

# TRAVEL.CH INTEGRATIONSTESTS

Simon Lang

24. Oktober 2015

Version 1.0.0

STUDIENGANG	Informatik 5 Ba 2012
SEMESTERARBEIT	Travel.ch Integrationstests
DOZENT	Matthias Bachmann
SCHULE	ZHAW - School of Engineering

Kurzfassung

**Schlagwörter:** Integration tests, Selenium, Testing

---

## Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ziele . . . . .	1
1.2	Begründung . . . . .	1
<b>2</b>	<b>Beschreibung der Aufgabe</b>	<b>2</b>
2.1	Ausgangslage . . . . .	2
2.2	Ziele der Arbeit . . . . .	2
2.3	Aufgabenstellung . . . . .	2
2.3.1	Recherche . . . . .	2
2.3.2	Anforderungen und Analyse . . . . .	2
2.3.3	Konzept . . . . .	3
2.3.4	Umsetzung . . . . .	3
2.4	Erwartete Resultate . . . . .	3
<b>3</b>	<b>Recherche</b>	<b>4</b>
3.1	Rahmenbedingungen . . . . .	4
3.1.1	Infrastruktur . . . . .	4
3.1.2	Tooling . . . . .	4
3.2	Testing Frameworks . . . . .	5
3.2.1	lokale Ausführung vs Verwendung eines Services . . . . .	5
3.2.2	Prototyp . . . . .	6
3.3	Zielgruppe . . . . .	6
<b>4</b>	<b>Anforderungen und Analyse</b>	<b>8</b>
4.1	Funktionalität . . . . .	8
4.2	Browserabdeckung . . . . .	9
4.3	Kosten & Zeit . . . . .	9
<b>5</b>	<b>Konzept</b>	<b>10</b>
5.1	Übersicht . . . . .	10
5.1.1	Kunde & Passagierangaben . . . . .	11
5.2	Happy Path . . . . .	11
5.2.1	Citytrip . . . . .	12

5.2.2	Flight	13
5.2.3	Hotel	14
5.3	Smoke Tests	14
5.3.1	Citytrip	15
5.3.2	Flight	16
5.3.3	Hotel	17
5.4	Main Tests	17
5.4.1	Citytrip	18
5.4.2	Flight	20
<b>6</b>	<b>Umsetzung</b>	<b>21</b>
6.1	Selenium	21
6.1.1	Beispiele	22
6.2	Infrastruktur	23
6.3	Design Patterns	24
6.3.1	Page Object	25
6.3.2	Page Factory	26
6.4	Architektur	28
6.4.1	Tests	29
6.4.2	Bootstrap	30
6.4.3	TestRunner	32
6.4.4	Pages	34
6.4.5	Modules	36
6.5	Probleme	36
6.5.1	Timing	36
<b>7</b>	<b>Schlussfolgerung</b>	<b>38</b>
<b>A</b>	<b>Testübersicht</b>	<b>39</b>
A.1	Testübersicht	40
<b>B</b>	<b>Funktionalitäten</b>	<b>45</b>
B.1	Diverses	46
B.2	Startseite	48
B.3	Engine Startseite	52
B.4	Hotelauswahl	53
B.5	Hotelzimmer Auswahl	56
B.6	Flugauswahl	59
B.7	Checkout: Passagiere	62
B.8	Checkout: Bezahlart	64
B.9	Checkout: Übersicht	65
B.10	Bestätigungsseite	66
	<b>Quellenverzeichnis</b>	<b>67</b>

---

## Abbildungsverzeichnis

---

4.1 Anforderungsdreieck . . . . .	8
6.1 Travelwindow AG Infrastruktur . . . . .	23
6.2 Klassendiagramm Übersicht . . . . .	28
6.3 Klassendiagramm Tests . . . . .	29
6.4 Klassendiagramm Bootstrap . . . . .	30
6.5 Klassendiagramm TestRunner . . . . .	32
6.6 Flow Chart Service-Anbieter Anbindung . . . . .	33
6.7 Klassendiagramm Pages . . . . .	34
6.8 Sequenzdiagramm für den Ablauf eines CityTrip Tests . . . . .	35
6.9 Klassendiagramm Modules . . . . .	36

---

## Tabellenverzeichnis

---

3.1 Top 10 Destinationen . . . . .	7
5.1 Testspezifikation: Happy Path - Citytrip . . . . .	12
5.2 Testspezifikation: Happy Path - Flight . . . . .	13
5.3 Testspezifikation: Happy Path - Hotel . . . . .	14
5.4 Testspezifikation: Smoke Test - Citytrip . . . . .	15
5.5 Testspezifikation: Smoke Test - Flight . . . . .	16
5.6 Testspezifikation: Smoke Test - Hotel . . . . .	17
5.7 Testspezifikation: Main Tests - Citytrip . . . . .	18
5.8 Testspezifikation: Main Tests - Flight . . . . .	20

## Akronyme

Bezeichnung	Beschreibung
DOM	Document Object Model
SVN	Subversion
URL	Uniform Resource Locator

## Glossar

### *Closure*

Eine Closure ist eine anonyme Funktion, welche Zugriff auf die Variablen aus dem Erstellungskontext enthält. In C# können diese mit den sogenannten Arrow-Functions erstellt werden:

```
Action addOne = number => number + 1;
```

Dies ist die Deklaration einer Closure. Sie ist vom Typ Action und nimmt einen Parameter entgegen, welcher "number" heisst. Die Closure wird über eine Arrow-Function definiert. Wenn Sie nur einen Ausdruck enthält, wie im obigen Beispiel, so wird dieser automatisch zurückgegeben und es ist keine *return*-Anweisung nötig.

```
addOne(counter);
```

Dies ruft die Closure schlussendlich aufgerufen. Counter hat zum Schluss den Wert 2.

### *HTML*

Hypertext Markup Language (HTML) ist eine textbasierte Auszeichnungssprache welche für die Erstellung von Webseiten verwendet wird. Die HTML-Dateien sind die Grundlagen des Internets und werden vom Browser dargestellt.

### *HTTP-Cookie*

Ein (HTTP-)Cookie ist eine Information, die der Server über den Browser auf dem Computer des Betrachters abspeichert. Die Cookie-Daten werden vom Browser bei jeder Anfrage an den Server zurück übermittelt.

### *IDE*

IDE steht für Integrated development environment und ist eine Sammlung von Anwendungsprogrammen, mit denen Software möglichst ohne Medienbrüche entwickelt werden kann.

### *IIS*

Die Internet Information Services (IIS) ist ein Web-, Mail-, FTP-, WebDAV-Server von Microsoft und wird benötigt um ASP- oder .NET Applikationen auszuführen.

### *Session*

Eine Session ist eine stehende Verbindung eines Clients mit einem Server. Da das Internet auf dem zustandslosen Protokoll HTTP (Hypertext Transfer Protocol)



basiert muss die Sitzung auf der Anwendungsschicht<sup>1</sup> etabliert werden. Dazu wird bei der ersten Kommunikation eine eindeutige Kennung vom Server an den Client übermittelt, welche dieser bei jedem Verbindungsaufbau zurück übermittelt.

#### *Versionierungssystem*

Wird zur Erfassung von Änderungen verwendet. Für jede Änderung wird eine Version angelegt wodurch diese auch rückgängig gemacht werden kann. Wenn mehrere Personen gleichzeitig an einem System arbeiten kann auch überprüft werden wer welche Anpassung vorgenommen hat.

#### *Virtual Machine*

Eine virtuelle Machine bildet einen Rechner einer real existierender Hardware in einer virtuellen Umgebung nach. Dabei handelt es sich um ein vollständiges Betriebssystem, welches sich exakt so verhält, als sei es direkt auf der Hardware installiert.

---

<sup>1</sup> *OSI-Modell* – *Wikipedia*. URL: [https://de.wikipedia.org/wiki/OSI-Modell#Schicht\\_7\\_.E2.80.93\\_Anwendungsschicht\\_.28Application\\_Layer.29](https://de.wikipedia.org/wiki/OSI-Modell#Schicht_7_.E2.80.93_Anwendungsschicht_.28Application_Layer.29) (besucht am 30.07.2015).

# KAPITEL 1

---

## Einleitung

---

1.1 Ziele

1.2 Begründung

# KAPITEL 2

---

## Beschreibung der Aufgabe

---

### 2.1 Ausgangslage

Alle 3 bis 4 Wochen wird auf der Webseite von Travelwindow AG eine neue Version eingespielt. Dies verlangt jedes Mal ein ausführliches Testen, was bislang immer manuell durchgeführt wurde. Dies ist eine sehr monotone und repetitive Arbeit, da die selben Testfälle auf unterschiedlichen Browsern überprüft werden müssen und dadurch oft Fehler übersehen werden. Der Testaufwand würde sich auf 3 bis 4 Tage belaufen, wenn alle kombinierbaren Möglichkeiten von Browser, Sprache und Produkten ausführlich geprüft werden müssen.

### 2.2 Ziele der Arbeit

Die Einführung von automatisierten Integrationstests sollen die repetitiven Testarbeiten minimieren. Diese sollen mittels Selenium umgesetzt werden. Das Ziel der Arbeit ist es, dass die Testpersonen bei jeder neuen Version nur noch testen müssen, was neu umgesetzt wurde. Dadurch sollen mehr Programmfehler aufgedeckt und menschliche Fehler ausgeschlossen werden. Später sollte durch die Automatisierung auch ein Continuous Delivery ermöglicht werden.

### 2.3 Aufgabenstellung

In der Arbeit werden während vier Phasen die automatisierbaren Testfälle entwickelt.

#### 2.3.1 Recherche

Zu Beginn werden die einzusetzenden Frameworks, welche für die Entwicklung und Ausführung der Testfälle benötigt werden, evaluiert.

#### 2.3.2 Anforderungen und Analyse

Als Grundlage für die Konzeptionierung werden bestmöglichst alle Testfälle spezifiziert. Diese sollen die Funktionalität der Webseite vollständig abdecken. Der Product Owner wird die Testfälle priorisieren, so dass entschieden werden kann nach welcher Reihenfolge die Testfälle konzeptioniert und implementiert werden können.

### 2.3.3 Konzept

Die Testfälle, welche während dieses Projektes umgesetzt werden, müssen aus spezifiziert werden. Sprich es sind Vorbedingungen, Testabläufe, Soll-Resultate und Nachbedingungen zu definieren. Zusätzlich wird die gesamte Infrastruktur beschrieben.

### 2.3.4 Umsetzung

Die Testfälle werden Implementiert und schlussendlich in den Application Lifecycle eingebunden, so dass sie automatisiert und zu definierten Zeitpunkten durchgeführt werden können. Je nach Anzahl der Testfälle werden alle oder ein Untermenge davon, in Abhängigkeit der erwähnten Priorisierung, umgesetzt.

## 2.4 Erwartete Resultate

Als Testvorbereitung werden die Testfälle spezifiziert und dokumentiert. Anschliessend werden verschiedene Testframeworks evaluiert und eine Architekturbeschreibung der Testinfrastruktur erstellt. Es werden alle möglichen Testfälle spezifiziert. Diese werden sehr umfangreich ausfallen weshalb Vorbedingungen, Testabläufe, Soll-Resultate und Nachbedingungen nur für jene Tests definiert werden, welche auch tatsächlich umgesetzt werden sollen. Die zu implementierenden Tests werden von der gesetzten Priorisierung bestimmt.

Der nächste Schritt ist die Implementierung der ausformulierten Tests. Dazu wird die Testinfrastruktur erstellt und die Tests programmiert. Ein vergleich mit den Ist-Resultate ist nicht nötig, da dies direkt von den automatischen Tests vorgenommen wird.

Schlussendlich müssen die Test noch in den Application Lifecycle integriert werden.

- Testfallspezifikation inkl. Priorisierung
- Evaluation von Testframeworks
- Architekturbeschreibung der Testinfrastruktur
- Aufsetzen der Testinfrastruktur
- Implementierte Testfälle
- Testfallintegration in den bestehenden Application Lifecycle

# KAPITEL 3

---

## Recherche

---

In diesem Kapitel werden die Rahmenbedingungen, die verschiedene Testing Frameworks (Service-Anbieter) sowie die Zielgruppe recherchiert.

### 3.1 Rahmenbedingungen

Getestet werden soll die Webseite der Firma *travelwindow AG*, welche unter <http://www.travel.ch> erreichbar ist. Die Seite ist mit C# implementiert, weshalb auch die Tests in dieser Programmiersprache umgesetzt werden sollen.

#### 3.1.1 Infrastruktur

Es gibt folgende drei Umgebungen:

- Test
- Quality
- Production

Bei der Entwicklung wird jeder Checkin in das Versionierungssystem (siehe Abschnitt [3.1.2 Tooling](#)) automatisch auf die Test-Umgebung aufgespielt. Sobald ein Stand erreicht ist, welcher vom Business der *travelwindow AG* getestet werden soll, wird diese Version auf die Quality-Umgebung geladen. Sobald diese fertig mit dem Testen sind, wird diese Version auf die Production-Umgebung gespielt und ist damit Live.

#### 3.1.2 Tooling

Der Sourcecode der Software ist auf einem eigenen [Subversion \(SVN\)](#)<sup>1</sup> Versioniert. Es ist jedoch eine Umstellung auf GIT<sup>2</sup> geplant, weshalb neuer Sourcecode fortan nur noch auf dem GIT Hosting Dienst BitBucket<sup>3</sup> gespeichert werden soll.

---

<sup>1</sup> *Apache Subversion*. URL: <https://subversion.apache.org/> (besucht am 26.07.2015).

<sup>2</sup> *Git*. URL: <http://www.git-scm.com/> (besucht am 26.07.2015).

<sup>3</sup> *Git and Mercurial code management for teams*. URL: <https://bitbucket.org/> (besucht am 26.07.2015).

Als Continuous Integration Server wird Bamboo<sup>1</sup> eingesetzt. Dieser kompiliert den Sourcecode und ist dafür verantwortlich, eine Version auf die Umgebungen Test, Quality und Production zu laden.

Das Tooling ist genauer beschrieben im Abschnitt [6.2 Infrastruktur](#).

### 3.2 Testing Frameworks

Der de facto Standard für Integrationstests von Benutzeroberfläche ist Selenium<sup>2, 3</sup>. Code für dieses Framework kann in C# programmiert werden, welcher einen Webbrowser startet und Benutzereingaben simuliert. Dadurch können repetitive Tests automatisiert und auf verschiedenen Browsern durchgeführt werden.

Die Tests können entweder auf einer eigens aufgesetzten Umgebung durchgeführt werden oder an einen Service ausgelagert werden.

#### 3.2.1 lokale Ausführung vs Verwendung eines Services

Werden die Tests lokal oder auf einem Server innerhalb der Firma ausgeführt, laufen die Tests sehr schnell ab und es muss nichts für die Ausführung der Tests bezahlt werden. Bei der Verwendung eines Services dauern die Tests länger und es fallen Laufzeitkosten an. Die Serviceanbieter rechnen pro Minute ab, welche ein Test auf ihrer Infrastruktur läuft.

Der Vorteil der Serviceanbieter liegt darin, dass sie virtuelle Umgebungen für die Durchführung der Tests zur Verfügung stellen. Diese sind mit verschiedenen Betriebssystemen und Browser Konfigurationen ausgestattet. Einige Beispiele solcher Zusammenstellungen:

- Windows 7, Firefox 39
- Windows 8.1, Firefox 39
- OS X Yosemite, Safari 8
- iPhone, iOS 8.4
- Android 4.4, Samsung Galaxy S4

Es ist ersichtlich, dass auch Mobile Browser unterstützt werden. Das Aufsetzen und Pflegen verschiedener Servern mit verschiedenen Betriebssystemen, Browsern und Emulatoren für mobile Endgeräte ist aufwändig und benötigt viel Knowhow. Aus diesem Grund wurde in Absprache mit dem Business von *travel.ch* entschieden, dass die Browser nicht auf einem Rechner installiert werden sollen, sondern ein Service-Anbieter verwendet wird.

Als Testing Framework für Selenium Tests wurden zwei Kandidaten näher in Betracht gezogen. SauceLabs<sup>4</sup> und CrossBrowserTesting<sup>5</sup>.

---

1 *Continuous Integration & Build Server - Bamboo* / Atlassian. URL: <https://www.atlassian.com/software/bamboo> (besucht am 26.07.2015).

2 *Selenium - Web Browser Automation*. URL: <http://docs.seleniumhq.org/> (besucht am 26.07.2015).

3 *Happy 10th Birthday, Selenium* / ThoughtWorks. URL: <http://www.thoughtworks.com/insights/blog/happy-10th-birthday-selenium> (besucht am 26.07.2015).

4 *Sauce Labs: Selenium Testing, Mobile Testing, JS Unit Testing and More*. URL: <https://saucelabs.com/> (besucht am 26.07.2015).

5 *Cross Browser Testing. Real mobile devices & browsers!* URL: <https://crossbrowsertesting.com/> (besucht am 26.07.2015).

Beide Anbieter verfügen über 500 verschiedene Betriebssystem/Browser Konfigurationen<sup>1, 2</sup>. Beide können automatisierte Tests ausführen und liefern als Resultat ein Video der Durchführung sowie Screenshots der einzelnen Tests.

Vom Preis her unterscheiden sie sich auch nicht markant<sup>3, 4</sup>. SauceLabs hat das kleinste Packet mit 120 Minuten Automatisierten Tests für \$12/Monat. Beide Anbietern verrrechnen für 10 Stunden Testing \$49/Monat. Bei den teureren Packet für \$149/Monat bietet CrossBrowserTesting mit 2000- mehr als Saucelabs mit 1800 Minuten.

Um sich für einen Serviceanbieter zu entscheiden wurde ein Prototyp entwickelt, welcher im nächsten Abschnitt beschrieben wird.

### 3.2.2 Prototyp

Um die beiden Frameworks, welche im Abschnitt 3.2.1 lokale Ausführung vs Verwendung eines Services beschrieben wurden, zu testen wurde ein Prototyp entwickelt. Zusätzlich kann dieser die Tests auch lokal laufen lassen. Der Browser sowie der Selenium Treiber müssen dazu jedoch auf dem eigenen Rechner installiert sein. Der Quellcode ist im BitBucket unter <https://bitbucket.org/soultemptation/tw-systemtests-prototype> einsehbar.

#### Auswertung

TODO!!! SauceLabs, da besseres Interface! Oben noch die anderen Service-Anbieter auflisten. Z.B. EU Anbieter um Latenz der US Anbieter zu testen.

### 3.3 Zielgruppe

Die Zielgruppe sind die Benutzer der Webseite, beziehungsweise die verschiedenen Browser, welche sie verwenden, sowie die Destinationen für die sie eine Reise buchen möchten. Denn für diese müssen Tests geschrieben werden.

Auf *travel.ch* wird Google Analytics eingesetzt. Die Auswertung des letzten halben Jahres (25.02.2015 - 25.07.2015) ergab das folgende Browserversionen am meisten verwendet wurden:

- Safari: 32.61%
  - 8.0: 53.40%
  - 7.0: 12.62%
- Internet Explorer: 26.62%
  - 11.0: 74.23%
  - 9.0: 13.95%
- Chrome: 21.14%

---

1 Platforms. URL: <https://saucelabs.com/platforms/> (besucht am 26.07.2015).

2 OS & Browser Configurations for Cross Browser Compatibility Testing | Pick an OS Pick a Browser Test a Website. URL: <https://crossbowseresting.com/browsers> (besucht am 26.07.2015).

3 Sauce Labs: Pricing. URL: <https://saucelabs.com/pricing> (besucht am 26.07.2015).

4 CrossBrowserTesting Pricing Model. URL: <https://crossbowseresting.com/pricing#plans> (besucht am 26.07.2015).

- 40.0: 19.24%
- 41.0: 15.82%
- Firefox: 14.22%
- 36.0: 25.95%
- 37.0: 21.53%

Die Webseite ist in drei Bereiche aufgeteilt: Citytrip (Flug & Hotel), Flug, Hotel. Folgend nun die zehn meist gesuchten Destinationen:

**Tabelle 3.1:** Top 10 Destinationen

#	Citytrip	Flug	Hotel
1	Berlin	Bangkok	London
2	Barcelona	Byculla	Rom
3	Wien	London	Paris
4	London	Berlin	New York City
5	Amsterdam	Miami	Berlin
6	Rom	Palma de Mallorca	Barcelona
7	Lissabon	Los Angeles	München
8	Hamburg	Wien	Palma de Mallorca
9	Prag	San Francisco	Wien
10	Istanbul	Barcelona	Venedig

Es ist demnach wichtig, dass die oben genannten Browser getestet werden sowie die aufgeführten Destinationen.



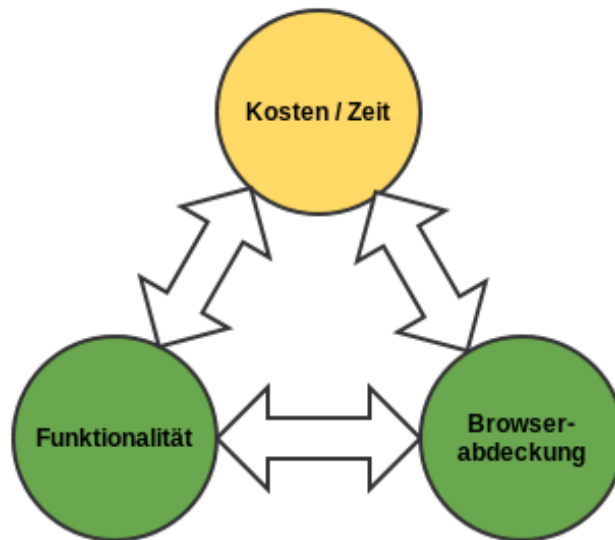
# KAPITEL 4

---

## Anforderungen und Analyse

---

Es gibt folgende Anforderungen an die Testsuite:



**Abbildung 4.1:** Anforderungsdreieck

Je mehr Funktionalität abgedeckt wird, desto länger dauern die Tests. Und für jeden Browser müssen nochmals alle Tests durchgeführt werden. Deshalb stehen „Kosten / Zeit“ im Widerspruch mit der Funktionalität und der Browserabdeckung.

Diese Anforderungen sind in den nächsten Abschnitten beschrieben.

### 4.1 Funktionalität

Die gesamte Funktionalität der Webseite, inklusive der Priorisierung, ist im Anhang [B Funktionalitäten](#) aufgelistet. Die Liste wurde im Wiki von der Hotelplan Management AG erstellt. Diese soll auch nach der Arbeit weiter gepflegt werden und wurde deshalb so aufgebaut, dass sie sehr Übersichtliche ist und einfach bearbeitet und erweitert werden kann.

Vor dieser Arbeit gab es keine Dokumente, welche die Funktionalität der Webseite beschrieb. Deshalb musste die gesamte Funktionalität durch analyse der bestehenden Seite rekonstruiert werden (reverse engineering).

## 4.2 Browserabdeckung

Im Abschnitt 3.2.2 Prototyp sind die meist verwendeten Browser aufgelistet. Der Vorteil an den Service-Anbietern ist, dass es sehr leicht ist eine Testsuite auf einem weiteren Webbrowser auszuführen. Es ist zu eruieren wie lange die Tests brauchen um durchzulaufen. Dann kann entschieden werden, im Betracht der Kosten, auf wie vielen Browsern die Testsuite durchgeführt wird. Diese Entscheidung wird mit dem Business der Travelwindow AG getroffen, da sie den Kostenrahmen der Tests bestimmen.

## 4.3 Kosten & Zeit

Kosten & Zeit sind wichtig, da die Tests oft gestartet werden müssen und deshalb nicht zu lange dauern sollten. Die Service-Anbietern rechnen pro Minute ab, welche die Tests auf ihren Systemen laufen.

Für jeden Test wird eine neue Virtual Machine gestartet, was Zeit beansprucht. Deshalb wird versucht, die Anzahl der Testfälle zu minimieren, um trotzdem die gesamte Funktionalität abdecken zu können.

Je nachdem wie lange die Tests benötigen kann über die Anzahl der Browser, auf denen die Tests laufen gelassen werden sollen, die Kosten gesteuert werden.

# KAPITEL 5

---

## Konzept

---

Dieses Kapitel spezifiziert die umzusetzenden Testfälle. Diese beinhalten die Ausführung, die erwarteten Resultate sowie allfällige Nachbedingungen. Die Vorbedingungen sind für alle Testfälle gleich, da jeder in einer neuen virtuellen Maschine (siehe [Virtual Machine](#)) gestartet wird und somit keine [Sessions](#) und [HTTP-Cookies](#) vorhanden sind.

Um die Kosten zu minimieren werden möglichst wenig Testfälle spezifiziert, jedoch trotzdem die gesamte Funktionalität der Webseite überprüft werden kann. Weitere Details dazu sind im Kapitel [4 Anforderungen und Analyse](#) aufgeführt.

Eine Übersicht der Testfälle wurde im Wiki von Hotelplan erstellt und wird zum Verständnis dort gepflegt. Die Liste ist im Anhang [A Testübersicht](#) unter „Test Cases“ beigefügt.

### 5.1 Übersicht

Die Seite der *travelwindow AG* besteht aus drei Bereichen (auch Engines genannt):

- Citytrip (Flug & Hotel)
- Flug
- Hotel

Die Tests werden in drei Kategorien unterteilt:

- Happy Path
- Smoke Tests
- Main Tests

Wenn die Tests ausgeführt werden, sollen als erstes die Happy Paths<sup>1</sup> überprüft werden. Dabei wird der gesamte Such- und Buchungsprozess pro Engine einmal durchlaufen, ohne eine Auswahl zu treffen oder auf Fehler zu überprüfen. Der Test ist erfolgreich wenn

---

<sup>1</sup> *Happy path* - Wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Happy\\_path](https://en.wikipedia.org/wiki/Happy_path) (besucht am 30.07.2015).

die letzte Seite des Buchungsprozesses erreicht werden kann. Damit wird sichergestellt dass die Webseite im standarddurchlauf keine Fehler produziert.

Danach folgen die Smoke- und Main Tests. Die Smoke Tests führen eine Suche für die in der Tabelle [3.1 Top 10 Destinationen](#) definierten Destinationen aus und überprüfen, ob Resultate geliefert werden.

Die Main Tests überprüfen schlussendlich die Funktionalität (siehe Anhang [B Funktionalitäten](#)) der Webseite.

#### 5.1.1 Kunde & Passagierangaben

Im Buchungsprozess müssen die Kunden- & Passagierangaben ausgefüllt werden. Für alle Tests werden die folgenden Daten verwendet.

Kundendaten:

- Vorname: Han
- Name: Sola
- Strasse & Nummer: Bederstrasse 66
- Postleitzahl: 8002
- Ort: Zürich
- E-Mail Adresse: info@travel.ch
- Telefonnummer: +41 44 200 26 26

Passagierangaben (Geschlecht, Vorname, Name, Geburtsdatum):

- Passagier 1: Männlich, Han, Solo, 24.03.1980
- Passagier 2: Weiblich, Leia Organa, Solo, 12.02.1978
- Passagier 3: Männlich, Luke, Skywalker, 07.08.1985

## 5.2 Happy Path

Der Happy Path testet, ob beim standard Szenario keine Fehler auftreten. Es werden demnach jede Engine einmal durchlaufen mit gültigen Angaben. Dies soll überprüfen ob keine grundlegenden Fehler vorhanden sind.

## 5.2.1 Citytrip

**Tabelle 5.1:** Testspezifikation: Happy Path - Citytrip

Seite	Aktion
-	Gehe auf die Startseite
Startseite	Destinationsfeld: Leeren Destinationsfeld: Gebe „ber“ ein Destinationsfeld: Warte bis die Vorschläge erscheinen Destinationsfeld: Klicke auf den ersten Vorschlag Zimmerangabe: Klicke auf das Feld Zimmerangabe: Wähle „1 Erwachsenen“ aus Zimmerangabe: Wähle „1 Kind“ aus Zimmerangabe: Wähle für das Geburtsdatum jeweils das erste Element aus den Dropdowns aus. Zimmerangabe: Füge ein Zimmer hinzu Zimmerangabe: Wähle „1 Erwachsenen“ aus Datumsangabe: Klicke auf das Abreisedatum Datumsangabe: Gehe 3 Monate in die Zukunft Datumsangabe: Wähle das erste Datum an Datumsangabe: Klicke auf das Rückreisedatum Datumsangabe: Wähle das dritte Datum an Klicke auf den Suchen Button
Hotel Suchresultatseite	Warte bis die Suchresultate dargestellt werden Wähle das erste Hotel aus
Hotel Konfigurationsseite	Warte bis der „Weiter“ Button anwählbar ist Klicke auf den „Weiter“ Button
Flug Konfigurationsseite	Warte bis die Seite geladen ist Warte bis der „Weiter“ Button anwählbar ist Klicke auf den „Weiter“ Button
Checkout: Passagierangabe	Warte bis der Warenkorb geladen ist Fülle die Passagierangaben wie im Abschnitt <a href="#">5.1.1 Kunde &amp; Passagierangaben</a> beschrieben ab. Fülle die Kundendaten wie im Abschnitt <a href="#">5.1.1 Kunde &amp; Passagierangaben</a> beschrieben ab. Klicke auf den „Weiter“ Button
Checkout: Bezahlart	Warte bis die Seite geladen ist Warte bis der Warenkorb geladen ist Klicke auf den „Weiter“ Button
Checkout: Übersicht	Warte bis die Seite geladen ist Warte bis der Warenkorb geladen ist

## 5.2.2 Flight

**Tabelle 5.2:** Testspezifikation: Happy Path - Flight

Seite	Aktion
-	Gehe auf die Startseite
Startseite	<p>Abflugsort Destinationsfeld: Leeren</p> <p>Abflugsort Destinationsfeld: Gebe „zurich“ ein</p> <p>Abflugsort Destinationsfeld: Warte bis die Vorschläge erscheinen</p> <p>Abflugsort Destinationsfeld: Klicke auf den ersten Vorschlag</p> <p>Ankunftsort Destinationsfeld: Leeren</p> <p>Ankunftsort Destinationsfeld: Gebe „ber“ ein</p> <p>Ankunftsort Destinationsfeld: Warte bis die Vorschläge erscheinen</p> <p>Ankunftsort Destinationsfeld: Klicke auf den ersten Vorschlag</p> <p>Passagierangabe: Klicke auf das Feld</p> <p>Passagierangabe: Wähle „2 Erwachsenen“ aus</p> <p>Passagierangabe: Wähle „1 Kind“ aus</p> <p>Passagierangabe: Wähle für das Geburtsdatum jeweils das erste Element aus den Dropdowns aus.</p> <p>Datumsangabe: Klicke auf das Abreisedatum</p> <p>Datumsangabe: Gehe 3 Monate in die Zukunft</p> <p>Datumsangabe: Wähle das erste Datum an</p> <p>Datumsangabe: Klicke auf das Rückreisedatum</p> <p>Datumsangabe: Wähle das dritte Datum an</p> <p>Klicke auf den Suchen Button</p>
Flug Suchresultatseite	<p>Warte bis die Suchresultate dargestellt werden</p> <p>Wähle den ersten Flug aus</p>
Checkout: Passagierangabe	<p>Warte bis der Warenkorb geladen ist</p> <p>Fülle die Passagierangaben wie im Abschnitt <a href="#">5.1.1 Kunde &amp; Passagierangaben</a> beschrieben ab.</p> <p>Fülle die Kundendaten wie im Abschnitt <a href="#">5.1.1 Kunde &amp; Passagierangaben</a> beschrieben ab.</p> <p>Klicke auf den „Weiter“ Button</p>
Checkout: Bezahlart	<p>Warte bis die Seite geladen ist</p> <p>Warte bis der Warenkorb geladen ist</p> <p>Klicke auf den „Weiter“ Button</p>
Checkout: Übersicht	<p>Warte bis die Seite geladen ist</p> <p>Warte bis der Warenkorb geladen ist</p>

## 5.2.3 Hotel

**Tabelle 5.3:** Testspezifikation: Happy Path - Hotel

Seite	Aktion
-	Gehe auf die Startseite
Startseite	Destinationsfeld: Leeren Destinationsfeld: Gebe „ber“ ein Destinationsfeld: Warte bis die Vorschläge erscheinen Destinationsfeld: Klicke auf den ersten Vorschlag Zimmerangabe: Klicke auf das Feld Zimmerangabe: Wähle „1 Erwachsenen“ aus Zimmerangabe: Wähle „1 Kind“ aus Zimmerangabe: Wähle für das Geburtsdatum jeweils das erste Element aus den Dropdowns aus. Zimmerangabe: Füge ein Zimmer hinzu Zimmerangabe: Wähle „1 Erwachsenen“ aus Datumsangabe: Klicke auf das Abreisedatum Datumsangabe: Gehe 3 Monate in die Zukunft Datumsangabe: Wähle das erste Datum an Datumsangabe: Klicke auf das Rückreisedatum Datumsangabe: Wähle das dritte Datum an Klicke auf den Suchen Button
Hotel Suchresultatseite	Warte bis die Suchresultate dargestellt werden Wähle das erste Hotel aus
Hotel Konfigurationsseite	Warte bis der „Weiter“ Button anwählbar ist Klicke auf den „Weiter“ Button
Checkout: Passagierangabe	Warte bis der Warenkorb geladen ist Fülle die Passagierangaben wie im Abschnitt <a href="#">5.1.1 Kunde &amp; Passagierangaben</a> beschrieben ab. Fülle die Kundendaten wie im Abschnitt <a href="#">5.1.1 Kunde &amp; Passagierangaben</a> beschrieben ab. Klicke auf den „Weiter“ Button
Checkout: Bezahlart	Warte bis die Seite geladen ist Warte bis der Warenkorb geladen ist Klicke auf den „Weiter“ Button
Checkout: Übersicht	Warte bis die Seite geladen ist Warte bis der Warenkorb geladen ist

## 5.3 Smoke Tests

Smoke Tests führen eine Aktion durch und schauen lediglich, ob kein Fehler aufgetreten ist. Der Begriff rührt daher, dass eine Maschine gestartet und getestet wurde, ob sie nicht

beginnt zu rauchen<sup>1</sup>

Bei den folgenden Smoke Tests werden pro Engine die jeweiligen Top 10 (siehe Tabelle 3.1 [Top 10 Destinationen](#)) Destinationen gesucht und überprüft, ob Suchresultate geliefert werden.

Die Tests sind pro Engine immer gleich aufgebaut. Sprich die Suchparameter sind dieselben und auch die Verifikation ob kein Fehler aufgetreten ist äquivalent. Das einzige was sich unterscheidet sind die Destinationen. Daher wird jeweils nur der Test für die erste Destination ausführlich spezifiziert. Für die restlichen 9 gilt der selbe Ablauf.

### 5.3.1 Citytrip

**Tabelle 5.4:** Testspezifikation: Smoke Test - Citytrip

Seite	Aktion
-	Gehe auf die Startseite
Startseite	Destinationsfeld: Leeren Destinationsfeld: Gebe „Wien“ ein Destinationsfeld: Warte bis die Vorschläge erscheinen Destinationsfeld: Klicke auf den ersten Vorschlag Zimmerangabe: Klicke auf das Feld Zimmerangabe: Wähle „2 Erwachsenen“ aus Datumsangabe: Klicke auf das Abreisedatum Datumsangabe: Gehe 3 Monate in die Zukunft Datumsangabe: Wähle das erste Datum an Datumsangabe: Klicke auf das Rückreisedatum Datumsangabe: Wähle das dritte Datum an Klicke auf den Suchen Button
Hotel Suchresultatseite	Warte bis die Suchresultate dargestellt werden <b>Verifikation:</b> Es muss mind. ein Hotel vorhanden sein.

<sup>1</sup> *Smoke testing (software)* - Wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Smoke\\_testing\\_\(software\)](https://en.wikipedia.org/wiki/Smoke_testing_(software)) (besucht am 01.08.2015).



## 5.3.2 Flight

**Tabelle 5.5:** Testspezifikation: Smoke Test - Flight

Seite	Aktion
-	Gehe auf die Startseite
Startseite	<p>Abflugsort Destinationsfeld: Leeren  Abflugsort Destinationsfeld: Gebe „Zurich“ ein  Abflugsort Destinationsfeld: Warte bis die Vorschläge erscheinen  Abflugsort Destinationsfeld: Klicke auf den ersten Vorschlag  Ankunftsort Destinationsfeld: Leeren  Ankunftsort Destinationsfeld: Gebe „Palma de Mallorca“ ein  Ankunftsort Destinationsfeld: Warte bis die Vorschläge erscheinen  Ankunftsort Destinationsfeld: Klicke auf den ersten Vorschlag</p> <p>Zimmerangabe: Klicke auf das Feld  Zimmerangabe: Wähle „2 Erwachsenen“ aus  Datumsangabe: Klicke auf das Abreisedatum  Datumsangabe: Gehe 3 Monate in die Zukunft  Datumsangabe: Wähle das erste Datum an  Datumsangabe: Klicke auf das Rückreisedatum  Datumsangabe: Wähle das dritte Datum an  Klicke auf den Suchen Button</p>
Flug Suchresultatseite	<p>Warte bis die Suchresultate dargestellt werden  <b>Verifikation:</b> Es muss mind. ein Flug vorhanden sein.</p>

## 5.3.3 Hotel

**Tabelle 5.6:** Testspezifikation: Smoke Test - Hotel

Seite	Aktion
-	Gehe auf die Startseite
Startseite	Destinationsfeld: Leeren Destinationsfeld: Gebe „ber“ ein Destinationsfeld: Warte bis die Vorschläge erscheinen Destinationsfeld: Klicke auf den ersten Vorschlag Zimmerangabe: Klicke auf das Feld Zimmerangabe: Wähle „1 Erwachsenen“ aus Zimmerangabe: Wähle „1 Kind“ aus Zimmerangabe: Wähle für das Geburtsdatum jeweils das erste Element aus den Dropdowns aus. Zimmerangabe: Füge ein Zimmer hinzu Zimmerangabe: Wähle „1 Erwachsenen“ aus Datumsangabe: Klicke auf das Abreisedatum Datumsangabe: Gehe 3 Monate in die Zukunft Datumsangabe: Wähle das erste Datum an Datumsangabe: Klicke auf das Rückreisedatum Datumsangabe: Wähle das dritte Datum an Klicke auf den Suchen Button
Hotel Suchresultatseite	Warte bis die Suchresultate dargestellt werden <b>Verifikation:</b> Es muss mind. ein Hotel vorhanden sein.

## 5.4 Main Tests

Bei den Happy Paths wird überprüft, ob die Funktionalität der Webseite gemäss Spezifikation (siehe Anhang [B Funktionalitäten](#)) funktioniert. Die Funktionalitäten im Anhang sind priorisiert und werden gemäss dieser umgesetzt. Begonnen wurde mit der höchsten Zahl.

## 5.4.1 Citytrip

**Tabelle 5.7:** Testspezifikation: Main Tests - Citytrip

Seite	Aktion
-	Gehe auf die Startseite
Startseite	Destinationsfeld: Leeren Destinationsfeld: Gebe „Amsterdam“ ein Destinationsfeld: Warte bis die Vorschläge erscheinen Destinationsfeld: Klicke auf den ersten Vorschlag Zimmerangabe: Klicke auf das Feld Zimmerangabe: Wähle „1 Erwachsenen“ aus Zimmerangabe: Wähle „1 Kind“ aus Zimmerangabe: Wähle für das Geburtsdatum jeweils das erste Element aus den Dropdowns aus. Zimmerangabe: Füge ein Zimmer hinzu Zimmerangabe: Wähle „1 Erwachsenen“ aus Datumsangabe: Klicke auf das Abreisedatum Datumsangabe: Gehe 3 Monate in die Zukunft Datumsangabe: Wähle das erste Datum an Datumsangabe: Klicke auf das Rückreisedatum Datumsangabe: Wähle das dritte Datum an Klicke auf den Suchen Button
Hotel Suchresultatseite	Warte bis die Suchresultate dargestellt werden Wähle das erste Hotel aus
Hotel Konfigurationsseite	Warte bis der „Weiter“ Button anwählbar ist Klicke auf den „Weiter“ Button
Flug Konfigurationsseite	Warte bis die Seite geladen ist Warte bis der „Weiter“ Button anwählbar ist Suche ein Flug welcher mehrere Hinflüge hat und wähle den zweiten aus Überprüfe ob sich in der Übersicht die Airline, das Datum und der Preis korrekt angepasst haben Suche ein Flug welcher mehrere Rückflüge hat und wähle den zweiten aus Überprüfe ob sich in der Übersicht die Airline, das Datum und der Preis korrekt angepasst haben



## 5.4.2 Flight

**Tabelle 5.8:** Testspezifikation: Main Tests - Flight

Seite	Aktion
-	Gehe auf die Startseite
Startseite	<p>Abflugsort Destinationsfeld: Leeren</p> <p>Abflugsort Destinationsfeld: Gebe „new york“ ein</p> <p>Abflugsort Destinationsfeld: Warte bis die Vorschläge erscheinen</p> <p>Abflugsort Destinationsfeld: Klicke auf den ersten Vorschlag</p> <p>Ankunftsart Destinationsfeld: Leeren</p> <p>Ankunftsart Destinationsfeld: Gebe „Zurich“ ein</p> <p>Ankunftsart Destinationsfeld: Warte bis die Vorschläge erscheinen</p> <p>Ankunftsart Destinationsfeld: Klicke auf den ersten Vorschlag</p> <p>Passagierangabe: Klicke auf das Feld</p> <p>Passagierangabe: Wähle „2 Erwachsenen“ aus</p> <p>Passagierangabe: Wähle „1 Kind“ aus</p> <p>Passagierangabe: Wähle für das Geburtsdatum jeweils das erste Element aus den Dropdowns aus.</p> <p>Datumsangabe: Klicke auf das Abreisedatum</p> <p>Datumsangabe: Gehe 3 Monate in die Zukunft</p> <p>Datumsangabe: Wähle das erste Datum an</p> <p>Datumsangabe: Klicke auf das Rückreisedatum</p> <p>Datumsangabe: Wähle das dritte Datum an</p> <p>Klicke auf den Suchen Button</p>
Flug Suchresultatseite	<p>Überprüfe ob beim Anpassen des Hinfluges der korrekte Flug angewählt wird:</p> <ul style="list-style-type: none"> <li>- Suche eine Fluggruppe mit mehreren Hinflügen</li> <li>- Speichere den ersten Flug der Fluggruppe</li> <li>- Wähle den zweiten Flug der Fluggruppe</li> <li>- Überprüfe ob der zweite Flug korrekt aktualisiert wurde</li> </ul> <p>Überprüfe ob beim Anpassen des Hinfluges der korrekte Flug angewählt wird:</p> <ul style="list-style-type: none"> <li>- Suche eine Fluggruppe mit mehreren Rückflügen</li> <li>- Speichere den ersten Flug der Fluggruppe</li> <li>- Wähle den zweiten Flug der Fluggruppe</li> <li>- Überprüfe ob der zweite Flug korrekt aktualisiert wurde</li> </ul> <p>Überprüfe ob bei allen Flügen der Totalpreis angezeigt wird</p> <p>Überprüfe ob das Datum des Fluges in der Übersicht dargestellt wird.</p> <p>Überprüfe für alle Flüge:</p> <ul style="list-style-type: none"> <li>- Flug ist anwählbar</li> <li>- Airline Bild des Hinfluges wird angezeigt</li> <li>- Airline Bild des Rückfluges wird angezeigt</li> </ul> <p>Wähle ein Rückflug mit mehreren Hin- und Rückflügen</p>

# KAPITEL 6

---

## Umsetzung

---

Dieses Kapitel beschreibt die technischen Aspekte, wie die Testfälle aus dem Kapitel 5 [Konzept](#) umgesetzt wurden.

Zuerst wird der Einsatz von Selenium sowie die Infrastruktur der Travelwindow AG beschrieben, um einen Überblick zu verschaffen. Danach folgen die Umsetzungsdetails und schlussendlich wie die Tests in den Entwicklungsprozess eingebunden werden.

### 6.1 Selenium

Die Testfälle werden mit Selenium<sup>1</sup> implementiert. Dieses automatisiert Browser, sprich es wird als erstes einer aufgestartet, danach Testschritte ausgeführt und schlussendlich einige Behauptungen aufgestellt. Sind diese korrekt so ist der Test erfolgreich. Wenn nicht schlägt der Test fehl. Angetrieben werden Selenium Tests mittels UnitTests. Diese können von Testtreibern ausgeführt werden und erlauben die Überprüfung von Behauptungen.t

Selenium selber ist in Java implementiert und bietet eine eigene [IDE](#) mit dem Namen Selenium IDE<sup>2</sup>, welche ein Plugin für den Firefox<sup>3</sup> darstellt. Damit können Aktionen im Browser aufgezeichnet werden und danach als Test automatisiert durchgeführt werden. Die gesamte Software ist Open Source unter der *Apache License 2.0* erhältlich<sup>4</sup>.

Die gesamte Webseitenentwicklung der Firma Travelwindow AG findet in der Programmiersprache C# statt. Aus Gründen der Konsistenz wurde entschieden, dass auch die Tests in C# umgesetzt werden sollen. Selenium WebDriver<sup>5</sup> ist ein Aufsatz auf Selenium, welcher eine API bietet damit mit verschiedenen Programmiersprachen Tests umgesetzt werden können. Vollständig unterstützt werden die Sprachen Python, Ruby, Java und C#.

---

<sup>1</sup> *Selenium - Web Browser Automation*. URL: <http://docs.seleniumhq.org/> (besucht am 26.09.2015).

<sup>2</sup> *Selenium IDE Plugins*. URL: <http://docs.seleniumhq.org/projects/ide/> (besucht am 26.09.2015).

<sup>3</sup> *Download Firefox — Free Web Browser — Mozilla*. URL: <https://www.mozilla.org/en-US/firefox/new/> (besucht am 26.09.2015).

<sup>4</sup> [\\_the\\_free\\_encyclopedia\\_2015-09-26;](#)

<sup>5</sup> [Selenium\\_WebDriver\\_2015-09-26.](#)

### 6.1.1 Beispiele

Hier sollen einige Beispiele dafür aufgezeigt werden, wie über Selenium mit dem Browser interagiert werden kann.

Um einen Browser zu starten muss ein entsprechender Driver instantiiert werden. Im folgenden Code wird ein Firefox aufgerufen.

```
1  IWebDriver driver = new FirefoxDriver();
```

---

Um auf eine Webseite zu navigieren muss dies dem Driver übermittelt werden:

```
1  driver.Navigate().GoToUrl(targetUrl);
```

---

Ist eine Seite im Browser geladen, kann über den [Document Object Model \(DOM\)](#)<sup>1</sup> Tree auf die HTML Elemente zugegriffen werden. Dazu werden Selektoren verwendet, welche entweder als *CSS Selector* oder als *XPath* definiert werden können. Alternativ kann auch JavaScript ausgeführt werden um ein Element zu finden.

```
1  ///// CSS selectors /////
2  // Get elements by class
3  IList<IWebElement> flightListElements = driver.FindElements(By.CssSelector(".
flight"));
4  // Get element by id
5  IWebElement submitButton = driver.FindElements(By.CssSelector("#submit"));
6  // Advanced usage
7  IWebElement cityTripButton = driver.FindElements(By.CssSelector(".toggle-
buttons button[value='Flight']"));
8
9  ///// XPath selectors /////
10 // Get all input elements
11 IList<IWebElement> inputs = driver.FindElements(By.XPath("//input"));
12 // Get the first image of each flight
13 IList<IWebElement> inputs = driver.FindElements(By.XPath("//child::flight/image
[1]"));
14
15 ///// JavaScript /////
16 IWebElement element = (IWebElement) ((IJavaScriptExecutor)driver).ExecuteScript
("return $('flight')[0]");
```

---

Mit den DOM Elementen kann auf verschiedene Arten interagiert werden.

```
1  // Klick auf einen Button
2  IWebElement button = GetButton();
3  button.Click();
4
5  // Text in ein Element schreiben
6  IWebElement inputField = GetInputField();
```

---

<sup>1</sup> *Document Object Model* - Wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Document\\_Object\\_Model](https://en.wikipedia.org/wiki/Document_Object_Model) (besucht am 04.10.2015).

```

7  inputField.SendKeys("`New York`");
8
9  // Eingabefeld leeren
10 inputField.Clear();
11
12 // Eingabefeld an den Webserver senden
13 inputField.Submit();

```

Falls nötig kann JavaScript Code ausgeführt werden.

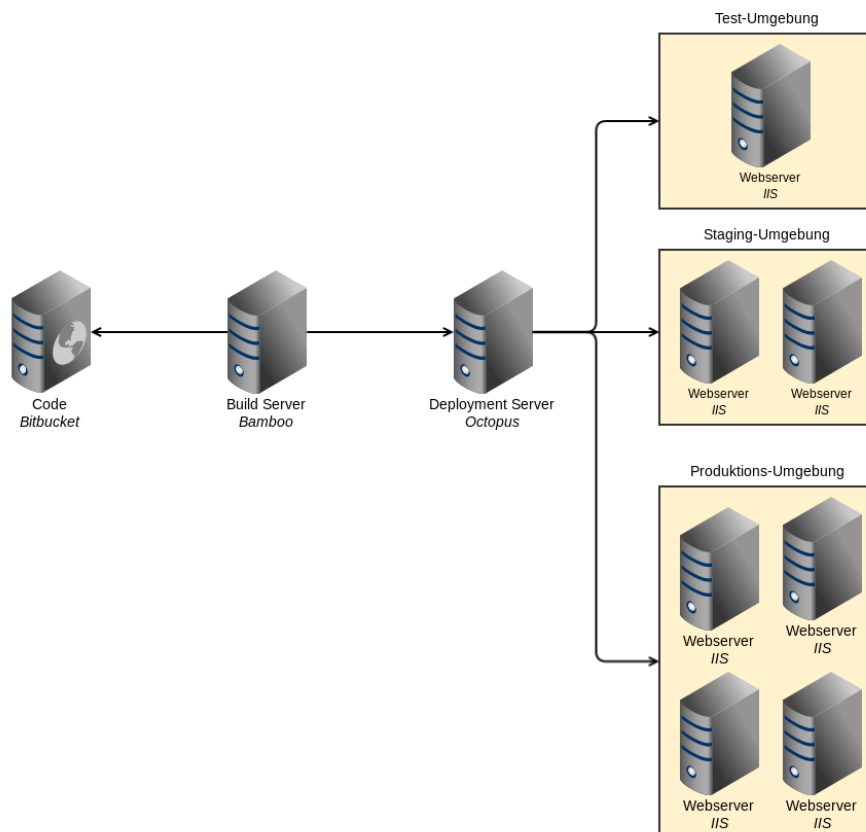
```

1  var jsExecutor = ((IJavaScriptExecutor)driver);
2  jsExecutor.ExecuteScript("var a=$('#chat');if (a!=null)a.parentNode.removeChild(a)");

```

## 6.2 Infrastruktur

Dieser Abschnitt beschreibt die Infrastruktur der Travelwindow AG.



**Abbildung 6.1:** Travelwindow AG Infrastruktur

Als [Versionierungssystem](#) wird BitBucket<sup>1</sup> von Atlassian eingesetzt. Der Code wird bei

<sup>1</sup> *Git and Mercurial code management for teams.*



jeder Änderung vom Build Server heruntergeladen und Compiliert, wodurch überprüft wird ob keine syntaktischen Fehler vorhanden sind. Dabei wird jeweils eine Revision angelegt. Auf dem Build Server läuft Bamboo<sup>1</sup>, welches ebenfalls von Atlassian hergestellt wird.

Es gibt folgende drei Umgebungen:

- Test
- Quality
- Production

Die erste ist für die Entwickler. Die zweite für den Kunden zum Testen und die letzte ist jene Umgebung, welche von den Kunden verwendet wird.

Soll eine Version der Webseite auf eine Umgebung hochgeladen werden, so kann der Prozess auf dem Build Server gestartet werden, wodurch eine Revision vom Bamboo Server auf die entsprechende Umgebung hochgeladen wird. Als Deployment Server wird Octopus<sup>2</sup> eingesetzt.

### 6.3 Design Patterns

Es werden die beiden Design Patterns *Page Object* und *Page Factory* vorgestellt. Die Verwendung soll anhand eines Beispiels erläutert werden. Dazu soll folgender Test optimiert werden:

```
1 public class CityTrip
2 {
3     [Test]
4     public IWebElement SearchReturnsResults()
5     {
6         // Setup driver
7         var driver = new FirefoxDriver();
8         driver.Navigate().GoToUrl("http://www.travel.ch/staedtereisen");
9         var wait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
10
11         // Fill in form
12         driver.FindElement(By.CssSelector(".form .destination")).SendKeys("Wien");
13         driver.FindElement(By.CssSelector(".form .submit")).Click();
14
15         // Wait for result page to be ready
16         wait.Until(d => d.FindElement(By.ClassName("js-results-item")));
17
18         // Check for search results
19         var results = driver.FindElement(By.CssSelector(".list .items"));
20         Assert.IsTrue(results.Any(), "`no search results`");
21     }
22 }
```

---

<sup>1</sup> *Continuous Integration & Build Server - Bamboo* / Atlassian.

<sup>2</sup> *Automated deployment for .NET - Octopus Deploy*. URL: <https://octopus.com/> (besucht am 28.09.2015).

Der Test gibt eine Destination in ein Suchformular ein, sendet das Formular ab und überprüft ob, ob mind. ein Resultat gefunden wurde.

### 6.3.1 Page Object

Das Page Object Design Pattern gibt vor, dass Webseiten in Objekten im Code repräsentiert werden sollen<sup>1</sup>. Diese Objekte werden Page Objects genannt. Sie kapseln die Logik, um mit einer Webseite zu interagieren, weg. Zum Beispiel das selektieren eines **DOM** Elements oder das drücken eines Buttons.

Das Design Pattern auf das in der Kapiteleinführung gegebene Beispiel angewendet ergibt folgender Code:

```
1 public class CityTrip
2 {
3     [Test]
4     public IWebElement SearchReturnsResults()
5     {
6         var driver = new FirefoxDriver();
7         driver.Navigate().GoToUrl("http://www.travel.ch/staedtereisen");
8         var wait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
9
10        var searchPage = new SearchPage(driver);
11        var resultsPage = searchPage.SubmitValidForm();
12
13        Assert.IsTrue(resultsPage.GetResults().Any(), "`no search results`");
14    }
15 }
16
17 public class StartPage
18 {
19     private readonly IWebDriver driver;
20
21     public StartPage(IWebDriver drivert)
22     {
23         this.driver = driver;
24     }
25
26     public SearchResultPage SubmitValidForm()
27     {
28         driver.FindElement(By.CssSelector(".form .destination")).SendKeys("Wien");
29         driver.FindElement(By.CssSelector(".form .submit")).Click();
30
31         return new SearchResultPage(driver);
32     }
33 }
34
```

---

1 *PageObjects and PageFactory – Selenium Design Patterns (C#, Java)*. URL: <http://relevantcodes.com/pageobjects-and-pagefactory-design-patterns-in-selenium/> (besucht am 05.10.2015).

```

35 public class SearchResultPage
36 {
37     private readonly IWebDriver driver;
38     private readonly WebDriverWait driverWait;
39
40     public StartPage(IWebDriver drivert)
41     {
42         this.driver = driver;
43         this.driverWait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
44         WaitForSearchResults();
45     }
46
47     private void WaitForSearchResults() {
48         driverWait.Until(d => d.FindElement(By.ClassName("js-results-item")));
49     }
50
51     public IList<IWebElement> GetResults()
52     {
53         return driver.FindElement(By.CssSelector(".list .items"))
54     }
55 }

```

Die Testmethode *SearchReturnsResults* (Zeile 4) ist aufgeräumt und übersichtlich. Es wird eine *SearchPage* instanziiert und auf der das Suchformular abgefüllt. Schlussendlich wird auf gefundene Resultate überprüft.

Dazu gekommen sind die beiden Page Objects *StartPage* (Zeile 17) und *SearchResultPage* (Zeile 35). Sie beinhalten die jeweiligen CSS Selektoren für die entsprechende Seite.

Für die Navigation von einer Seite auf die nächste sind ebenfalls die Page Objects zuständig, da sie wissen auf welche Seite ein Link oder ein Formular führt. Die Methode *SubmitValidForm* (Zeile 26) gibt ein weiteres Page Object zurück.

### 6.3.2 Page Factory

Eine *Page Factory* erleichtert den Umgang mit einem *Page Object*. Es initialisiert die Elemente eines Page Objects. Der Code kann auf folgende Weise umgebaut werden:

```

1 public class CityTrip
2 {
3     [Test]
4     public IWebElement SearchReturnsResults()
5     {
6         var driver = new FirefoxDriver();
7         driver.Navigate().GoToUrl("http://www.travel.ch/staedtereisen");
8         var wait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
9
10        var searchPage = new SearchPage(driver);
11        var resultsPage = searchPage.SubmitValidForm();
12
13        Assert.IsTrue(resultsPage.Results.Any(), "`no search results`");
14    }

```

```
15 }
16
17 public class StartPage
18 {
19     private readonly IWebDriver driver;
20
21     [FindsBy(How = How.CssSelector, Using = ".form .destination")]
22     private IWebElement destination;
23
24     [FindsBy(How = How.CssSelector, Using = ".form .submit")]
25     private IWebElement submitButton;
26
27     public StartPage(IWebDriver drivert)
28     {
29         this.driver = driver;
30         PageFactory.InitElements(driver, this);
31     }
32
33     public SearchResultPage SubmitValidForm()
34     {
35         destination.SendKeys("Wien");
36         submitButton.Click();
37
38         return new SearchResultPage(driver);
39     }
40 }
41
42 public class SearchResultPage
43 {
44     private readonly IWebDriver driver;
45     private readonly WebDriverWait driverWait;
46
47     [FindsBy(How = How.CssSelector, Using = ".list .items")]
48     public IList<IWebElement> Results;
49
50     public StartPage(IWebDriver drivert)
51     {
52         this.driver = driver;
53         this.driverWait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
54         WaitForSearchResults();
55         PageFactory.InitElements(driver, this);
56     }
57
58     private void WaitForSearchResults() {
59         driverWait.Until(d => d.FindElement(By.ClassName("js-results-item")));
60     }
61 }
```

---

Als erstes wurde auf den Zeilen 21-25 sowie 47-48 Felder mit einer Annotation eingefügt. Diese Teile der PageFactory mit, wie die Elemente gefunden werden sollen.

Im zweiten Schritt wird auf den Zeilen 30 und 55 die PageFactory aufgerufen, welche die für die Felder entsprechenden Elemente im [DOM](#) sucht.

Auf den Zeilen 13, 35 und 36 werden schlussendlich die Felder verwendet.

Durch die Page Factory wird die Lesbarkeit des Codes verbessert sowie dessen wartbarkeit, da die Elementdefinitionen einheitlich zu beginn der Klasse aufgeführt werden.

## 6.4 Architektur

Das Kapitel widmet sich der Architektur. Das Programm ist in einzelne Bereiche, so genannte Namespaces aufgliedert. Diese werden für sich beschrieben, sowie die Interaktion zwischen ihnen.

Folgend nun die Übersicht über die Namespaces in Form eines Klassendiagramms:

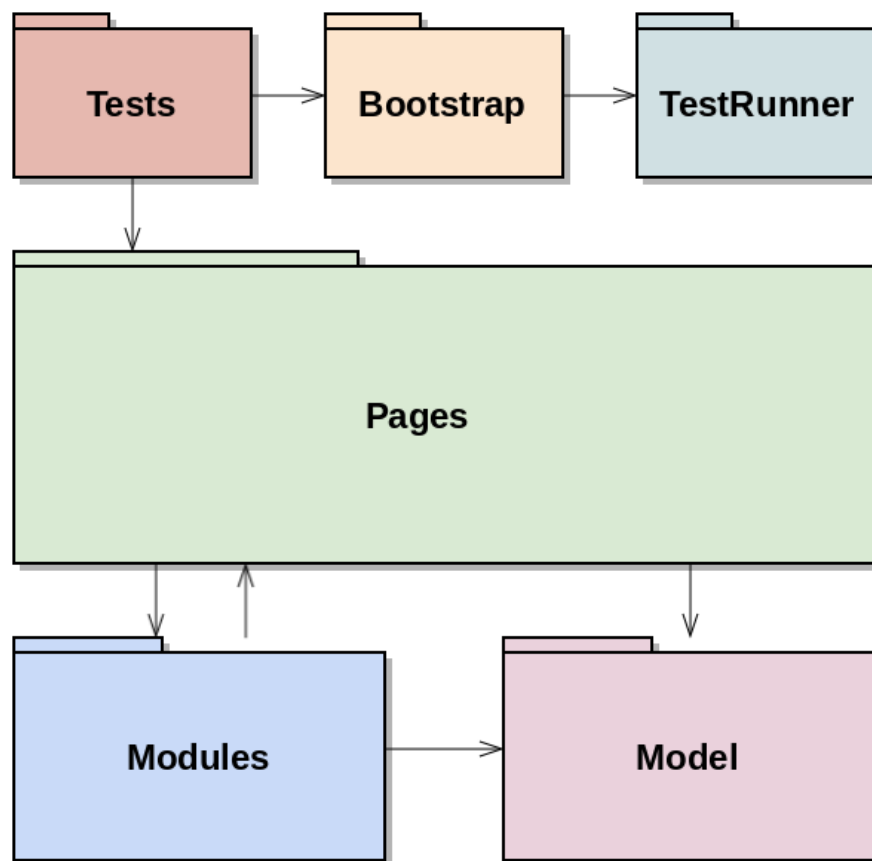


Abbildung 6.2: Klassendiagramm Übersicht

## 6.4.1 Tests

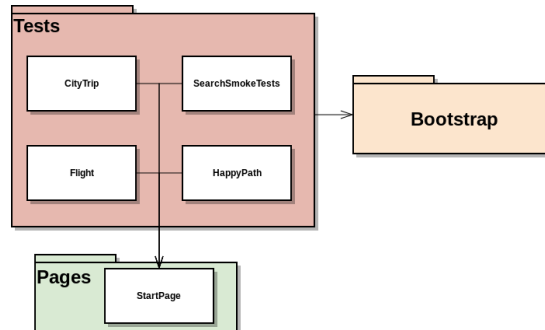


Abbildung 6.3: Klassendiagramm Tests

Im Ordner Tests sind die Tests gespeichert. Zur Erstellung des Drivers wird die Klasse *DriverFactory* aus dem Namespace *Bootstrap* benötigt (siehe Abschnitt 6.1 Selenium). Für die Interaktion mit dem Browser werden die *Pages* verwendet (siehe Abschnitt 6.3.1 Page Object).

Die HappyPath Tests sind in der Datei HappyPath.cs im spezifiziert, die SearchSmoke-Tests im File SearchSmokeTests.cs und für die Main Tests gibt es für die jeweilige Engine eine eigene Datei (Citytrip.cs und Flight.cs).

Wie im Abschnitt 6.1 Selenium beschrieben werden die Tests mit C# umgesetzt und mittels des UnitTest Frameworks NUnit<sup>1</sup> angetrieben.

```

1 [SetUp]
2 public void SetUp()
3 {
4     driverFactory = new DriverFactory();
5     targetUrl = driverFactory.TargetUrl;
6     driver = driverFactory.GetDriver();
7     wait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
8 }
9
10 [TearDown]
11 public void TearDown()
12 {
13     driverFactory.FailTestIfNotSucceeded();
14     driver.Quit();
15 }
  
```

Die Methoden SetUp und TearDown werden zu Beginn respektive zum Ende jedes Tests ausgeführt. Die SetUp Methode lädt den Treiber mittels der DriverFactory und erstellt die Klasse *WebDriverWait* (siehe Abschnitt 6.5.1 Timing). Die TearDown Methode sendet den Status über Erfolg oder Misserfolg an den Service-Anbieter und schliesst den Treiber wenn

<sup>1</sup> NUnit - Home. URL: <http://www.nunit.org/> (besucht am 26.09.2015).

der Test abgeschlossen ist.

Eine Testmethode wird mit dem *Test* Attribut gekennzeichnet. Diese werden von NUnit erkannt und ausgeführt.

```

1 [Test]
2 public void Citytrip_ToAmsterdam_FromZurich_Room1_1Adults1Child_Room2_1Adult()
3 {
4     // Setup
5     driver.Navigate().GoToUrl(targetUrl);
6     var startPage = new StartPage(driver, wait);
7
8     // Tests & Assertions
9     ...
10
11    // Teardown
12    driverFactory.TestSucceeded();
13 }

```

In jedem Test wird zuerst auf die Startseite Navigiert (Zeile 5) und die *StartPage* instanziiert (Zeile 6).

Danach folgen die Testinstruktionen und die Behauptungen.

Wenn der Test durchläuft, sprich es gab keine Behauptung welche fehlgeschlagen ist, so wird zum Schluss noch *driverFactory.TestSucceeded()*; aufgerufen. Dies signalisiert das der Test erfolgreich war was dem Service-Anbieter übermittelt werden kann (weitere Informationen dazu im Abschnitt 6.4.3 *TestRunner*).

#### 6.4.2 Bootstrap

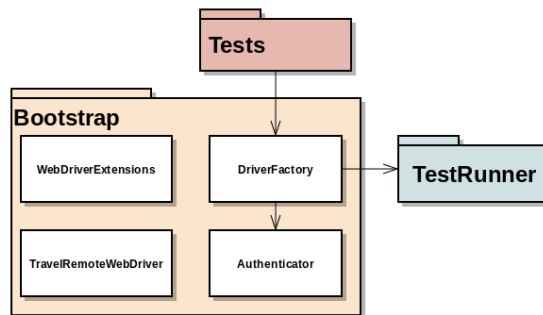


Abbildung 6.4: Klassendiagramm Bootstrap

Das Bootstrapping ist dazu da, die Grundlage für den Test bereitzustellen. Dazu gehört die Erstellung des Drivers, die Authentifizierung an der Webseite. Diese Punkte können über eine Konfiguration angepasst werden.

#### DriverFactory

Die *DriverFactory* ist der Einstiegspunkt in die ganze initialisierung (Bootstrapping) eines Tests und zuständig für die Erstellung eines Treibers. Dazu wird ein *TestRunner* benötigt (siehe Abschnitt 6.4.3 *TestRunner*). Welcher erstellt werden soll wird über die Konfiguration

(siehe Abschnitt [6.4.2 Konfiguration](#)) gesteuert. Zusätzlich initialisiert die DriverFactory auch die Authentifizierung, welche durch den *Authenticator* durchgeführt wird.

### Konfiguration

Es gibt zwei Orte wo die Ausführung konfiguriert werden kann. Dies ist zum einen in den *app.config*-Dateien, sowie über die Kommandozeile.

Die *app.config*-Dateien sind im obersten Ordner des Projektes abgelegt. Für jede Umgebung (siehe Abschnitt [6.2 Infrastruktur](#)) gibt es ein dediziertes File, welche folgenden Aufbau haben:

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
3   <testConfigurations xdt:Transform="Replace">
4     <testConfiguration>
5       <testRunner>crossbrowsertesting</testRunner>
6       <targetUrl>http://travel.local.ch</targetUrl>
7       <authenticationType>netScaler</authenticationType>
8     </testConfiguration>
9   </testConfigurations>
10 </configuration>
```

Die der *app.config* kann der Service-Anbieter, die Webseite welche getestet werden soll, sowie den Typ der Authentifizierung festgelegt werden.

Der Browser, die Browser-Version und die Plattform sind für alle Tests die selben und werden über die Kommandozeile-Parameter definiert. Wenn die Test-Suite mit mehreren Konfigurationen getestet werden soll, so muss das Test-Projekt mehrfach mit verschiedenen Parametern ausgeführt werden. Dies wird entsprechend im Bamboo (siehe Abschnitt [6.2 Infrastruktur](#)) definiert.

Die Kommandozeile-Parameter haben folgende Form:

browser=Chrome version=38 platform=Windows8

Die Werte der Parameter müssen mit den Werten in dem Ordner */Bootstrap/Devices* übereinstimmen. Eine Ausnahme ist die Version. In dem genannten Ordner haben diese einen Unterstrich vorangesetzt (zum Beispiel *\_7*, *\_11*, etc.), da die Werte in C# nicht mit einer Zahl beginnen dürfen.

### Authentifizierung

Es gibt drei verschiedene Arten der Authentifizierung:

- Keine
- BasicAuthentication
- netScaler

Keine Authentifizierung wird für die lokale Testausführung verwendet. Sprich wenn der Test auf dem eigenen Rechner ausgeführt werden soll.



BasicAuthentication ist die Authentifizierungsmethode des IIS Servers. Kommt man auf eine entsprechend Konfigurierte Webseite wird man vom Browser zur Eingabe eines Benutzernamens und Passwort aufgefordert. Alternativ können die beiden Werte auch über die Uniform Resource Locator (URL) mitgegeben werden in folgender Form:

scheme://user:passwort@host

Die DriverFactory (siehe Abschnitt 6.4.2 DriverFactory) stellt sicher, dass die Authentifizierung für HTTP sowie für HTTPS durchgeführt wird.

Der netScaler<sup>1</sup> ist ein Application Delivery Controller der Firma Citrix und wird in Hotelplan Gruppe verwendet um unter anderem die Authentifizierung sicherzustellen. Kommt man auf eine vom netScaler geschützten Webseite erscheint eine HTML-Seite welche die Eingabe eines Passwort verlangt. Die DriverFactory gibt das Passwort ein und leitet den User auf die Startseite weiter. Danach kann mit Testausführung begonnen werden.

### 6.4.3 TestRunner

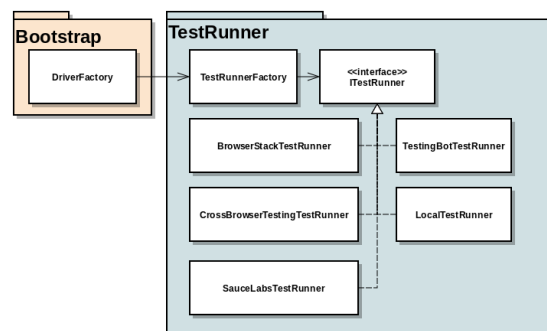


Abbildung 6.5: Klassendiagramm TestRunner

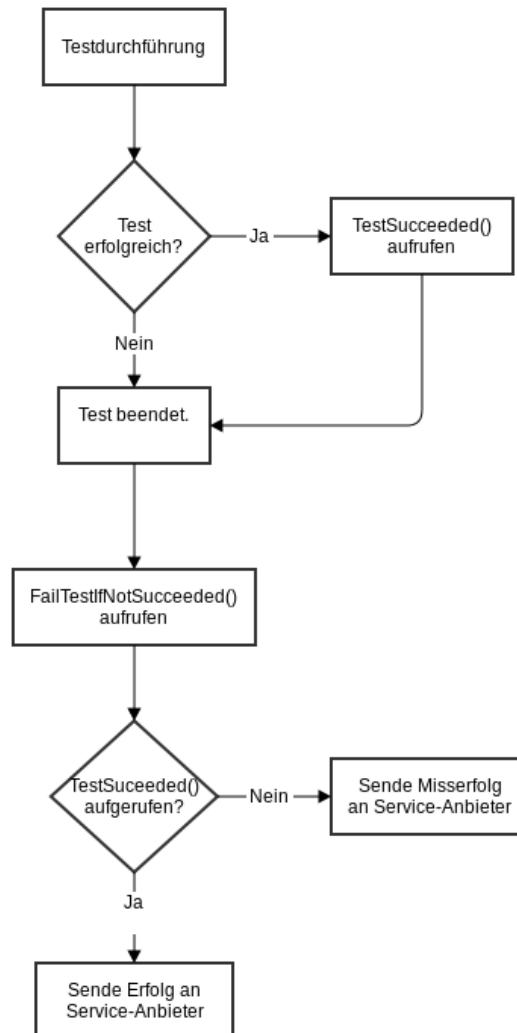
TestRunners treiben die Tests an und sind die Anbindung an die Service-Anbieter (siehe Abschnitt 3.2 Testing Frameworks). Sie sind dafür zuständig, einen passenden und konfigurierten Treiber zu erstellen sowie eine Antwort über Erfolg oder Misserfolg an den Service-Anbieter zu senden.

Für die Konfiguration eines Treibers können über den Konstruktor ein Browser, eine Browser-Version sowie die Plattform/Betriebssystem übergeben werden. Die möglichen Werte sind im Ordner /Bootstrap/Devices in Objekte gekapselt. Da jedoch jeder Anbieter eigene Werte definiert, müssen dies internen auf jene des Anbieters innerhalb des TestRunners gemapped werden.

Um einen Status über den Erfolg oder Misserfolg an den Service-Anbieter zu senden, sind im Interface ITestRunner die zwei Methoden *TestSucceeded* und *FailTestIfNotSucceeded* definiert. Sobald ein Test erfolgreich war wird die erste Methode aufgerufen. Die zweite wird

<sup>1</sup> NetScaler Application Delivery Controller - Application Security and Cloud Scalability - Citrix. URL: <https://www.citrix.com/products/netscaler-application-delivery-controller/overview.html> (besucht am 05. 10. 2015).

zum Schluss jedes Tests ausgeführt und liefert einen Test-Erfolg an den Service anbieter, wenn TestSucceeded einmal ausgeführt wurde, sonst wird ein Test-Misserfolg gesendet. Dieser Ablauf wird im folgenden Flow Chart verbildlicht:



**Abbildung 6.6:** Flow Chart Service-Anbieter Anbindung

Ein spezieller TestRunner stellt der LocalTestRunner. Dieser führt den Test nicht bei einem Service-Anbieter durch, sondern auf der lokalen Maschine. Demnach muss der Browser, der Browser-Treiber und das Projekt auf dem eigenen Computer lauffähig sein. Dieser wird hauptsächlich für die Entwicklung verwendet da die Tests schneller abgearbeitet werden können. Die beiden Methoden TestSucceeded und FailTestIfNotSucceeded machen bei diesem TestTreiber nichts.

## 6.4.4 Pages

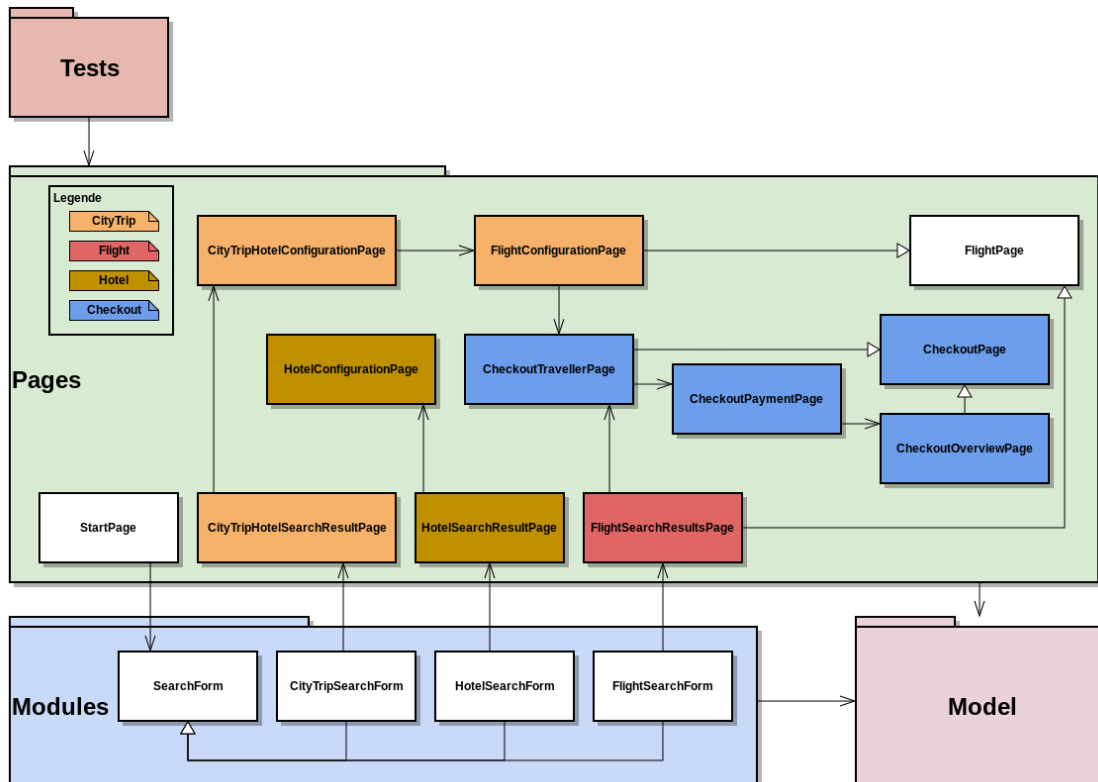


Abbildung 6.7: Klassendiagramm Pages

Die funktionsweise der Page Objekte wurde im Abschnitt [6.3.1 Page Object](#) bereits erläutert.

Für die Tests ist der Einstieg immer das Objekt *StartPage*. Von der Startseite kann ein *SearchForm* angefordert werden, welches in Form einer konkreten Implementation zurückgegeben wird. Das *SearchForm* liefert dann eine Suchresultatseite. So angelt man sich weiter vor bis zur letzten Seite.

Das folgende Sequenzdiagramm zeigt den Ablauf am Beispiel der CityTrip Engine:

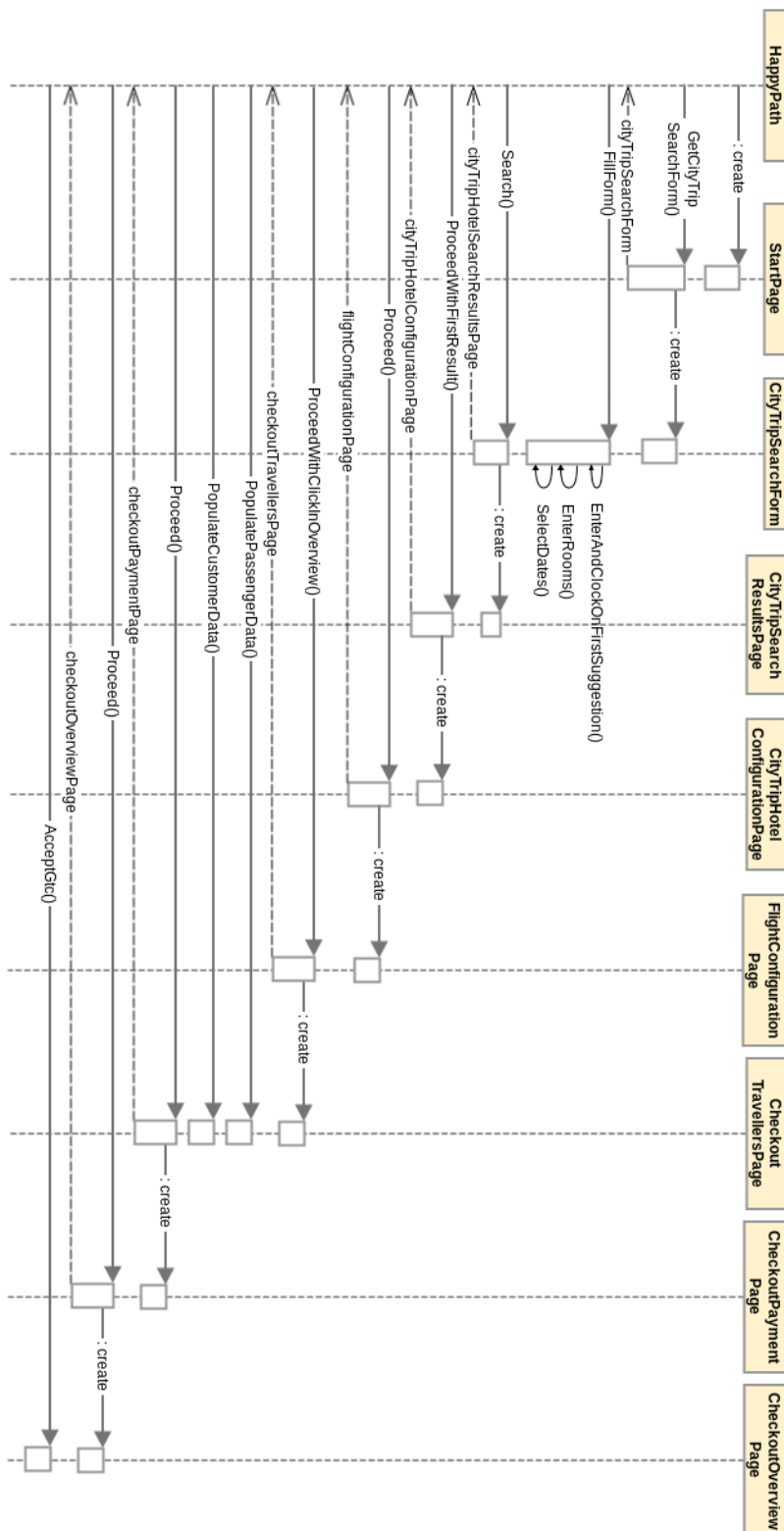


Abbildung 6.8: Sequenzdiagramm für den Ablauf eines CityTrip Tests

Dass Diagramm soll aufzeigen, dass man über eine Page oder ein Modul immer eine Seite weiter gelangt.

#### 6.4.5 Modules

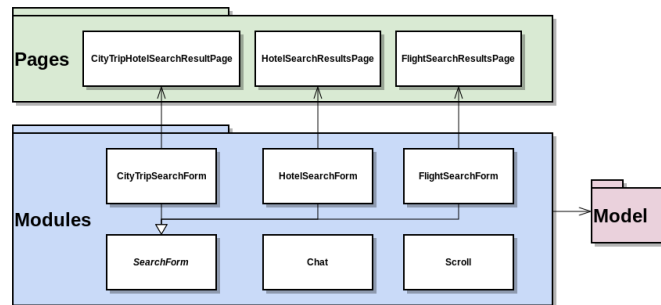


Abbildung 6.9: Klassendiagramm Modules

Module sind wiederkehrende Teile einer Page (siehe Abschnitt 6.3.1 Page Object). Beispiele dafür sind die Suchformulare der verschiedenen Engines, da diese auf der Start-, auf der Engine-Einstiegs- sowie auf der Suchresultatseite vorkommen.

## 6.5 Probleme

### 6.5.1 Timing

Wichtig für den Erfolg von Selenium Tests sind die Timings. Einige Teile der Webseite brauchen mehr Zeit bis sie geladen sind, da viel Rechenleistung benötigt wird oder sie mit JavaScript nachgeladen werden. Die Pages (siehe ?? Page Object) warten deshalb im Konstruktor, bis die Webseite geladen ist. Dazu wird die Klasse WebDriverWait von Selenium verwendet. Es wird gewartet, bis ein HTML-Element gefunden wird.

Die Timings sind sehr aufwändig, da man zu Beginn meistens nicht genau weiss, wo die Tests schneller sind als die Webseite. Dies merkt man meistens erst bei der Ausführung und muss danach die Timings nachpflegen, was sehr aufwändig werden kann.

Ein Beispiel dafür ist die Flug-Konfigurations Seite. Dort ist ein Button erst nach wenigen Sekunden anwählbar. Solange er es noch nicht ist hat er die Klasse „btn-loading“.

```

1 public void WaitForPageToBeReady()
2 {
3     wait.Until(d => d.FindElement(By.CssSelector(".js-configuration-submit:not(.
4     btn-loading)")));
  
```

Mit diesem Code wird sichergestellt, dass gewartet wird bis der Button die Klasse nicht mehr hat.

Auf der Flugresultat Seite werden die Flüge über JavaScript nachgeladen, weshalb gewartet werden muss bis eine Antwort eingetroffen ist und diese in dem Browser dargestellt wird.

```

1 public void WaitForResults()
  
```

```
2 {  
3     wait.Until(d => d.FindElement(By.CssSelector(".js-search-content.content-  
        loaded")));  
4 }
```

---

Diese Methode wartet, bis der Behälter für die Suchresultate die Klasse „content-loaded“ besitzt, denn erst dann darf mit dem Test weitergeführt werden.

Die `wait.Until()` Methode nimmt eine [Closure](#) entgegen.

# KAPITEL 7

---

## Schlussfolgerung

---

dasdfad  
aadddd meeeeee!!!

## ANHANG A

---

### Testübersicht

---



## A.1 Testübersicht

## TEMP: Automated Tests for Travel

### Status:

OPEN

DONE

CLARIFICATION REQUIRED

### Prios

1: Tief

10: Hoch

### Browsers

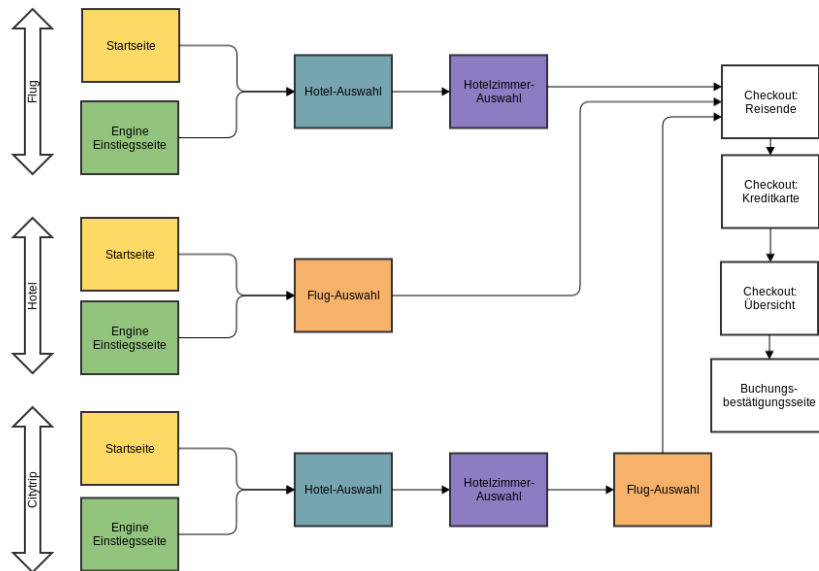
CLARIFICATION REQUIRED

### Mobile

Die Mobile Seite wird redesigned. Deswegen werden Mobile Browsers vorerst nicht getestet.



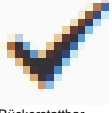


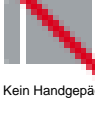
### Ablauf




Klick auf die Boxen um die Tests zu sehen...



## Test Cases

### Citytrip

Beschreibung	Destination	PAX	Abflug Ab	Hotel	Rückerstattbar	Zimmertyp	Verpflegungstyp	Flug
Happy Path Nicht Expedia Hotel 	Berlin	1. Zimmer a. 2 Erwachsene	Zürich	Erstes nicht Expedia Hotel		Erstes rückerstattbares Zimmer	Standardauswahl	Standardauswa
Rückerstattbar Expedia Hotel 	Amsterdam	1. Zimmer a. 1 Erwachsener, b. 1 Kind 2. Zimmer a. 1 Erwachsener	Zürich	Star Inn Hotel Wien Schonbrunn		Standard-einzelzimmer	-	Swiss Direktflug  Kein Handgepä
Kein Handgepäck 								







Nicht Expedia Hotel Nur 1 Zimmert- & Verpflegungstyp Flug mit Zwischenstopp	Barcelona	1. Zimmer a. 2 Erwachsene 2. Zimmer a. 1 Erwachsener b. 1 Kind	Basel	LLORET				Lufthansa 1 Zwischenstopp
Expedia Hotel  Nicht Rückerstattbar Zimmerauswahl mit nicht anwählbaren Optionen	London	1. Zimmer a. 2 Erwachsene	Genève-Cointrin	Hilton London Metropole		Deluxe-zimmer, 1 king-bett	-	Easy Jet
Suche Smoketest	Wien	1. Zimmer a. 2 Erwachsener	Zürich					
Suche Smoketest	Rom	1. Zimmer a. 2 Erwachsener	Zürich					
Suche Smoketest	Lissabon	1. Zimmer a. 2 Erwachsener	Zürich					
Suche Smoketest	Hamburg	1. Zimmer a. 2 Erwachsener	Zürich					
Suche Smoketest	Prag	1. Zimmer a. 2 Erwachsener	Zürich					
Suche Smoketest	Istanbul	1. Zimmer a. 2 Erwachsener	Zürich					

## Flug

Beschreibung	Destination	PAX	Abflug Ab	Flug
Happy Path	Berlin	2 Erwachsene	Zürich	Standardauswahl
	New York City	2 Erwachsene 1 Kind	Zürich	
	Bangkok	1 Erwachsener	Basel	
Suche Smoketest	Palma de Mallorca	1 Erwachsener	Zürich	
Suche Smoketest	London	1 Erwachsener	Zürich	
Suche Smoketest	Miami	1 Erwachsener	Zürich	
Suche Smoketest	Los Angeles	1 Erwachsener	Zürich	
Suche Smoketest	Wien	1 Erwachsener	Zürich	
Suche Smoketest	San Francisco	1 Erwachsener	Zürich	
Suche Smoketest	Barcelona	1 Erwachsener	Zürich	

## Hotel

Beschreibung	Destination	PAX	Hotel	Rückerstattbar	Zimmertyp	Verpflegungstyp	Bezahlung
--------------	-------------	-----	-------	----------------	-----------	-----------------	-----------

Happy Path	Berlin	1. Zimmer a. 2 Erwachsene	Erstes nicht Expedia Hotel		Erstes rückerstattbares Zimmer	Standardauswahl	Rechnung
Expedia Hotel  Rückerstattbar	Rom	1. Zimmer a. 1 Erwachsene	Hotel Quirinale		Standard-doppelzimmer	Frühstücksbuffet	Rechnung
Nicht Expedia Hotel Nur 1 Zimmertyp- & Verpflegungstyp	New York City	1. Zimmer a. 2 Erwachsene 2. Zimmer a. 1 Erwachsene b. 1 Kind	Novotel New York Times Square				CC
Expedia Hotel  Nicht Rückerstattbar Zimmerauswahl mit nicht anwählbaren Optionen	Paris	1. Zimmer a. 2 Erwachsene	Novotel Paris Centre Tour Eiffel		Standard-zweibettzimmer	-	Rechnung
Suche Smoketest	London	1. Zimmer a. 2 Erwachsene					
Suche Smoketest	Barcelona	1. Zimmer a. 2 Erwachsene					
Suche Smoketest	München	1. Zimmer a. 2 Erwachsene					
Suche Smoketest	Palme de Mallorca	1. Zimmer a. 2 Erwachsene					
Suche Smoketest	Wien	1. Zimmer a. 2 Erwachsene					
Suche Smoketest	Venedig	1. Zimmer a. 2 Erwachsene					

## Top 10 Destinationen

Citytrip	Flug	Hotel
BER	BKK	LON
BCN	BYC	ROM

VIE	LON	PAR
LON	BER	NYC
AMS	MIA	BER
ROM	PMI	BCN
LIS	LAX	MUC
HAM	VIE	PMI
PRG	SFO	VIE
IST	BCN	VCE

## ANHANG B

---

### Funktionalitäten

---

## B.1 Diverses

## Diverses

Status	Kategorie	Engine	Beschreibung	Prio	Notizen	Implemented in:
CLARIFICATION REQUIRED	Happy Path	City	Happy Path (Suchen bis Buchen)	10	wie kann sicher gestellt werden, dass die Buchung wieder storniert wird auf P? (mit Nicole abklären?)	HappyPath.City
CLARIFICATION REQUIRED		Flug	Happy Path (Suchen bis Buchen)	10	wie kann sicher gestellt werden, dass die Buchung wieder storniert wird auf P? (mit Nicole abklären?)	HappyPath.Flight
CLARIFICATION REQUIRED		Hotel	Happy Path (Suchen bis Buchen)	10	wie kann sicher gestellt werden, dass die Buchung wieder storniert wird auf P? (mit Nicole abklären?)	HappyPath.Hotel
OPEN	Header		Travel Logo: Wird angezeigt	8		
OPEN			Travel Logo: Verlinkt auf Startseite	3		
OPEN			Flyout: Kann geöffnet & geschlossen werden	3		
OPEN			Flyout: Offen liegt es über dem restlichen Inhalt	3		
OPEN			Kontakt: Telefonnummer wird angezeigt	8		
OPEN			Kontakt: Bürozeiten werden angezeigt	3		
OPEN	Footer		Ansprechspartner: 16 Bilder von Travel Mitarbeitern werden angezeigt	1		
OPEN			Ansprechspartner: Flyout kann geöffnet & geschlossen werden	1		
OPEN			Ansprechspartner: Flyout schiebt restlichen Inhalt nach unten	4		
OPEN			Ansprechspartner: Flyout kann mit einem klick ausserhalb des Flyouts geschlossen werden.	1		
OPEN			Newsletter: Eingabefeld & Button wird angezeigt.	4		
OPEN			Newsletter: Wird eine ungültige E-Mail Adresse eingegeben und "Anmelden" Button gedrückt, erscheint eine Fehlermeldung.	3		
OPEN			Newsletter: Wird eine gültige E-Mail Adresse eingegeben und "Anmelden" Button gedrückt, erscheint eine Bestätigung der Newsletteranmeldung	2		
OPEN	Main-Navigation		Städtereisen: Main Navigation hat einen Link zur Städtereisen Engine	5		
OPEN			Flüge: Main Navigation hat einen Link zur Flüge Engine	5		
OPEN			Hotels: Main Navigation hat einen Link zur Hotels Engine	5		
OPEN			La-Mi: Main Navigation hat einen Link zur La-Mi Engine	5		
OPEN			Kreuzfahrten Führt auf die Seite kreuzfahrten.travel.ch	5		
OPEN			Mehr: Öffnet & schliesst ein Flyout	1		
OPEN			Mehr: Flyout kann mit einem Klick ausserhalb des Flyouts geschlossen werden	1		
OPEN			Mehr - Ferienthemen: Öffnet einen Link auf die Ferienthemen	1		
OPEN						



OPEN			Mehr - Ferienwohnungen: Öffnet einen Link auf die Ferienwohnungen	1		
OPEN			Mehr - Mietwagen: Öffnet einen Link auf die Mietwagen	1		
OPEN			Mehr - Sprachreisen: Öffnet einen Link auf die Sprachreisen	1		
OPEN			Mehr - Touren und Ausflüge: Öffnet einen Link auf die "Touren und Ausflüge"	1		
OPEN	Progress Tracker	Citytrip	Enthält folgende Punkte: <ul style="list-style-type: none"> <li>• Suche</li> <li>• Hotel</li> <li>• Zimmer</li> <li>• Flugverbindung</li> <li>• Personalien</li> <li>• Bezahlung</li> <li>• Zusammenfassung</li> </ul>	5		
OPEN		Hotel	Enthält folgende Punkte: <ul style="list-style-type: none"> <li>• Suche</li> <li>• Hotel</li> <li>• Zimmer</li> <li>• Personalien</li> <li>• Bezahlung</li> <li>• Zusammenfassung</li> </ul>	5		
OPEN		Flug	Enthält folgende Punkte: <ul style="list-style-type: none"> <li>• Suche</li> <li>• Flugverbindung</li> <li>• Personalien</li> <li>• Bezahlung</li> <li>• Zusammenfassung</li> </ul>	5		
OPEN		Alle	Icons: Icons links vom aktiven Step haben einen Haken als Logo	3		
OPEN			Icons: Der aktive Step und alle Steps rechts davon haben die Step-Nummer als Logo	3		
OPEN			Icons: Der aktive Step ist hervorgehoben	3		
OPEN						








## B.2 Startseite


## Startseite

URL: [www.travel.ch](http://www.travel.ch)

Status	Kategorie	Engine	Beschreibung	Prio	Notizen	Implemented in
DONE	Suche - Top Destinations	City/Flug/Hotel	Suche nach Berlin	10		SearchSmokeTests
DONE			Suche nach Barcelona	10		SearchSmokeTests
DONE			Suche nach .....	10	Top 10 Destinationen aufgeführt unter: <a href="#">TE MP: Automated Tests for Travel</a>	SearchSmokeTests
OPEN	Suche - Filterung	City	Passagiere: Wird berücksichtigt (3 Erwachsene, 1 Kind in 2 Zimmer)	7	Zimmerverteilung ist bei einigen Hotelprovidern relevant -> 2 Erwachsene in Zimmer 1, 1 Erwachsener 1 Kind in Zimmer 2	
OPEN			Passagiere: Kind Geburtsdatum: Darf max. 18 Jahre alt sein.			
OPEN			Hotelklasse: Wird berücksichtigt	4		
OPEN			Bewertung: Wird berücksichtigt	4		
OPEN			Verpflegung: Wird berücksichtigt	4		
OPEN			Direktflug: Wird berücksichtigt	4		
OPEN			Abflugdatum: Wird berücksichtigt			
OPEN			Abflugdatum in der Vergangenheit nicht anwählbar	8		
OPEN			Abflugdatum muss vor dem Ankunftsdatum liegen	7		
OPEN			Mit / ohne Gepäck: Wird berücksichtigt	4	 <b>TWWEB-566</b> - Mit / ohne Gepäck: Filter auf Suchergebnisseite	<b>RESOLVED</b>
OPEN		Flug	Passagiere: Wird berücksichtigt (3 Erwachsene, 1 Kind)	7		
OPEN			Abflugdatum in der Vergangenheit nicht anwählbar	8		
OPEN			Direktflug: Wird berücksichtigt	4		
OPEN			Flugklasse: Wird berücksichtigt	4		
OPEN			Mit / ohne Gepäck: Wird berücksichtigt	4	 <b>TWWEB-566</b> - Mit / ohne Gepäck: Filter auf Suchergebnisseite	<b>RESOLVED</b>
OPEN						
OPEN		Hotel	Passagiere: Wird berücksichtigt (3 Erwachsene, 1 Kind in 2 Zimmer)	7	Zimmerverteilung ist bei einigen Hotelprovidern relevant -> 2 Erwachsene in Zimmer 1, 1 Erwachsener 1 Kind in Zimmer 2	
OPEN			Hotelklasse: Wird berücksichtigt	4		
OPEN			Bewertung: Wird berücksichtigt	4		
OPEN			Verpflegung: Wird berücksichtigt	4		
OPEN						
DONE	Suche - JavaScript	City	Destination Auswahl: Erscheint wenn 3 Buchstaben eingegeben wurden	8		HappyPath.Citytrip

DONE	Suche - JavaScript	Flug	Destination Auswahl: Schliesst wenn Item angeklickt wird.	5		HappyPath.Citytrip
OPEN			Destination Auswahl: Schliesst wenn ausserhalb der Liste geklickt wird.	4		
OPEN			Filterung: Kann auf- & zugeklappt werden.	4		
OPEN			Filterung: Aufgeklappt liegt die Filterung oberhalb dem Rest.	3	 TWWEB-571 - erweiterte Suche verdeckt Suchergebnisse <span>CLOSED</span>	
DONE			Abreise Kalender: Kann angezeigt werden.	7		HappyPath.Citytrip
DONE			Abreise Kalender: Schliesst sich bei Auswahl & Auswahl wird übernommen.	8		HappyPath.Citytrip
OPEN			Abreise Kalender: Schliesst sich wenn ausserhalb des Kalenders geklickt wird.	4		
DONE			Rückreise Kalender: Kann angezeigt werden.	7		HappyPath.Citytrip
DONE			Rückreise Kalender: Schliesst sich bei Auswahl & Auswahl wird übernommen.	8		HappyPath.Citytrip
OPEN			Rückreise Kalender: Schliesst sich wenn ausserhalb des Kalenders geklickt wird.	4		
DONE			Personen: Auswahl-Popup wird angezeigt.	4		HappyPath.Citytrip
DONE			Personen: Zimmer hinzufügen	6		HappyPath.Citytrip
OPEN			Personen: Zimmer entfernen	6		
OPEN			Personen: Tooltip erscheint beim Mouseover über dem 	1		
OPEN			Abflugsort Destination Auswahl: Erscheint wenn 3 Buchstaben eingegeben wurden	4		
OPEN			Abflugsort Destination Auswahl: Schliesst wenn Item angeklickt wird.	5		
OPEN			Abflugsort Destination Auswahl: Schliesst wenn ausserhalb der Liste geklickt wird.	4		
DONE			Ankunftsort Destination Auswahl: Erscheint wenn 3 Buchstaben eingegeben wurden	8		HappyPath.Flight
DONE			Ankunftsort Destination Auswahl: Schliesst wenn Item angeklickt wird.	4		HappyPath.Flight
OPEN			Ankunftsort Destination Auswahl: Schliesst wenn ausserhalb der Liste geklickt wird.	4		
OPEN			Filterung: Kann auf- & zugeklappt werden	4		
OPEN			Filterung: Aufgeklappt liegt die Filterung oberhalb dem Rest.	3	 TWWEB-571 - erweiterte Suche verdeckt Suchergebnisse <span>CLOSED</span>	

DONE			Hinflug Kalender: Kann angezeigt werden.	7		HappyPath.Flight
DONE			Hinflug Kalender: Schliesst sich bei Auswahl & Auswahl wird übernommen.	8		HappyPath.Flight
DONE			Rückflug Kalender: Kann angezeigt werden.	7		HappyPath.Flight
DONE			Rückflug Kalender: Schliesst sich bei Auswahl & Auswahl wird übernommen.	8		HappyPath.Flight
DONE			Personen: Auswal-Popup wird angezeigt	4		HappyPath.Flight
OPEN			Personen: Tooltip erscheint beim Mouseover über dem 	1		
OPEN			Airline Dropdown: Kann angezeigt werden.	4		
OPEN			Airline Dropdown: Schliesst sich wenn ausserhalb des Dropdowns geklickt wird.	3		
OPEN			Airline Dropdown: Filtrierung schränkt Auswahl ein.	4		
OPEN			Airline Dropdown: Filtrierung darf keine Einträge doppelt anzeigen.	2	Swiss ist in der gesamten Liste 2x vorhanden. Filtrierung darf jedoch nicht 2x Swiss anzeigen.	
OPEN			Airline Dropdown: "Alle Airlines" entfernt bestehende Auswahl.	2		
CLARIFICATION REQUIRED					Die Airlines im Dropdown sind gruppiert nach: <ul style="list-style-type: none"> <li>Airline Allianzen</li> <li>Beliebte Airlines</li> <li>Airlines</li> </ul> Muss dies getestet werden?	
DONE	Suche - JavaScript	Hotel	Destination Auswahl: Erscheint wenn 3 Buchstaben eingegeben wurden	8		HappyPath.Hotel
OPEN			Destination Auswahl: Schliesst wenn Item angeklickt wird.	5		
OPEN			Destination Auswahl: Schliesst wenn ausserhalb der Liste geklickt wird.	4		
OPEN			Filtrierung: Kann auf- & zugeklappt werden	4		
OPEN			Filtrierung: Aufgeklappt liegt die Filtrierung oberhalb dem Rest.	3	 <b>TWWEB-571</b> - erweiterte Suche verdeckt Suchergebnisse <b>CLOSED</b>	
DONE			Abreise Kalender: Kann angezeigt werden.	7		HappyPath.Hotel
DONE			Abreise Kalender: Schliesst sich bei Auswahl & Auswahl wird übernommen	7		HappyPath.Hotel
OPEN			Abreise Kalender: Schliesst sich wenn ausserhalb des Kalenders geklickt wird	4		
DONE			Rückreise Kalender: Kann angezeigt werden	7		HappyPath.Hotel
DONE			Rückreise Kalender: Schliesst sich bei Auswahl & Auswahl wird übernommen	7		HappyPath.Hotel

OPEN	Banner		Rückreise Kalender: Schliesst sich wenn ausserhalb des Kalenders geklickt wird.	4		
DONE			Personen: Auswahl-Popup wird angezeigt.	4		HappyPath.Hotel
DONE			Personen: Zimmer hinzufügen	6		HappyPath.Hotel
OPEN			Personen: Tooltip erscheint beim Mouseover über dem 	1		
OPEN		Städtereisen	Es werden 8 Städtereisen-Banner mit Bild, Preis und Stadtname angezeigt	4		
OPEN			Link "Mehr Angebote" führt zur City-Engine Startseite	4		
OPEN		Flüge	Es werden 3 Flug-Banner angezeigt mit Bild, Preis und Stadtname	4		
OPEN			Link "Mehr Angebote" führt zur Flug-Engine Startseite	4		
OPEN		Hotels	Es werden 3 Hotel-Banner angezeigt mit Bild, Preis und Stadtname	4		
OPEN			Link "Mehr Angebote" führt zur Hotel-Engine Startseite	4		
OPEN		La-Mi	Es werden 3 Lami-Banner mit Bild, Preis und Destination angezeigt	4		
OPEN			Link "Mehr Angebote" führt zur Lami-Engine Startseite	4		

## B.3 Engine Startseite

## Engine Einstiegsseite

URL's:

- Citytrip: <http://www.travel.ch/de/staedtereisen/>
- Flug: <http://www.travel.ch/de/fluege/>
- Hotel: <http://www.travel.ch/de/hotel/>

Status	Kategorie	Engine	Beschreibung	Prio	Notizen
OPEN	JavaScript	City	<a href="#">Startseite</a> -> Suche - JavaScript - City	10	
OPEN			Suche liefert Resultate	10	
OPEN		Flug	<a href="#">Startseite</a> -> Suche - JavaScript - Flug	10	
OPEN			Suche liefert Resultate	10	
OPEN		Hotel	<a href="#">Startseite</a> -> Suche - JavaScript - Hotel	10	
OPEN			Suche liefert Resultate	10	

## B.4 Hotelauswahl

## Hotel-Auswahl

Status	Kategorie	Engine	Beschreibung	Prio	Notizen
OPEN	Progress Tracker		Diverses -> Progress Tracker	5	
OPEN			Aktiver Schritt: Hotel	5	
OPEN	Suchresultate - Liste - Hotel	Citytrip/Hotel	Bewertungen: Link "XXX Bewertungen" öffnet Bewertungsansicht	4	
OPEN			Bewertungen: "Bewertungen" öffnet & schliesst Bewertungen	4	
OPEN			Bilder: Link "Bilder" öffnet & schliesst Bilderansicht	6	
OPEN			Bilder: Wird ein Thumbnail angeklickt wird es links in gross dargestellt.	6	
OPEN			Bilder: Scrolling nach oben & unten möglich.	4	
OPEN			Karte: Link "Karte" öffnet & schliesst Kartenansicht.	6	
OPEN			Karte: Marker ist zentriert ersichtlich.	6	
OPEN			1. Geöffnet: Kartenansicht 2. Klick auf: "Bilder" 3. Kartenansicht wird geschlossen & Bilderansicht geöffnet.	4	
OPEN			1. Geöffnet: Kartenansicht 2. Klick auf: "Bewertungen" 3. Kartenansicht wird geschlossen & Bewertungsansicht geöffnet.	4	
OPEN			1. Geöffnet: Kartenansicht 2. Klick auf: "XXX Bewertungen" 3. Kartenansicht wird geschlossen & Bewertungsansicht geöffnet.	4	
OPEN			1. Geöffnet: Bilderansicht 2. Klick auf: "Karte" 3. Bilderansicht wird geschlossen & Kartenansicht geöffnet.	4	
OPEN			1. Geöffnet: Bilderansicht 2. Klick auf: "Bewertungen" 3. Bilderansicht wird geschlossen & Bewertungsansicht geöffnet.	4	
OPEN			1. Geöffnet: Bilderansicht 2. Klick auf: "XXX Bewertungen" 3. Bilderansicht wird geschlossen & Bewertungsansicht geöffnet.	4	
OPEN			1. Geöffnet: Bewertungsansicht 2. Klick auf: "Karte" 3. Bewertungsansicht wird geschlossen & Kartenansicht geöffnet.	4	
OPEN			1. Geöffnet: Bewertungsansicht 2. Klick auf: "Bilder" 3. Bewertungsansicht wird geschlossen & Bilderansicht geöffnet.	4	
OPEN			1. Geöffnet: Kartenansicht 2. Klick auf "Bilder" eines anderen Suchresultates 3. Kartenansicht wird geschlossen & Bilderansicht geöffnet.	4	
OPEN			Hoteltypen: Es wird maximal 1 Hoteltyp aufgeführt.	4	
OPEN			Feature-Liste: Es werden maximal 3 features aufgeführt.	6	
OPEN	Suchresultate - Liste - Hotel - Karte		Es kann heraus und herein gezoomt werden.	4	
OPEN			Kartenausschnitt kann per Maus verschoben werden.	4	
OPEN			Kartenausschnitt kann per Tastatur verschoben werden.	2	
OPEN			Kartenausschnitt kann mittels den entsprechenden Buttons verschoben werden	2	
OPEN			Satellit Ansicht kann gewählt werden.	1	
OPEN			Hotel-Marker ist zentriert ersichtlich.	6	
OPEN	Suchresultate - Liste - Paging	Citytrip/Hotel	Auf Seite 1: Klick auf "2" lädt die zweite Seite	6	

OPEN			Auf Seite 1: Klick auf "weiter" lädt die zweite Seite.	6	
OPEN			Auf Seite 2: Klick auf "1" lädt die erste Seite	6	
OPEN			Auf Seite 2: Klick auf "zurück" lädt die erste Seite	6	
OPEN			Auf Seite 1 bis (MAX-5) werden rechts "..." angezeigt	2	
OPEN			Auf Seite 5 bis MAX werden links "..." angezeigt	2	
OPEN			Auf Seite 1/2/3/4: Es werden links keine "..." angezeigt.	2	
OPEN			Auf den 1/2/3/4 letzten Seiten: Es werden rechts keine "..." angezeigt.	2	
OPEN	Suchresultate - Liste - Sortierung	Citytrip/Hotel	Dropdown: Klick auf "Beliebteste" öffnet das Dropdown mit allen Sortiermöglichkeiten	3	
OPEN			Bewertung: Klick auf "Bewertung" im Dropdown sortiert die Liste nach Bewertungen	3	
OPEN			Preis: Klick auf "Preis" im Dropdown sortiert die Liste nach Preisen	3	
OPEN			Lage: Beim Klick auf "Lage" erscheint ein aufgeklapptes Dropdwon mit POI's.	5	
OPEN			Lage - POI: Klick auf einen POI sortiert die Liste nach abständen zu dem POI	5	
OPEN			Lage - POI: Ist nach einem POI sortiert erscheint bei jedem Suchresultat der Abstand zu dem POI.	5	
OPEN			Lage - POI: Ist der Abstand eines Hotels zum gewählten POI zwischen 1-999 Meter, so wird die Distanz in Meter angegeben.	3	
OPEN			Lage - POI: Ist der Abstand eines Hotels zum gewählten POI grösser als 999 Meter, so wird die Distanz in Kilometer angegeben.	3	
OPEN	Suchresultate - Karte	Citytrip/Hotel	Karte: Kann heraus und herein gezoomt werden.	4	
OPEN			Karte: Kartenausschnitt kann per Maus verschoben werden.	4	
OPEN			Karte: Kartenausschnitt kann per Tastatur verschoben werden.	1	
OPEN			Karte: Kartenausschnitt kann mittels den entsprechenden Buttons verschoben werden	1	
OPEN			Karte: Satellit Ansicht kann gewählt werden.	1	
OPEN			POI: POIs werden als solche auf der Karte angezeigt.	4	
OPEN			POI: Wird ein POI angeklickt wird dessen Name angezeigt.	4	
OPEN			Gruppierung: Liegen mehrere Hotels nahe zusammen, werden diese zusammen gruppiert und die Anzahl angezeigt.	3	
OPEN			Gruppierung: Wird eine Gruppe angeklickt wird in diese hinein gezoomt.	3	
OPEN			Hotels: Hotels werden als solche auf der Karte angezeigt.	4	
OPEN			Hotels: Wird ein Hotel angeklickt erscheinen folgende Informationen: <ul style="list-style-type: none"> <li>Name</li> <li>Hotelklasse</li> <li>Bewertung</li> <li>Anzahl Bewertungen</li> <li>Preis</li> <li>Button um das Hotel zu buchen</li> </ul>	5	
OPEN			Hotels: Klick auf ein Hotel und anschliessend ein Klick auf den Button führen zur Hotelzimmer Auswahl.	5	
OPEN			Preis: Untere Grenze kann eingeschränkt werden.	4	
OPEN			Preis: Obere Grenze kann eingeschränkt werden.	4	
OPEN	Suchresultate - Filtrierung	Citytrip/Hotel	Hotelklasse: Kann eingeschränkt werden.	4	
OPEN			Bewertung: Kann eingeschränkt werden.	4	

OPEN
OPEN
OPEN
OPEN
OPEN

Verpflegung: Kann eingeschränkt werden.	4	
Hoteltypen: Kann eingeschränkt werden.	4	
Hoteltypen: Wird nur nach einem Hoteltypen gefiltert, wird bei jedem angezeigten Suchresultat der gewählte Hoteltypen dargestellt.	4	
Ausstattung: Kann eingeschränkt werden.	4	
Hotelname: Es kann nach einem Hotelnamen gesucht werden.	4	
Anzahl Resultate: die Anzahl der jeweiligen Resultate werden in Klammern hinter der Filtrierung angezeigt	3	



## B.5 Hotelzimmer Auswahl

## Hotelzimmer-Auswahl



### Preismatrix

Zum besseren Verständnis der Tests nachfolgend ein PrtScr zu einem Expedia Hotel (Hilton Hotel - Berlin) mit einer grossen Preismatrix:

	rückerstattbar ⓘ	Frühstücksbuffet	Großes Frühstück	Kontinentales Frühstück
Zweibettzimmer ⓘ	✗	<input checked="" type="radio"/> CHF 354	—	—
	✓	<input type="radio"/> CHF 393	<input type="radio"/> CHF 415	—
Zimmer Mit king-bett ⓘ	✗	<input type="radio"/> CHF 354	—	—
	✓	<input type="radio"/> CHF 393	<input type="radio"/> CHF 415	—
Einzelzimmer ⓘ	✗	<input type="radio"/> CHF 354	—	—
	✓	<input type="radio"/> CHF 393	<input type="radio"/> CHF 415	—
Plus, King-bett ⓘ	✗	<input type="radio"/> CHF 395	—	—
	✓	<input type="radio"/> CHF 443	—	—
Zweibettzimmer, Blick Auf Die Kathedrale ⓘ	✗	<input type="radio"/> CHF 395	—	—
	✓	<input type="radio"/> CHF 443	—	—
Deluxe, Zweibettzimmer ⓘ	✗	<input type="radio"/> CHF 408	—	—
	✓	<input type="radio"/> CHF 459	—	—
Deluxe, King-bett ⓘ	✗	<input type="radio"/> CHF 408	—	—
	✓	<input type="radio"/> CHF 459	—	—
Executive, 1 einzelbett ⓘ	✗	<input type="radio"/> CHF 426	—	—
Executive, 1 king-bett, Blick Auf Kathedrale ⓘ	✗	<input type="radio"/> CHF 440	—	—
Executive Zweibettzimmer, Blick Auf Die Kathedrale ⓘ	✗	<input type="radio"/> CHF 440	—	—
Executive Studio, King-bett ⓘ	✗	—	—	<input type="radio"/> CHF 453
Executive, 1 king-bett ⓘ	✗	—	<input type="radio"/> CHF 454	—
<b>Totalpreis (Hotel + Flug)</b>		<b>CHF 354</b>		

### Test Cases

Status	Kategorie	Engine	Hotelprovider	Beschreibung	Prio	Notizen
OPEN	Progress Tracker			Diverses -> Progress Tracker	5	
OPEN				Aktiver Schritt: Zimmer	5	

OPEN	Warenkorb	Citytrip		Flug-Auswahl -> Warenkorb	5	
OPEN				Zimmer wählen: "Zimmer wählen" Link springt zur Preismatrix	5	
OPEN	Hotelinfos	Citytrip/Hotel		Titel wird angezeigt	4	
OPEN				Hotelkategorie wird angezeigt	4	
OPEN				Hoteladresse wird angezeigt	4	
OPEN				Auf Karte anzeigen: Der Link "auf Karte anzeigen" scrollt zur Kartenansicht	6	
OPEN				Bilder: Wird ein Thumbnail angeklickt wird es links in gross dargestellt.	6	
OPEN				Bilder: Scrolling nach oben & unten möglich.	4	
OPEN				Gesamt Bewertung wird angezeigt	6	
OPEN				Anzahl Bewertungen werden angezeigt.	4	
OPEN				Klick auf die "Anzahl der Bewertungen" scrollt zu den Bewertungen	6	
OPEN				Hoteltypen: Es werden 3 Hoteltypen angezeigt	6	
OPEN				Hoteltypen: Wenn nach 1 bis 3 Hoteltypen gefiltert wurde, müssen die gewählten Hoteltypen dargestellt werden. Absteigend sortiert nach Anzahl der Treffer in der Suchresultatliste der jeweiligen Hoteltypen.	5	
OPEN				Hoteltypen: Wenn nach mehr als 3 Hoteltypen gefiltert wurde, müssen die Hoteltypen dargestellt werden, welche die meisten Treffer in der Suchresultatliste aufweisen. Absteigend nach dieser Trefferanzahl sortiert.	5	
OPEN				Feature-Liste: Es werden maximal 3 features aufgeführt.	4	
OPEN	Preismatrix	Citytrip/Hotel		Rückerstattbar: Tooltip erscheint beim Mouseover über dem 	5	
OPEN				Rückerstattbar: Ist das selbe Zimmer einmal als "rückerstattbar" und einmal als "nicht rückerstattbar" buchbar, müssen die beiden Zimmer zusammengeführt werden.	4	
OPEN		Expedia		Zimmertyp: Tooltip erscheint beim Mouseover über dem 	3	
OPEN				Auswahl: Die Vorauswahl wird hervorgehoben	4	
OPEN				Auswahl: Wird ein anderer Radio-Button angeklickt wird dieser als angewählt markiert	4	
OPEN				Auswahl: Wird ein anderer Radio-Button angeklickt wird das ganze Feld hervorgehoben.	4	
OPEN				Auswahl: Wird innerhalb der selben Spalte ein Radio-Button angeklickt, wechselt der Zimmerpreis am Ende dieser spalte den Preis entsprechend.	4	
OPEN				Auswahl: Wird in einer anderen Spalte ein Radio-button angeklickt, verschwindet der alte Preis am Ende der vorherigen Spalte. Der neue Preis erscheint am Ende der neuen Spalte.	4	
OPEN		Alle (ausser Expedia)		Auswahl: Wird eine andere Kombination angewählt, wird der Verpflegungstyp im Warenkorb aktualisiert	4	
OPEN		Expedia		Auswahl: Wird eine andere Kombination angewählt, in deren Spaltentitel ein Verpflegungstyp angegeben wird, wird dieser Verpflegungstyp im Warenkorb dargestellt.	4	
OPEN		Expedia		Auswahl: Wird eine andere Kombination angewählt, in deren Spaltentitel kein Verpflegungstyp angegeben wird, wird kein Verpflegungstyp im Warenkorb dargestellt.	4	
OPEN				Auswahl: Wird eine andere Kombination angewählt, wird der Zimmertyp im Warenkorb aktualisiert.	4	
OPEN				Auswahl: Wird eine andere Kombination angewählt, wird derselbe Preis im Warenkorb angezeigt/aktualisiert.	4	

OPEN				Auswahl: Felder welche ein " — " enthalten können nicht angewählt werden.	4	
OPEN	Karte			Die selben Tests wie unter: <a href="#">Hotel-Auswahl</a> -> Suchresultate - Liste - Hotel - Karte		
OPEN	Hoteldetails			Es werden Informationen dargestellt		
OPEN	Bewertungen			Es werden Informationen dargestellt		
OPEN	Hotelfacts			Es werden Informationen dargestellt		
OPEN	Check-in Anweisungen			Es werden Informationen dargestellt		
OPEN	Klimadaten			Es werden Klimadaten von der Stadt ausgewählt, in der sich das Hotel befindet		
OPEN	Weitere Angebote			Es werden Informationen dargestellt		

## B.6 Flugauswahl

## Flug-Auswahl

Status	Kategorie	Engine	Hotelprovider	Beschreibung	Prio	Notizen
OPEN	Progress Tracker			Diverses -> Progress Tracker	5	
OPEN				Aktiver Schritt: Flugverbindung	5	
OPEN	Warenkorb	Citytrip		Destination wird angezeigt	5	
OPEN				Reisezeitraum wird angezeigt	5	
OPEN				Hotelname des gewählten Hotels wird angezeigt.	5	
OPEN				Hotelklasse des gewählten Hotels wird angezeigt.	3	
OPEN				Zimmertyp: Gewählter Zimmertyp wird angezeigt.	3	
OPEN			Alle (ausser Expedia)	Verpflegungstyp: Der gewählte Verpflegungstyp wird angezeigt.	4	
OPEN			Expedia	Verpflegungstyp: Ist ein Verpflegungstyp ausgewählt, wird dieser angezeigt.	3	
OPEN			Expedia	Verpflegungstyp: Ist kein Verpflegungstyp ausgewählt, wird keiner angezeigt.	3	
DONE				Hinflug: Gewählter Hinflug wird angezeigt.	7	Citytrip.AssertThatSelectingAnSegmentChangesTheOverview
CLARIFICATION REQUIRED				Rückflug: Gewählter Rückflug wird angezeigt.	7	Citytrip.AssertThatSelectingAnSegmentChangesTheOverview Funktioniert nicht!! Test ist auskommentiert
OPEN				Preis wird angezeigt.	7	Citytrip.AssertThatSelectingAnSegmentChangesTheOverview
OPEN				Preis pro Person wird angezeigt.	5	
OPEN				Personen: Anzahl Erwachsenen wird angezeigt.	4	
OPEN				Personen: Anzahl Kinder wird angezeigt.	4	
OPEN	Suchresultat	Citytrip/Flug		Gewähltes Suchresultat ist hervorgehoben.	5	
DONE				Gewähltes Suchresultat: Gewählter Hinflug sowie Rückflug ist angewählt.	7	Flight.AssertIfChangingRadiosDoesNotAffectOtherFlights
OPEN				Nicht gewähltes Suchresultat: Keine Flüge angewählt.	5	


DONE		Hinflüge ist nur einer anwählbar.	7	Flight.AssertIfChangingRadiosDoesNotAffectOtherFlights
DONE		Rückflüge ist nur einer anwählbar.	7	Flight.AssertIfChangingRadiosDoesNotAffectOtherFlights
OPEN		Hinflug: Datum wird angezeigt.	7	Flight.AssertionsForEachSearchResult
OPEN		Hinflug: Carrier wird angezeigt.	4	Flight.AssertThatImagesBeingDisplayed
DONE		Hinflug: Hat mind. 1 anwählbarer Flug.	7	Flight.AssertionsForEachSearchResult
OPEN		Flugklasse (Economy/Business) wird angezeigt.	4	
DONE		Totalpreis wird angezeigt.	7	Flight.AssertThatTotalpricesBeingDisplayed
OPEN		Preis/Person wird angezeigt.	4	
OPEN		Tooltips: Korrekte Reisedauer wird angezeigt.	4	
OPEN		Tooltips: Hinflugzeit mit entsprechendem Carrier wird angezeigt.	4	
OPEN		Tooltips: Rückflugzeit mit entsprechendem Carrier wird angezeigt.	4	
OPEN		Tooltips: Carrier des gesamten Fluges wird angezeigt.	4	
OPEN		Tooltips: Flugnummer wird angezeigt.	4	
OPEN		Tooltips: Aufenthaltsdauer wird bei Flügen mit mind. 1 Stopp angezeigt.	4	
OPEN		Gepäck: Ist kein Gepäck im Flug inbegriffen, wird beim entsprechenden Flug mit einem Icon angezeigt.	4	
OPEN		Gepäck: Fährt man mit der Maus über das "kein Gepäck inbegriffen" Icon, wird ein Tooltip angezeigt.	4	
OPEN		Flugauswahl: Wird ein Flug ausgewählt und der Button neben dem Flug gedrückt, so wird auf der nächsten Seite der gewählte Flug angezeigt.	7	Flight.Flight_ToNewYork_FromZurich_1Adult

OPEN	Citytrip		Flugauswahl: Wird ein anderer Hinflug ausgewählt verändert sich entsprechend der Warenkorb. Preis im Warenkorb bleibt gleich.	7	
OPEN			Flugauswahl: Wird ein anderer Rückflug ausgewählt verändert sich entsprechend der Warenkorb. Preis im Warenkorb bleibt gleich.	7	
OPEN			Flugauswahl: Wird ein anderes Suchresultat ausgewählt, verändert sich entsprechend der Warenkorb. Preis im Warenkorb wird angepasst.	7	
OPEN			Flugauswahl: Wird ein Flug ausgewählt und dann im Warenkorb den fortfahren Button gedrückt, so wird auf der nächsten Seite der gewählte Flug angezeigt.	7	

## B.7 Checkout: Passagiere

## Checkout: Reisende


Status	Kategorie	Engine	Hotelpvider	Beschreibung	Prio	Notizen
OPEN	Progress Tracker			Diverses -> Progress Tracker	5	
OPEN				Aktiver Schritt: Personalien	5	
OPEN	Warenkorb	Citytrip/Hotel		Hotelname wird angezeigt.	7	
OPEN				Hotelklasse wird angezeigt.	5	
OPEN				Nächte: Anzahl Nächte wird angezeigt.	7	
OPEN				Personen: Anzahl Erwachsene wird angezeigt.	7	
OPEN				Personen: Anzahl Kinder wird angezeigt.	7	
OPEN				Zimmertypen werden angezeigt.	5	
OPEN			Alle (ausser Expedia)	Verpflegungstypen werden angezeigt.	5	
OPEN			Expedia	Verpflegungstypen werden angezeigt, wenn einer in der Hotelzimmer Auswahl ausgewählt wurde.	5	
OPEN			Expedia	Verpflegungstypen werden nicht angezeigt, wenn keiner in der Hotelzimmer Auswahl ausgewählt wurde.	5	
OPEN				Abreise Datum wird angezeigt.	7	
OPEN				Rückreise Datum wird angezeigt.	7	
OPEN		Hotel		Zimmertypen werden separat aufgeführt mit jeweiligem Preis.	5	
OPEN		Citytrip/Flug		Abreise Datum wird angezeigt.	7	
OPEN				Rückreise Datum wird angezeigt.	7	
OPEN				Hinflug Abflughafen IATA wird angezeigt.	7	
OPEN				Rückflug Abflughafen IATA wird angezeigt.	7	
OPEN				Hinflug Abflugszeit wird angezeigt.	7	
OPEN				Rückflug Abflugszeit wird angezeigt.	7	
OPEN		Alle		Totalpreis wird angezeigt.	7	
OPEN	Reisende	Alle		Pro Zimmer müssen die Personen-Angaben eingegeben werden.	7	
OPEN				Geburtsdatum: Muss für Erwachsene eingegeben werden.	7	
OPEN				Geburtsdatum: Muss für Kinder nicht eingegeben werden.	7	
OPEN				Anrede Dropdown kann aufgeklappt werden.	7	
OPEN				Anrede: Im Dropdwon kann ein Wert ausgewählt werden.	7	
OPEN				Anrede: Fehler wenn: <ul style="list-style-type: none"> <li>Anrede aufgeklappt</li> <li>Nichts anwählen</li> <li>Ausserhalb des Dropdowns geklickt wird.</li> </ul>	5	
OPEN				Anrede: Fehler wenn: <ul style="list-style-type: none"> <li>Anrede aufgeklappt</li> <li>Oberstes Item ausgewählt wird</li> <li>Ausserhalb des Dropdowns geklickt wird.</li> </ul>	5	
OPEN				Textfeld: Text kann in ein Textfeld eingegeben werden.	7	

OPEN	Cumulus	Alle	Textfeld: Fehler wenn:	7	
			<ul style="list-style-type: none"> <li>• Ins Feld geklickt wird</li> <li>• Nichts eingegeben wird</li> <li>• Ausserhalb des Feldes geklickt wird.</li> </ul>		
OPEN			Geburtsdatum Dropdown kann aufgeklappt werden.	7	
OPEN			Geburtsdatum: Im Dropdwon kann ein Wert ausgewählt werden.	7	
OPEN			Wird Cumulus Checkbox angeklickt erscheint:	3	
			<ul style="list-style-type: none"> <li>• Infotext mit der Anzahl zu vergebenen Punkte</li> <li>• Eingabefeld für die Cumulus Nummer</li> </ul>		
OPEN			Cumulus Textfeld: Text kann in ein Textfeld eingegeben werden.	4	
OPEN			Cumulus Textfeld: Fehler wenn:	4	
			<ul style="list-style-type: none"> <li>• Ins Feld geklickt wird</li> <li>• Nichts eingegeben wird</li> <li>• Ausserhalb des Feldes geklickt wird.</li> </ul>		
OPEN	Formular absenden		Cumulus Textfeld: Fehler wenn keine gültige Cumulus Nummer eingegeben wurde.	4	
OPEN			Tooltip erscheint beim MouseOver über dem 	3	
OPEN			Wird das Formular abgesendet werden alle Felder validiert.	7	



## B.8 Checkout: Bezahlart

## Checkout: Kreditkarte

Status	Kategorie	Beschreibung	Prio	Notizen
OPEN	Progress Tracker	Diverses -> Progress Tracker		
OPEN		Aktiver Schritt: Bezahlung		
OPEN	Warenkorb	Checkout: Reisende -> Warenkorb	7	
OPEN	Kreditkarte	Bei Auswahl des Kreditkarten Radio Buttons erscheint das CC-Eingabeformular mit folgenden Feldern: <ul style="list-style-type: none"> <li>Kartenummer</li> <li>Gültig bis (Monat)</li> <li>Gültig bis (Jahr)</li> <li>Kartenprüfnummer</li> <li>Karteninhaber</li> </ul>	10	
OPEN		Kreditkarten-Formular verschwindet, wenn "Zahlung per Rechnung" angeklickt wird.	4	
OPEN		Kartenummer: Wenn eine Zahl mit 16 Stellen eingegeben wird, wird kein Fehler angezeigt, wenn das Feld verlassen wird.	7	
OPEN		Kartenummer: Wenn eine Zahl mit weniger als 16 Stellen eingegeben wird, wird ein Fehler angezeigt, wenn das Feld verlassen wird.	8	
OPEN		Gültig bis (Monat): Wenn eine Zahl zwischen 01 und 12 (mit führender 0) eingegeben wird, wird kein Fehler angezeigt, wenn das Feld verlassen wird.	5	
OPEN		Gültig bis (Monat): Wenn einer der folgenden Werte eingegeben wird, wird ein Fehler angezeigt, wenn das Feld verlassen wird: <ul style="list-style-type: none"> <li>0</li> <li>00</li> <li>Grösser als 12</li> <li>Zahl zwischen 1 und 9 ohne führende Null.</li> </ul>	8	
OPEN		Gültig bis (Jahr): Wenn zwei Ziffern eingegeben werden, wird kein Fehler angezeigt, wenn das Feld verlassen wird.	5	
OPEN		Gültig bis (Jahr): Wenn nicht zwei Ziffern (bsp: a, 9, 9a, a9) eingegeben werden, wird ein Fehler angezeigt, wenn das Feld verlassen wird.	8	
OPEN		Kartenprüfnummer: Wenn drei Zifferneingegeben werden, wird kein Fehler angezeigt, wenn das Feld verlassen wird.	5	
OPEN		Kartenprüfnummer: Wenn nicht drei Zifferneingegeben (bsp: a, aa, aaa, 9, 99) werden, wird ein Fehler angezeigt, wenn das Feld verlassen wird.	8	
OPEN		Alle Felder zeigen einen Fehler an, wenn: <ol style="list-style-type: none"> <li>Ins Feld geklickt wird</li> <li>Nichts eingegeben wird</li> <li>Ausserhalb des Feldes geklickt wird.</li> </ol>	8	
OPEN	Formular absenden	Wird das Formular abgesendet werden alle Felder validiert.	8	
OPEN		CC Validierung: Ist die CC ungültig, wird ein Fehler angezeigt und das CC Formular ist aufgeklappt.	8	<div>  <b>TWWEB-570</b> - CC Formular anzeigen wenn CC falsch eingegeben wurde. <a href="#">OPEN</a> </div>

## B.9 Checkout: Übersicht

## Checkout: Übersicht

Status	Kategorie	Bedingungen	Beschreibung	Prio	Notizen
OPEN	Progress Tracker		Diverses -> Progress Tracker	5	
OPEN			Aktiver Schritt: Zusammenfassung	5	
OPEN	Warenkorb		Checkout: Reisende -> Warenkorb	7	
OPEN			Zahlungsmethode: Wenn Rechnung gewählt wurde wird dies dargestellt.	7	
OPEN			Zahlungsmethode: Wenn CC gewählt wurde werden die CC Gebühren dargestellt.	7	
OPEN	Reisende		<ul style="list-style-type: none"> <li>Alle Reisende aufgeführt</li> <li>Alle Daten so aufgeführt wie sie eingegeben wurden.</li> </ul>	4	
OPEN	Kontaktdaten		Daten werden angezeigt wie sie eingegeben wurden.	6	
OPEN	Zahlungsmethode		Rechnung: Zahlung per Rechnung wird angezeigt	4	
OPEN			CC: Es werden folgende Infos angezeigt: <ul style="list-style-type: none"> <li>Kartentyp</li> <li>Kartenummer maskiert (xxxx xxxx xxxx 1234)</li> <li>Gültigkeit</li> </ul>	4	
OPEN	Stornierungsbedingungen	Non-Refundable Hotel	NonRef Info wird angezeigt.	4	
OPEN			Annulationskosten ab heute	5	
OPEN		Refundable Hotel	Mind. 1 Annulationskosten Info.	5	
OPEN	AGB	Engine ist City/Hotel UND Hotelprovider nicht Expedia	Travel AGS's sind verlinkt	4	
OPEN		Engine ist City/Hotel UND Hotelprovider ist Expedia	Expedia AGB's sind verlinkt	5	
OPEN		Engine ist City/Flug	Info darüber das die Flugtickets nach dem Buchen direkt ausgestellt werden (keine Änderungen möglich).	5	
OPEN			Wenn AGB Checkbox abgewählt wird erscheint eine Fehlermeldung	8	
OPEN	Formular absenden		Wenn AGB Checkbox nicht gesetzt ist erscheint eine Fehlermeldung.	8	

## B.10 Bestätigungsseite

## Buchungsbestätigungsseite

Status	Kategorie	Beschreibung	Prio	Notizen
OPEN	Warenkorb	Checkout: Übersicht -> Warenkorb	7	
OPEN	Stornierungsbedingungen	Checkout: Übersicht -> Stornierungsbedingungen	7	
OPEN	Buchungsinformationen	Buchungsnummer wird angezeigt	7	
OPEN		Reisende werden aufgeführt	5	
OPEN		Hotelbild wird angezeigt	3	
OPEN		Hotelname wird angezeigt	7	
OPEN		Hotelkategorie wird angezeigt	3	
OPEN		Hotelbewertung wird angezeigt	3	
OPEN		Anzahl Hotelbewertungen wird angezeigt	3	
OPEN		Abreisedatum wird angezeigt	7	
OPEN		Rückreisedatum wird angezeigt	7	
OPEN		Destination wird angezeigt	7	
OPEN		Anzahl Nächte wird Angezeigt	4	
OPEN		Print Link: Es wird ein Print Link angezeigt	1	
OPEN		Kalender: Es wird ein Link angezeigt zum Downloaden eines iCal Files	1	
OPEN		Facebook Share Link wird angezeigt	1	
OPEN		Newsletter Anmeldung Link wird angezeigt.	1	
OPEN	Cross-Selling	Mietwagen: 3 Angebote werden angezeigt	5	
OPEN		Mietwagen: Angebote haben folgende Informationen: <ul style="list-style-type: none"> <li>• Preis</li> <li>• Bild</li> <li>• Link zum Angebot</li> </ul>	5	

---

## Quellenverzeichnis

---

- [1] (Siehe S. 21).
- [2] *Apache Subversion*. URL: <https://subversion.apache.org/> (besucht am 26.07.2015) (siehe S. 4).
- [3] *Automated deployment for .NET - Octopus Deploy*. URL: <https://octopus.com/> (besucht am 28.09.2015) (siehe S. 24).
- [4] *Continuous Integration & Build Server - Bamboo | Atlassian*. URL: <https://www.atlassian.com/software/bamboo> (besucht am 26.07.2015) (siehe S. 5, 24).
- [5] *Cross Browser Testing. Real mobile devices & browsers!* URL: <https://crossbrowsertesting.com/> (besucht am 26.07.2015) (siehe S. 5).
- [6] *CrossBrowserTesting Pricing Model*. URL: <https://crossbrowsertesting.com/pricing#plans> (besucht am 26.07.2015) (siehe S. 6).
- [7] *Document Object Model - Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Document\\_Object\\_Model](https://en.wikipedia.org/wiki/Document_Object_Model) (besucht am 04.10.2015) (siehe S. 22).
- [8] *Download Firefox — Free Web Browser — Mozilla*. URL: <https://www.mozilla.org/en-US/firefox/new/> (besucht am 26.09.2015) (siehe S. 21).
- [9] *Git*. URL: <http://www.git-scm.com/> (besucht am 26.07.2015) (siehe S. 4).
- [10] *Git and Mercurial code management for teams*. URL: <https://bitbucket.org/> (besucht am 26.07.2015) (siehe S. 4, 23).
- [11] *Happy 10th Birthday, Selenium | ThoughtWorks*. URL: <http://www.thoughtworks.com/insights/blog/happy-10th-birthday-selenium> (besucht am 26.07.2015) (siehe S. 5).
- [12] *Happy path - Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Happy\\_path](https://en.wikipedia.org/wiki/Happy_path) (besucht am 30.07.2015) (siehe S. 10).
- [13] *NetScaler Application Delivery Controller - Application Security and Cloud Scalability - Citrix*. URL: <https://www.citrix.com/products/netScaler-application-delivery-controller/overview.html> (besucht am 05.10.2015) (siehe S. 32).
- [14] *NUnit - Home*. URL: <http://www.nunit.org/> (besucht am 26.09.2015) (siehe S. 29).

- [15] *OS & Browser Configurations for Cross Browser Compatibility Testing / Pick an OS Pick a Browser Test a Website*. URL: <https://crossbrowsertesting.com/browsers> (besucht am 26.07.2015) (siehe S. 6).
- [16] *OSI-Modell – Wikipedia*. URL: [https://de.wikipedia.org/wiki/OSI-Modell#Schicht\\_7\\_.E2.80.93\\_Anwendungsschicht\\_.28Application\\_Layer.29](https://de.wikipedia.org/wiki/OSI-Modell#Schicht_7_.E2.80.93_Anwendungsschicht_.28Application_Layer.29) (besucht am 30.07.2015) (siehe S. vii).
- [17] *PageObjects and PageFactory – Selenium Design Patterns (C#, Java)*. URL: <http://relevantcodes.com/pageobjects-and-pagefactory-design-patterns-in-selenium/> (besucht am 05.10.2015) (siehe S. 25).
- [18] *Platforms*. URL: <https://saucelabs.com/platforms/> (besucht am 26.07.2015) (siehe S. 6).
- [19] *Sauce Labs: Pricing*. URL: <https://saucelabs.com/pricing> (besucht am 26.07.2015) (siehe S. 6).
- [20] *Sauce Labs: Selenium Testing, Mobile Testing, JS Unit Testing and More*. URL: <https://saucelabs.com/> (besucht am 26.07.2015) (siehe S. 5).
- [21] *Selenium - Web Browser Automation*. URL: <http://docs.seleniumhq.org/> (besucht am 26.07.2015) (siehe S. 5).
- [22] *Selenium - Web Browser Automation*. URL: <http://docs.seleniumhq.org/> (besucht am 26.09.2015) (siehe S. 21).
- [23] *Selenium IDE Plugins*. URL: <http://docs.seleniumhq.org/projects/ide/> (besucht am 26.09.2015) (siehe S. 21).
- [24] *Smoke testing (software) - Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Smoke\\_testing\\_\(software\)](https://en.wikipedia.org/wiki/Smoke_testing_(software)) (besucht am 01.08.2015) (siehe S. 15).