

**Projekthandbuch**Projekt: WebInterface Reisebüro Graf

[1. Einrichten der Netzwerkumgebung 4](#_Toc463427351)

[1.1. Workstations konfigurieren 4](#_Toc463427352)

[1.2. Domäne 7](#_Toc463427353)

[1.3. Serverstruktur 8](#_Toc463427354)

[1.4. Aufstufung des Domänencontrollers 10](#_Toc463427355)

[1.5. DHCP Server aufsetzen 12](#_Toc463427356)

[1.6. Netzwerkdiagramm(Grafik) 14](#_Toc463427357)

[2. Anwendungsfälle Benutzer 15](#_Toc463427358)

[3. Anwendungsfälle Mitarbeiter 16](#_Toc463427359)

[4. Datenbankentwurf 17](#_Toc463427360)

[4.1. Entity Relationship Model (ERM) 17](#_Toc463427361)

[4.2. Beschreibung 18](#_Toc463427362)

[5. Aufbau der Applikation 19](#_Toc463427363)

[5.1. Model View Controller 19](#_Toc463427364)

[5.1. Datenbankzugriff 19](#_Toc463427365)

[5.2. Authentifizierung und Autorisierung 19](#_Toc463427366)

[5.3. Grafische Gestaltung 21](#_Toc463427367)

[5.4. Unterscheidung zwischen Test- und Echtsystem 25](#_Toc463427368)

[6. Usecases Kunden 26](#_Toc463427369)

[6.1. Reisen suchen und Reisen filtern 26](#_Toc463427370)

[6.2. Buchung tätigen 26](#_Toc463427371)

[6.3. Registrierung 26](#_Toc463427372)

[6.4. Einloggen 26](#_Toc463427373)

[6.5. Profil bearbeiten 26](#_Toc463427374)

[6.6. Reisen bewerten 26](#_Toc463427375)

[7. Usecases Mitarbeiter 26](#_Toc463427376)

[7.1. Neue Reise anlegen 26](#_Toc463427377)

[7.2. Reise ändern 26](#_Toc463427378)

[7.3. Reise löschen 26](#_Toc463427379)

[7.4. Buchungen stornieren 26](#_Toc463427380)

[7.5. Stornierungen rückgängig machen 26](#_Toc463427381)

[8. Businesslogik 26](#_Toc463427382)

[8.1. Benutzerverwaltung 26](#_Toc463427383)

[8.2. Reisenverwaltung 26](#_Toc463427384)

[8.3. Buchungsverwaltung 26](#_Toc463427385)

[8.4. EmailVerwaltung 26](#_Toc463427386)

[8.5. Tools 27](#_Toc463427387)

1. Einrichten der Netzwerkumgebung
   1. Workstations konfigurieren

Ausgangssituation besteht aus 3 Virtuelle Festplatten (VHD)

* IT IN 19
* IT IN 18 🡪 diese muss gelöscht werden und durch IT IN 20 ersetzt werden
* Backup VHD 🡪 diese soll auf Daten:\ kopiert und dort in Volume2 umbenannt werden

IT IN19 booten

Die entstandene einheitliche Ordnerstruktur soll folgendermaßen aufgebaut sein:

C:\

Daten:\

Sicherung:\

Volume1 – IT IN19

Volume2 – IT IN20

EasyBCD

Backup

….

In EasyBCD unter Add new Entry einen neuen Eintrag festlegen. Bei der Pfadangabe den folgenden Pfad auswählen (Daten:\Volume2) und folgenden Namen (IT IN20 WIN 8.1) vergeben.



Daten:\Volume2

IT IN20 WIN 8.1

In den Tools Edit Boot Menu den Eintrag IT IN18 löschen und

anschließend durch Save Settings bestätigen

Anschließend die VHD Volume2 löschen und die Backup VHD auf Daten kopieren

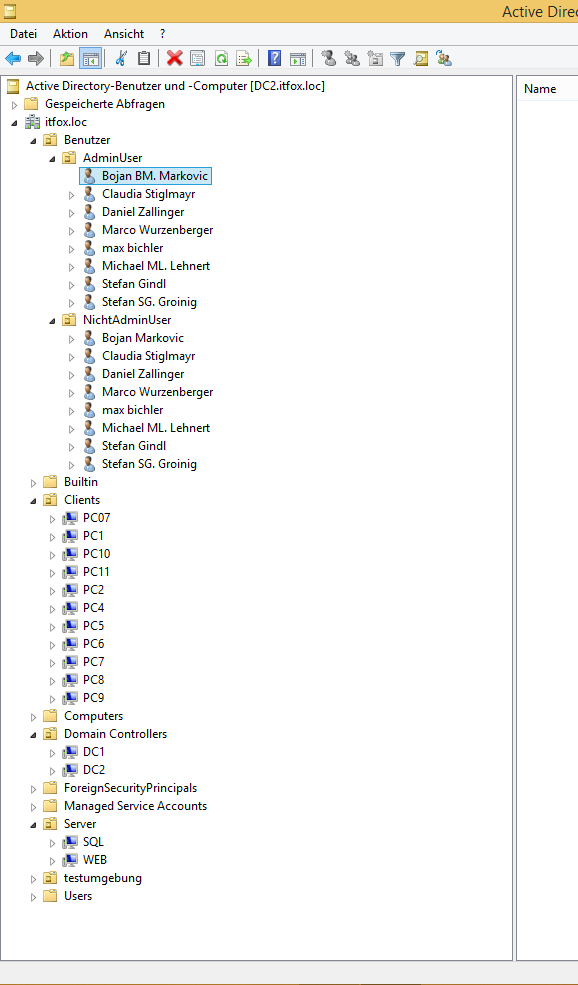
Vor dem Neustart die VHD Backup in Daten:\ in Volume2 umbenennen

TCP/IP Konfiguration

IP-Adressen der Clients: ab 192.168.120.101

Subnetmaske: 255.255.255.0

Standardgateway: 192.168.120.254

* 1. Domäne

Domäne: itfox.loc

* Domänencontroller
  + DC 1: 192.168.120.1 (DNS Server)
  + DC 2: 192.168.120.2
* Organisationseinheit Server
  + SQL: 192.168.120.3
  + Web: 192.168.120.4
* Organisationseinheit Clients
  + PC 1- PC11
* Organisationseinheit Benutzer
  + Organisationseinheit AdminUser
    - Gruppe DomänenAdmins und Domänenbenutzer
    - Lesen und Ändern
    - Anmeldung: ersten 3 Buchstaben Nachname und 3 Buchstaben Vorname und „-ad“  
      (bspw. Sticla-ad@itfox.loc)
  + Organisationseinheit NichtAdminUser
    - Gruppe Domänenbenutzer
    - Lesen
    - Anmeldung: ersten 3 Buchstaben Nachname und 3 Buchstaben Vorname   
      (bspw. Sticla@itfox.loc)
  1. Serverstruktur

Server/Clients in die Domäne aufnehmen

* Überprüfung, ob Server bereits aktiviert sind
  + Windows Update: Product Key ändern
  + Product Key: „7PJBC-63K3J-62TTK-XF46D-W3WMD“
* Server in Domain aufnehmen
  + Unter System – Einstellungen ändern – Systemkonfiguration
  + In die Domäne wird „itfox.loc“ eingetragen, durch die Windows-Sicherheit muss man sich als Administrator anmelden
* Clients in Domäne hinzufügen
  + Zuweisung eines DNS-Servers
  + Prüfen, ob die Domäne gefunden wird (mittels ping itfox.loc)
  + Zur Domäne hinzufügen unter System – Einstellungen ändern
  + Bei Domäne wird „itfox.loc“ eingetragen
* Remotedesktop aktivieren
  + Unter Systemsteuerung – RemoteApp/Desktopverbindung einrichten
  + Unter Server-Manager: Remoteverwaltung für Server konfigurieren – die höchste Sicherheit wählen
* Remotedesktopverbindung
  + IP für Domäne eintragen : Bei DC1 = 192.168.120.1 – als Administrator anmelden
* Remotedesktoptools unter Windows 8.1
  + Als Administrator von itfox.loc anmelden
    - Itfox.loc/Administrator
    - Administrator@itfox.loc
  1. Aufstufung des Domänencontrollers



Installation der Rolle „Active-Directory-Domänendienste“. Auswahl einer Neue Domäne in neuer Gesamtstruktur. Als Domäne wird itfox.loc verwendet. Anmeldeinformation ist ITFOX\Administrator. Als Standort wird „Default-First-Site-Name“ ausgewählt. Der Domänencontroller DC1 fungiert auch als DNS Server und Globaler Katalog

Die Speicherorte sind vorgegeben, hier wird nichts geändert.  
DC2 hochstufen: Festlegen eines Computernamens und Mitgliedschaft einer Domäne

* 1. DHCP Server aufsetzen



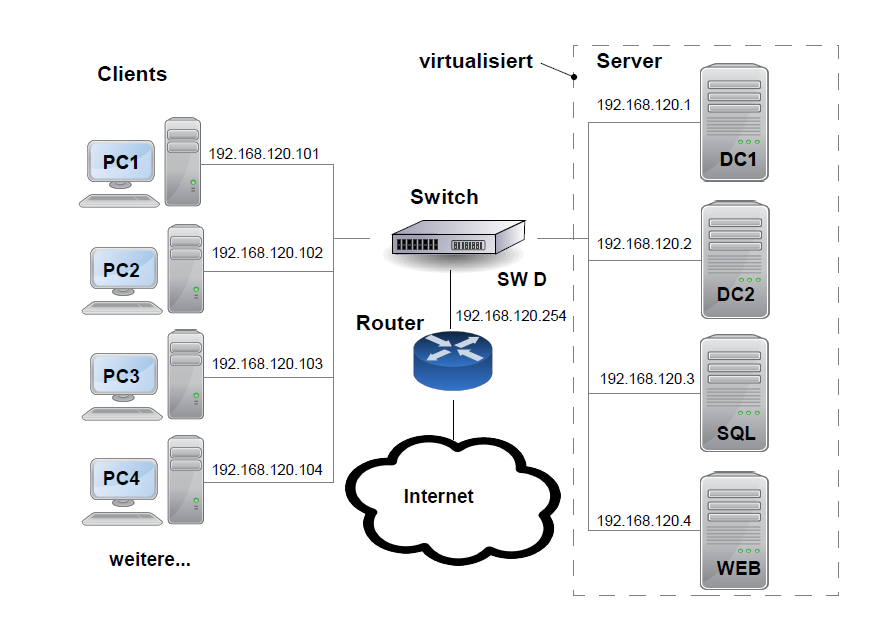
Vergabe der Rolle „DHCP-Server“ unter dem Assistenten „Rollen hinzufügen“. Als statische IP-Adresse wurde folgende Adresse definiert: 192.168.120.1 (DC1)  
Als Netzwerkverbindung wird die IP-Adresse 192.168.120.1 vom Typ IPv4 genommen. Übergeordnete Domäne ist itfox.loc. IPv4-Adresse des bevorzugten DNS-Servers: 192.168.120.1(DC1). Ein alternativer DNS Server wird nicht vergeben. Ein WINS Server wird nicht benötigt, daher wird dieser auch nicht vergeben.

DCHP-Bereiche:

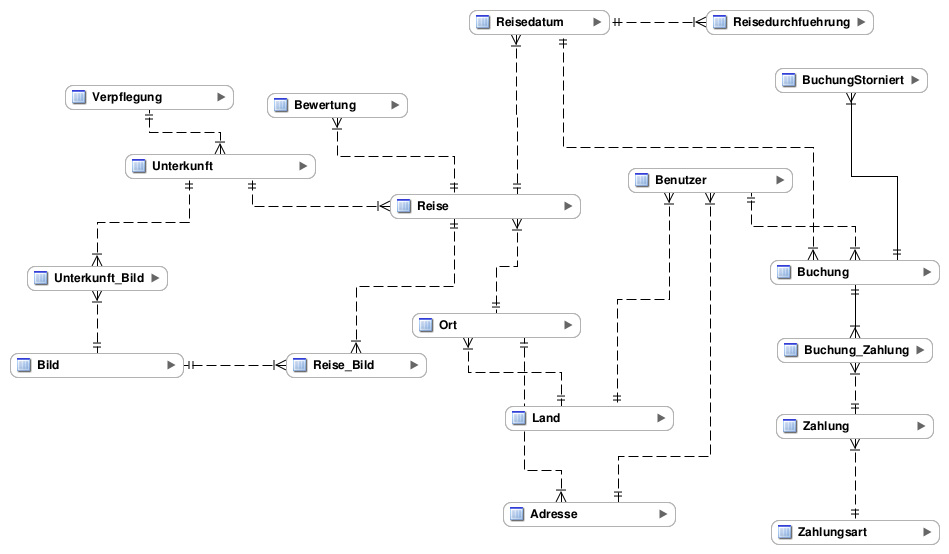
|  |  |
| --- | --- |
| Bereichsnamen | AKTRAUM |
| IP Adressbereich | 192.168.120.101 – 192.168.120.116 |
| Leasedauer | 8 Tage |
| Subnetzmaske | 255.255.255.0 |
| Standardgateway | 192.168.120.254 |



Deaktivierung des statusfreien DHCP-v6-Modus. Einbindung des DHCP Servers in Active Directory zur Autorisierung in AD – Domänendiensten. Autorisierung in unseren AD-Domänendiensten, hierzu werden die aktuellen Anmeldeinformationen verwendet

* 1. Netzwerkdiagramm(Grafik)

Die Server sind virtualisiert. Das Netzwerk ist mittels des Patchpanels D mit dem Switch verbunden. Als Standardgateway ist 192.168.120.254 definiert. EVTL. NOCH MEHR ERKLÄRUNG

1. Datenbankentwurf
   1. Entity Relationship Model (ERM)
   2. Beschreibung
      1. Erfassung der Benutzerdaten

Die Daten eines Benutzers werden einerseits in der Benutzertabelle gespeichert, wobei die Adresse über Beziehungen in die Tabellen Adresse, Land, Ort eingetragen wird. Die direkte Beziehung zur Landtabelle dient zur Erfassung der Staatsbürgerschaft.

* + 1. Speicherung der Buchungsdaten

Während einer Buchung werden die Buchungsdaten in die Tabelle Buchung eingetragen, wobei jede Buchung einem Benutzer zugeordnet werden kann. Die Zahlungsdaten werden in einer eigenen Tabelle gespeichert, die wiederum auf die jeweilige Zahlungsart verweist (bspw. Überweisung, Visa, etc.). Die Zuordnung, welche Buchung mit welcher Zahlung in Beziehung steht, geschieht über eine Zwischentabelle (Buchung\_Zahlung).  
Sollte eine Buchung storniert werden, wird die ID der Buchung in die Tabelle Buchung\_Storniert eingetragen.

* + 1. Erfassung der Reisen

Jede Reise besteht aus Titel, Bezeichnung, Preis und der Beziehung zur jeweiligen Unterkunft mit deren Verpflegung. Die Bewertung durch die Benutzer wird in einer eigenen Tabelle gespeichert. Da Reisen zu unterschiedlichen Zeitpunkten angeboten werden, wird dies über die Reisedatum Tabelle realisiert. Hier wird jede Reise pro Datum einmal eingetragen. Die Anzahl der Plätze auf einer Reise zu einem Datum wird über die Anzahl der eingetragenen Datensätze in der Tabelle Reisedurchführung realisiert.

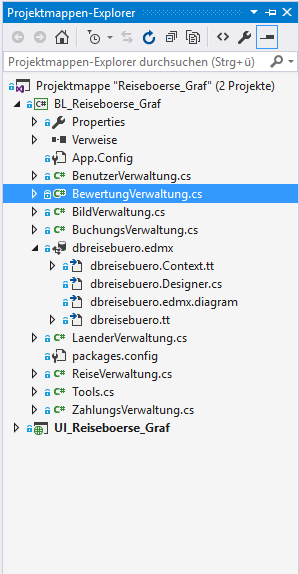
* + 1. Speicherung von Bild

Zu jede Reise, sowie zu jeder Unterkunft gibt es mehrere Bilder. Diese werden in einer Bildtabelle gespeichert, die Zuordnung erfolgt mittels zwei Zwischentabellen (Reise\_Bild und Unterkunft\_Bild).

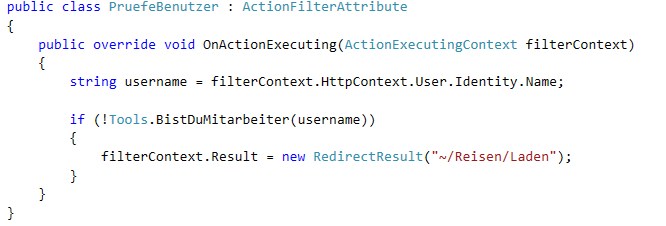
1. Aufbau der Applikation
   1. Model View Controller

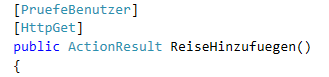
Die Webanwendung wurde mittels dem Entwurfsmuster Model – View – Controller realisiert.   
[Asp.net 🡪 1-2 Sätze]

* 1. Datenbankzugriff
     1. Entity Framework

Die Anbindung der Datenbank im Projekt sowie der Zugriff darauf erfolgten mittels Entity Framework. Aus Gründen der Wiederverwendbarkeit von Code sowie der besseren Strukturierung wurden diese Teile der Anwendung in eine eigene Klassenbibliothek (*BL\_Reiseboerse\_Graf*) ausgelagert.

* 1. Authentifizierung und Autorisierung

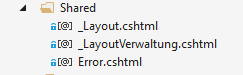
Die Unterscheidung zwischen Mitarbeiter und Kunden erfolgt in der Datenbank mittels Bitfeld. Bei der Anmeldung wird der Benutzer – unabhängig ob Kunde oder Mitarbeiter – anhand seiner Email und seines Passworts authentifiziert. Ob der aktuelle Benutzer Mitarbeiter ist wird mit einer Methode in der Klasse Tools der Businesslogik abgefragt. Die Autorisierung der Mitarbeiter erfolgt mittels eines benutzerdefinierten Action Filters.  
  
Die Anwendung bei den einzelnen Actionmethoden, die die Rechte eines Mitarbeiters erfordern erfolgt über das erstellte Attribut.



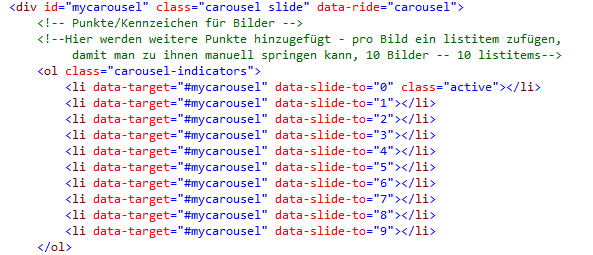
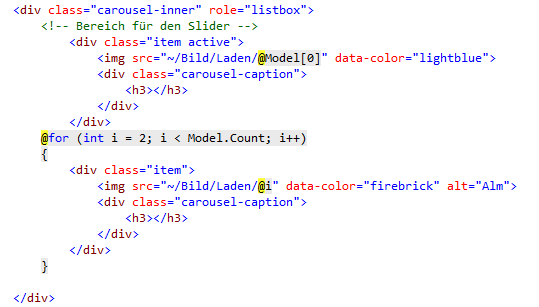
* 1. Grafische Gestaltung
     1. Bootstrap

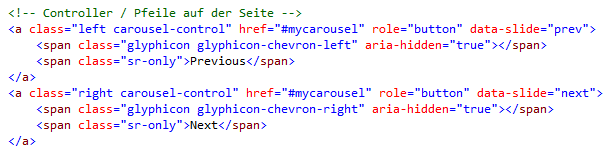
Bootstrap wurde verwendet, um in künftigen Versionen die Software einfacher auch für mobile Plattformen zugänglich zu machen.

* + 1. Geteilte Vorlagen

Sowohl für die Oberflächen für die Kunden, als auch für Verwaltung durch die Mitarbeiter, wurden geteilte Vorlagen erstellt, um ein einheitliches Aussehen zu gewährleisten.

* + 1. Bildslider auf der Startseite (Home/Index)

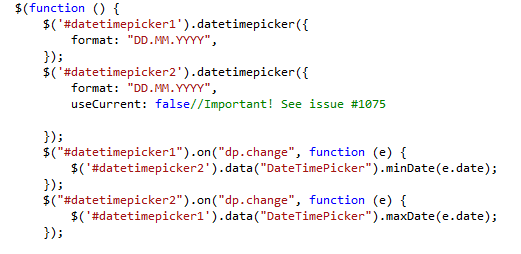
Quelle: <http://codepen.io/SitePoint/pen/ZbGwqe>   
  
Über die Klasse *carousel* wird der Slider dargestellt. Mit <li data-target=““> wird ein neuer Aufzählungspunkt für ein Bild hinzugefügt.  
  
Die Liste aller Bilder aus der Datenbank wird als Model an die View übergeben und dann in den Slider integriert.  
Des Weiteren wurden Pfeile zum Navigieren zwischen den Bildern hinzugefügt werden.



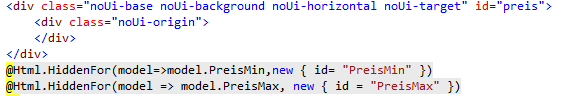
Mit *interval: 3400* wird festgelegt , dass das nächste Bild nach 3,4 Sekunden angezeigt werden soll.

*var $wHeight = $(window).height()* definiert, dass *$wHeight* die Höhe vom Fenster ausnutzen soll.  
*$item.height($wHeight)* definiert, dass *$wHeight* als Höhe jedes Elements verwendet werden soll.

* + 1. DateTimePicker

Quelle: <https://eonasdan.github.io/bootstrap-datetimepicker/>   
  


* + 1. Preisslider bei Reise filtern

Quelle: http://refreshless.com/nouislider/

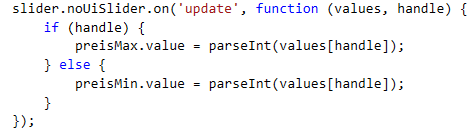
*noUiSlider* ruft die Methode *create* auf um einen Slider zu erstellen  
Folgende Parameter werden gesetzt:

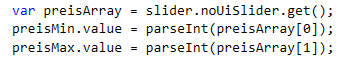
* *start*: Werte an denen die Handler starten
* *range*: Spanne an Werten
* *connect*: true (farbiger Balken als ausgewählte Spanne)
* *pips*: Skala anzeigen (mode und density: Durchsichtigkeit??)
* *step*: in 200er Stufen
* *margin*: zum Rand (man kann nicht weniger als 200 als Spanne auswählen)



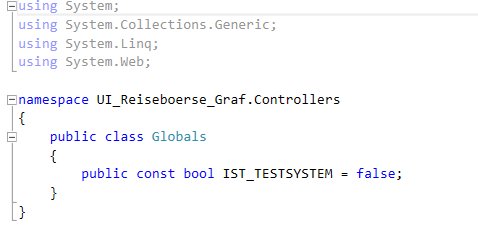
Die Html Elemente, auf denen die Werte gespeichert werden sollen, um diese an die Action zu schicken, werden den Variablen zugewiesen.

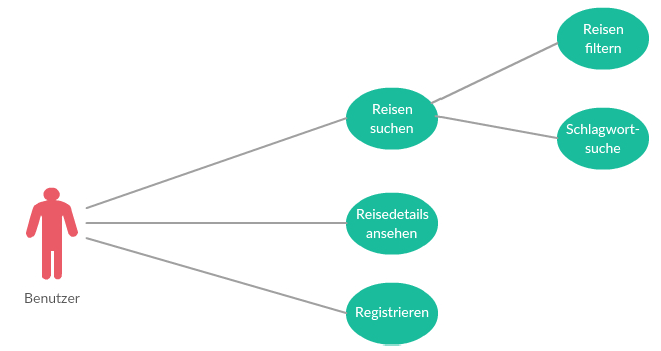
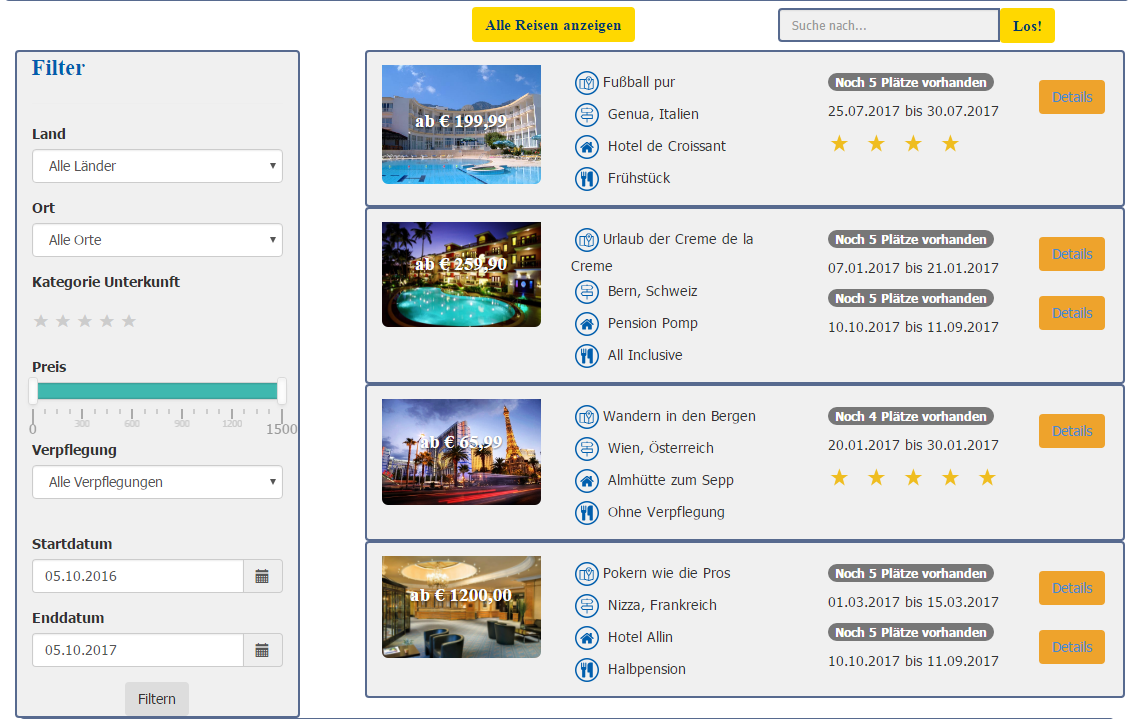


Beim Verändern der Handler im Slider wird der Wert automatisch geändert und den Variablen *preisMin* und *preisMax* zugewiesen.

Dann werden die aktuellen Werte ausgelesen, in *int* umgewandelt und auf dem *value* Attribut zugewiesen.

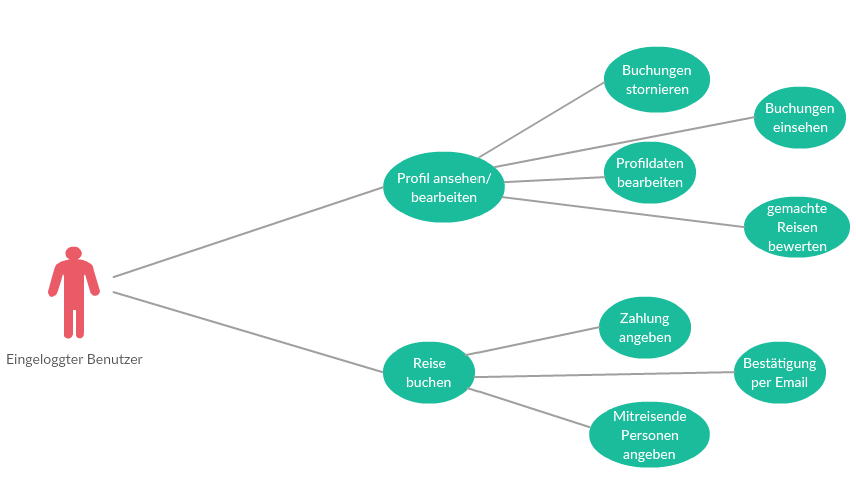
* 1. Unterscheidung zwischen Test- und Echtsystem

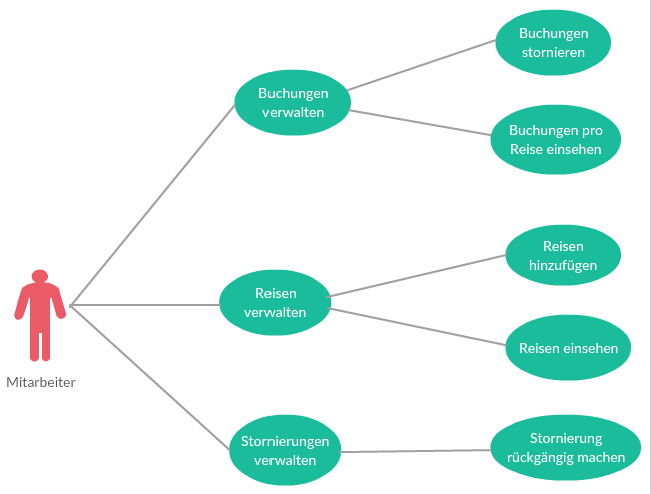
Um beim Programmieren nicht von der Datenbank abhängig zu sein, wurde im ersten Sprint eine Unterscheidung zwischen Testsystem (mit hartcodierten Werten) und Echtsystem (Datenbankzugriff) eingeführt. Diese wurde über die Klasse *Globals.cs* realisiert.

1. Usecases Benutzer
   1. Diagramm
   2. Reisen suchen und Reisen filtern

Die Anzeige der Reisen wird aus der Datenbank geladen, allerdings werden nur die Reisen angezeigt, die aktuell angeboten werden und die noch über Restplätze verfügen. Diese Reisen lassen sich weiter eingrenzen, sowohl über eine Filtermaske (1) mithilfe bestimmter Kriterien als auch mittels Schlagworten (2). Die Suche der Schlagworte beschränkt sich dabei auf die Beschreibung der jeweiligen Reise. Über den Button „Alle Reisen anzeigen“ (3) lassen sich wieder alle Reisen anzeigen.

* 1. Reisedetails ansehen
  2. Registrieren

1. Usecases Registrierter Benutzer
   1. Diagramm
   2. Einloggen
2. Usecases Eingeloggter Benutzer
   1. Diagramm
   2. Profil ansehen/bearbeiten
   3. Reise buchen
3. Usecases Mitarbeiter
   1. Diagramm



* 1. Buchungen stornieren
  2. Stornierungen rückgängig machen