

# NameSplitter

Softwarequalität  
HOR-TINF2021

Erstellt von:  
Schmidt Julian, Blattau Markus, Janocik Luka

Abgabetermin: 10.05.2024

## Inhaltsverzeichnis

User Stories.....	3
Erste User Story .....	3
Zweite User Story.....	3
Dritte User Story .....	3
Vierte User Story .....	3
Fünfte User Story .....	3
Testkonzept .....	4
<i>Ziele:</i> .....	4
Testkriterien:.....	4
Abdeckung: .....	4
Äquivalenzklassen:.....	4
Definition of Done (DoD) .....	5
Design: .....	5
Design (UI):.....	5
Code: .....	5
Release Instructions and Limitations (RiLI):.....	5
Unit Tests (UT):.....	5

## User Stories

### *Erste User Story*

Als Entwickler wünsche ich mir, dass eine eingegebene Anrede in ihre Bestandteile zerlegt werden kann. Dies ermöglicht eine effiziente Speicherung dieser innerhalb der anhängenden Datenbank. Ich weiß, dass das Ziel erreicht ist, wenn sie die Eingabe in die Bestandteile Titel, Anrede, standardisierte Briefanrede, Vorname, Nachname und Geschlecht unterteilt.

Story Points: 5

Priorität: 1

### *Zweite User Story*

Als Anwender möchte ich eine Möglichkeit zur manuellen Eingabe, falls eine automatische Zerteilung nicht möglich ist. Dadurch möchte ich sicherstellen, dass meine Angaben valide sind. Ich weiß, dass das Ziel erreicht ist, wenn ich fehlende Informationen manuell ergänzen kann.

Story Points: 3

Priorität: 2

### *Dritte User Story*

Als Anwender möchte ich eine einfache funktionale Benutzeroberfläche. Dies soll dabei helfen, eine Anrede einfach und schnell eingeben und speichern zu können. Ich weiß, dass das Ziel erreicht ist, wenn die Benutzeroberfläche intuitiv, vollständig und performant ist.

Story Points: 4

Priorität: 1

### *Vierte User Story*

Als Anwender möchte ich einen neuen Titel innerhalb der Datenbank abspeichern können, um diesen bei der nächsten Anwendung erneut verwenden zu können, wodurch die Effizienz gesteigert werden würde. Ich weiß, dass das Ziel erreicht ist, wenn das System einen neu hinzugefügten Titel bei der nächsten Anwendung erkennt.

Story Points: 3

Priorität: 2

### *Fünfte User Story*

Als Anwender möchte ich Anreden für den DACH-Markt abspeichern können, um das System auch in angrenzenden deutschsprachigen Ländern verwenden zu können. Ich weiß, dass das Ziel erreicht ist, wenn die typischen Anreden des DACH-Marktes erkannt werden.

Story Points: 3

Priorität: 3

## Testkonzept

### *Ziele:*

Das Hauptziel der Tests ist, die Funktionalität und Nutzbarkeit des Kontakt-Splitters zu gewährleisten. Es soll sichergestellt werden, dass die Applikation Anreden korrekt in ihre Bestandteile zerlegt kann oder Benutzer korrekt informiert.

### *Testkriterien:*

1. Funktionalität: Die Applikation soll alle definierten Anforderungen erfüllen.
2. Zuverlässigkeit: Die Ergebnisse sind konsistent und reproduzierbar. Zudem ist die Applikation fehlertolerant.
3. Benutzbarkeit: Die Applikation soll intuitiv und benutzerfreundlich sein.
4. Leistung: Die Applikation sollte innerhalb einer kurzen Zeit auf die Benutzereingabe reagieren (für den Fall, dass das Backend lokal gehostet wird in unter 0,5 Sekunden).
5. Wartbarkeit: Der Code ist wartbar (nach State of the Art). Die internen Module müssen gut modifizierbar sein.
6. Übertragbarkeit: Das System ist auf Windows 10 und 11 vollständig funktionsfähig.
7. Sicherheit: Falscheingaben werden nicht versehentlich abgespeichert (für die Datensicherheit ist das CRM-System verantwortlich).
8. Kompatibilität: Front- und Backend sind jeweils getrennt voneinander austauschbar und erweiterbar.

### *Abdeckung:*

Es werden sowohl vollständig zerlegbare als auch nicht vollständig zerlegbare Anreden getestet, um sicherzustellen, dass die Applikation unter allen möglichen Umständen korrekt funktioniert.

### *Äquivalenzklassen:*

1. Gängige Anreden: Es werden die gewöhnlichen Anreden wie "Herr", "Frau" und "Dr." überprüft.
2. Falsche Anreden: Es werden offensichtlich falsche und nicht passende Eingaben getestet.
3. Keine Eingabe: Es wird versucht ein leerer String oder „null“ zu parsen.
4. Ungewöhnliche Namen: Namen mit mehreren Bestandteilen wie Doppelnamen oder Namen mit Präfixen.
5. Kein Vorname: Der Vorname fehlt komplett, Nachname ist allerdings vorhanden.
6. Kein Vorname und Nachname: Vorname und Nachname fehlen in der Eingabe. Somit ist die Eingabe nicht valide und kann nicht geparkt werden.

## Definition of Done (DoD)

### *Design:*

- Die im Design enthaltenen Komponenten sind in der Lage alle definierten Anforderungen zu erfüllen
- Das Design ist einfach und lesbar gestaltet.
- Das Design ermöglicht eine Übersicht über das Gesamtsystem.
- Das Design enthält alle Relevanten Komponenten und Klassen.
- Das Backend und Frontend sind voneinander unabhängig ersetzbar.
- 

### *Design (UI):*

- Die UI ist intuitiv und Benutzerfreundlich gestaltet.
- Alle in den Anforderungen definierten Funktionen sind vorhanden.

### *Code:*

- Der Code ist funktional und erfüllt die definierten UserStories.
- Der Code enthält keine Fehler, welche zum Abstürzen der Applikation oder zur Beschädigung bereits gespeicherter Daten führt.
- Der Code folgt den gängigen Codierungsstandards und Konventionen.
- Der Code ist verständlich und gut dokumentiert.
- Alle definierten Anforderungen sind erfüllt und korrekt umgesetzt.
- Fehler werden sinnvoll behandelt oder der Benutzer darauf hingewiesen.
- Das Backend und Frontend sind voneinander unabhängig ersetzbar.

### *Release Instructions and Limitations (RiLI):*

- Die RiLI enthalten alle notwendigen Informationen zur Inbetriebnahme der Applikation.
- Die RiLI erklären klare und korrekte Annahmen und Limitierungen der Applikation.
- Die RiLI sind verständlich formuliert, so dass auch ein Benutzer mit wenig Erfahrung die Applikation Starten und bedienen kann.

### *Unit Tests (UT):*

- Die Tests erfüllen das First Prinzip (Fast, Independent, Repeatable, Self-validating, Timely).
- Die Tests sind verständlich und gut dokumentiert.
- Die im Testkonzept definierten Tests bestätigen die erwarteten Funktionalitäten.
- Die Testabdeckung ist wie im Testkonzept definiert erreicht.
- Die Tests zeigen eine hohe Abdeckung von Funktionen und Logik.

## Releasenotes