

NameSplitter

Softwarequalität
HOR-TINF2021

Erstellt von:
Schmidt Julian, Blattau Markus, Janocik Luka

Repository:
<https://github.com/ProgrammierenHorb/name-splitter>

Abgabetermin: 10.05.2024

Inhaltsverzeichnis

NameSplitter.....	1
User Stories.....	3
Erste User Story	3
Zweite User Story.....	3
Dritte User Story	3
Vierte User Story	3
Fünfte User Story	3
Design	4
Auswahl der Schnittstelle.....	4
Testkonzept	5
Ziele:.....	5
Testkriterien:.....	5
Abdeckung:	5
Äquivalenzklassen:.....	5
Definition of Done (DoD)	6
Design:	6
Design (UI):.....	6
Code:	6
Release Instructions and Limitations (RiLI):	6
Unit Tests (UT):.....	6
Releasenotes.....	7
Benutzung.....	7
Backend ist nicht erreichbar:.....	7
Hauptanzeige.....	7
Rückgabe des Parsers	9
Titel löschen.....	9
Titel hinzufügen	10
Funktionsumfang	11
Gender parsen	11
Akademische Titel parsen.....	12
Namen parsen	13
Formatierungen	14
Standardisierte Briefanrede	14
Einschränkungen.....	15

User Stories

Erste User Story

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

Story Points: 5

Priorität: 1

Zweite User Story

Als Anwender wünsche ich mir eine Möglichkeit zur manuellen Eingabe, falls eine automatische Zerteilung nicht möglich ist, weil ich sicherstellen möchte, 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 wünsche ich mir eine einfache funktionale Benutzeroberfläche, weil es dabei hilft, eine Anrede einfach und schnell einzugeben und zu speichern. 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 wünsche ich mir eine Funktion zur Abspeicherung eines neuen Titels in der Datenbank, um diesen bei der nächsten Anwendung erneut verwenden zu können, weil ich hierdurch den neuen Titel nur einmal eintragen muss und die Effizienz somit gesteigert wird. 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 wünsche ich mir, dass ich Anreden für den DACH-Markt abspeichern kann, weil ich das System auch in angrenzenden deutschsprachigen Ländern verwenden möchte. Ich weiß, dass das Ziel erreicht ist, wenn die typischen Anreden des DACH-Marktes erkannt werden.

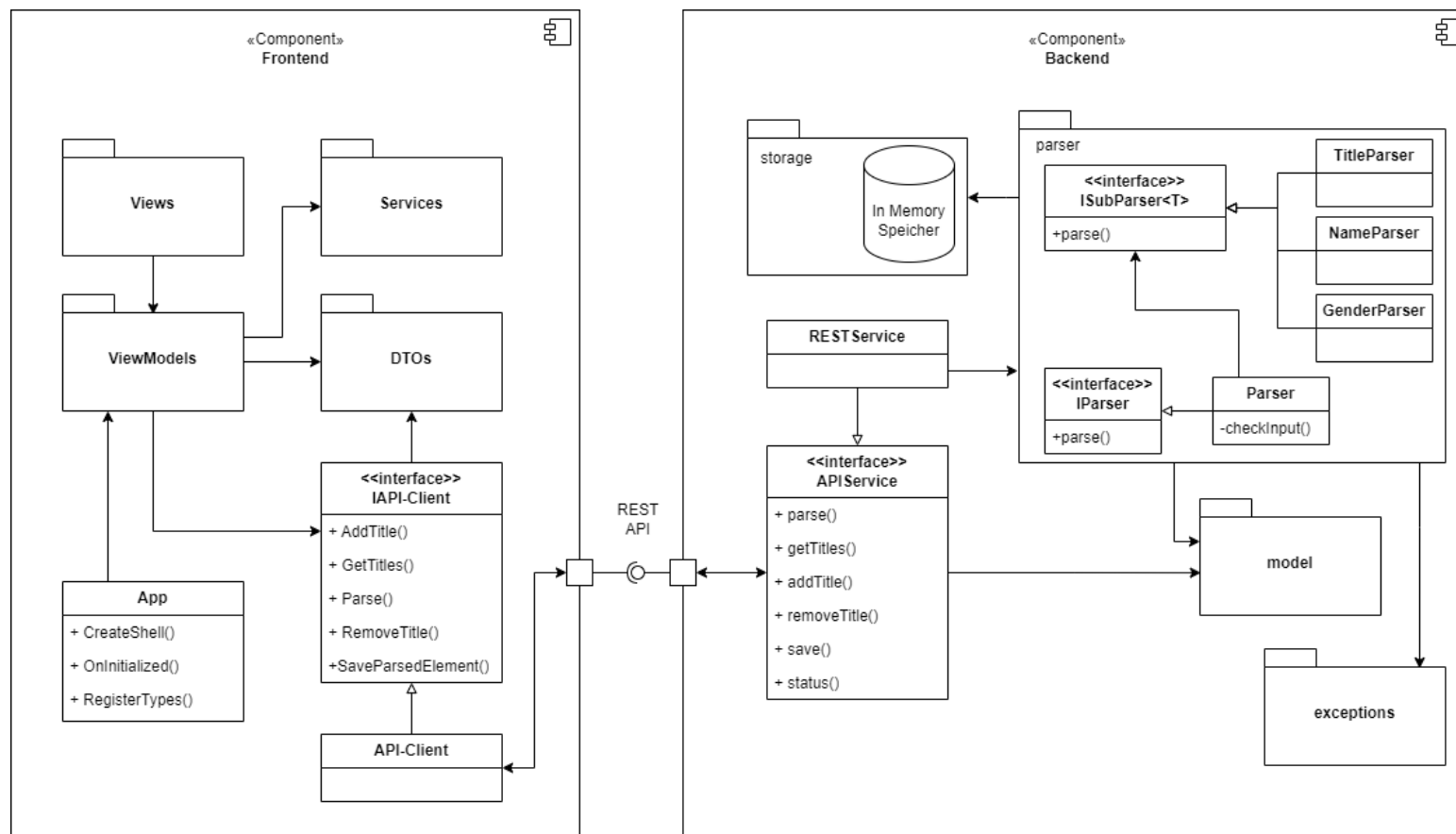
Story Points: 3

Priorität: 3

Design

Auswahl der Schnittstelle

Als Schnittstelle zwischen dem Frontend und dem Backend wurde REST gewählt. Rest bietet den Vorteil, dass Frontend und Backend unabhängig voneinander entwickelt und einfach ausgetauscht werden können. Dadurch kann diese Applikation einfach an ein beliebiges CRM-System angehängt werden. Über die REST-Schnittstelle werden Daten im JSON Format ausgetauscht.



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. Einfache Eingabe: Es werden die gewöhnlichen Anreden wie "Herr", "Frau" und "Dr." überprüft.
2. Falsche Eingabe: 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 als Eingabe: Namen mit mehreren Bestandteilen wie Doppelnamen oder Namen mit Präfixen.
5. Kein Vorname in der Eingabe: Der Vorname fehlt komplett, Nachname ist allerdings vorhanden.
6. Kein Vorname und kein Nachname in der Eingabe: 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

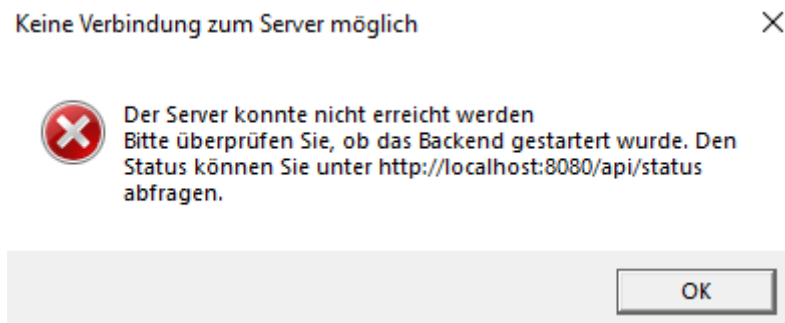
Benutzung

Zu dieser Dokumentation haben Sie ebenfalls eine verpackte Datei namens „NameSplitter_Release.zip“ erhalten. Wird diese entpackt, so erhalten Sie einen Ordner, welcher die Dateien „NameSplitter_Release.exe“, „NameSplitter_Release.jar“ sowie „NameSplitter_Release.bat“ enthält. Die Datei „NameSplitter_Release.bat“ können Sie nun durch einen doppelten Linksklick öffnen. Dadurch wird das Backend („NameSplitter_Release.jar“), sowie das Frontend („NameSplitter_Release.exe“) in jeweils einzelnen Prozessen gestartet. Nun öffnet sich die grafische Benutzeroberfläche, mit welcher Sie nun Ihre Namen parsen und zur Laufzeit speichern können. Durch diese grafische Oberfläche soll eine Speicherung leicht und intuitiv gestaltet werden können [Dritte User Story].

Im Folgenden bekommen Sie einen Einblick in die Benutzung der grafischen Benutzeroberfläche:

Backend ist nicht erreichbar:

Falls das Backend nicht erreichbar ist, bekommen Sie eine Fehlermeldung. Um die Erreichbarkeit manuell zu testen, können Sie in einem beliebigen Browser die URL <http://localhost:8080/api/status> aufrufen. Erhalten Sie nicht die Rückgabe „Online“, so starten sie die Datei „NameSplitter_Release.bat“ erneut.



Hauptanzeige

Dies ist das initial geöffnete Fenster, nachdem die Applikation gestartet wurde. Hier haben Sie verschiedene Möglichkeiten, um den NameSplitter zu verwenden. Anhand der in Rot dargestellten Zahlen werden die Funktionalitäten kurz erläutert:

1: Innerhalb dieser TextBox können Sie Ihre Eingabe tätigen. Durch einen Klick auf den Button „Parsen“ können Sie anschließend die Eingabe formatieren, wodurch sich ein weiteres Fenster öffnet. Eine manuelle Eingabe können Sie durch das Klicken auf den Button „Manuelle Eingabe“ machen.

2: Wird eine Eingabe geparsed und anschließend gespeichert, so werden alle zur Laufzeit gespeicherten Eingaben innerhalb dieser Tabelle ersichtlich sein. Diese Einträge können Sie durch das Klicken des Buttons „Zurücksetzen“ löschen, wodurch alle Einträge innerhalb dieser Tabelle verschwinden werden. Möchten Sie eine gespeicherte Eingabe noch einmal bearbeiten, so können Sie auf den gewünschten Eintrag einen Doppelklick ausführen. Dadurch erscheint ein weiteres Fenster, innerhalb welchem Sie das ausgewählte Element anpassen können.

3: Hier erhalten Sie einen Überblick über alle bisher bekannten Titel. Um einen weiteren hinzuzufügen, Klicken Sie auf den Button „Hinzufügen“, wodurch sich ein weiteres Fenster öffnet. Das Löschen eines Titels erfolgt ebenfalls hier durch einen Doppelklick auf den Titel. Anschließend erscheint ein Bestätigungsfenster.

NameSplitter

1

Eingabe:

Parsen

Manuell eintragen

Durch einen Doppelklick auf einen Eintrag innerhalb der Tabelle, können Sie Ihre Einträge anpassen.

2

Titel	Vorname	Nachname	Briefanrede	Geschlecht
Dr.	Max	Mustermann	Sehr geehrter Herr Dr. M	Männlich

Zurücksetzen

Durch einen Doppelklick auf ein Titel können Sie diesen löschen


3

Regex	Verfügbare Titel	Geschlecht	Priorität
Prof.\?s	Prof.		10
Professor\s?	Prof.	MALE	10
Professorin\s?	Prof.	FEMALE	10
Dr\.\?s*habil\.\?s	Dr.habil.		20
Dr\.\?s*rer\.\?s*nat\.\?s\?s=	Dr.rer.nat.		30
Dr\.\?s*med\.\?s	Dr.med.		40
Dr\.\?s*phil\.\?s	Dr.phil.		50
Dr\.\?s*rer\.\?s*pol\.\?s	Dr.rer.pol.		60
Dr\.\?s*oec\.\?s	Dr.oec.		70
Dr\.\?s*ing\.\?s	Dr.-Ing.		80
Dr\.\?s	Dr.		90

Hinzufügen

Rückgabe des Parsers

Dieses Fenster erscheint, nachdem innerhalb der Hauptanzeige ein gespeichertes Element bearbeitet, eine Eingabe geparsed oder eine manuelle Eingabe erfolgen soll. Der Titel „Das Parsen war erfolgreich“ signalisiert den Status, nachdem eine Eingabe erfolgt ist. Darunter stehen die Fehler, falls welche aufgetreten sind. Dadurch kann dieser auch seine Eingaben manuell nachträglich eintragen oder verändern, um sicherzustellen, dass diese wie erwartet gespeichert werden [Zweite User Story]. Ansonsten können Sie hier einen Vor- und Nachnamen, das Geschlecht, sowie Titel eingeben. Innerhalb des Dropdowns für die Titel ist das Element „-Keine Auswahl-“, wodurch Sie signalisieren, dass dieses einzelne Dropdown nicht mehr beachtet werden soll. Nach dem Speichern wird nicht nur die Tabelle innerhalb der Hauptanzeige, sondern auch die standardisierte Briefanrede generiert.

 Rückgabe des Parsers— □ ×


Das Parsen war erfolgreich

Im Folgenden stehen alle gefundenen Elemente, welche Sie nun noch vor dem Speichern anpassen können.


Eingabe:

Dr. Max Mustermann


Vorname




Nachname:



Geschlecht:


 ☒ Männlich ☐ Weiblich ☐ Divers



▼

Titel löschen

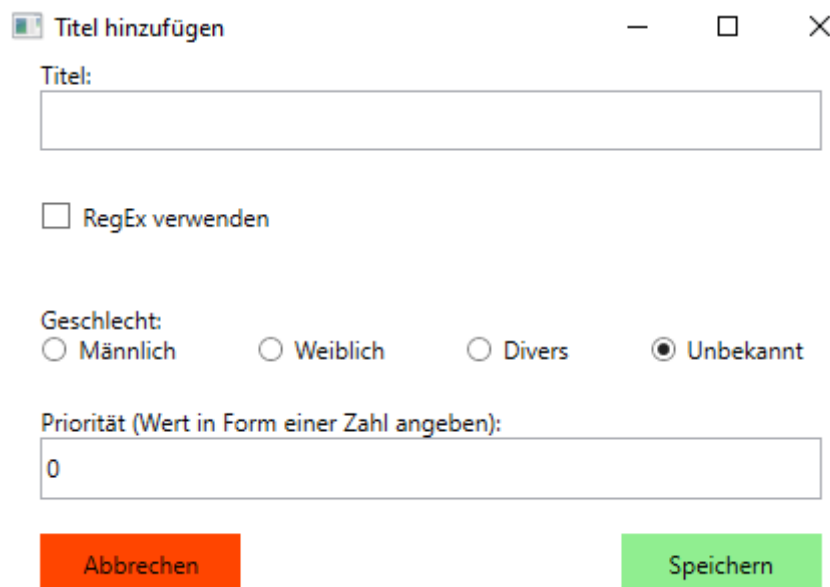
Möchten Sie einen vorhandenen Titel zur Laufzeit löschen, so können Sie dies durch einen Doppelklick auf diesen innerhalb des Hauptfensters machen. Anschließend öffnet sich ein Bestätigungsfenster, innerhalb welchem Sie den ausgewählten Titel final löschen können.

 Titel löschen— □ ×

Sind Sie sicher, dass sie "Dr.rer.pol." löschen möchten?

Titel hinzufügen

Innerhalb des Hauptfensters können Sie ebenfalls neue Titel hinzufügen, welche innerhalb des Backends gespeichert werden und somit bei einem Neustart der grafischen Benutzeroberfläche weiterhin angezeigt werden können (ein Neustart des Backends löscht die selbst eingetragenen Titel wieder). Dadurch öffnet sich ein weiteres Fenster, in welchem Sie den gewünschten Titel eingeben können. Durch die Checkbox „RegEx verwenden“ erscheint eine weitere TextBox, in welcher Sie einen regulären Ausdruck definieren können, mit welchem der NameSplitter beim Parsen nach dem Titel suchen soll. Standardmäßig sucht dieser nur den Titel, wenn er alleinsteht. Des Weiteren können Sie zu jedem Titel ein Geschlecht, sowie die Priorisierung festlegen. Anhand der Priorisierung wird festgelegt, an welcher Stelle dieser Titel in der standardisierten Briefanrede steht. Desto niedriger die eingegebene Zahl (im positiven Bereich), desto höher die Priorisierung. Wird die Eingabe gespeichert, so erscheint Ihr Titel innerhalb der Tabelle „Verfügbare Titel“ in dem, Hauptfenster [Vierte User Story].



Titel hinzufügen

Titel:

☐ RegEx verwenden

Geschlecht:

☐ Männlich ☐ Weiblich ☐ Divers ☒ Unbekannt

Priorität (Wert in Form einer Zahl angeben):

0

Abbrechen Speichern

Funktionsumfang

Der Parser kann einen komplexen Eingabestring (begrenzt auf die Unicode-Klassen L, M und Z, sowie den Punkt (.), das Komma (,), einen Bindestrich (-) und Apostroph (')) entgegennehmen und diesen in die Bestandteile (Geschlecht, Titel, Vorname, Nachname) umwandeln. Außerdem kann mit den strukturierten Daten eine standardisierte Briefanrede generiert werden. Titel bezieht sich hierbei auf eine Liste mit beliebig vielen hinterlegten akademischen Titeln [Erste User Story].

Gender parsen

Das Geschlecht einer Person kann anhand folgender Kriterien erkannt werden:

- 1) Eine genderspezifische Anrede wird eingegeben (z.B. Herr Max Mustermann)
- 2) Ein genderspezifischer Titel wird eingegeben (z.B. Professorin Erika Mustermann)
- 3) Wenn keines der beiden Kriterien in der Eingabe gegeben ist, versucht das Programm mithilfe des eines Datensatzes das Geschlecht „zu schätzen“.

Der Datensatz beinhaltet rund 150000 gängige Vornamen und weist diesen ein Geschlecht zu. Er wird automatisch mit der Applikation ausgeliefert. Wenn der Parser den ersten Vornamen des eingegebenen Namens in der Datenbank findet. Wird das Geschlecht, welches diesem Namen im Datensatz zugeordnet ist, verwendet. Zu beachten ist, dass diese Methode nur eingesetzt wird, wenn 1) und 2) nicht gegeben sind. Der Benutzer kann dann in der GUI das Geschlecht für die Eingabe noch anpassen, sollte es nicht das gewünschte sein.

Beispiel:

„Herr Andrea Pirlo“

Geschlecht	Titel	Vorname	Nachname
M	-	Andrea	Pirlo

„Andrea Pirlo“

Geschlecht	Titel	Vorname	Nachname
F ¹	-	Andrea	Pirlo

¹Andrea wird im Datensatz als weiblicher Name geführt

Akademische Titel parsen

Der Parser kann beliebig viele akademische Titel in der Eingabe erkennen. Zum Start der Applikation sind bereits einige Titel hinterlegt. Weitere Titel kann der Benutzer über die GUI hinzufügen. Der Name wird mithilfe eines regulären Ausdrucks erkannt, sodass gewisse Variationen einer Eingabe möglich sind. Der später ausgegebene Titel hat jedoch eine einheitliche Form.

Beispiel:

„Dr.rer.nat. John Doe“

„Dr. rernat John Doe“

„Drrernat John Doe“

führen alle zu

Geschlecht	Titel	Vorname	Nachname
M	Dr. rer. nat.	John	Doe

Akademische Titel können auch Informationen zum Geschlecht der Person beinhalten (vgl. 11), dies muss aber nicht der Fall sein.

Professor → M

Professorin → F

Prof. → <Keine Geschlechtsinformation>

Außerdem hat jeder Titel eine Priorität, die seine Position in der standardisierten Briefanrede angibt. Der gleiche Titel kann in der Eingabe mehrmals vorkommen.

Beispiel:

Herr B.Sc. Dr.-Ing. Dr. rer. nat. Prof. Paul Steffens

Geschlecht	Titel	Vorname	Nachname
M	Prof. Dr. rer. nat Dr.-Ing. B.Sc.	Paul	Steffens

Wie man sehen kann, werden die Titel in der Ausgabe entsprechend ihrem akademischen Grad angeordnet.

Namen parsen

Der Parser unterstützt folgende Arten von Namen

- 1) Einfacher Name: John Doe
- 2) Mehrfache Vornamen: John Martin Doe
- 3) Adelige Präpositionen: John van Doe
- 4) Präpositionen in Kombination mit Adelstiteln. Adelstitel werden in diesem Fall als Teil des Nachnamens und nicht als Titel angesehen: John Freiherr von Doe
- 5) Doppelte/Mehrfache Nachnamen (werden in der Ausgabe des Parsers mit einem Bindestrich verbunden):
 - 1) Können erkannt werden, wenn sie hinter einer adeligen Präposition stehen, z.B. „John van Maier Müller“
 - 2) Können in der Schreibweise <Nachname, Vorname> erkannt werden z.B. „Maier Müller, John“
 - 3) Müssen ansonsten bei der Eingabe mit einem Bindestrich angegeben werden, da ansonsten nicht entscheidbar ist, ob es sich um einen weiteren Vornamen oder Nachnamen handelt (der Parser zählt es dann standardmäßig als Vorname): z.B. „John Maier-Müller“

Durch diese Funktionen können komplexe Namen geparsed werden.

Beispiel:

„Karl-Theodor Maria Nikolaus Johann Jacob Philipp Franz Joseph Sylvester Freiherr von und zu Guttenberg“

Geschlecht	Titel	Vorname	Nachname
-2		Karl-Theodor Maria Nikolaus Johann Jacob Philipp Franz Joseph Sylvester	Freiherr von und zu Guttenberg

Formatierungen

Der Parser ist in der Lage viele unschön formatierte Eingaben zu erkennen und selbstständig zu verschönern.

Beispiel:

„prOf DR. JOHN JoE dOe-SmltH“

Geschlecht	Titel	Vorname	Nachname
M	Prof. Dr.	John Joe	Doe-Smith

Standardisierte Briefanrede

Die App ist in der Lage eine standardisierte Briefanrede aus dem Namen zu generieren [Fünfte User Story], dabei gibt es einige Szenarien, je nach:

- 1) Das Geschlecht ist männlich oder weiblich
 - 1) Weiblich: „Sehr geehrte Frau [Titel] <Nachname>“
 - 2) Männlich: „Sehr geehrter Herr [Titel] <Nachname>“
- 2) Das Geschlecht ist divers/ nicht angegeben
 - 1) Es ist ein Vorname angegeben: „Guten Tag <Vorname> <Nachname>“
 - 2) Es ist KEIN Vorname angegeben: „Sehr geehrte Damen und Herren“

²Der Vorname „Karl-Theodor“ ist nicht in der Namensdatenbank vorhanden

<Vorname> ist ein Platzhalter für den spezifischen Vornamen

<Nachname> ist ein Platzhalter für den spezifischen Nachnamen

[Titel] ist ein Platzhalter für alle angegebenen Titel sortiert nach ihrer Priorität (Je kleiner die Priorität desto weiter vorne der Titel) in der Form „Titel1 Titel2 ... TitelN“

Einschränkungen

- Es können aktuell nur akademische Titel vom Benutzer erweitert werden. Adelstitel, Präfixe (van, von, ...), Anreden (Herr, Frau, ...) sind nicht über die GUI erweiterbar
- Wenn der Benutzer z.B. einen Titel oder Anrede eingibt, der nicht im System hinterlegt ist, wird der Parser diese als Teil des Namens werten. Es gibt allerdings zumindest einen Kontrollmechanismus, der einen Fehler wirft, wenn der unerkannte Titel Punkte („.“) enthält
- Adelstitel können aktuell nur in Zusammenhang mit Präfixen (van, von, ...) erkannt werden. Er kann daher nicht allein stehen. Außerdem werden Adelstitel als Teil des Nachnamens und nicht als Titel angesehen (dies ist eine bewusste Entscheidung und keine Einschränkung). Beispiel: Bei „Prinz Markus von Anhalt“ wird „Prinz“ nicht als Adelstitel erkannt, bei „Markus Prinz von Anhalt“ allerdings schon