

REPORT

1. Use Cases und Szenarios

In dieser Übung haben wir gelernt, was man unter Use Cases und Szenarios versteht und wie man diese entwirft. Use Cases benutzt man, um Software entwerfen zu können und gleichzeitig die Anforderungen an diese festzuhalten. Dabei sollte jeder diesen Use Case verstehen können, unabhängig vom technischen Hintergrund. Sie sind losgelöst von der technischen Umsetzung. Ein Use Case setzt sich aus bestimmten Parametern zusammen wie einem Namen, Akteuren oder auch die Folgen. Ein Use Case fasst mehrere Szenarien zusammen, die eine detaillierte Beschreibung eines konkreten Ablaufs darstellen und die Schritte derer auflistet.

Wir haben gelernt, dass es nicht DAS EINE Use Case Schema gibt, nach dem sich jeder richtet, sondern dass diese in ihren Details von Benutzer zu Benutzer unterschiedlich aussehen können. Jedoch gibt es die UML (Unified Modeling Language), mit Hilfe derer Konventionen geschaffen wurden, wie ein Use Case Diagramm auszusehen hat (nein, man kann nicht einfach hübschere Icons verwenden).

2. Erstellen der Use Cases & Szenarios

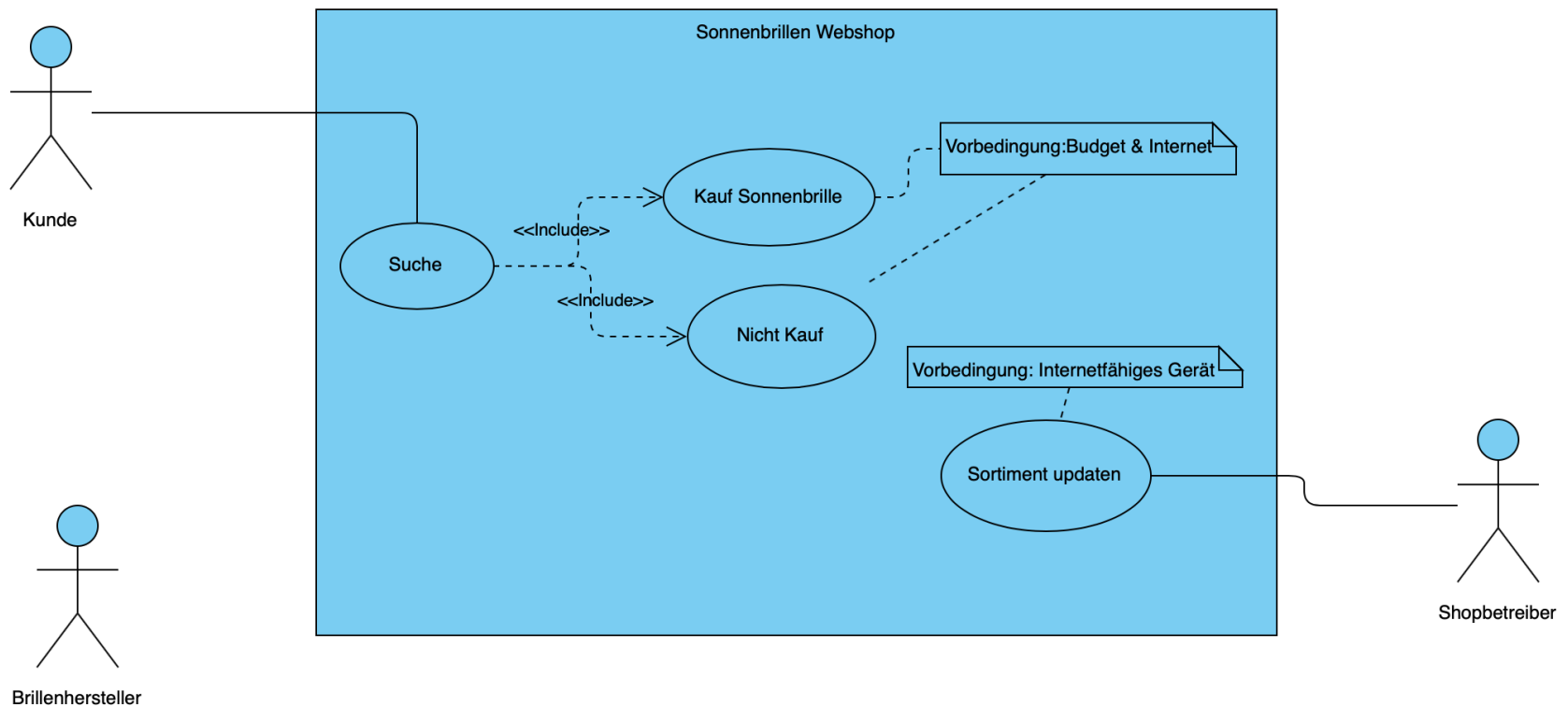
Als wir mit dem Erstellen der Use Cases begonnen haben, nahmen wir uns als erstes den Use Case "Kunde sucht eine Sonnenbrille" vor. Schnell merkten wir aber, dass wir diesen Case für die weitere Beschreibung zu ungenau benannt hatten. Deshalb änderten wir diesen in "Kunde kauft eine Sonnenbrille auf unserer Seite" und als Gegenstück "Kunde kauf keine Sonnenbrille auf unserer Seite". Wir erstellten ein paar Szenarios, die sich auf das Filtern der Suchergebnisse auf der Webseite bezogen und bemerkten schnell, wie anfangs bereits richtig vermutet, dass wir eigentlich noch einen übergeordneten Use Case brauchen, der sich nur mit der Suche beschäftigt. Auf diesem ließen wir dann verschiedene Szenarien mit verschiedenen Filterparametern aufbauen.

Bei den Use Cases, die vom 1. abhängig waren, war uns nicht ganz klar, wie wir bei den nächsten Schritten mit der Nummerierung fortfahren sollten. Wir entschieden uns schlussendlich dafür, wieder bei 1. anzufangen, das der Use Case an sich ja unique ist.

Die Szenarien schrieben sich leicht und da hätten wir auch noch einige mehr ausarbeiten können, aber für den Umfang einer Übung fanden wir vier Szenarios genügend.

3. Erstellen des Use Case Diagramms

Das anschließende Erstellen des Use Case Diagramms war dank einer Webanwendung (<https://online.visual-paradigm.com/diagrams.jsp>) nicht weiter schwierig, auch wenn nicht alle Pfeile vorhanden waren (z.B. die für die Vorbedingungen, in unserer Grafik nur gestrichelt dargestellt).



4. Erstellen des Klassendiagramms

Das System Substantive, Verben und Adjektive zu unterstreichen um aus einer Beschreibung die benötigten Klassen, Methoden und Attribute herauszufinden kannten wir bereits vom Anfang des letzten Semesters. Um das Klassendiagramm für diese Aufgabe zu erstellen haben mit dem ersten Use Case und den dazugehörigen Szenarios angefangen diese Methode anzuwenden. Nachdem wir durch alle Use Cases durch waren hatten wir die folgenden Worte gesammelt:

- die Substantive waren Kunde, Sonnenbrille, Gerät, Verbindung, Webshop, Ergebnisse, Betreiber des Shops, Brillenhersteller, Sonne, Sommer, Ergebnisse, Stärke, Filterfunktion, Erwachsene, Kinder, Budget, Mitarbeiter/Admin, Sortiment, Rechten.
- die Adjektive waren internet-fähiges, gefilterte, passenden, verschiedene, neue.
- die Verben waren sucht, anschauen, gesehen, findet, wählt, klickt, filtern, lädt neu, angezeigt, kauft, gefällt, bezahlt, bekommt, wird verschickt, kommt an, merkt sich, aktualisiert, zieht an, loggt sich ein, fügt hinzu, speichert, loggt sich aus.

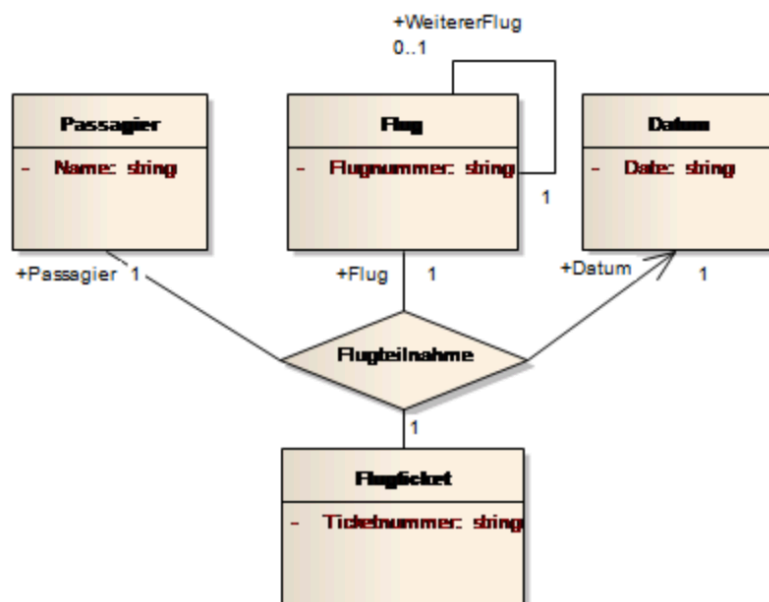
Im nächsten Schritt haben wir dann alle überflüssigen Begriffe ausgemistet. Am ende blieben 7 Klassen für unser Diagramm übrig. Es war eher schwierig zu entscheiden, ob Methoden die Notwendigkeit aufweisen, public, private oder protected zu sein. Meistens haben wir uns auf public geeinigt.

5. Assoziationsknoten

Zu Assoziationsknoten haben wir uns vor allem auf dieser Website belesen: <https://www.sparxsystems.de/ressourcen/literatur/leseprobe-zu-projektabwicklung-mit-uml-und-enterprise-architect/klassendiagramm-class-diagram/>

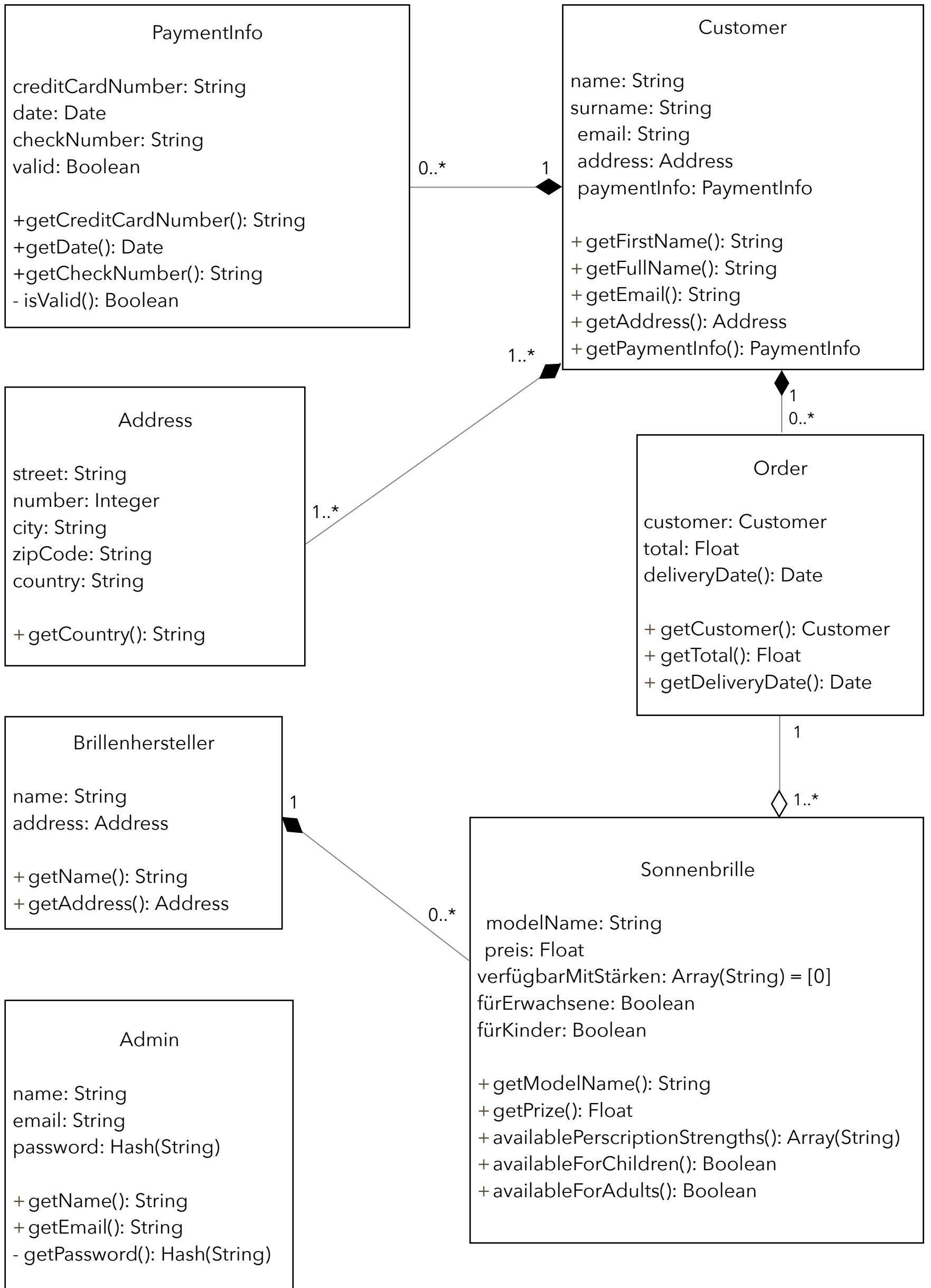
Laut dieser Quelle Wird ein Assoziationsknoten dann benutzt, wenn nicht nur zwei Klassen in direkter Beziehung zueinander stehen, sondern wenn es mehrere Klassen sind und eine weitere Klasse sich zusammensetzt aus den anderen Klassen und nicht etwa mit diesen direkt assoziiert ist. Für jede Referenz darf es nur eine Instanz der Assoziationsklasse geben. In der Abbildung unten heißt das also, dass es nur eine Instanz von Flugteilnahme für eine Kombination von Passagier, Flug, Datum und Fluggesellschaft geben darf. Diese Notation stammt von der Beziehungsnotation in ER-Modellen ab.

Siehe Abb. 38 der Quelle Stand 14.04.19



Beim erstellen eines Assoziationsknotens müssen, wie in einem ER-Modell, die Kardinalitäten der Beziehungen beachtet und mit angegeben werden.

Wir könnten einen solchen Assoziationsknoten durchaus gebrauchen, und zwar für unsere Klasse "Order". Diese beinhaltet Referenzen zu Sonnenbrille, Customer und PaymentInfo. Wir könnten uns vorstellen, dass es dafür auch nur eine Instanz geben darf und somit die Bedingungen für eine Assoziationsklasse erfüllt sind.



USE CASE 1:

Name:	Kunde sucht nach einer Sonnenbrille .
Precondition:	internet-fähiges Gerät und Verbindung
Primary actor:	Kunde
Scope:	Webshop
Summary:	Der Kunde sucht auf der Webseite nach einer Sonnenbrille und hat gefilterte Ergebnisse .
Level:	Überblick
Stakeholders:	Kunde - möchte eine Sonnenbrille / möchte Sonnenbrillen anschauen Betreiber des Shops - Gewinn/Anteile Brillenhersteller - Gewinn
Minimal guarantee:	Kunde hat Sonnenbrillen gesehen
Success guarantee:	Kunde findet eine Sonnenbrille
Trigger:	Sonne, Sommer
Basic flow:	1. Kunde geht auf die Webseite 2. Kunde sucht/filtert Ergebnisse

Szenario 1

Name:	Kunde sucht nach Brillen mit Stärke
Precondition:	Kunde ist auf der Website und sucht nach Sonnenbrillen
Sequence of Steps:	1. Kunde findet die Filterfunktion . 2. Kunde wählt passende Stärke aus. 3. Kunde klickt auf filtern . 4. Die Seite lädt den Anzeigebereich neu .
Postcondition:	Nur die passenden Modelle werden angezeigt .

Szenario 2

Name:	Kunde sucht nach Brillen ohne Stärke
Based on:	Kunde sucht nach Brillen mit Stärke
Exception:	2. Kunde wählt aus, dass nur Brillen ohne Stärke angezeigt werden.

Szenario 3

Name:	Kunde sucht nach Brillen für Erwachsene
Precondition:	Kunde ist auf der Website und sucht nach Sonnenbrillen
Sequence of Steps:	1. Kunde findet die Filterfunktion. 2. Kunde wählt aus, dass nur Brillen für Erwachsene angezeigt werden. 3. Kunde klickt auf filtern. 4. Die Seite lädt den Anzeigebereich neu.
Postcondition:	Nur die passenden Modelle werden angezeigt.

Szenario 4

Name: Kunde sucht nach Brillen für **Kinder**
Based on: Kunde sucht nach Brillen für Erwachsene
Exception: 2. Kunde wählt aus, dass nur Brillen für Kinder angezeigt werden.

USE CASE 2:

Name: Kunde kauft eine Sonnenbrille.
Precondition: Man hat **Budget** für eine Sonnenbrille und ein internet-fähiges Gerät.
Primary actor: Kunde
Scope: Webshop
Summary: Der Kunde sucht nach Sonnenbrillen in unserem Webshop und **kauft** eine.
Level: Überblick
Stakeholders: Kunde - möchte eine Sonnenbrille
Betreiber des Shops - Gewinn/Anteile
Brillenhersteller - Gewinn
Minimal guarantee: Kunde findet eine Sonnenbrille
Success guarantee: Kunde findet eine Sonnenbrille, die ihm sehr **gefällt**.
Trigger: Use Case 1
Basic flow: baut auf UC 1 auf
1. Kunde sucht sich eine Brille aus
2. Kunde **bezahlt** die Brille -> Extension Point
3. Kunde **bekommt** Bestätigung -> Extension Point
4. Brille **wird verschickt** -> Extension Point
5. Brille **kommt** beim Kunden **an** -> Extension Point

USE CASE 3:

Name: Kunde kauft keine Sonnenbrille.
Precondition: Internetfähiges Gerät und Verbindung
Primary actor: Kunde
Scope: Webshop
Summary: Der Kunde sucht nach Sonnenbrillen in unserem Webshop und kauft keine.
Level: Überblick
Stakeholders: Kunde - Interesse an Sonnenbrillen
Betreiber des Shops - Gewinn/Anteile
Brillenhersteller - Gewinn
Minimal guarantee: Kunde hat sich **verschiedene** Sonnenbrillen angeschaut
Success guarantee: Kunde **merkt sich** die Seite für späteren Sonnenbrillen Kauf
Trigger: Use Case 1
Basic flow: baut auf UC 1 auf
1. Kunde sucht sich keine Brille aus
2. Kunde kauft keine Brille

USE CASE 4:

Name:	Shopbetreiber fügt Sonnenbrille dem Sortiment hinzu.
Precondition:	Internetfähiges Gerät und Verbindung; Adminrechte
Primary actor:	Shopbetreiber
Scope:	Webshop
Summary:	Ein Mitarbeiter fügt neue Sonnenbrillen zum Sortiment hinzu.
Level:	Überblick
Stakeholders:	Kunde - Aktuelles Angebot Betreiber des Shops - Gewinn/Anteile, Aktuelles Angebot Brillenhersteller - Gewinn, Aktuelles Angebot
Minimal guarantee:	Das Angebot des Webshops ist aktualisiert
Success guarantee:	Das aktuelle Angebot zieht neue Kunden an
Trigger:	Neues Sortiment
Basic flow:	<ol style="list-style-type: none">1. Admin geht auf die Webseite2. Admin loggt sich mit seinen Rechten ein3. Admin fügt neue Produkte hinzu4. Admin speichert Aktualisierung5. Admin loggt sich aus