

Hotelrezeption

Lehrziele:

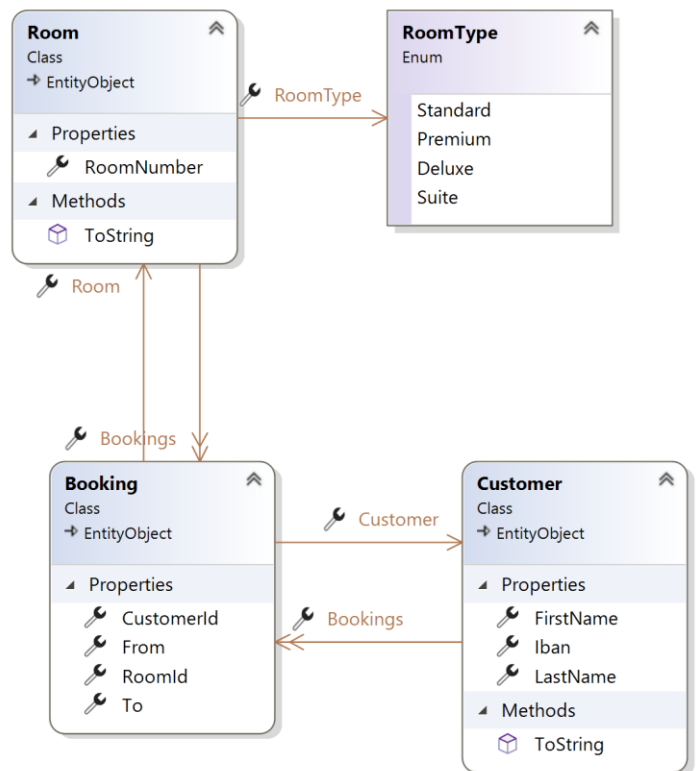
- Generieren von Testdaten mit Bogus
- WPF MVVM
- Webanwendungen mit Razor Pages

Aufgabenstellung

- Sie sollen für die **Rezeption** eines Hotels eine **WPF-Anwendung** schreiben, die das Verbuchen von Anreisen und Abreisen von Hotelgästen ermöglicht.
- Das Hotel-Management hingegen benötigt Zugriff auf sämtliche Buchungen und Kundendaten. Im ersten Prototyp sollen Sie daher in einer **RazorPages-Webanwendung** die Möglichkeit schaffen,
 - Kunden zu suchen, Kundendaten zu ändern sowie neue Kunden anzulegen
 - Alle Zimmerbuchungen von Kunden sowie deren Rechnungsbeträge einzusehen.

Datenmodell

- **Room:**
 - Das Hotel bietet vier unterschiedliche Zimmerkategorien (**RoomType**) an: Standard, Premium, Deluxe und Suite.
 - Die Zimmerbezeichnungen (**RoomNumber**) sind sehr einfach gehalten: Sie starten mit „R-„ und enden mit einer pro Stockwerk fortlaufenden Nummer.
- **Customer:**
 - Zur Vereinfachung können Sie davon ausgehen, dass sämtliche Hotelgäste bereits namentlich und mit Kontonummer in Ihrem System erfasst sind (siehe Import).
- **Booking:**
 - Bei der Ankunft des Gastes wird eine Zimmerbuchung für das gewünschte Zimmer angelegt => dabei wird das **From**-Datum erfasst (heutiges Datum), das Ende-Datum (**To**) der Buchung bleibt noch offen (keine Vorreservierungen!).
 - Erst bei der Abreise des Gastes wird das **To**-Datum erfasst und somit die Buchung abgeschlossen. Man kann jedoch auch schon am Tag vor der Abreise auschecken.



Die Entities und Repositories sind bereits angelegt. Machen Sie sich mit dem Datenmodell vertraut!

Testdaten-Import

In der **ImportConsole**-App wird bereits Bogus als Testdatengenerator für eine festgelegte Anzahl von Zimmern (Rooms) und Hotelgästen (Customers) verwendet.

Um ausreichend Buchungsdaten für Testzwecke zur Verfügung zu haben, müssen auch noch für alle Zimmer abgeschlossene Buchungen in der Vergangenheit erstellt werden, für einige Zimmer soll es auch aktuelle (noch offene) Buchungen geben.

Dabei gilt:

- Ein Zimmer kann gleichzeitig nur von einem Gast belegt sein!
- Umgekehrt darf aber ein Gast mehrere Zimmer gleichzeitig buchen – z.B. wenn alle Zimmer für eine Familie auf einen Gast-Namen lauten.

Ihre Aufgabe:

- Speichern Sie alle erzeugten Zimmer und Gäste.
- Generieren Sie zwischen 3 und 10 abgeschlossene Buchungen für jedes Zimmer, sowie für ca. 10 Zimmer aktuelle (noch offene Buchungen).

```
Datenbank migrieren
```

```
Daten generieren
```

```
Daten speichern
```

```
40 Hotelgäste, 20 Zimmer, 146 Zimmerbuchungen wurden aus DB gelesen.  
Davon sind 13 Buchung offen.
```

Hinweis:

- Eine beigelegte Batch-Datei erledigt das Importieren von korrekten Testdaten, sodass Sie diesen Punkt auch überspringen könnten.

WPF: MainWindow**Anzeige aller Zimmer**

Das WPF-Hauptfenster zeigt alle Zimmer des Hotels (initial nach Zimmernummern sortiert) in einem *ItemsControl* (*scrollable Grid*) an. Für jedes Zimmer wird dargestellt:

- Zimmernummer
- Zimmertyp
- Ist das Zimmer derzeit (= am heutigen Tag) belegt oder frei?
- Wenn das Zimmer derzeit belegt ist, seit wann ist es belegt?

Zimmer-Nr.	Zimmertyp	Frei	Belegt seit	Bis
R-40	Deluxe	<input checked="" type="checkbox"/>		
R-41	Premium	<input checked="" type="checkbox"/>		
R-42	Premium	<input checked="" type="checkbox"/>		
R-43	Standard	<input type="checkbox"/>	29.01.2022	
R-44	Suite	<input type="checkbox"/>	27.01.2022	
R-45	Standard	<input checked="" type="checkbox"/>		
R-46	Deluxe	<input checked="" type="checkbox"/>		
R-47	Standard	<input type="checkbox"/>	25.01.2022	
R-48	Standard	<input checked="" type="checkbox"/>		
R-49	Standard	<input type="checkbox"/>	28.01.2022	
R-50	Deluxe	<input checked="" type="checkbox"/>		

Ihre Aufgaben:

- Entwerfen Sie eine geeignete Datenstruktur (*DataTransferObject*) für das DataBinding zwischen *ViewModel* und dem *MainWindow* und implementieren Sie den Datenbankzugriff im *ViewModel* und dem betreffenden *Repository*.
- Vervollständigen Sie die Anzeige des DataGrids und konfigurieren Sie das DataBinding.

Filtern der Zimmer

Fügen Sie im Layout für das MainWindow eine Filterzeile hinzu, die es ermöglicht, die angezeigten Zimmer anhand einer Nummernsuche bzw. anhand des Zimmertyps einzuschränken.

Suche Zimmernummer:
 Zimmertyp:

Zimmer-Nr.	Zimmertyp	Frei	Belegt seit	Bis
R-40	Deluxe	<input checked="" type="checkbox"/>		
R-41	Premium	<input checked="" type="checkbox"/>		
R-42	Premium	<input checked="" type="checkbox"/>		
R-43	Standard	<input type="checkbox"/>	29.01.2022	

Bei der Zimmernummer reicht es, eine oder mehrere Ziffern einzugeben, um die Anzeige der Zimmer einzuschränken.

Beispiel: Wird die Ziffer „5“ eingegeben, werden nur mehr Zimmer im 5. Stockwerk angezeigt. Erst bei Eingabe von „50“ ist dann nur mehr „R-50“ gelistet.

Suche Zimmernummer: Zimmertyp: Alle

Zimmer-Nr.	Zimmertyp	Frei	Belegt seit	Bis
R-50	Deluxe	<input checked="" type="checkbox"/>		
R-51	Suite	<input type="checkbox"/>	26.01.2022	
R-52	Standard	<input checked="" type="checkbox"/>		
R-53	Suite	<input checked="" type="checkbox"/>		
R-54	Deluxe	<input checked="" type="checkbox"/>		
R-55	Suite	<input checked="" type="checkbox"/>		
R-56	Standard	<input checked="" type="checkbox"/>		
R-57	Premium	<input type="checkbox"/>	29.01.2022	
R-58	Deluxe	<input type="checkbox"/>	26.01.2022	
R-59	Suite	<input checked="" type="checkbox"/>		

Die Auswahl eines speziellen Zimmertyps ist mit dem Nummerfilter kombiniert, Beispiel:

Suche Zimmernummer: Zimmertyp: Deluxe

Zimmer-Nr.	Zimmertyp	Frei	Belegt seit	Bis
R-50	Deluxe	<input checked="" type="checkbox"/>		
R-54	Deluxe	<input checked="" type="checkbox"/>		
R-58	Deluxe	<input type="checkbox"/>	26.01.2022	

Button-Belegung

Sobald ein Zimmer im DataGrid ausgewählt wurde, kann einer der beiden folgenden Buttons betätigt werden:

- Ist das ausgewählte Zimmer frei, kann eine **Anreise** gebucht werden (Button „Anreise“ ist aktiv).
- Ist das ausgewählte Zimmer belegt, kann eine **Abreise** gebucht werden (Button „Abreise“ ist aktiv).

Suche Zimmernummer: Zimmertyp: Deluxe

Zimmer-Nr.	Zimmertyp	Frei	Belegt seit	Bis
R-50	Deluxe	<input checked="" type="checkbox"/>		
R-54	Deluxe	<input checked="" type="checkbox"/>		
R-58	Deluxe	<input type="checkbox"/>	26.01.2022	

CheckoutBookingWindow: Abreise Buchen

Soll nun wie im Beispiel oben die Abreise für das derzeit belegte Zimmer R-58 gebucht werden, so werden zuvor die Einzelheiten der Buchung in folgendem Dialog dargestellt:

Abreise

Zimmertyp: Deluxe

Kunde: Rohrer Kerim (Iban: AT382040954640080045)

Ankunft: Mittwoch, 26. Januar 2022

Abreise: 30.01.2022

Aufenthaltsdauer: 4 Nächte

Abbrechen Abreise speichern

Das Abreisedatum ist auf das heutige Datum voreingestellt, kann aber auf ein beliebiges Datum (**frühestens jedoch einen Tag nach Ankunftsdatum**) verändert werden. Dabei wird sofort die entsprechende Anzahl von Übernachtungen als Aufenthaltsdauer angezeigt.

Erst beim Betätigen des Buttons „Abreise speichern“ wird das eingestellte Abreisedatum zur Buchung in die Datenbank gespeichert und somit die Verfügbarkeit des Zimmers geändert.

Hinweise:

- Die Aufenthaltsdauer wird nur zur Darstellung berechnet und nicht in die DB gespeichert.
- Achten Sie darauf, dass das MainWindow nach Schließen des modalen Checkout-Dialogs aktualisiert wird! Beispiel:

R-58	Deluxe	<input type="checkbox"/>	26.01.2022	03.02.2022
------	--------	--------------------------	------------	------------

Anreise Buchen

Der modale Dialog zum Buchen einer Anreise soll wie folgt aussehen: Zimmernummer, Typ und Anreisedatum (heutiges Datum) sind fix vorgegeben und eine ComboBox für die Auswahl eines Gastes ist noch mit keinem Wert befüllt. Solange kein Gast ausgewählt wurde, ist auch der „Anreise speichern“ Button nicht aktiv.

CheckinBo

Zimmertyp: Premium

Anreisedatum: 06.02.2022

Gast:

Abbrechen Anreise speichern

Razor Webanwendung: Gästeliste

In der Gästeverwaltung zeigt die Index-Seite eine alphabetisch sortierte Liste von Hotelgästen mit deren Gesamtanzahl an Buchungen an.

Eine Filterzeile ermöglicht die Textsuche nach Kundennamen (case-insensitive) sowie die Einschränkung, ob nur Kunden mit einer aktuellen Buchung gezeigt werden sollen.

Beispiel:

- „lo“ wird in 4 Namen gefunden

WebApp Home Gäste

Kundenübersicht

Filter Name: Nur mit aktuellen Buchungen: ☐ [Aktualisieren](#)

Nachname	Vorname	Anzahl Buchungen	
Floder	Giuseppe	5	Bearbeiten
Logsch	Nicolas	2	Bearbeiten
Lohse	Amelia	4	Bearbeiten
Loy	Jenna	2	Bearbeiten

- Davon haben jedoch nur 2 auch eine aktuelle Buchung offen:

WebApp Home Gäste

Kundenübersicht

Filter Name: Nur mit aktuellen Buchungen: ☒ [Aktualisieren](#)

Nachname	Vorname	Anzahl Buchungen	
Floder	Giuseppe	5	Bearbeiten
Loy	Jenna	2	Bearbeiten

Kunden-Details bearbeiten

Clickt man in der Kundenübersicht auf „Bearbeiten“, gelangt man auf eine Detail-Ansicht des Kunden, in der man die einzelnen Attribute eines Kunden ändern kann.

Beachten Sie dabei, dass alle Attribute verpflichtend sind und die Email-Adresse und die Kreditkartennummer gültig sein müssen!

Zusätzlich wird eine Liste aller Zimmerbuchungen dieses Gastes, sortiert nach Buchungszeitraum, angezeigt.

Bearbeite Giuseppe Floder

Vorname

Nachname

Email-Adresse

Kreditkartennummer

Kreditkartennummer ist ungültig

[Speichern](#)

Bearbeite Giuseppe Floder

Vorname	<input type="text" value="Giuseppe"/>
Nachname	<input type="text" value="Floder"/>
Email-Adresse	<input type="text" value="Giuseppe_Floder@gmail.com"/>
Kreditkartennummer	<input type="text" value="6767-2843-2814-2450"/>
<input type="button" value="Speichern"/>	

Zimmerbuchungen

Von	Bis	Zimmer	Zimmertyp	Anzahl Übernachtungen
03.02.2022		R-53	Premium	
31.10.2020	06.11.2020	R-45	Premium	6
31.08.2020	02.09.2020	R-45	Premium	2
14.06.2020	22.06.2020	R-58	Standard	8
20.02.2020	22.02.2020	R-51	Premium	2

[Zurück](#)

Kreditkartenprüfung (optional!)

Zur Prüfung einer Kreditkartennummer anhand des Luhn-Algorithmus gibt es ein UnitTest-Projekt, das bei der Entwicklung des Prüfalgorithmus hilft.

Hier ist die Beschreibung des Prüfalgorithmus nach Luhn:

Die Kreditkartennummer ist durch eine Prüfziffer (letzte Stelle, die bei der Berechnung nicht mitverwendet wird) wie folgt gegen Fehleingaben abgesichert:

- Aus der Nummer wird die Prüfziffer (letzte Stelle) und die Payload (alle Ziffern davor) extrahiert.
- Die Payload wird von rechts beginnend iteriert und jede Ziffer wie folgt behandelt
 - Jede zweite Ziffer (beginnend bei 0, also der ersten Stelle) werden verdoppelt und deren Ziffernsumme wird aufsummiert
 - Die Ziffern an den Stellen mit ungerader Nummer werden aufsummiert
- Die beiden erhaltenen Zahlen werden addiert und die Differenz zur nächsten Zehnerzahl wird ermittelt, die gleich groß oder größer ist als die Summe. Diese Ziffer (0-9) ist die Prüfziffer
- Ist die berechnete Prüfziffer ident mit der Prüfziffer (letzte Stelle Kreditkartennummer), so handelt es sich um eine gültige Kreditkartennummer.

Ein Beispiel für die Kreditkartennummer 2718 2818 2845 8567:

[illegible]

Viel Erfolg!