allgemeine Einschätzung des Nutzers in Bezug auf seine Fähigkeit im Umgang mit dem 10-Finger-Tippen. Varchar.

Public: **Multiplayerstatistik**: Auswertung der Daten des Nutzers ausschließlich auf den Multiplayermodus bezogen. Diagramm/Integer.

3.2.2.3. Beziehungen

1 Profil gehört zu 1 Benutzer.

3.2.3. Übung

3.2.3.1. Beschreibung

Die Übung ist der Hauptbestandteil des Programms, in den verschiedenen Übungsmodi kann der Nutzer seine 10-Finger-Tipp-Fähigkeit trainieren, die Übungsmodi sind: Das Anfängertraining, das Fortgeschrittenentraining, der Zeitwettkampf und der Multiplayermodus. Die Daten der Übung werden im Benutzerprofil gespeichert.

3.2.3.2. Attribute

Public: **WPM**, die Anzahl der getippten Wörter pro Minute. Integer/Diagramm.

Public: **FPM**, die Anzahl der falsch getippten Zeichen pro Minute. Integer/Diagramm (Das Diagramm wird zeigen wie man sich im Vergleich zu anderen Übungen verändert hat).

Public: **Nutzer**, derjenige, der die Übung bestreitet und in dessen Statistik die Daten einfließen. Varchar, zwischen 5 und 15 Zeichen.

Public: **geübte Finger**, für die Anfangsübungen kann ein Übungsbereich auf der Tastatur gesetzt werden. Varchar, pro Finger ein bestimmter Wert, die Übungen werden dann aus den Werten erstellt.

Public: **Übungstext**, der in der Übung zu bearbeitende Text. Unterschiedliche Länge, Varchar.

Public: **Laufzeit**, die Zeit die der Nutzer üben mag. Kann unterschiedlich gesetzt werden, im Moment werden wir bestimmte Intervalle setzen von denen der Nutzer auswählen kann. Time.

3.2.3.3. *Beziehungen*

1 Übung besitzt 1 On Screen Keyboard.

1 Übung besitzt 1 Eingabefeld.

1 Übung wird von 1 Benutzer bestritten.

3.2.4. *Zeitwettkampf*

3.2.4.1. *Beschreibung*

Der Zeitwettkampf ist eine Unterform der Übung, deswegen erbt er von dieser Klasse. Beim Zeitwettkampf werden Texte für das ganze Keyboard getippt, und dabei eine Zeit gemessen. Ziel ist es eine möglichst hohe WPM-Rate zu erreichen.

3.2.4.2. *Attribute*

Public: **Kampfzeit**, die Zeit, die der bestreitende Nutzer hat, um Wörter zu Tippen. Time.

Public: **Kampfstatistik**, die Daten der derzeitigen Wettkampfbestreitung, diese werden in die persönliche Benutzerstatistik eingefügt.

3.2.4.3. *Beziehungen*

Zeitwettkampf erbt von Übung.

1 Zeitwettkampf wird von 1 Benutzer bestritten.

3.2.5. *Multiplayerwettkampf*

3.2.5.1. *Beschreibung*

Der Multiplayerwettkampf ist das gleichzeitige Bestreiten eines Textippwettkampfes von zwei unterschiedlichen Benutzern. Dabei tippen zwei Benutzer in einer festgelegten Zeit den gleichen Text ab. Wer den Text schneller und mit weniger Fehlern beendet bzw. nach der Zeit weniger falsche Zeichen getippt hat, gewinnt.

3.2.5.2. *Attribute*

Public: **Kampftext**, der Text, bei beiden Spielern derselbe, der getippt werden soll. Varchar.

Public: **Wettkampfteilnehmer**, zwei Benutzer, die gegeneinander antreten. Nur einer kann Gewinnen! Varchar, jeweils 5-15 Zeichen.

Public: **Serververbindung**, für den Multiplayer muss ein Server bereitstehen, der die Nutzer verbindet und ihnen den Text zuspilt und die Ergebnisse auswertet. IP-Adresse.

Public: **Kampfstatistik**, die Daten der Spieler, diese werden in die persönlichen Benutzerstatistiken eingefügt.

3.2.5.3. *Beziehungen*

Multiplayerwettkampf erbt von Übung.

1 Multiplayerwettkampf wird von zwei Benutzern bestritten.

3.2.6. *Eingabefeld*

3.2.6.1. *Beschreibung*

Das Feld zur Eingabe der Zeichen. Es werden nur Buchstaben, die Shift taste, die Sonderzeichen links vom rechten Shift, die Zahlen, Backspace und Enter akzeptiert.

3.2.6.2. *Attribute*

Public: **Eingabezeichen**, die Zeichen, die von dem Feld akzeptiert werden. Varchar.

Public: **Zeichengröße**, die Anzeigegröße der eingetippten Zeichen.

3.2.6.3. Beziehungen

1 Eingabefeld existiert immer mit 1 On Screen Tastatur.

1 Eingabefeld existiert in 1 Übung (und in den erbenden Klassen).

3.2.7. On Screen Keyboard**3.2.7.1. Beschreibung**

Das Keyboard, dass die eingetippten Zeichen auf dem Bildschirm anzeigt und sowohl die Finger, die das Zeichen tippen sollten farblich zu erkennen gibt, als auch fehlerhafte Eingaben farblich markiert.

3.2.7.2. Attribute

Public: Fehlererkennung, fehlerhafte Eingaben werden angezeigt. Farbcodiert.

Public: Eingabeanzeige, die getippten Zeichen werden angezeigt und farblich markiert. Farbcodiert.

Public: Fingeranzeige, es wird angezeigt, welcher Finger für welchen Bereich auf der Tastatur zuständig ist und mit welchem Finger bestimmte Eingaben betätigt werden sollen.

3.2.7.3. Beziehungen

1 On Screen Keyboard existiert mit 1 Eingabefeld.

1 On Screen Keyboard existiert in 1 Übung (und in den erbenden Klassen).

3.2.8. Benutzerhandbuch**3.2.8.1. Beschreibung**

Das Benutzerhandbuch, das das Programm Typo, den Vorgang des 10-Finger-Tippens und den Programmablauf, sowie den Umgang mit dem Programm erklärt.

3.2.8.2. Attribute

Public: Inhalt, das Benutzerhandbuch ist gefüllt mit Informationen zu dem Programm und dem Programmablauf, der Lernstatistik etc. Varchar.

Public: Suchoption, mit der Suchoption kann man das Benutzerhandbuch nach bestimmten Artikeln oder Informationen durchsuchen.

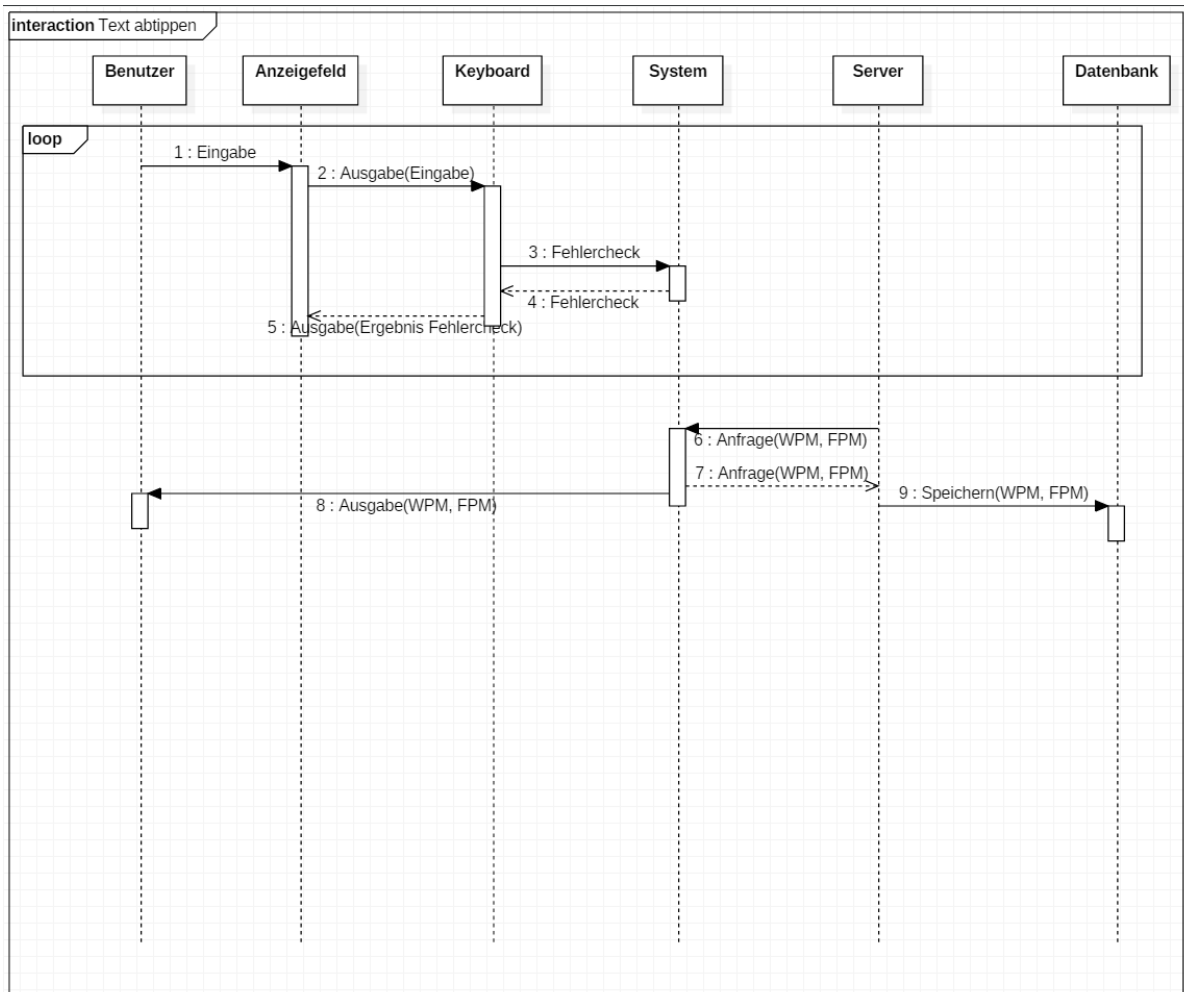
3.2.8.3. Beziehungen

Das Benutzerhandbuch ist in den Optionen Verfügbar, besitzt jedoch keine besondere Verbindung zu anderen Klassen.

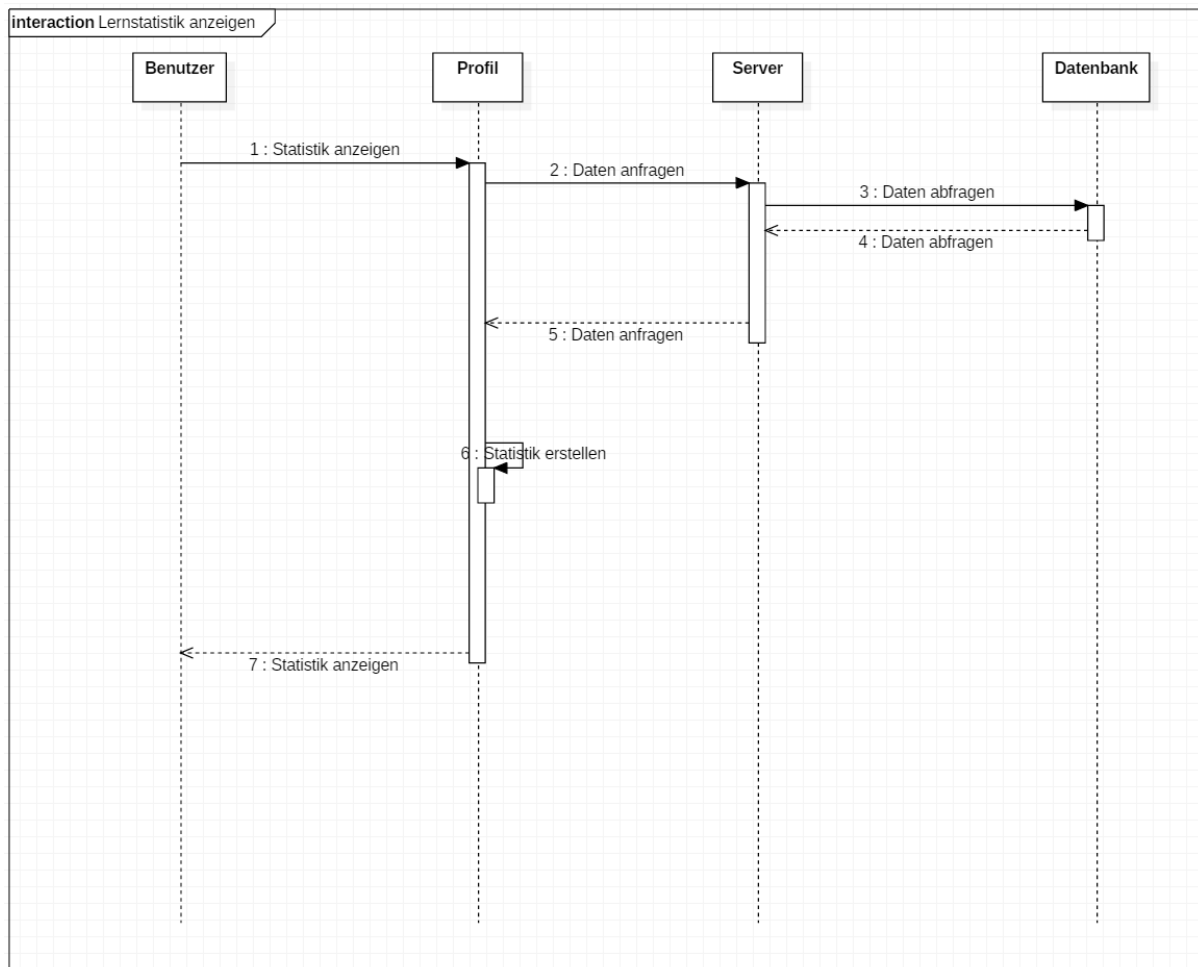
4. Systemsequenzdiagramme

<SSDs der wichtigsten, sowie der komplizierteren Use Cases>

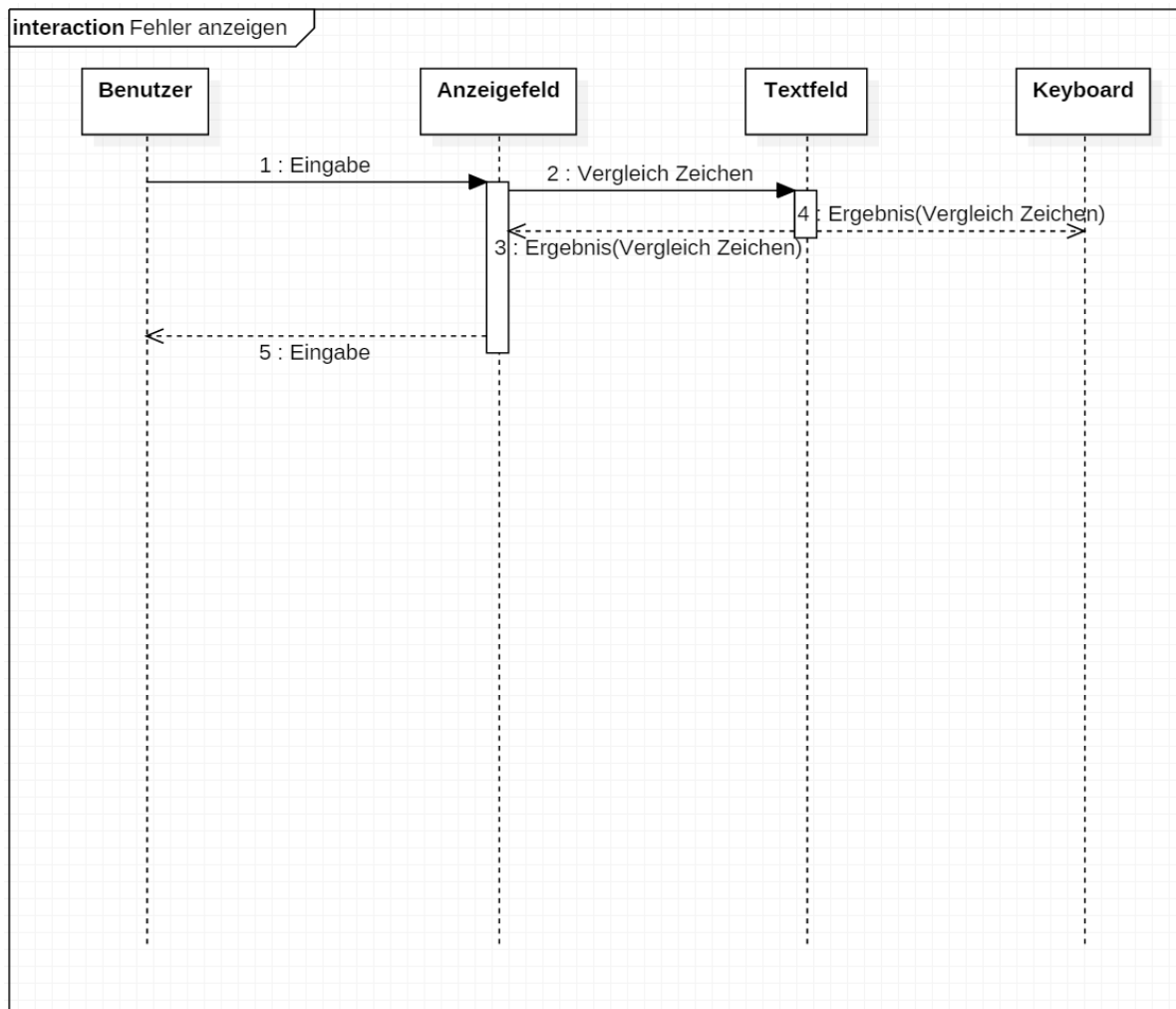
4.1. Text abtippen



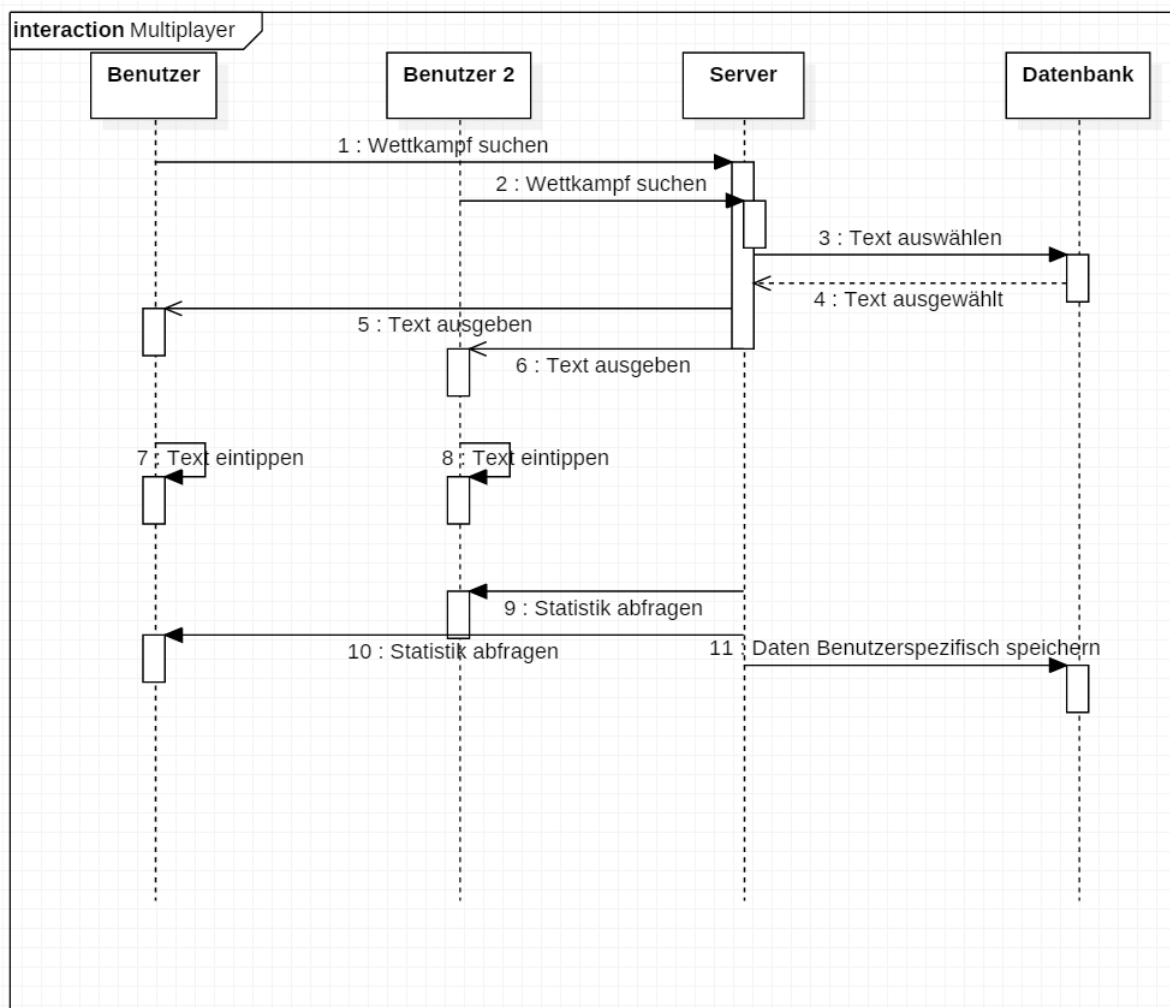
4.2. Lernstatistik abfragen



4.3. Fehler anzeigen



4.4. Multiplayermodus



5. Systemoperationen

<Beschreibung der Systemoperationen aus den SSDs>

5.1.1. Vertrag Bestimmten Finger lernen

Operationsbeschreibung: Einen bestimmten Finger mit einer ausgewählten Übung lernen

Use Case Referenzen: Anfängerübung

Vorbedingung: Wille zum Üben ist existent.

Nachbedingung:

-Übung mit dem bestimmten Finger ist abgeschlossen.

<Vertragskonstrukt mit Name, Operationsbeschreibung, Vorbedingungen, Nachbedingungen, etc...
[Larman S. 177]>

5.1.2. Vertrag persönlichen Text einspeichern

Operationsbeschreibung: Text wird in die Datenbank eingespeichert

Use Case Referenzen: Text einspeichern

Vorbedingung: Text existiert in .txt Format

Nachbedingung:

-Textdatei ist in der Datenbank erstellt.

-Textdatei ist einem Benutzer zugewiesen.

-Textdatei hat einen geschätzten Schwierigkeitsgrad

5.1.3. Vertrag Fehler anzeigen

Operationsbeschreibung: Fehler werden im Anzeigefeld markiert.

Use Case Referenzen: Fehler anzeigen

Vorbedingung: Texteingabe wird vorgenommen

Nachbedingung:

- Fehler ist markiert
- Fehler ist in Statistik gespeichert

<Vertragskonstrukt>