

Projekt: Typo

Anforderungsspezifikation (Requirement Specification)

[Dokumentstruktur basiert auf RUP „Requirement Specification“]

1. Dokumentinformationen

1.1. Änderungsgeschichte

<i>Datum</i>	<i>Version</i>	<i>Änderung</i>	<i>Autor</i>
14.04.2017	1.0	Erstellen des Dokuments	Lennart Frank
27.04.2017	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 (Definitions, Acronyms, Abbreviations)	5
2.2. Referenzen (References).....	5
2.3. Übersicht (Overview).....	5
3. Allgemeine Beschreibung (Overall Description)	5
3.1. Produkt Perspektive (Product Perspective)	5
3.2. Produkt Funktion (Product Functions)	5
3.3. Benutzer Charakteristik (User Characteristics).....	5
3.4. Einschränkungen (Constraints).....	5
3.5. Annahmen (Assumptions).....	5
3.6. Abhängigkeiten (Dependenencies)	5
3.7. Use Case Überblick (Use Case Survey).....	6
4. Spezifische Anforderungen (Specific Requirements)	6
4.1. Funktionale Anforderungen (Functionality)	6
4.1.1. Funktionale Anforderung F-1	6
4.1.2. Funktionale Anforderung F-2	6
4.1.3. Funktionale Anforderung F-3	6
4.1.4. Funktionale Anforderung F-4	6
4.2. Bedienbarkeit (Usability)	6
4.2.1. Bedienbarkeitsanforderung U-1	6
4.2.2. Bedienbarkeitsanforderung U-2.....	6
4.3. Zuverlässigkeit (Reliability)	7
4.3.1. Zuverlässigkeitsanforderung R-1	7
4.3.2. Zuverlässigkeitsanforderung R-2.....	7
4.3.3. Zuverlässigkeitsanforderung R-3.....	7
4.3.4. Zuverlässigkeitsanforderung R-4.....	7
4.4. Leistung (Performance).....	7
4.4.1. Leistungsanforderung P-1	7
4.4.2. Leistungsanforderung P-2	7
4.4.3. Leistungsanforderung P-3	7
4.5. Wartbarkeit (Supportability)	7
4.5.1. Wartbarkeitsanforderung S-1.....	7
4.5.2. Wartbarkeitsanforderung S-2.....	7
4.6. Installation	7
4.6.1. Installationsanforderung I-1	7
4.6.2. Installationsanforderung I-2	7
4.7. Internationalisierung / Lokalisierung	7
4.7.1. Internationalisierungsanforderung I18L-1	7
4.7.2. Internationalisierungsanforderung I18L-2	7
4.8. Schnittstellen (Interfaces).....	8
4.8.1. Benutzerschnittstelle (User Interface).....	8
4.8.2. Hardwareschnittstelle (Hardware Interface)	8

4.8.3.	Softwareschnittstellen (Software Interface)	8
4.8.4.	Datenbankschnittstelle (Database Interface).....	8
4.8.5.	Kommunikationsschnittstelle (Communication Interface).....	8
4.9.	Lizenzanforderungen (Licensing Requirement)	8
4.10.	Verwendete Standards (Applicable Standards)	8
5.	Use Cases.....	8
5.1.	Use Case Diagramm.....	8
5.2.	Aktoren & Stakeholders	8
5.3.	Lernen	8
5.4.	Üben.....	8
5.5.	Fingerbereich auswählen	9
5.6.	Zeitrennen bestreiten	9
5.7.	Multiplayermodus bestreiten	9
5.8.	Text abtippen	9
5.9.	WPM abrufen	9
5.10.	Fehler anzeigen.....	9
5.11.	Zeit abrufen	9
5.12.	Daten in Statistik speichern.....	9
5.13.	Eigenen Text einfügen	10
5.14.	Lernstatistik anzeigen.....	10

2. Einführung (Introduction)

2.1. Definitionen und Abkürzungen (Definitions, Acronyms, Abbreviations)

WPM: Wörter pro Minute

FPM: Fehler pro Minute

2.2. Referenzen (References)

-

2.3. Übersicht (Overview)

Dieses Dokument gibt eine Übersicht über die Anforderungen, die das Programm „Typo“ erfüllen soll. Zuerst wird eine Allgemeine Beschreibung gegeben, in der das Produkt vorgestellt wird und die Funktionen sowie Besonderheiten hervorgehoben werden. Es werden Abhängigkeiten aufgezeigt und danach ein Überblick über die Use Cases gegeben. Dann werden unterschiedliche Anforderungen beschrieben, die erfüllt werden sollen. Danach werden die verschiedenen Schnittstellen von Typo angegeben. Zum Schluss werden die Use Cases nochmal detailliert beschrieben.

3. Allgemeine Beschreibung (Overall Description)

3.1. Produkt Perspektive (Product Perspective)

Typo hat einen Vorsprung zu anderen 10-Finger-Tipp Lernprogrammen, denn Typo besitzt einen Multiplayer-Modus zusätzlich zu anderen Zusatzfunktionen wie der Bildschirmtastatur und der Lernstatistik, auch das üben mit eigenen Texten.

Der Multiplayermodus sorgt für einen Konkurrenzkampf unter Spielern, was diese zu längerem Üben und zu einer Weiterbildung ihrer Fähigkeiten sorgt.

3.2. Produkt Funktion (Product Functions)

Das Produkt kann dem Nutzer das 10-Finger-Tippen erklären und ihm Übungen bereitstellen um diese Fähigkeit zu erlernen. Unser Programm erstellt eine Statistik, mit der der Nutzer seinen Fortschritt betrachten kann. Typo kann dem Nutzer eine Tastatur anzeigen, mit der dieser nichtmehr auf die physische Tastatur blicken muss, wodurch er ein besseres Gefühl für das 10-Finger-Tippen erhält. Außerdem wird ein Server bereitgestellt, über den mehrere Nutzer ihre Fähigkeiten messen können. Zusätzlich stellt eine Wettkampfoption bereit, über den die Fähigkeit auch alleine getestet werden kann.

3.3. Benutzer Charakteristik (User Characteristics)

Die Zielgruppe des Produktes sind Menschen, die das 10-Finger-Tippen erlernen wollen, möglich sind Schreibtischarbeitende Personen, die ihre Effizienz steigern wollen, Menschen die den besseren Umgang mit ihrem Computer erlernen wollen, junge Menschen, denen eine wertvolle Fähigkeit angelernt werden soll.

3.4. Einschränkungen (Constraints)

Das Produkt kann nicht prüfen, ob eine Person für einen besonderen Test bereit ist.

Das Produkt kann kein Zertifikat für eine Fähigkeit ausstellen.

3.5. Annahmen (Assumptions)

Wir nehmen an, dass das Programm einen Lernfortschritt erzielt, wodurch alle Übungen bestritten werden können.

3.6. Abhängigkeiten (Dependenencies)

Das Produkt benötigt eine Serververbindung für das richtige Funktionieren des Multiplayers.

Für die Nutzerstatistik, die Benutzerbezogen ist, ist eine Datenbank vonnöten, die jedem Benutzer seine eigene Statistik zuordnet.

Für die Übungstexte muss eine Datenbank dazu fähig sein die Texte für unterschiedliche Übungsschwierigkeiten zuzuordnen und diese für den ausgewählten Schwierigkeitsgrad und übenden Finger bzw. Tastaturbereich auszugeben.

Für die Datenbank ist ein laufender Server vonnöten.

3.7. Use Case Überblick (Use Case Survey)

Use Case 1: Lernen

Use Case 2: Üben

Use Case 3: Fingerbereich auswählen

Use Case 4: Zeitrennen bestreiten

Use Case 5: Multiplayermodus bestreiten

Use Case 6: Text abtippen

Use Case 7: WPM abrufen

Use Case 8: Zeit abrufen

Use Case 9: Fehler anzeigen

Use Case 10: Daten in Statistik speichern

Use Case 11: Eigenen Text einfügen

Use Case 12: Lernstatistik anzeigen

4. Spezifische Anforderungen (Specific Requirements)

4.1. Funktionale Anforderungen (Functionality)

Die Anforderungen wurden durch die Entwicklergruppe von Typo gewonnen, sowie auch durch „Tippenakademie“, eine Web-Anwendung, die auch das 10-Finger-Tippen erklärt.

4.1.1. Funktionale Anforderung F-1

Das 10-Finger-Tippen soll einfach und von Grund auf erklärt werden. Es soll also nicht nur ein Programm für das Üben und Lernen existieren, sondern auch eine Erklärung des Vorgangs bereitstehen.

4.1.2. Funktionale Anforderung F-2

Es soll das Üben in verschiedenen Schwierigkeitsstufen möglich sein, sodass der Übende sich an einem stetig steigenden Schwierigkeitsgrad verbessern kann.

4.1.3. Funktionale Anforderung F-3

Das Programm soll eine Netzwerkverbindung zwischen mehreren Spielern ermöglichen, durch die diese gegeneinander live Wettfeiern können.

4.1.4. Funktionale Anforderung F-4

Das Programm soll eine Bildschirmtastatur besitzen, durch die das Hinabblicken auf die physische Tastatur abtrainiert wird und die Übenden während des Schreibens den Blick auf dem Bildschirm belassen können.

4.2. Bedienbarkeit (Usability)

4.2.1. Bedienbarkeitsanforderung U-1

Das Programm soll nach wenigen Minuten Arbeitszeit benutzbar sein. In dieser Zeit soll das Erstellen eines Benutzerprofils beinhaltet sein.

4.2.2. Bedienbarkeitsanforderung U-2

Um alle bereitgestellten Funktionen des Programms nutzen zu können muss die Grundfähigkeit des 10-Finger-Tippens erlernt werden. Dafür muss entweder die Anfänger-Übung absolviert und/oder der Vorgang des 10-Finger-Tippens im Benutzerhandbuch gelernt werden. Dies kann mehrere Stunden dauern, da das Üben der Grundfertigkeiten sehr aufwändig sein kann.

4.3. Zuverlässigkeit (Reliability)

4.3.1. Zuverlässigkeitsanforderung R-1

Es soll nicht mehr als ein Verbindungsfehler pro Verbindungsversuch geschehen

4.3.2. Zuverlässigkeitsanforderung R-2

Ein Übungstext soll nicht mehr als zweimal hintereinander wiederholt werden.

4.3.3. Zuverlässigkeitsanforderung R-3

Das Programm soll mindestens zwei Personen gleichzeitig über einen Server für die Nutzung des Multiplayermodus verbinden können.

4.3.4. Zuverlässigkeitsanforderung R-4

Das Programm soll bis zu einen Monat nach Abgabe des Projekts noch den Multiplayermodus unterstützen.

4.4. Leistung (Performance)

4.4.1. Leistungsanforderung P-1

Das Programm soll Texte der Länge von bis zu 1500 Zeichen speichern können.

4.4.2. Leistungsanforderung P-2

Das Programm soll genug Texte für jede Übung beinhalten, sodass mehrere Übungen hintereinander mit einem zufälligen Text ausgeführt werden können, ohne den gleichen Text nochmal zu erhalten.

4.4.3. Leistungsanforderung P-3

Es soll eine durchgehend gute Verbindung der zwei Spieler im Onlinemodus verfügbar sein.

4.5. Wartbarkeit (Supportability)

<Wartbarkeit, Anpassbarkeit, Konfigurierbarkeit, etc...>

4.5.1. Wartbarkeitsanforderung S-1

Die einzelnen Teile des Programms sollen als Widgets zusammen in das MainWindow eingefügt werden, um die Wartbarkeit zu verbessern.

4.5.2. Wartbarkeitsanforderung S-2

Für die Wartung ihrer persönlichen Bereiche sind die Programmierer zuständig, die den Bereich erstellt haben.

4.6. Installation

4.6.1. Installationsanforderung I-1

Das Programm wird als Download für Windows-Desktops zur Verfügung gestellt.

4.6.2. Installationsanforderung I-2

Das Programm wird als Installationsdatei gedownloadet und automatisch installiert, danach kann das Programm über die .exe Datei ausgeführt werden.

4.7. Internationalisierung / Lokalisierung

4.7.1. Internationalisierungsanforderung I18L-1

Das Produkt soll nur auf Deutsch vermarktet werden.

4.7.2. Internationalisierungsanforderung I18L-2

Das Produkt soll nur als Desktopanwendung zur Verfügung stehen

4.8. Schnittstellen (Interfaces)

4.8.1. Benutzerschnittstelle (User Interface)

Es gibt die Benutzerschnittstelle per Maus-Klick-eingabe im Menu und per Tastatureingabe in den Übungen.

4.8.2. Hardwareschnittstelle (Hardware Interface)

Die Tastatur wird als Eingabemethode für die Zeichen in den Übungen benutzt.

Die Mauseingabe wird zur Direktion im Menu benutzt.

4.8.3. Softwareschnittstellen (Software Interface)

Es existieren Schnittstellen zu den im Programm benutzten Qt-Bibliotheken. Es werden z.B. die QLabel und QTextStream benutzt.

Für den persönlichen Übungstext wird ein Editor benötigt.

4.8.4. Datenbankschnittstelle (Database Interface)

Als Datenbank wird eine MySQL Datenbank genutzt. Diese wird über unseren Server mit Daten gespeist.

4.8.5. Kommunikationsschnittstelle (Communication Interface)

Wir nutzen für unseren Online-Service einen gemieteten Root-Server von Gamed.de .

4.9. Lizenzanforderungen (Licensing Requirement)

Es wird eine von uns angeeignete Lizenz für den Server benötigt.

Für das Verwalten des Servers wird eine Plesk-Web-Admin Lizenz benötigt.

Diese wurden schon erhalten.

4.10. Verwendete Standards (Applicable Standards)

-

5. Use Cases

5.1. Use Case Diagramm

Diagramm „Use Case Typo.mdj“ im Repository verfügbar, zu öffnen mit dem Programm StarUML, dies ist auch dort verfügbar.

5.2. Aktoren & Stakeholders

Aktor 1: User, Der Benutzer des Programms, Hauptakteur bei dem Umgang mit Typo

Aktor 2: Wettkampfgegner, Ein Gegner im Multiplayermodus der den Wettkampf gegen den User bestreitet

Aktor 3: Server, Der Onlineserver des Programms, der Typo mit seiner Datenbank verbindet, die Statistiken und Texte an die lokalen PCs sendet und bei dem Multiplayerwettkampf die Verbindung zwischen den Spielern herstellt.

5.3. Lernen

In diesem Teil des Programms führt der Lernende Übungen für Anfänger durch. Es wird ein Text aus der Datenbank entnommen, der spezifisch für den ausgewählten Finger zu bearbeiten ist. Dieser wird dann von dem Benutzer über die Tastatur eingetippt, die Eingabe wird auf dem Widget „Keyboard“ angezeigt und im Anzeigefeld werden Fehler markiert. Der Use Case ist ähnlich dem „Üben“ Use Case.

5.4. Üben

Der fortgeschrittene Übungsteil des Programms wird hier absolviert. Texte spezifisch für alle Finger und mit höherem Schwierigkeitsgrad werden aus der Datenbank geladen, diese danach von dem Benutzer über die Tastatur eingetippt. Die Eingabe wird auf dem Widget „Keyboard“ angezeigt und im Anzeigefeld werden Fehler markiert. Der Use Case ist ähnlich dem „Lernen“ Use Case.

5.5. Fingerbereich auswählen

Nach der Entscheidung zum "Lernen" muss ein bestimmter Fingerbereich (z.B. kleiner Finger, rechte Hand) ausgewählt werden, den es zu bearbeiten gilt. Der zu tippende Text beinhaltet nur Buchstaben aus diesem Teil der Tastatur. Der Use Case ist von dem Benutzer in der GUI auszuwählen.

5.6. Zeitrennen bestreiten

In dem ersten Wettkampfmodus ist der einzelne Benutzer gefragt. Er soll einen Text in einer bestimmten Zeit abtippen. Der Text wird aus der Datenbank gezogen, er ist nicht von dem Anfängerschwierigkeitsgrad. Es muss eine Zeit angegeben werden, die der Benutzer tippen will. Die Eingabe wird über die physische Tastatur getätigt und auf dem Widget „Keyboard“ angezeigt, Fehler werden im Anzeigefeld angezeigt und in die Statistik von dem Benutzer nach abschließen der Übung eingetragen. Der Wettkampfmodus ist ähnlich dem Multiplayermodus.

5.7. Multiplayermodus bestreiten

In der zweiten Wettkampfmart treten zwei Benutzer gleichzeitig über eine Serververbindung gegeneinander an. Sie erhalten beide automatisch den gleichen Text von dem Server zugewiesen und tippen gleichzeitig diesen ab. Nach Ablauf der Zeit werden die FPM und WPM verglichen. Dieser Vergleich wird als Diagramm angezeigt. Das Abtippen ist konsistent mit den anderen Abtippvorgängen. Dieser Wettkampfmodus ist ähnlich dem Zeitrennen modus.

5.8. Text abtippen

Der Use Case beschreibt den Vorgang beim Abtippen von Texten. Dieser ist bei "Lernen", "Übung", "Zeitwettkampf bestreiten" und "Multiplayermodus bestreiten" analog. Es wird eine Eingabe über die Tastatur ausgelesen und im Anzeigefeld angezeigt. Falls Ungleichheiten mit dem Ursprungstext bestehen wird ein Fehler im Anzeigefeld markiert. Gleichzeitig wird die Eingabe im Widget „Keyboard“ erkenntlich gemacht, falls es eine fehlerhafte Eingabe war so wird diese Speziell markiert. Fehler werden gezählt und am Ende eines Eingabevorgangs in die Statistik gespeichert. Die Anschläge bzw. Wörter die geschrieben wurden werden mit der Zeit verglichen und auch diese Daten werden gespeichert. Die Benutzerstatistik wird erst mit der Datenbank abgeglichen und danach erst ausgegeben.

5.9. WPM abrufen

Der Vorgang beschreibt, wie der Server die WPM für die Statistik berechnet. Die geschriebenen Wörter werden anhand der Leerzeichen gezählt. Danach wird die tatsächlich getippte Zeit genommen und die Anzahl(Wörter) mit der Zeit Dividiert. Der gefolgerte Wert ist danach in der Datenbank Nutzerspezifisch abzuspeichern. Dieser Use Case ist ähnlich dem „Fehler anzeigen“.

5.10. Fehler anzeigen

Der Use Case beschreibt den Vorgang, wie die Fehlerquote errechnet wird. Für jeden Fehler wird ein Zähler hochgerechnet, die Anzahl davon wird nach dem Ende des Eingabevorgangs mit der getippten Zeit dividiert. Der errechnete Wert wird danach in der Datenbank Nutzerspezifisch abgespeichert. Dieser Use Case ist ähnlich dem „WPM abrufen“.

5.11. Zeit abrufen

Es wird die getippte Zeit von der Eingabe abgerufen. Dabei wird nicht die vergangene Zeit gezählt, sondern die Zeit in der der Nutzer getippt hat. Nach der Eingabe eines Zeichens muss innerhalb der nächsten 1,5s eine weitere Eingabe getätigt werden, sonst wird die zählende Uhr angehalten. Das Programm nutzt die abgerufene Zeit für die Berechnung der FPM und WPM.

5.12. Daten in Statistik speichern

Die gesammelten Nutzerspezifischen Daten müssen in der Datenbank zugeordnet gespeichert werden und auch für jeden Benutzer persönlich abgerufen werden können. Dafür wird jeder Benutzer mit einem einzigartigen Schlüssel versehen, der an die Daten gehängt wird. Somit können diese eindeutig zugeordnet werden. Die Daten werden von dem Server automatisch in die Datenbank eingespeichert und automatisch mit dem Schlüssel versehen. Dieser Use Case ist ähnlich zu „Eigene Text einfügen“.

5.13. Eigenen Text einfügen

Jeder Nutzer kann zur eigenen Übung persönliche Texte einspeichern. Diese werden in der Datenbank gespeichert und können nur von dem Nutzer genutzt werden. Daher werden sie mit dem Nutzerschlüssel versehen und sonst automatisch in die Datenbank eingepflegt. Dieser Use Case ist ähnlich zu „Daten in Statistik speichern“.

5.14. Lernstatistik anzeigen

Jeder Nutzer kann in seinem Profil seine Persönliche Nutzerstatistik abfragen. Dafür werden die aktuellen Daten aus der Datenbank ausgewertet und in einem Diagramm dargestellt. Der Nutzer kann zwischen verschiedenen Diagrammen und Datensätzen wechseln. (z.B. Balkendiagramm der FPM der letzten Woche, Kreisdiagramm der Fehler zu WPM der ganzen Zeit, etc.)

