



**HTL - Perg**

**Höhere Abteilung für Informatik**

**Diplomarbeit**



**HRM – Hotel Room Manager**

Projektteam: Stefan Pilz

Fabian Radinger

Projektbetreuer: Prof. Mag. Rupert Obermüller

In Zusammenarbeit mit Gasthof Schiefer

Betreuer Herr Josef Schiefer

Bearbeitungszeitraum: 14.09.2015 – 08.04.2016



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis.....</b>	<b>2</b>
<b>1 Eidesstattliche Erklärung.....</b>	<b>7</b>
<b>2 Danksagung .....</b>	<b>8</b>
<b>3 Impressum.....</b>	<b>9</b>
<b>4 Einleitung .....</b>	<b>10</b>
4.1 Kurzbeschreibung .....	10
4.2 Abstract .....	11
<b>5 Das Team.....</b>	<b>12</b>
5.1 Stefan Pilz .....	12
5.2 Fabian Radinger .....	12
<b>6 Motivation .....</b>	<b>13</b>
<b>7 Projektdefinition .....</b>	<b>14</b>
7.1 Aufgabenstellung.....	14
7.2 Geschäftsziel.....	14
7.3 Projektziel.....	14
7.4 Projektumfeld .....	15
7.4.1 Schule .....	15
7.4.2 Betreuungslehrer.....	16
7.4.3 Auftraggeber.....	16
<b>8 Entstehung und Planung .....</b>	<b>17</b>
8.1 Vorbereitung.....	17
8.2 Zeitplanung .....	17
8.3 IVM – Matrix.....	19
8.4 Projektstrukturplan.....	19
8.5 Konfigurationsplan.....	21
8.5.1 booking.com .....	22
8.5.2 WuBook.....	22

8.5.3	Internet.....	22
8.5.4	User .....	22
8.5.5	Landgasthof „Zur Zugbrücke“ .....	23
<b>9</b>	<b>Realisierung.....</b>	<b>24</b>
9.1	Überblick Gesamtsystem .....	24
9.1.1	Wubook Funktionen: .....	25
9.2	Von der XML-Struktur zum Datenbankmodell .....	30
9.2.1	Zimmer .....	30
9.2.2	Reservierungen.....	31
9.3	Datenbankmodell .....	32
9.3.1	Benutzer .....	33
9.3.2	Zimmertyp .....	33
9.3.3	Zimmer .....	33
9.3.4	Kunde .....	33
9.3.5	Buchung.....	34
9.3.6	Status.....	34
9.3.7	Sperrung .....	34
9.4	Verbindung mit der Datenbank.....	35
9.5	Darstellung.....	36
9.5.1	Menüleiste.....	36
9.5.2	Navigationsbereich .....	36
9.5.3	Darstellungsframe .....	36
9.6	Funktionalitäten.....	37
9.6.1	Starten der Applikation.....	37
9.6.2	Login .....	38
9.6.3	Verwalten von Kunden .....	38
9.6.4	Eintragung von Buchungen .....	39
9.6.5	Übersicht .....	40
9.6.6	Verwalten von Buchungen.....	42
9.6.7	Rechnungen:.....	44

9.6.8	Schnittstelle .....	45
9.7	Code.....	46
9.7.1	Start der Applikation:.....	46
9.7.2	createRectangles() .....	47
9.7.3	RefreshPage() .....	50
9.7.4	Export CSV .....	50
9.7.5	Lock_rooms().....	51
9.7.6	createBill() .....	51
9.7.7	Delegates.....	52
9.8	Risiken .....	52
9.8.1	Aufwand von Schnittstellenkonfiguration und Datenaustausch zu groß (WuBook) .....	53
9.8.2	Funktionsumfang zu groß .....	53
9.8.3	Qualitätsverlust .....	54
<b>10</b>	<b>Technologien .....</b>	<b>55</b>
10.1	WPF .....	55
10.2	XAML .....	55
10.3	C# (C – Sharp) .....	55
10.4	GIT Hub.....	55
10.5	Dropbox.....	56
10.6	Visual Studio .....	56
10.7	DBDesigner Fork .....	56
10.8	ChannelManager (WuBook).....	56
10.9	Booking.com .....	56
10.10	Datenbank ADO.NET (Access Data Object).....	57
10.10.1	ADO.NET Architektur .....	57
10.10.2	Entity Framework.....	58
10.10.3	Microsoft SQL – Server:.....	58
10.10.4	Datenbankverwaltungssysteme (DBVS) .....	59
10.11	Language Integrated Query (LINQ) .....	60

10.11.1	Lambda Ausdrücke.....	60
10.12	Delegates .....	61
10.13	XML-Technologie .....	61
10.13.1	XML – String:.....	61
10.13.2	Request & Response: .....	62
<b>11</b>	<b>Qualitätssicherung .....</b>	<b>64</b>
11.1	Qualitätsmerkmale .....	64
<b>12</b>	<b>Testen – Inbetriebnahme .....</b>	<b>65</b>
12.1	Einleitung.....	65
12.2	Testfälle .....	66
12.2.1	Reservieren.....	67
12.2.2	Reservierung bearbeiten.....	67
12.2.3	Übersicht anzeigen .....	68
12.2.4	Rechnung erstellen .....	69
12.2.5	Zimmer zuordnen .....	70
12.2.6	Kunden bearbeiten .....	71
12.2.7	Kunden E-Mail Adressen exportieren .....	71
12.2.8	Buchungsliste drucken .....	72
12.2.9	Zimmer sperren .....	73
12.2.10	Reservierung anfordern .....	73
12.2.11	Zimmertypen anfordern.....	73
<b>13</b>	<b>Auszug aus dem Projekthandbuch.....</b>	<b>74</b>
13.1	Meilensteine.....	74
<b>14</b>	<b>Installation .....</b>	<b>75</b>
14.1	Anmeldung bei flexerasoftware.com: .....	75
14.2	Installation:.....	75
14.3	Verwendung des Tools.....	76
<b>15</b>	<b>Zusammenfassung .....</b>	<b>78</b>
15.1	Ergebnis .....	78
15.2	Resümee .....	79

## INHALTSVERZEICHNIS

15.2.1	Stefan Pilz .....	80
15.2.2	Fabian Radinger .....	80
15.3	Aufwandsverteilung .....	81
<b>16</b>	<b>Literatur und Quellen Verzeichnis: .....</b>	<b>82</b>
16.1	Abbildungsverzeichnis .....	82
16.2	Tabellenverzeichnis.....	84
16.3	Literaturverzeichnis .....	84
16.4	Abkürzungsverzeichnis.....	86
<b>17</b>	<b>Im Anhang .....</b>	<b>86</b>

# 1 Eidesstattliche Erklärung

---

Hiermit versichern wir, die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von uns angegebenen Quellen angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Perg, \_\_\_\_\_ Unterschrift \_\_\_\_\_  
(Stefan Pilz)

Perg, \_\_\_\_\_ Unterschrift \_\_\_\_\_  
(Fabian Radinger)

## 2 Danksagung

---

An dieser Stelle möchten wir uns bei allen Personen bedanken, die uns bei der Erstellung der Diplomarbeit zur Seite standen und uns Hilfe boten.

Besonderer Dank gilt unseren Betreuungslehrer Prof. Mag. Rupert Obermüller, der uns die gesamte Projektdauer für technische Fragen zur Seite stand und somit viele Ideen für unsere Arbeit beisteuerte.

Abschließend bedanken wir uns bei allen anderen Lehrkräften, die uns während der Arbeit hilfsbereit unterstützten.

Herzlichen Dank!

# 3 Impressum

---

## **Schule**

- HTL Perg für Informatik

## **Schuljahr**

- 2015/2016

## **Klasse**

- 5AHIF

## **Projekttitle**

- HRM
- Hotel Room Manager

## **Projektteam**

- Stefan Pilz
- Fabian Radinger

## **Beratungslehrer**

- Prof. Mag. Rupert Obermüller

# 4 Einleitung

---

## 4.1 Kurzbeschreibung

Der Gasthof „Zur Zugbrücke“ Schiefer in Bad Kreuzen führt neben dem „Wirtshausbetrieb“, einen Hotelbetrieb mit 23 Zimmern. Diese werden entweder mittels booking.com oder telefonisch gebucht. Aufgrund der verschiedenen Buchungsarten, kam es zu Terminüberschneidungen. Teilweise sind Zimmer telefonisch gebucht worden, die aber bereits auch in booking.com reserviert worden sind. Dies lag daran, dass die Rezeptionistin nicht immer die aktuellste Liste der Online Buchungen zur Verfügung hatte.

Daher wollte Herr Schiefer eine einfache Lösung um dieses Problem zu beseitigen. Eine Desktop-Applikation die mit booking.com direkt verbunden ist, soll diese Aufgabe mit einer einfachen Bedienung lösen. Grundsätzlich wird das Programm von der Rezeption verwendet und gesteuert.

Neue Reservierungen können durch eine einfache Bedienung eingetragen, sowie bearbeitet werden. Dazu ist auch eine Kundenverwaltung notwendig, das heißt, Kunden, sollen erstellt, bearbeitet und gelöscht werden können.

Die getätigten Reservierungen sollen anschließend in einer Kalenderübersicht anschaulich dargestellt werden.

Wünscht der Kunde eine Rechnung so ist es dem Auftraggeber möglich, diese über unsere Applikation (HRM) zu erstellen und zu drucken.

Die Aufgabe von HRM (Hotel Room Manager) ist Terminüberschneidungen zu vermeiden, welche zuvor aufgetreten sind.

Die Einspeisung der Buchungsportale (booking.com) erfolgt mithilfe eines externen Unternehmens. Der Name der externen Schnittstelle lautet WuBook (<https://wubook.net/>) und die Daten werden durch JSON oder XML übertragen.

Die Einzigartigkeit unseres Projektes wird durch die Kommunikation und Zusammenarbeit mit verschiedenen Buchungsportalen mit einer externen Schnittstelle erfüllt.

## 4.2 Abstract

The inn “Zur Zugbrücke“ Schiefer in Bad Kreuzen has 23 rooms, which can be booked via booking.com or by telephone. As you can book online as well as by telephone there were some clashes of dates. Sometimes the receptionist hasn't the actual list of online booked rooms. That's why some rooms were also booked by telephone at the same date.

Therefore Mister Schiefer needed an easy solution to solve the problem. He decided to get a desktop application, which stays in a permanent connection with booking.com and is easy to handle. The programme is only used on the reception computer, which is why the database is also installed on the same computer.

The main task of our Software is to avoid schedule conflicts. Our goal is that there will not be rooms in the future, which are booked twice. If someone books online via booking.com, our database should get a message and save it. To notice that someone booked online we needed a channel manager, which works together with booking.com and manages the online bookings. The name of the company is WuBook. Furthermore, the receptionist should be able to see a short overview of the rooms and print a list of the booked ones. Moreover, the user can manage their booking and customers which means he is able to create, update and delete them.

There should also be a function in our Software, which allows the user to create an Excel Sheet that contains the bill and print it for the costumer if he or she insists on it.

The uniqueness of the project is given by the communication with the booking website booking.com.

# 5 Das Team

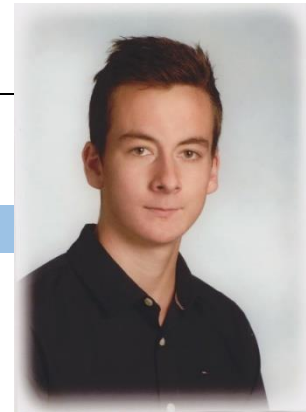


Abb. 1: Herr Pilz Stefan

## 5.1 Stefan Pilz

### Persönliche Daten

Name: Stefan Christian Pilz  
Geburtsdaten: Amstetten, 05.Jänner 1997  
Straße: Bad Kreuzen 173  
Ort: Bad Kreuzen  
E-Mail: pilzstefan97@gmail.com

### Erfahrungen

4 Jahre Volksschule in Bad Kreuzen  
4 Jahre Hauptschule in Bad Kreuzen  
seit Sep. 2011 bis Juli 2016 HTL für Informatik in Perg  
  
August 2013 Praktikum bei der Firma Helvetia Versicherungen AG  
Juli 2014 Praktikum bei der Firma Bilfinger

### Kenntnisse und Fähigkeiten

Sprachen: Deutsch (Muttersprache)  
Englisch (fließend)  
EDV-Kenntnisse: Java, C#, SQL, PLSQL, Microsoft Office, SAP

## 5.2 Fabian Radinger



Abb. 4: Herr Radinger Fabian

### Persönliche Daten

Name: Fabian Heribert Radinger  
Geburtsdaten: Dorfstetten, 12.Juli 1997  
Straße: Wimbergeramt 38  
Ort: Dorfstetten  
E-Mail: fabian.radinger@gmail.com

### Erfahrungen

4 Jahre Volksschule in Dorfstetten  
4 Jahre Hauptschule in Waldhausen  
seit Sep. 2011 bis Juli 2016 HTL für Informatik in Perg  
  
August 2013 & 2014 Praktikum bei der Firma Synthesa

### Kenntnisse und Fähigkeiten

Sprachen: Deutsch (Muttersprache)  
Englisch (fließend)  
EDV-Kenntnisse: Java, C#, PHP, SQL, PLSQL, Microsoft Office, SAP

## 6 Motivation

---

Für das Personal an der Rezeption ist es meistens sehr mühsam, wenn ein Kunde ein Zimmer telefonisch buchen möchte, da auch die Möglichkeit besteht via booking.com zu buchen. So hat die Rezeptionistin immer zuerst die Aufgabe, die Verfügbarkeit des Zimmers in booking.com im gewünschten Zeitraum zu prüfen und dieses dann als gebucht zu kennzeichnen. Bei einer Unachtsamkeit führt dies schnell zu einer Terminüberschneidung zweier Buchungen im selben Zeitraum.

Der Fokus unserer Diplomarbeit liegt darauf, dass Terminüberschneidungen in Zukunft vermieden werden können. Der Benutzer verfügt über einen Übersichtskalender aller Buchungen, telefonisch und online. Zusätzlich kann eine Buchungsliste generiert und ausgedruckt werden. Die implementierte Kundenverwaltung erleichtert es dem Auftraggeber von Zeit zu Zeit seine Angebote zu versenden. Wenn ein Kunde „auscheckt“ und eine Rechnung wünscht muss der Benutzer nur die gewünschte Buchung wählen und eine Excel Rechnung mit den gespeicherten Daten wird erstellt.

# 7 Projektdefinition

---

## 7.1 Aufgabenstellung

HRM ist eine Desktop-Applikation, die dem Auftraggeber eine einfache Verwaltung seiner Hotelzimmer ermöglichen soll.

Zuvor kam es aufgrund telefonischer und online durchgeführter Reservierungen zu Terminüberschneidungen. Die beschriebenen online Reservierungen wurden über das Buchungsportal booking.com getätigt. Daher wird eine Lösung benötigt, welche online Buchungen automatisch erfasst und nebenbei den Angestellten ermöglicht telefonische Buchungen einzutragen. Neben den Buchungen ist auch eine Verwaltung der Kunden notwendig, damit Angebote gelegentlich an bestehende Kunden versendet werden können.

Somit kann eine einfache Übersicht über telefonische, wie auch online Buchungen dargestellt werden. Außerdem kann es nicht mehr vorkommen, dass ein Zimmer am gleichen Zeitpunkt zweimal gebucht wird.

## 7.2 Geschäftsziel

Dem Auftraggeber soll eine einfache Verwaltung seiner Zimmer ermöglicht werden. Somit soll es auch zu keinen Terminüberschneidungen mehr kommen egal ob ein Kunde telefonisch oder über das Buchungsportal booking.com bucht. Zusätzlich soll auch eine Kundenverwaltung implementiert sein. Um den Receptionist beim Check-Out zu unterstützen, kann eine Excel Rechnung automatisch erstellt werden.

## 7.3 Projektziel

Der Anwender soll über eine benutzerfreundliche Desktop Applikation, Buchungen eintragen sowie verwalten können. Um Terminüberschneidungen zu vermeiden, soll die Software telefonische sowie auch online Buchungen des Buchungsportal booking.com zusammenfassen.

## 7.4 Projektumfeld

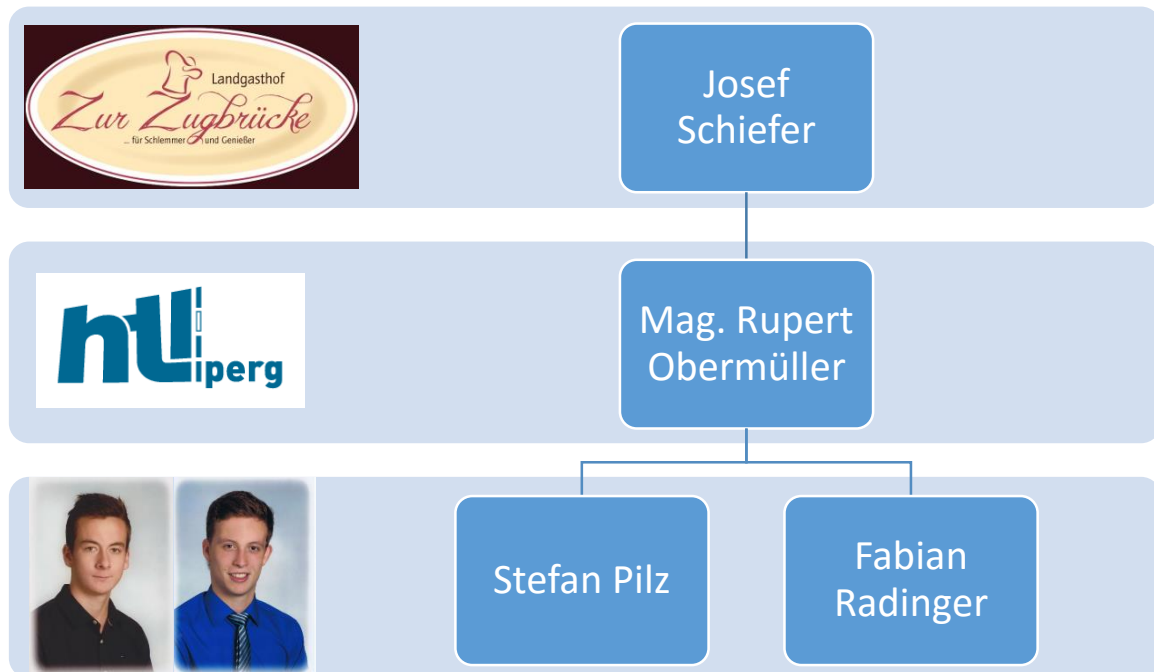


Abb. 7 Projektumfeld, Hierarchie

Der Auftraggeber dieser Diplomarbeit ist, der Landgasthof „Zur Zugbrücke“, wo uns Herr Josef Schiefer als Ansprechpartner zur Seite stand.

Da diese Diplomarbeit im Rahmen der Reife- & Diplomarbeit an der HTL Perg durchgeführt wird, steht uns Herr Prof. Mag. Rupert Obermüller als Betreuungslehrer zur Seite.

Dieses Projekt wird von Stefan Pilz und Fabian Radinger durchgeführt, die auch zugleich die Verfasser dieser Arbeit sind.

### 7.4.1 Schule

Diese Diplomarbeit wurde im Rahmen der Reife und Diplomprüfung an der HTL in Perg, Schwerpunkt Informatik, im Schuljahr 2015/2016 durchgeführt.

Kontakt	
<b>Name:</b>	HTBLA Perg
<b>Straße:</b>	Machlandstraße 48
<b>Adresse:</b>	4320 Perg
<b>Telefon:</b>	07262 539 26
<b>Webseite:</b>	<a href="http://www.htl-perg.ac.at">www.htl-perg.ac.at</a>

Tabelle 1 Kontaktdaten HTL Perg



Abb. 8: HTL – Perg (vgl. [HTL Perg])

## 7.4.2 Betreuungslehrer

Unsere Betreuungslehrkraft ist Herr Prof. Mag. Rupert Obermüller der ebenfalls den Gegenstand „Programmieren und Software-Engineering“ seit der 4. Schulstufe leitet. Herrn Obermüller gelang es immer wieder uns mit seiner hilfsbereiten Art anzutreiben um das Bestmögliche zu geben.

Persönliche Daten	
<b>Name</b>	Prof. Mag. Rupert Obermüller
<b>E-Mail:</b>	r.obermueller@htl-perg.ac.at

Tabelle 2 Kontaktdaten Rupert Obermüller



Abb. 9: Herr Prof. Mag. Rupert Obermüller (vgl. [LEHRER, HTL Perg])

## 7.4.3 Auftraggeber



Abb. 10: Logo „Zur Zugbrücke“ (vgl. [SCHIEFER, Landgasthof])

Der Auftraggeber dieser Diplomarbeit ist der Landgasthof „Zur Zugbrücke“ mit Sitz in Bad Kreuzen. Unser Ansprechpartner ist Herr Josef Schiefer. Der Gasthof verfügt neben dem normalen „Wirtshausbetrieb“ noch über 23 geräumige Komfortzimmer für mehr als 50 Gäste.

Kontakt	
<b>Firmenname:</b>	Gasthof zur Zugbrücke "Schiefer"
<b>Besitzer:</b>	Josef Schiefer
<b>Straße:</b>	Bad Kreuzen 9
<b>Adresse:</b>	4362 Bad Kreuzen
<b>Telefon:</b>	07266 6212
<b>E-Mail:</b>	info@gasthof-schiefer.at
<b>Webseite:</b>	www.gasthof-schiefer.at

Tabelle 3 Kontaktdaten Gasthof Schiefer



Abb. 11: Herr Josef Schiefer (vgl. [SCHIEFER, Landgasthof])

# 8 Entstehung und Planung

---

## 8.1 Vorbereitung

Die Idee für diese Diplomarbeit stammt von Herrn Josef Schiefer der Geschäftsführer vom Gasthof „Zur Zugbrücke“ und zugleich Stefan Pilz Onkel ist. Wir entschieden uns bereits in der 4. Klasse für diese Diplomarbeit.

Aufgrund der terminlichen Schwierigkeiten im Sommer begannen wir erst Ende August mit der Planung für dieses Projekt bzw. Diplomarbeit. Vom Gegenstand Projektentwicklung wussten wir, dass eine strukturiert durchgeführte Planungsphase der Grundstein für einen erfolgreichen Abschlusses eines Projekts ist.

Als dies Abgeschlossen war begannen wir auch schon mit der Arbeit und verteilten die einzelnen Tasks. Da der Auftraggeber eine Desktopapplikation wünschte, fiel uns die Wahl eine WPF Anwendung zu entwickeln nicht schwer. Da wir die Entwicklung solcher Anwendungen im Programmierunterricht erlernten. Fabian Radinger musste sich nebenbei mit dem Datenaustausch der Schnittstelle welcher mit JSON und XML arbeitet vertraut machen.

## 8.2 Zeitplanung

Die Zeitplanung ist eine der wichtigsten Instrumente in der Projektentwicklung, deshalb sollte man sich dafür ein sorgfältiges Management überlegen. Dabei können Technische Hilfsmittel vom Vorteil sein, wie zum Beispiel „GanttProject“, welches uns seit Beginn des Gegenstandes Projektentwicklung begleitet.

Die Grafik auf der nächsten Seite zeigt unseren Zeitplan. Diese Abbildung beinhaltet alle Meilensteine. Dieser Plan erleichterte uns die Fertigstellung unserer Diplomarbeit. Die Aufgaben, welche pro Meilenstein notwendig waren, haben wir uns in regelmäßigen Meetings zugeteilt.

Die Meilensteine sind im unten gezeigten Bild mit Rauten gekennzeichnet.

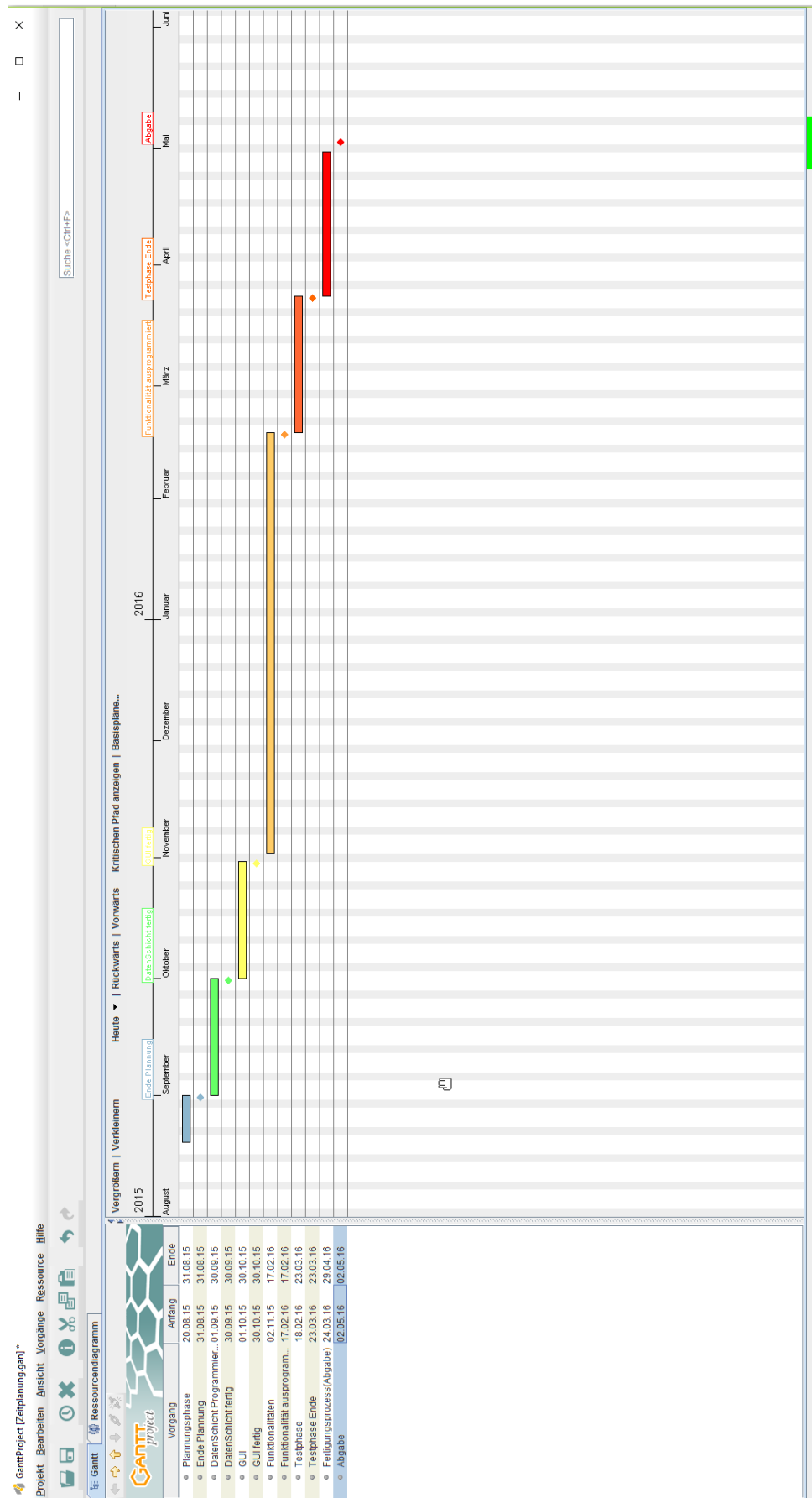


Abb. 12: Gant Project Plan

### 8.3 IVM – Matrix

Die IVM Matrix gibt einen schnellen Überblick über verschiedene Teile des Projekts. Man sieht auf einen Blick wer für eine Arbeit verantwortlich (V), mitwirkt (M) oder informiert (I) werden soll.

	Stefan Pilz	Fabian Radinger	Auftraggeber	Betreuungslehrer
Planung	V	V	I	I
Datenbank Konfiguration	I	V		
GUI	V	M		I
Funktionalitäten	V	M	I	I
Schnittstellen Anbindung	M	V	I	I
Dokumentation	V	V	I	I

*Tabelle 4 IVM – Matrix*

### 8.4 Projektstrukturplan

Ebenso wie die Zeitplanung ist auch ein Projektstrukturplan eine hilfreiche Methode zur erfolgreichen Fertigstellung eines Projekts. Dieser gliedert unsere Diplomarbeit in kleine Aufgaben und sorgt somit für einen besseren Überblick über das Gesamtprojekt.

Projektstrukturpläne werden üblicherweise hierarchisch gebildet. Die nachfolgende Grafik zeigt den Projektstrukturplan für unsere Diplomarbeit. Er gibt einen kurzen Überblick über das gesamte Projekt.

Das wichtigste Element des Plans ist das Projektmanagement, da es das Fundament für ein gutes Endergebnis bildet.

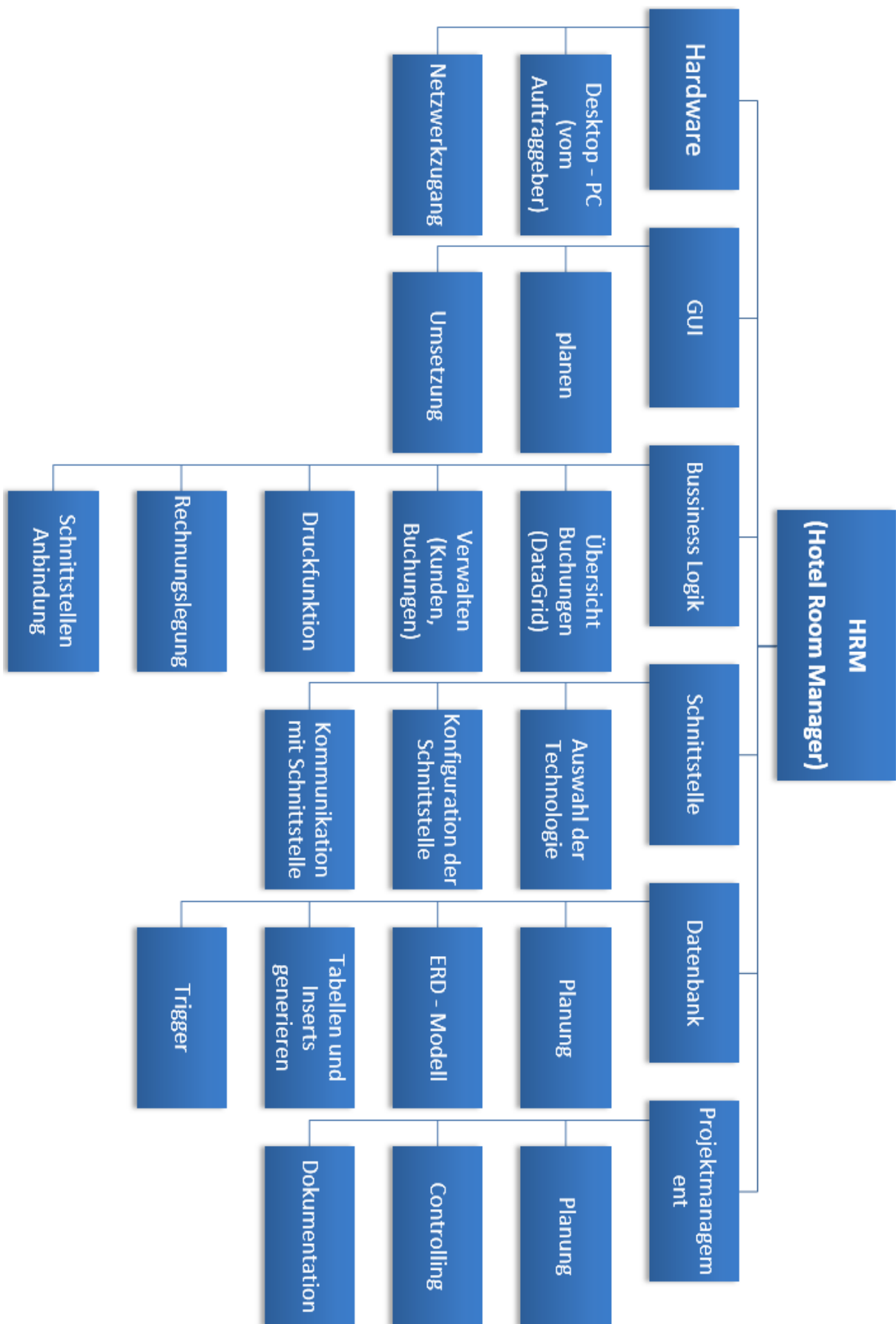


Abb. 13 Projektstrukturplan

## 8.5 Konfigurationsplan

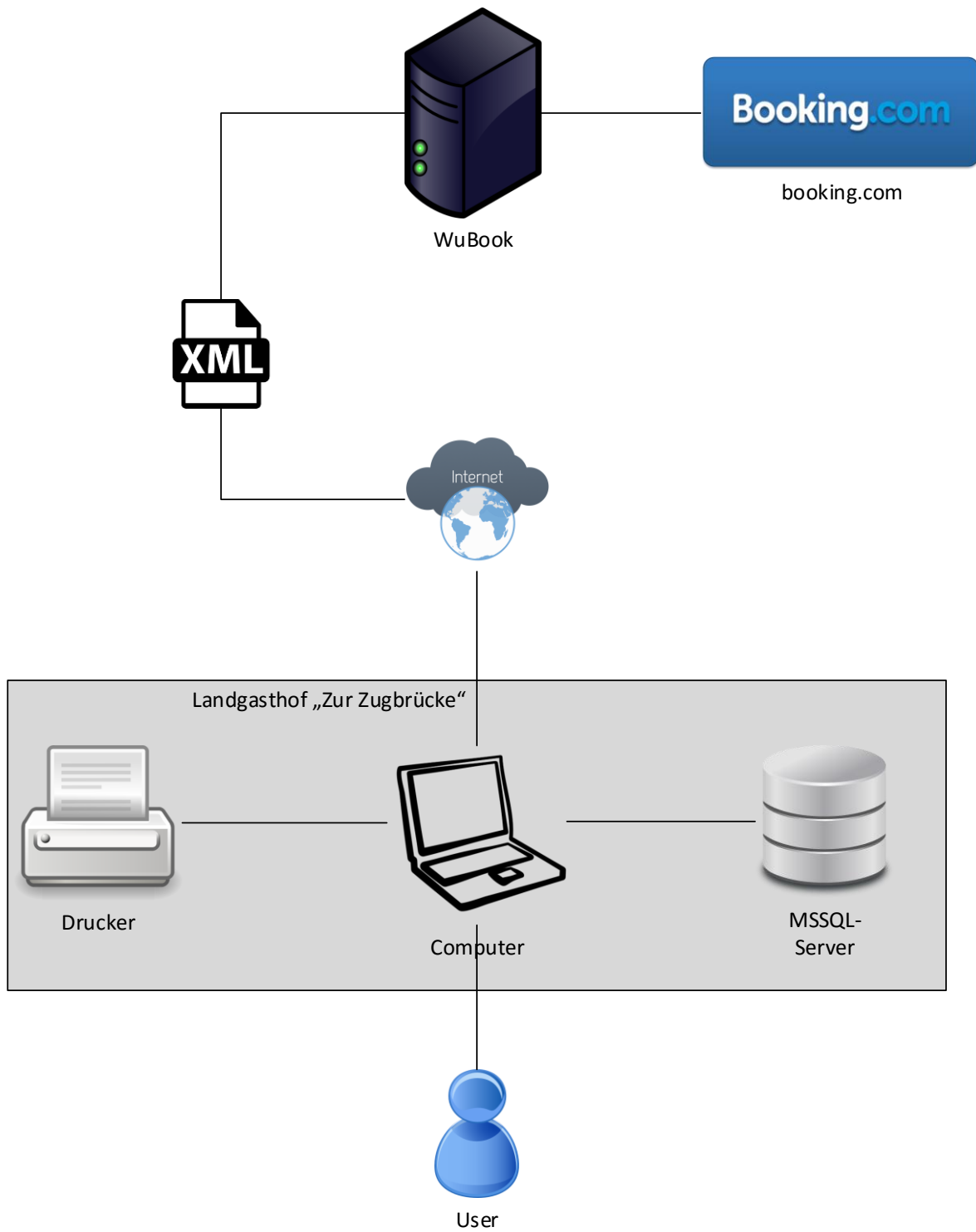


Abb. 14 Konfigurationsplan

### 8.5.1 booking.com

Die Zimmer unseres Auftraggebers können sowohl telefonisch als auch online gebucht werden. Booking.com ist also die Komponente, die Online Buchungen entgegennimmt.

### 8.5.2 WuBook

Um die Online Buchungen in eine externe Software zu bekommen, benötigt man die entsprechenden Daten. Diese können mithilfe einer Schnittstelle entgegengenommen werden. Booking.com stellt verschiedenste Schnittstellen zur Verfügung. Wir entschieden uns für WuBook.

Damit der Datenaustausch funktionieren kann, muss man zuerst den booking.com und den WuBook Account korrekt konfigurieren.

#### 8.5.2.1 XML

Ist die Konfiguration von WuBook und booking.com erfolgreich erledigt, kann man via XML Anfragen die Buchungen von Booking.com anfordern oder die Buchungen, die über HRM erfolgt sind, an WuBook per JSON senden. Die Anforderungen erfolgen via AlienPlugin (senden von neuen Buchungen) und der Wired Funktion (abrufen von Buchungen), auf welches wir später noch näher eingehen werden.

### 8.5.3 Internet

Das Internet ist die Schnittstelle für die Kommunikation, welche für die Verbindung mit WuBook notwendig ist. Es wird benötigt, um die telefonischen und die online Buchungen miteinander abzugleichen. Somit können Überbuchungen in Zukunft vermieden werden.

### 8.5.4 User

Der User ist in unserem Fall die Person die an der Rezeption arbeitet und sich entweder einen Überblick über die Buchungen machen will oder selber für einen Kunden reservieren will. Außerdem verfügt er über diverse Nebenfunktionen, die wir später noch erläutern werden.

### 8.5.5 Landgasthof „Zur Zugbrücke“

Der Landgasthof „Zur Zugbrücke“ fungiert, wie bereits oben erwähnt, als unser Auftraggeber.

Hier werden alle Reservierungen zusammengefasst, sowohl online, als auch telefonische. Diese können ebenso wie die Kunden entsprechend verwaltet werden.

#### 8.5.5.1 MS Sql-Server

Damit die Daten (Kunden, Buchungen, Zimmer, usw.) auch dauerhaft gespeichert werden können, benötigt unser Auftraggeber eine eigene Datenbank. Diese ist am Computer an der Rezeption installiert.

#### 8.5.5.2 Computer

Der Computer ist bei unserer Arbeit die wichtigste Komponente. Auf ihm wird unsere Software ausgeführt. Durch unsere Software werden XML Anfragen an WuBook gesendet. Wenn Daten zurückkommen, werden diese korrekt in der Datenbank (MS Sql-Server) gespeichert.

#### 8.5.5.3 Drucker

Da unsere Software auch über Druckfunktionen (Buchungsliste, Rechnung) verfügt ist ein Drucker ebenfalls notwendig. Dieser befindet sich ebenfalls an der Rezeption.

# 9 Realisierung

## 9.1 Überblick Gesamtsystem

In dem folgenden Use-Case Diagramm wird das Gesamtsystem grafisch dargestellt. Dieses setzt sich aus drei verschiedenen Akteuren zusammen. Zu beachten ist, dass der Akteur booking.com nur bei online Buchungen aktiv ist. Wubook ist hingegen bei online und auch telefonischen Buchungen aktiv, da es als Schnittstelle fungiert und somit alle Buchungen verwalten muss.

Zu beachten ist, dass Wubook für die Online Buchungen, die via booking.com getätigt worden sind, zuständig ist. Wenn somit ein Kunde bucht, kann unsere Software die getätigten online Buchungen durch die Schnittstellen Anbindung an Wubook in unsere Datenbank speichern. Somit ist ein Zimmer für den gewählten Zeitraum gesperrt.

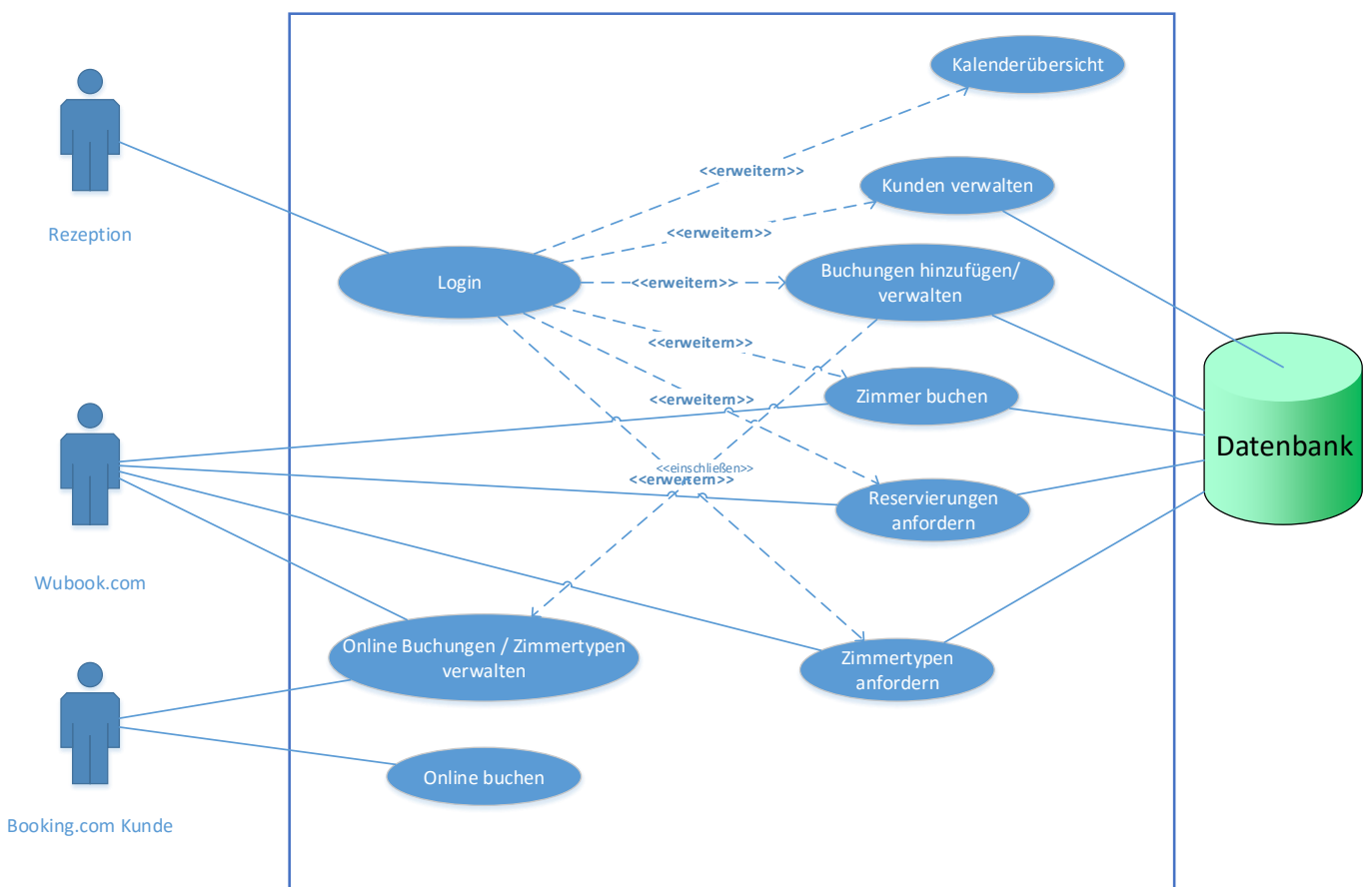


Abb. 15 Use-Case Gesamtsystem

## 9.1.1 Wubook Funktionen:

### 9.1.1.1 Wired API:

Im Zuge dieser Diplomarbeit werden die Daten von Wubook mittels der Wired API Funktion zu Verfügung gestellt. Um Buchungen oder sonstige Informationen von Wubook-Server zum Client zu senden, kann das Protokoll XMLRpc verwendet werden. Man kann auch mit verschiedenen Hilfsmittel XML – Strukturen selbst zusammenstellen und diese an den Server senden. Microsoft hat für diese Problemstellung verschiedene Aufzeichnungen und Dokumentation, wie diese Technologie verwendet wird. Im Zuge dieser Diplomarbeit haben wir von Herrn Professor Mag. Obermüller ein Script bekommen welches zusammenfassend alle wichtigen Funktionen über das Thema „Die Erstellung von XML – Strukturen“, beinhaltet.

Die URL für den Zugriff auf die Wired API Funktion lautet „<https://wubook.net/xrws/>“.

#### 1) token

Ein Token wird bei jedem Request mitgesendet um den Client eindeutig zu identifizieren. Der Token ist eine vierstellige Gleitkommazahl.

Es gibt verschiedene Einschränkungen für Tokens:

- Ein Token kann nur für 60 Operationen verwendet werden
- Ein Token erlischt nach 3600 Sekunden (1 Stunde)

Da ein Token nur 1 Stunde aktiv ist, muss jedes Mal wenn Reservierungen abgeholt werden, ein neuer Token angefordert werden.

#### 2) lcode

Der lcode wird direkt auf der Homepage von Wubook unter dem Menüpunkt „Unterkunft/Seite“ zur Verfügung gestellt. Dieser Code wird verwendet, um die Channels (booking.com, ...) eindeutig zu identifizieren.

#### 3) acquire\_token

Um einen Token zu erwerben wird die „acquire\_token“ Methode aufgerufen, die den Usernamen, das Passwort und den Provider\_Key als Parameter enthalten muss. Der Provider\_Key wird extern von Wubook nach einer Anfrage per E-Mail zugewiesen. Der Rückgabewert dieser Methode ist ein XML-String mit einem aktiven Token.

`acquire_token(user, password, provider_key)`

*Abb. 16: Screenshot von tdocs.Wubook (vgl. [DOC, Wubook Wired Authentication])*

```

<methodCall>
  <methodName>acquire_token</methodName>
  <params>
    <param>
      <value>
        <string>your_account_username</string>
      </value>
    </param>
    <param>
      <value>
        <string>your_account_password</string>
      </value>
    </param>
    <param>
      <value>
        <string>your_provider_key</string>
      </value>
    </param>
  </params>
</methodCall>

<methodResponse>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>
              <int>0</int>
            </value>
            <value>
              <string>1949536118.6632</string>
            </value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodResponse>

```

Abb. 17: Screenshot von XML-String (acquire\_token)

Dieser Screenshot zeigt, wie die Request und Response XML – Struktur bei einer erfolgreichen Kommunikation aussehen soll.

#### 4) release\_token:

Diese Methode ermöglicht einen, aktive Token zu schließen, damit der Server nicht überlastet wird. Außerdem hilft es, zu vermeiden, dass man als „flooder“ markiert wird.

### 5) `fetch_Rooms`:

Diese Funktion liefert uns alle Zimmer von einem Gasthof oder Hotel und deren Informationen. Diese werden als Rückgabeparameter in einem XML-String gesendet.

Es gibt 2 verschiedenen Arten von Zimmer: „Virtual room“ & „Standard room“

Ein Virtuelles Zimmer ist immer mit einem Standard Zimmer verbunden und erbt von seiner verfügbaren Anzahl von Betten. Ein Virtuelles Zimmer wird als „subroom“ mitgesendet. „Virtual room“ besitzt Herr Schiefer nicht, daher wurden diese Informationen vernachlässigt.

Die Methode „`fetch_rooms`“ benötigt nur einen aktiven Token und den lcode.

```
fetch_rooms(token, lcode, ancillary= 0)
```

*Abb. 18: Screenshot von `tdocs.Wubook` (vgl. [DOC, Wubook Wired Fetching reservations] )*

`ancillary` (ergänzend) ... wenn `ancillary` auf 1 gesetzt wird, werden alle Namen und Beschreibungen in jeder zur Verfügung stehenden Sprache zurückgegeben

### 6) `fetch_new_bookings`:

Vorab wichtige Informationen über die Funktion „`fetch_new_bookings`“:

#### a. **Marking**

„Marking“ bedeutet, dass alle abgeholten Reservierungen auf dem Wubook Server gekennzeichnet werden und diese bei einer normalen Request-Anfrage nicht mehr mitgesendet werden. Dies hat zur Folge, dass Reservierungen nicht doppelt gesendet und gespeichert werden.

#### b. **Push notification**

Damit nicht ständig eine Anfrage zum Server gesendet werden muss, ob neue Reservierungen getätigt worden sind, benachrichtigt uns diese Funktion bei Änderungen im Reservierungsbereich.

### c. Neue Reservierungen abholen:



```
fetch_new_bookings(token, lcode[, ancillary= 0, mark= 1])
```

Abb. 19: Screenshot von *tdocs.Wubook* (vgl. [DOC, Wubook Wired Fetching reservations])

Diese Methode gibt nur jene Reservierungen zurück, die noch nicht markiert worden sind. Wenn der Parameter „mark“ auf eins gesetzt ist, ist die Funktion „marking“ aktiviert. Wenn der Parameter „ancillary“ auf eins gesetzt wird, werden Zusatzinformationen bei jeder Reservierung mitgegeben.

#### Wichtige Parameter:

- status: Der Parameter „status“ gibt eine Zahl, zwischen eins und fünf zurück.
  - 1 ... Bestätigt
  - 2 ... Wartet auf Bestätigung
  - 3 ... Verweigert
  - 4 ... Akzeptiert
  - 5 ... Widerrufen

Die wichtigsten Werte sind eins und fünf, weil sowohl bei einer neuen Reservierung, als auch bei einer Widerrufung, die gleiche XML-Struktur gesendet wird und nur der Parameter „Status“ verändert wird.

- reservation\_code: Dieser Parameter ist die Reservierungs-ID, die eine Reservierung eindeutig identifiziert.
- modified\_reservations: Dieser Parameter ist normalerweise immer leer. Nur bei einer Änderung bekommt er den rcode zugewiesen.
- was\_modified: Dieser Parameter ist entweder 0 oder 1.
  - 1 ... wurde verändert
  - 0 ... keine Änderungen
- date\_received, date\_arrival, date\_departure
  - eingegangen Buchung
  - Anreise
  - Abreise
- rooms: Der Parameter „rooms“ enthält die ID vom gebuchten Raum Typ.

#### **d. Reservierungen mit Datum abholen:**

Wenn Reservierungen benötigt werden die schon markiert worden sind, können diese mittels der Funktion „fetch\_bookings“ erledigt werden. Da nur eine begrenzte Anzahl von Reservierungen gesendet werden können (max. 120 Reservierungen), sollte ein Zeitraum festgelegt werden.

Alte Reservierungen, die noch nicht mit dieser Applikation gebucht worden sind, werden, nach Absprache mit dem Auftraggeber, vernachlässigt.

#### **9.1.1.2 Alien Plugin**

Das Alien Plugin erlaubt es neue Reservierungen zu speichern oder zu löschen. Dieses Plugin muss extern unter dem Menüpunkt „Verwalten/Plugins“ installiert werden. Wenn es erfolgreich installiert wurde, wird ein „ALIENS CODE“ generiert und zugewiesen. Diese Zusatzfunktion kann mit zwei verschiedenen Technologien verwendet werden, entweder mit JSON POST oder mit XmlRpc. Im Zuge dieser Diplomarbeit haben wir JSON-Post verwendet, weil JSON-Strukturen einfacher zu generieren sind.

#### **Alien Reservation**

Diese Funktion ist dafür zuständig, dass neue Reservierungen erfolgreich zum Wubook – Server gesendet und gespeichert werden.

Folgende Parameter werden benötigt:

- dfrom (Ankunftsdatum)
- dto (Abreisedatum)
- alien\_code
- lcode
- rooms
- customer (Kundendaten)
- ancillary\_data
- ccard
- amount
- origin (kleine Nachricht, ist notwendig um sich zu identifizieren, wer die Reservierung eingefügt hat)

#### **XmlRpc - Request:**

alienbook(tok, lcode, alien\_code, dfrom, dto, room, customer, ccard, amount, origin, ancillary)

### Alien Deletion

Diese Funktion ist dafür zuständig, Reservierungen zu löschen. Folgende Parameter werden benötigt:

- lcode
- alien\_code (wird beim Installieren generiert)
- reservation\_id (Reservierungs ID von Wubook)
- origin
- reason (optional)

## 9.2 Von der XML-Struktur zum Datenbankmodell

### 9.2.1 Zimmer

XML-Struktur - Zimmer				
Attributname	Datentyp	Beschreibung		Zimmertypen
id	integer			idZimmertyp
name	varchar			typBezeichnung
shortname	varchar			kurzBezeichnung
occupancy	integer	Belegung		anzahlBetten
men	integer	Anzahl Erwachsene		anzahlErw
children	integer			anzahlKinder
subroom				
anchorate		price anchorage. The field contains the Parent Room		
price		Default Preis		Preis
availability		Default Availability		bettenVorhanden
board				
boards				
wodoo				

Abb. 20 XML Struktur

Grün markierte Felder, die Attributnamen beinhalten, sind in das Datenbankmodell übernommen worden. Gelbe Felder sind nur optional und für unser Programm nicht erforderlich. Damit wir in unserer Diplomarbeit diese Felder ausschließen konnten, wurden diese mit unserem Auftraggeber besprochen und danach erst herausgenommen. Die roten Felder wurden schon im Vorhinein ausgeschlossen. Die Spalte Zimmertypen stellt die Attribute vom Datenbankmodell der Tabelle „ZimmerTypen“ dar.

## 9.2.2 Reservierungen

XML-Struktur - Reservierungen			
Attributname	Datentyp	Beschreibung	Buchung
reservation_code	INTEGER NOT NULL		reservationCode
status	INTEGER	Reservierungen!S55	idStatus
channel_reservation_code			
id_channel		If originated by an OTA, that's the channel type (booking, expedia, ....)	id_channel
fount	INTEGER	Fount ID	
modified_reservations		wenn geändert wurde dann rcode gesetzt	welcheBuchung
was_modified		ob es geändert wurde (0 ok, 1 not)	wurdeKorrigiert
amount	DOUBLE	Geldbetrag	preisBuchung
booked_rate			
orig_amount	DOUBLE		originalPreisBuchung
date_received	Date	Wann reserviert wurde	eingegangenBuchung
date_arrival	Date	Anreise	vonBuchung
date_departure	Date	Abreise	bisBuchung
boards		Verpflegung, bb: bed and breakfast fb: full board	boards
men		adults	erwachseneBuchung
children		children	kinderBuchung
customer_city			Ort
customer_country			Land
customer_mail			emailKunde
customer_name			nachnameKunde
customer_surname			vornameKunde
customer_notes			notizen
customer_phone			telefonKunde
customer_address			adresseKunde
customer_language			Sprache
customer_zip		Postleitzahl	PLZ
rooms		A string contatining booked rooms IDs, comma separated	idZimmer

Abb. 21: XML-Struktur

Wie oben bei den Zimmern schon erwähnt, wurden die grün markierten Felder, die ebenfalls den Attributnamen enthalten, in das Datenbankmodell übernommen. Die dick geschriebenen Attribute, wurden in die Tabelle „Buchung“ übernommen. Beim Senden jeder Buchung von Wubook werden Kundeninformationen auch mitgesendet. Daher sind Felder, die ebenfalls grün sind und nicht dick geschrieben sind in die Tabelle „Kunde“ übernommen worden.

Die Spalte „idStatus“ gibt an was genau mit der Buchung durchgeführt werden soll:

1. bestätigt (Buchung kann gespeichert werden)
2. Wartet auf Bestätigung (Buchung kann nicht gespeichert werden)
3. Verweigert (Buchung kann nicht gespeichert werden)
4. akzeptiert (braucht nicht berücksichtigt werden)
5. widerrufen (Buchung wurde gelöscht)

### 9.3 Datenbankmodell

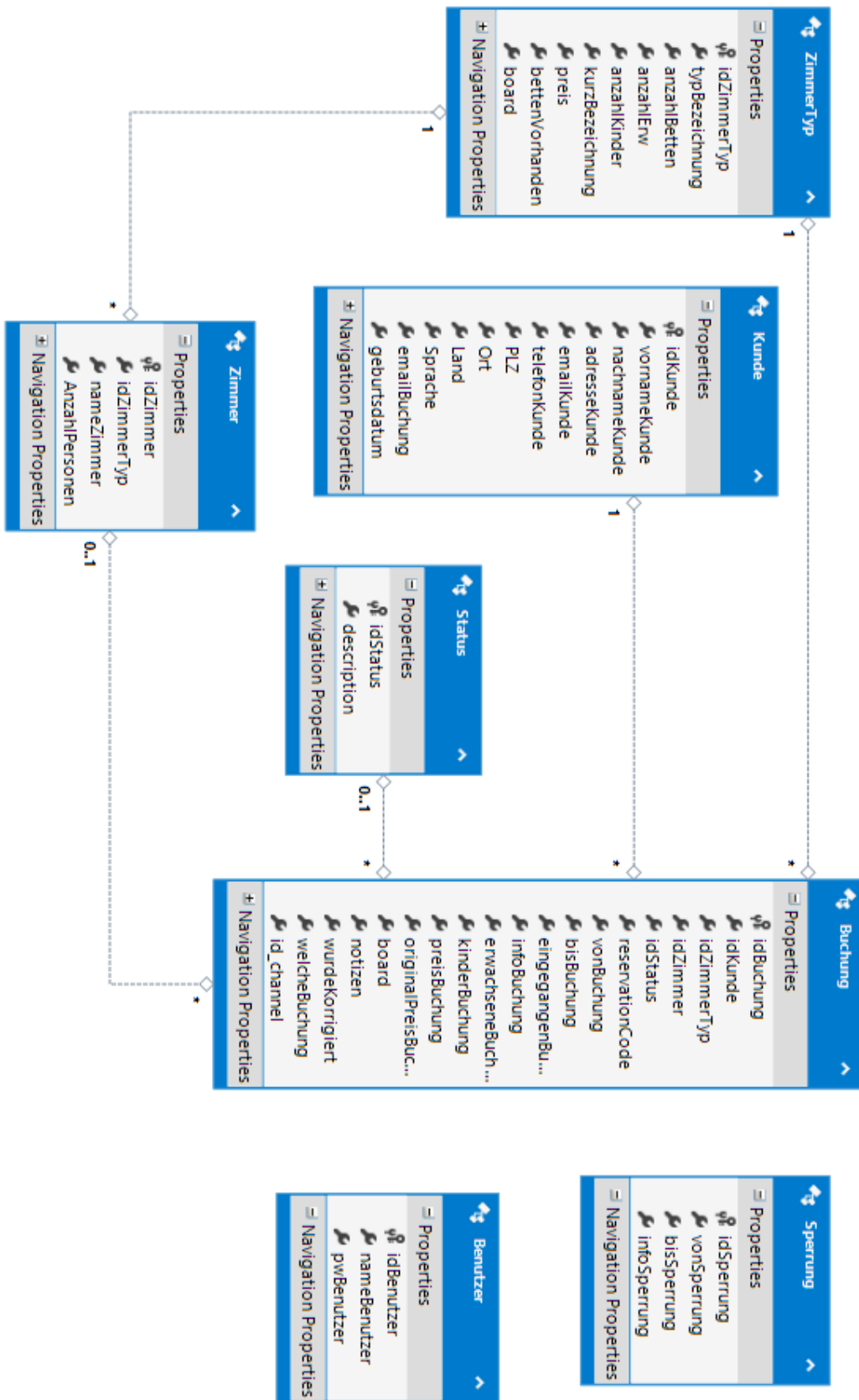


Abb. 22 Datenbankmodell

Das Datenbankmodell ist mit der Programmierumgebung Visual Studio erstellt worden. Das Modell zeigt sieben Tabellen inklusive ihrer Attribute und Beziehungen. Die einzelnen Tabellen werden in den Unterkapiteln erklärt. Einige Tabellen sind mit der XML Struktur der externen Schnittstelle abgeglichen. Die XML Struktur wurde im vorigen Kapitel erklärt.

### 9.3.1 Benutzer

Da unsere Software einen Login beinhaltet ist eine Benutzerverwaltung notwendig. Die verschiedenen Benutzer werden in der Tabelle Benutzer gespeichert. Diese Tabelle steht in keiner Beziehung zu den restlichen Tabellen. Sie beinhaltet nur den Namen des Benutzers und das Passwort.

### 9.3.2 Zimmertyp

Da unser Auftraggeber seine Zimmer verschieden vermieten kann, sind verschiedene Typen notwendig. Ein Zimmertyp hat einen Namen (Familienzimmer, Dreibettzimmer, Doppelbettzimmer, Einzelbettzimmer), außerdem hat jeder Typ einen fixen Preis und eine bestimmte Anzahl von Betten. Sie wird neben dem Zimmer und den Kunden benötigt um eine Buchung zu vervollständigen. Diese Tabelle wurde wie die Buchungstabelle mit der XML Struktur der Schnittstelle abgeglichen, damit für jedes Attribut eine Spalte vorhanden ist, da die Zimmertypen auch von booking.com stammen.

### 9.3.3 Zimmer

In dieser Tabelle werden lediglich die verschiedenen Zimmer mit ihren Namen gespeichert. Sie wird neben dem Zimmertyp und den Kunden benötigt, um eine Buchung zu vervollständigen. Ein Zimmer hat keinen fixen Zimmertyp. Der Typ ergibt sich aus der Anzahl der Personen, die in einer Buchung eingetragen sind.

### 9.3.4 Kunde

Damit der Auftraggeber auch über eine Kundenverwaltung verfügt, ist eine Tabelle Kunde notwendig. Hier werden alle nötigen Infos zu einem Kunden gespeichert (Adresse, Name, Email, Telefonnummer). Außerdem ist ein Kunde ebenfalls für eine Buchung notwendig.

### 9.3.5 Buchung

Diese Tabelle ist notwendig um die nötigen Infos von einer Buchung zusammenzufassen. Hier wird der Aufenthaltszeitraum (von – bis), die Anzahl von Kindern bzw. Erwachsenen, das Zimmer(`idZimmer`), der Kunde(`idKunde`) bekannt gegeben. Optional kann eine Info zur Buchung hinzugefügt werden. Der Zimmertyp(`idZimmerTyp`) wird automatisch von einem Trigger gesetzt. Dieser wird anhand der Anzahl von Erwachsenen und Kindern ermittelt. Diese Tabelle beinhaltet alle Daten die ein Kunde über booking.com eintragen kann. Sie ist mit der XML Struktur der Schnittstelle abgeglichen, damit für jedes Attribut eine Spalte vorhanden ist, da die Buchungen von booking.com stammen können.

### 9.3.6 Status

Die Tabelle Status ist aufgrund der Schnittstellen Anbindung notwendig. Es gibt darüber Auskunft ob eine Buchung neu, verändert oder gelöscht wurde. Es ist auch für Buchungen über HRM notwendig, da diese ebenfalls über das sogenannte Alien Plugin an die Schnittstelle gesendet werden.

### 9.3.7 Sperrung

Da es bestimmte Ruhetage und Betriebsurlaube gibt, ist die Tabelle Sperrung notwendig. Hier gibt man das „von und bis Datum“ der gewünschten Sperrung ein. Zusätzlich kann noch eine Info hinzugefügt werden. Diese Tabelle steht in keiner Beziehung zu den anderen.

## 9.4 Verbindung mit der Datenbank

In dieser Diplomarbeit verwendeten wir die Funktion „Code First from database“ für die Konfiguration unseres Programms mit der Datenbank. Dies wird dann verwendet, wenn das Datenmodell schon entworfen wurde und eine fertige Datenbank mit dem entworfenen Modell existiert. Diese Funktion erstellt das Modell und die Entities automatisch, also die Klassen und deren Eigenschaften. Das Erstellen der Entities wird in den folgenden Screenshots gezeigt.

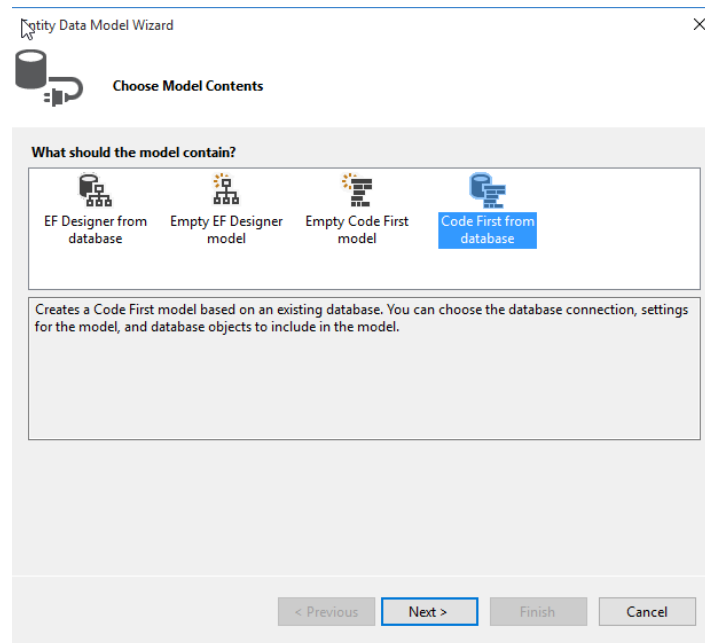


Abb. 23: Code First from database in VisualStudio

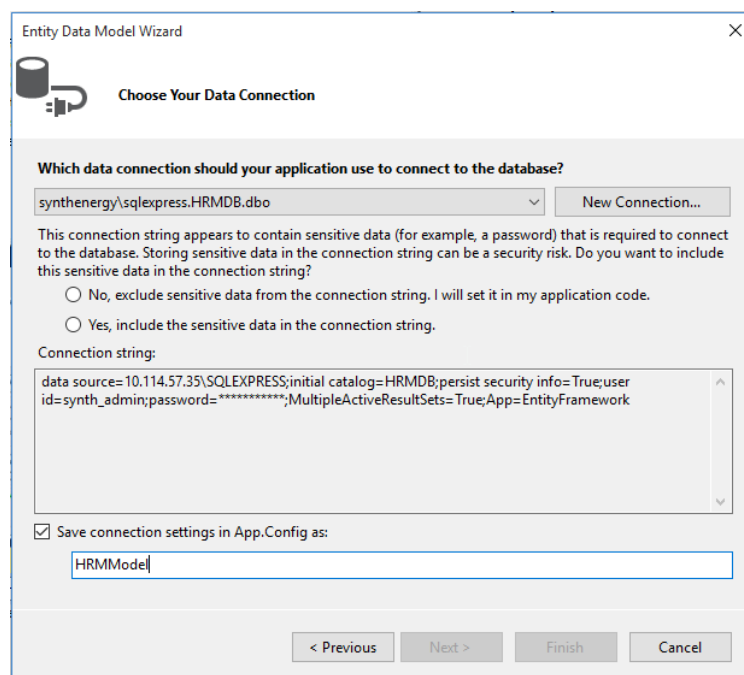


Abb. 24: Code First from database in VisualStudio – Auswahl von Datenbank

## 9.5 Darstellung

Einer der wichtigsten Aspekte unserer Diplomarbeit ist die Gewährleistung einer Benutzerfreundlichen Bedienung der Software. Außerdem ist es wichtig, alle wichtigen Funktionen schnell aufrufen zu können.

Um diese Aspekte zu erfüllen haben wir unsere Desktop Applikation in drei Hauptbereiche unterteilt, diese werden in den Unterkapiteln näher erklärt

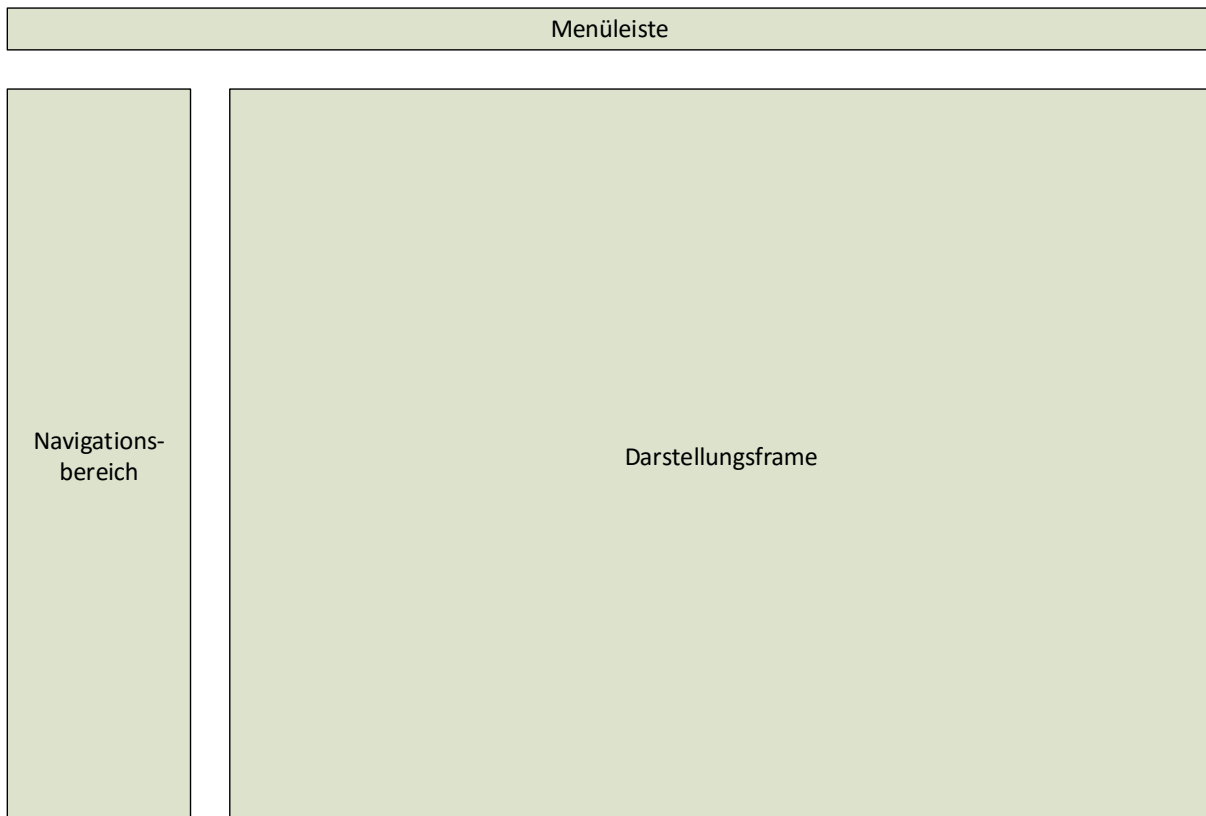


Abb. 25 GUI Aufbau der Desktop Applikation

### 9.5.1 Menüleiste

Die Menüleiste befindet sich wie gewöhnlich am obersten Bereich unserer Anwendung. Mit dieser Leiste können alle Funktionen aufgerufen werden.

### 9.5.2 Navigationsbereich

In diesem Bereich findet der Benutzer die wichtigsten Funktionen. Dies gewährleistet eine schnellere Navigation zu diesen Funktionen.

Außerdem befinden sich die Uhrzeit, eine Information über Check-Ins, Check-Outs und Warnungen in diesem Bereich.

### 9.5.3 Darstellungsframe

Wie der Name schon verrät, wird in diesem Bereich der gewünschte Inhalt nach Auswahl eines Menüpunkts dargestellt.

## 9.6 Funktionalitäten

In dem folgenden Use-Case, sieht man lediglich die Funktionen, über die der Benutzer der die Software bedient, verfügt. Zu beachten ist, dass im Hintergrund der verschiedenen Funktionen eine Verbindung zu einer Schnittstelle nötig ist, die booking.com und telefonische Buchungen zusammenfasst.

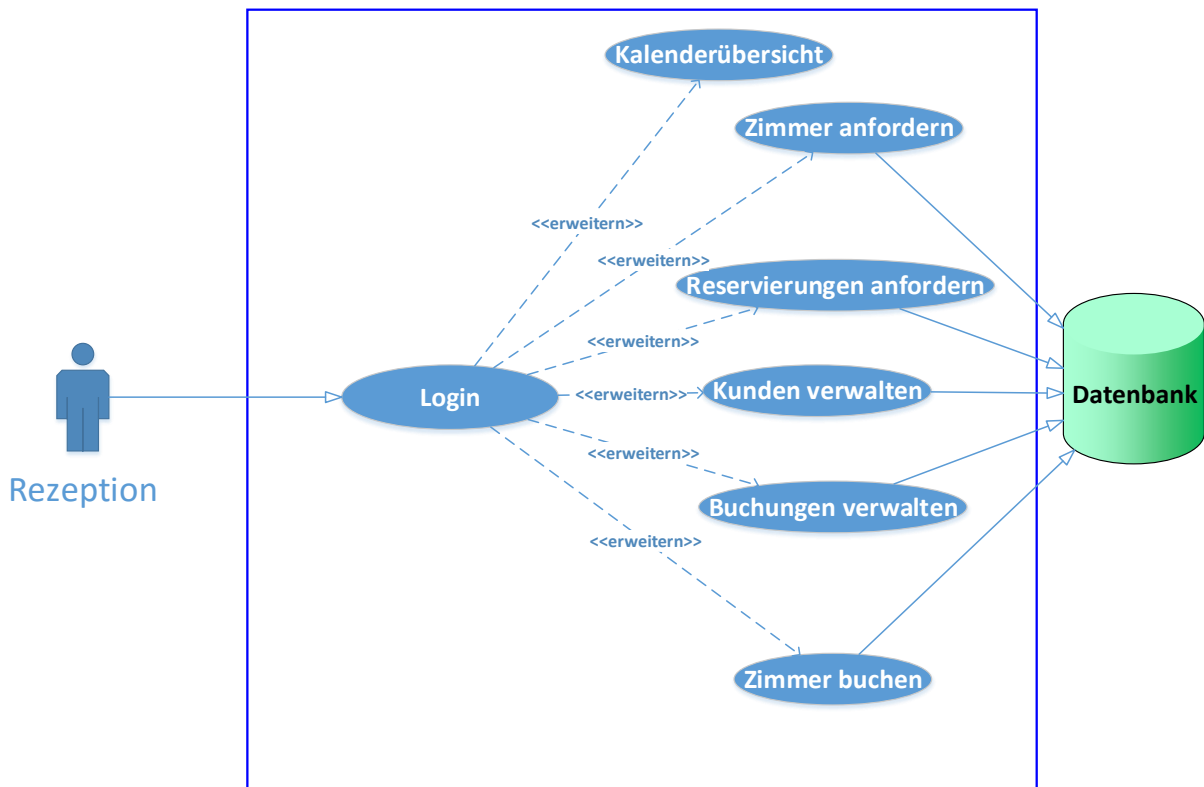


Abb. 26 Use-Case Funktionalitäten

### 9.6.1 Starten der Applikation

Unsere Anwendung beginnt mit einem Splash Screen um die Ladeoperationen zu initialisieren und die Wartezeit so kurz und angenehm wie möglich zu gestalten. Ein Splash Screen ist ein grafischer Platzhalter um die Ladezeiten zu überbrücken. Um den Splash Screen auszulösen, wird ein Startup Event in der App.xaml gesetzt und bei der App.xaml.cs ausprogrammiert.

### 9.6.2 Login

Damit nur bestimmte Personen Zutritt zum Hauptfenster haben, ist ein Login erforderlich. Beim Start der Software erscheint deshalb ein kleines Dialog-Fenster.



Abb. 27 Login

### 9.6.3 Verwalten von Kunden

Damit der Auftraggeber seine Kunden zentral gesammelt hat, ist eine Kundenverwaltung erforderlich. Außerdem ermöglicht diese auch eine schnellere Verfassung der Reservierungen. Ein neuer Kunde kann entweder direkt bei einer Buchung angelegt werden oder über eine eigene Seite.



Abb. 28: Kundenverwaltung

### 9.6.3.1 Export der Kunden E-Mail Adressen

Der Benutzer hat neben der Übersicht seiner Kunden, die Möglichkeit die E-Mail Adressen seiner Kunden in eine CSV Datei zu exportieren, damit er von Zeit zu Zeit seine Angebote versenden kann.

Kunden Übersicht				
Name Kunde	Adresse	Plz - Ort	Telefonnummer	E-Mail
Fabian Radinger	Musterstraße 1	4392 Dorfstetten	00000	a.a@a.at
Tobias Radinger	asdf	4392 Dorfstetten	06448461	tobias@yahoo.com
Stefan Pilz	Musterstraße 1	4362 Bad Kreuzen	00000	a.a@a.at
Simon Kroissmayr	Straße 2	1231 Teststadt	06648453456	simon@aon.at
Oliver	Mußerstraße 1	4141 Stadt	00000	oliver.voß@aon.at
Florian Schimmer	aaaa	1234 Stadt	0064 / 4564	a.a@a.at
Hans Huber	Linzerstraße 22	4020 Linz	0664 25122	huber@hans.at
Josef Schiefer	Bk 9	4362 Bad Kreuzen	0664869255899	josef@aon.at
Fritz Josef	Pergerstraße 12	4320 Perg	0664	josef@gmail.com
Robert Neiss	Linzerstraße 2	4780 Schärding	0650&#47;8513307	
Jerry Gustafson	P.O. Box 4	SOC 1P0 Macoun	306-634-9231	
Claudia Stark	Claudia	4910 Stark	00436764530340	
Sepp Aigner	Bad Kreuzen	4362 Bad Kreuzen 56	0664 55623	seppaigner@gmx.at
Sepp Aigner	Bad Kreuzen 23	4362 Bad Kreuzen	0664 52232	aigner@fmx.at
Kurt Schreiber	Mitterlach 5a	5274 Burgkirchen	+ 4366473527949	
Johann Egger	Bahnhofstrasse 14	4775 Taufkirchen an der Pr	06649246701	
berthold löschner	alpenstraße 1	5020 salzburg	+436602739255	
Anna Tetera	Czajkowskiego 78	51147 Wroclaw	+48500111301	
Hans Hedrich	Haus-Endt-Str. 158	40593 Düsseldorf	00491722409914	

[Exportieren](#)

Abb. 29 Übersicht der Kunden und Export der Mail Adressen

### 9.6.4 Eintragung von Buchungen

Neben den online getätigten Buchungen soll der Rezeptionist auch selbst Buchungen eintragen können, falls ein Kunde telefonisch reservieren möchte. Geschieht dies, wird die Buchung über das Alien Plugin von wubook.com an booking.com gesendet und somit die Verfügbarkeit der Zimmer verringert.

## Buchung

Bestehender Kunde  
 Neuer Kunde

Suchen

---

Vorname:   
 Nachname:   
 Adresse:   
 Ort:

Aufenthalt von:    
 Aufenthalt bis:    
 Zimmer:    
 Zusatzinfos:   
 Anzahl Erwachsene:   
 Anzahl Kinder:   
 Verpflegung:

[Reservieren](#)

Abb. 30: Buchen von Zimmern

### 9.6.5 Übersicht

Mithilfe dieser Funktion sieht der Benutzer eine schnelle Übersicht über die noch verfügbaren oder besetzten Zimmer. Er kann den Zeitraum mittels den beiden Datumsfelder im oberen Bereich bestimmen. Außerdem können hier neue Buchungen ganz bequem per Mausklick hinzugefügt, bestehende verändert oder gelöscht werden. Nach einem Klick auf ein Feld öffnet sich ein Dialog Fenster. Die Textfelder sind bei einem neuen Eintrag leer und bei einem bestehenden beinhalten die Textfelder die vorhandenen Daten zur Buchung.

Zimmer	15.02.2016	16.02.2016	17.02.2016	18.02.2016	19.02.2016	20.02.2016	21.02.2016	22.02.2016	23.02.2016	24.02.2016	25.02.2016
Zimmer 1											
Zimmer 2	Sepp Ayler alleinlich gegen Nude										
Zimmer 3											
Zimmer 4											
Zimmer 5											
Zimmer 6											
Zimmer 7											
Zimmer 8											
Zimmer 9											
Zimmer 10											
Zimmer 11											
Zimmer 12											
Zimmer 13											
Zimmer 14											
Zimmer 15											
Zimmer 16											
Zimmer 17											
Zimmer 18											

Abb. 31: Übersicht von Buchungen in Kalenderansicht

### 9.6.5.1 Drucken der Buchungen vom Übersichtsraster

Neben der Übersicht, kann sich der Benutzer noch eine Liste der gebuchten Zimmer erstellen lassen. Dies geschieht mittels dem Button „Drucken“. Wählt er diese Funktion so öffnet sich zuerst eine Druckvorschau, klickt er dann auf das Druckersymbol, so öffnet sich ein Druckdialog wo der gewünschte Drucker ausgewählt werden kann.

Buchungsliste (11.02.2016 - 11.04.2016) Stand vom: 11.02.2016 11:01:49

Name	Aufenthalt von	Aufenthalt bis	Zimmer	Erw/Kind	
Fabian Radinger	17.02.2016	20.02.2016	10	2/2	(Test)
Stefan Pilz	10.02.2016	13.02.2016	7	1/1	(1)
Stefan Pilz	10.02.2016	11.02.2016	3	2/0	(Halbpension)
Stefan Pilz	12.02.2016	13.02.2016	7	1/0	(gee)
Stefan Pilz	19.02.2016	20.02.2016	7	1/0	0
Simon Kroissmayr	18.02.2016	19.02.2016	7	1/0	0
Stefan Pilz	13.02.2016	14.02.2016	7	1/0	(Test)
Stefan Pilz	14.02.2016	18.02.2016	7	1/0	(Test Buchungs Bug)
Tobias Radinger	03.02.2016	05.02.2016	2	2/1	0

Seite 1/1

Abb. 32 Druckvorschau

Drucken

Allgemein

Drucker auswählen

- Fax
- KONICA MINOLTA C364e PCL 6 1.0G an saturn
- Microsoft Print to PDF
- Microsoft XP

Status: Bereit

Standort:

Kommentar:

Einstellungen

Drucker suchen...

Seitenbereich

Alles

Markierung

Aktuelle Seite

Seiten:

Anzahl Exemplare: 1

Sortieren

1 1 2 2 3 3

Drucken Abbrechen Übernehmen

Abb. 33 Druckdialog

### 9.6.5.2 Zimmer Sperren

Über den Button „Zimmer Sperren“ kann der Benutzer über ein Dialog Fenster alle Zimmer sperren. Dies ist notwendig wenn ein Betriebsurlaub oder sonstiges vorliegt. Über den aufklappbaren Menüpunkt „Bestimmte Tage“ ist es auch möglich, nur bestimmte Tage in einem Zeitraum zu sperren. Dies ist zum Beispiel beim Eintragen des Ruhetages sehr hilfreich. Wird eine Sperrung irrtümlich eingetragen, so muss der Benutzer lediglich die beiden Daten erneut eintragen und der Button ändert seinen Wert von „Sperren“ auf „Freigeben“

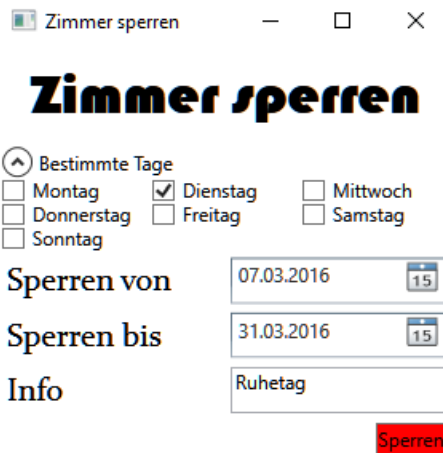


Abb. 34 Dialog Fenster: Zimmer sperren

### 9.6.6 Verwalten von Buchungen

Bereits eingetragene Buchungen können einfach verändert werden. Es besteht die Möglichkeit nach Zimmer und Datum oder nur nach einem Kunden zu filtern. Anschließend muss lediglich die gewünschte Buchung angeklickt werden.

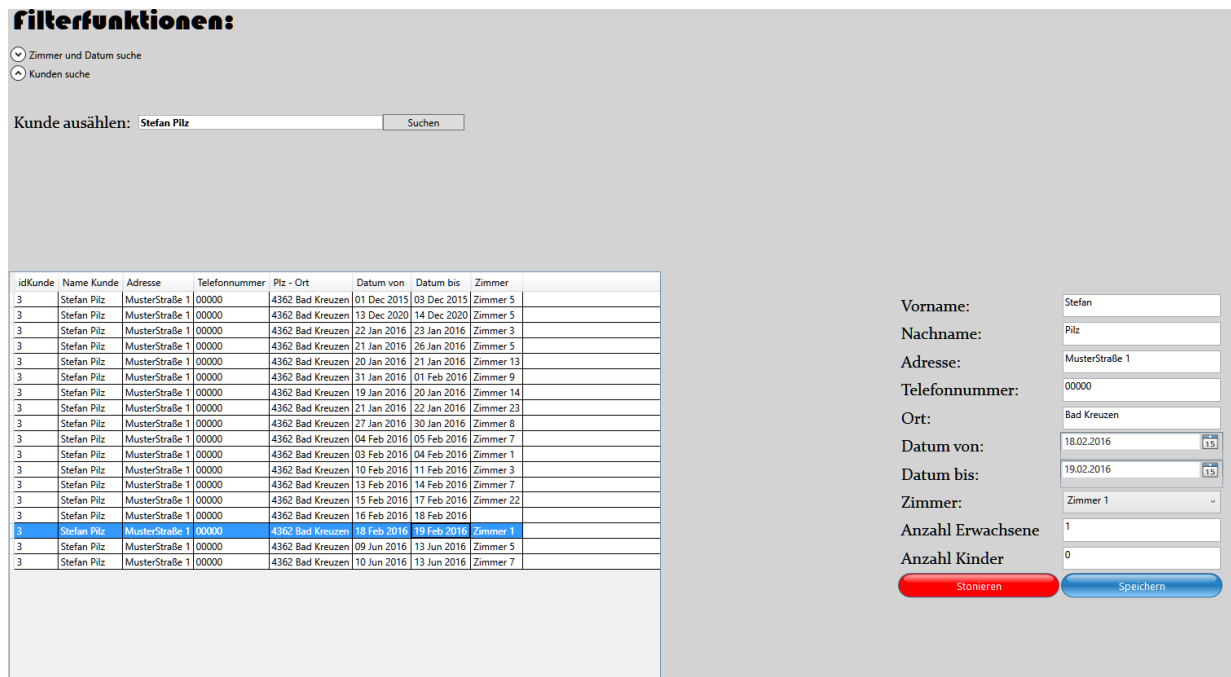


Abb. 35: Bearbeiten von Reservierungen

### 9.6.6.1 Online Buchungen Zimmer zuordnen

Online Buchungen sollen nicht automatisch einem Zimmer zugeordnet werden, sondern soll von der Rezeptionistin durchgeführt werden. Daher ist eine zusätzliche Funktion notwendig die das ermöglicht.

Auf dieser Seite sieht man also alle Buchungen die noch keinem Zimmer zugeordnet sind. Hier kann man nur das Zimmer bestimmen. Alle anderen Werte dürfen nicht Editierbar sein. Die online getätigten Buchungen werden in 30-minütigen Abschnitten aktualisiert. Sie können aber auch über den Menüpunkt „Reservierungen anfordern“ aktualisiert werden.

Damit der Benutzer nicht auf die Zuordnungen vergisst, wird in der linken TextBox des Hauptfensters eine Warnung angezeigt.

**Zimmer zuordnen:**

Name Kunde	Telefonnummer	Plz - Ort	Adresse	Datum von	Datum bis	Zimmer
Stefan Pilz	00000	4392 Dorfstetten	Musterstraße 1	16 Feb 2016	18 Feb 2016	

Vorname:

Nachname:

Adresse:

Telefonnummer:

Ort:

Datum von:

Datum bis:

Zimmer:

Anzahl Erwachsene

Anzahl Kinder

- Zimmer 1
- Zimmer 2
- Zimmer 3
- Zimmer 4
- Zimmer 5
- Zimmer 6
- Zimmer 8
- Zimmer 9
- Zimmer 11
- Zimmer 12
- Zimmer 13
- Zimmer 14
- Zimmer 15
- Zimmer 16
- Zimmer 17
- Zimmer 18
- Zimmer 19
- Zimmer 20

Abb. 36 Zimmer zuordnen

### 9.6.7 Rechnungen:

Mit dieser Funktion wird automatisch ein Excel File erstellt. Zuvor muss der Zeitraum ausgewählt werden und auf die Buchung geklickt werden, für die man die Rechnung benötigt. Mittels einem Speicher – Dialog kann der Benutzer die Rechnung an einem beliebigen Ort abspeichern. Danach wird eine Excel Datei generiert, welche die notwendigen Daten der gewählten Buchung beinhaltet.

## Rechnung erstellen

Aufenthalt von:  15

Aufenthalt bis:  15

Name Kunde	Adresse	Telefonnummer	Plz - Ort	Datum von	Datum bis	Zimmer	Info
Fabian Radinger	MusterStraße 1	00000	4392 Dorfstetten	04 Feb 2016	05 Feb 2016	Zimmer 1	mit Haustier
Stefan Pilz	MusterStrase 1	+3964329779	4362 Bad Kreuzen	31 Jan 2016	01 Feb 2016	Zimmer 9	Halbpension
Florian Schimmer	aaaa	0064 / 4564	1234 Stadt	04 Feb 2016	05 Feb 2016	Zimmer 2	Wieder ein Test
Simon Kroissmayr	Straße 2	06648453456	1231 Teststadt	06 Feb 2016	07 Feb 2016	Zimmer 12	
Stefan Pilz	MusterStrase 1	+3964329779	4362 Bad Kreuzen	04 Feb 2016	05 Feb 2016	Zimmer 7	nur Frühstück
Stefan Pilz	MusterStrase 1	+3964329779	4362 Bad Kreuzen	03 Feb 2016	04 Feb 2016	Zimmer 1	nur Frühstück
Simon Kroissmayr	Straße 2	06648453456	1231 Teststadt	07 Feb 2016	10 Feb 2016	Zimmer 1	Halbpension
Fabian Radinger	MusterStraße 1	00000	4392 Dorfstetten	05 Feb 2016	07 Feb 2016	Zimmer 5	
Stefan Pilz	MusterStrase 1	+3964329779	4362 Bad Kreuzen	10 Feb 2016	11 Feb 2016	Zimmer 3	Halbpension
Florian Schimmer	aaaa	0064 / 4564	1234 Stadt	06 Feb 2016	07 Feb 2016	Zimmer 10	nur Frühstück

Abb. 37: Erzeugen von Rechnungen


			
Josef Leopold Schiefer, 4362 Bad Kreuzen 9 Tel. +43 / (0)7266 / 6212 info@gasthof-schiefer.at www.gasthof-schiefer.at UID: ATU 6539 4815			
		Bad Kreuzen,	14.03.2016
		Rgnr.	376
Aufenthalt vom		06.02.2016	bis 07.02.2016
<b>RECHNUNG</b>			
2 Übernachtung Doppel Zimmer		€	72,00
2 Ortstaxe pro Tag und erw. Person	á €	1,00 €	2,00
<b>Zwischensumme</b>			€ 74,00
abzügliche Anzahlung			€
<b>Gesamtsumme brutto</b>			€ 74,00
<b>Mehrwertssteueraufstellung:</b>			
Steuer-	Netto	MWSt.	Brutto
0% Ortstaxe	€ 2,00	-	€ 2,00
13% Übernachtung	€ 63,72	€ 8,28	€ 72,00
Danke für Ihren Besuch Auf Wiedersehen in Bad Kreuzen Josef Schiefer und sein Team			

Abb. 38: Ergebnis von der Erstellung der Rechnung

## 9.6.8 Schnittstelle

Damit der Austausch der Daten mit booking.com funktioniert, ist eine Schnittstelle erforderlich. Unser Schnittstellen Anbieter WuBook, welcher mit booking.com zusammenarbeitet, bietet diverse Funktionen an, um die Reservierungen und die Zimmertypen von booking.com zu exportieren. Die Funktionen der Schnittstelle sind in der Menüleiste abrufbar.

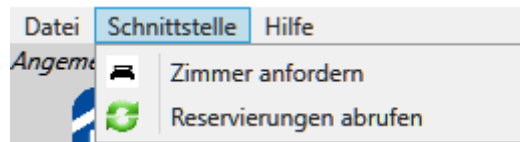


Abb. 39 Menüpunkt Schnittstelle

### 9.6.8.1 Zimmer anfordern

Da über booking.com verschiedene Zimmertypen (Einzelbett-, Doppelbett-, Dreibett-, Familienzimmer) reserviert werden können und sich diese gelegentlich ändern können, besteht die Möglichkeit, die aktuellen Zimmertypen anzufordern. Diese werden dann in die Datenbank gespeichert.

### 9.6.8.2 Reservierung anfordern

Damit der Benutzer die aktuellen Buchungen von booking.com erhält, muss er in der Menüleiste auf die Funktion „Reservierung anfordern“ klicken. Die neuen Reservierungen werden aber auch beim Start der Anwendung und alle 30 Minuten automatisch aufgerufen. Nach Aufruf dieser Funktion, werden die neuen Reservierungen in die Datenbank übernommen. Die angeforderten Reservierungen sind aber noch keinem Zimmer zugeordnet. Diese müssen erst von der Rezeption dem gewünschten Zimmer zugeordnet werden.

## 9.7 Code

### 9.7.1 Start der Applikation:

Damit Ladezeiten nicht störend empfunden werden, kann mittels einem „Splash Screen“ ein Logo oder sonstige Grafiken angezeigt werden. In dieser Diplomarbeit wird beim Starten das selbst entworfene Logo angezeigt.

App.xaml:

```
<Application x:Class="Diplomarbeit.App"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             Startup="App_Startup">
    <Application.Resources>

    </Application.Resources>
</Application>
```

Abb. 40: App.xaml Startup Event

App.xaml.cs:

```
public partial class App : Application
{
    void App_Startup(object sender, StartupEventArgs e)
    {
        SplashScreen screen = new SplashScreen("../images/logo.png");
        screen.Show(true);

        Thread.Sleep(1000);

        MainWindow mainWindow = new MainWindow();
    }
}
```

Abb. 41: App.xaml.cs Splash Screen

## 9.7.2 createRectangles()

Diese Methode ist für die OverviewPage notwendig. Dies ist eine Seite, in welcher freie und reservierte Zimmer farblich gekennzeichnet sind.

createRectangle() ist eine Methode die von uns selbst geschrieben wurde. Wie der Name schon verrät, ist die Methode dafür zuständig die Rechtecke auf der OverviewPage richtig darzustellen. Außerdem werden die Labels links (Zimmer) und oberhalb (Datum) von dieser Methode erstellt.

Dadurch, dass sich die Buchungen zur Laufzeit ändern können, muss die Erstellung der Rectangles und Labels im CodeBehind geschehen. Für die Positionierung wurden Hilfsvariablen verwendet (**helpLeft**, **helpTop**, **roomTop**, **dateLeft**). Um die Koordinaten auszurechnen benötigten wir eine Formel. Da der Benutzer die Möglichkeit einer Datumseingabe hat, ist es nötig die Tage zwischen den beiden Daten zu ermitteln und in die Formel einfließen zu lassen, ebenso die Anzahl der Zimmer.

```
int zim = rooms.Count();
TimeSpan daysBetween = dpBis.SelectedDate.Value - dpVon.SelectedDate.Value;
int Tage = Math.Abs(daysBetween.Days) + 1;
//Hilfsvariable für rectangles Koordinaten
int helpLeft = -(Tage * 150 - 150); //Tage*Breite-150(--> 150 dewegen weil die breite einmal weggerechnet wird)
int helpTop = -(zim * 50 - 50); //gleiches wie oben höhe =50
```

Abb. 42: Code (Variablendeklaration für Rectangles)

## REALISIERUNG

Beim Durchlaufen der äußersten for-Schleife müssen zuerst immer die Zimmer angelegt werden `labelRoom`. In der nächsten for-Schleife wird für jeden Tag ein Rechteck angelegt `r1`, zuerst werden alle grün gefärbt (grün= Zimmer ist frei) und erst später wird abgefragt ob das Zimmer zu diesem Datum belegt ist. Ist dies der Fall wird das Rechteck rot gefärbt. Außerdem wurde auf jedes Rechteck ein EventHandler gegeben (`r1.MouseDown += r1_MouseDown`).

```
void createRectangles()
{
    Rectangle r1 = new Rectangle();

    //In Dieser Region werden alle notwendigen Variablen angelegt die im Code benötigt werden
    #region

    //for schleife für Zimmer
    for (int i = 0; i < zim; i++)
    {
        labelRoom = new Label()
        {
            Name = "room_" + i,
            Content = rooms.ElementAt(i).nameZimmer,
            Margin = new Thickness(
                left: 0,
                top: roomTop,
                right: 0,
                bottom: 0),
            FontSize = 14
        };
        labelRoom.FontWeight = FontWeights.Bold;
        roomTop += 50;
        zimmerGrid.Children.Add(labelRoom);
        //For schleife für Tage
        for (int j = 0; j < Tage; j++)
        {
            r1 = new Rectangle()
            {
                Height = 50,
                Width = 150,
                Name = "rect_" + i + "_" + j,
                Stroke = Brushes.Green,
                Fill = Brushes.LightGreen,
                Margin = new Thickness(
                    left: helpLeft,
                    top: helpTop,
                    right: 0,
                    bottom: 0),
                HorizontalAlignment = HorizontalAlignment.Stretch,
                VerticalAlignment = VerticalAlignment.Stretch,
                //Tag = new Object{dateList.ElementAt(j),i},
                ToolTip = "ich bin Rect des Zimmer " + i + " und des Datums " + j
            };
        }
    }
}
```

Abb. 43: Code (Erstellung von Rectangles)

Am Ende der äußersten for-Schleife, wird dann noch das Datum oberhalb hinzugefügt (**labelDate**).

```

if (i == 0)
{
    labelDate = new Label()
    {
        Name = "date" + j,
        Content = dateList.ElementAt(j).ToShortDateString(),
        Margin = new Thickness(
            left: dateLeft,
            top: 0,
            right: 0,
            bottom: 0),
        FontSize = 14,
        Tooltip = "ich bin Datum " + j
    };

    dateLeft += 150;
    dateGrid.Children.Add(labelDate);
}

```

Abb. 44: Code (Format für Überschriften)

### 9.7.2.1 fillDateList()

Wie in der oberen Abbildung zu sehen ist, wird für das Datums Label eine Liste von Daten verwendet. Diese werden zuerst mithilfe der Methode **fillDateList()** befüllt. Diese Liste ist abhängig von der Datumsauswahl, dies wird dem Benutzer mit zwei Datepicker (**dpvon**, **dpbis**) ermöglicht.

```

List<DateTime> fillDateList()
{
    TimeSpan daysBetween = dpBis.SelectedDate.Value - dpVon.SelectedDate.Value;
    int Tage = Math.Abs(daysBetween.Days) + 1;
    List<DateTime> List = new List<DateTime>();
    DateTime date = dpVon.SelectedDate.Value;
    for (int i = 0; i < Tage; i++)
    {
        List.Add(date);
        date = date.AddDays(1);
    }
    return List;
}

```

Abb. 45 Code (füllen der Datumsliste)

### 9.7.3 RefreshPage()

Da der Benutzer den Datumsbereich der Übersichtsseite verändern kann, ist die Methode `refreshPage()` notwendig.

Sie hat die Aufgabe, die Liste `booking` mit den passenden Buchungen, die in dem eingegeben Zeitraum fallen, zu befüllen und die Grids für die Zimmer(`roomGrid`), Rechtecke(`rectGrid`) und Daten(`dateGrid`) zu löschen, weil diese wieder von `createRectangles()` befüllt werden.

```
void refreshPage()
{
    booking = hm.repBuchung.Get(includes: "Kunde").Where(a => a.bisBuchung >= dpVon.SelectedDate).Where(b => b.vonBuchung <= dpBis.SelectedDate).ToList();

    rectGrid.Children.Clear();
    roomGrid.Children.Clear();
    dateGrid.Children.Clear();
    createRectangles();
}
```

Abb. 46: Code (Aktualisiert die Übersicht)

### 9.7.4 Export CSV

In der Methode `export_Click` wird der Export der E-Mail Adressen der Kunden realisiert. Die Datei kann dann mittels einem Speicher(`saveFileDialog1`) Dialog an einem beliebigen Ort gespeichert werden.

```
private void export_Click(object sender, RoutedEventArgs e)
{
    List<Kunde> listK = hm.repKunde.Get().ToList();

    string delimiter = ",";
    StringBuilder sb = new StringBuilder();
    for (int index = 0; index < listK.Count(); index++)
        sb.AppendLine(string.Join(delimiter, listK.ElementAt(index).emailKunde));
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    string filter = "CSV file (*.csv)|*.csv| All Files (*.*)|*.*";
    saveFileDialog1.Filter = filter;
    StreamWriter writer = null;

    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        filter = saveFileDialog1.FileName;
        writer = new StreamWriter(filter);
        writer.WriteLine(sb.ToString());
        writer.Close();
    }
}
```

Abb. 47 Code (Export CSV)

### 9.7.5 Lock\_rooms()

**Lock\_room()** ist die Methode, die für das Sperren der Zimmer notwendig ist. Diese ermöglicht es, mit dem vorgefertigten Property **BlackoutDates** alle Daten in den Datepickern (**dpVon**, **dpBis**) nicht selektierbar zu machen, für die eine Sperrung vorgesehen ist. Die Sperrungen sind in der Datenbank gespeichert.

```
public void Lock_rooms()
{
    sper = (List<Sperrung>)hm.repSperrung.Get().ToList();
    foreach (var s in sper)
    {
        dpVon.BlackoutDates.Add(new CalendarDateRange(s.vonSperrung, s.bisSperrung));
        dpBis.BlackoutDates.Add(new CalendarDateRange(s.vonSperrung, s.bisSperrung));
    }
}
```

Abb. 48 Code(Daten für Zimmer sperren)

### 9.7.6 createBill()

Diese Methode ist zuständig für die Erstellung der Rechnung. Dabei wird eine Excel Datei generiert und mit den korrekten Daten befüllt. Um eine Excel Datei zu erstellen, ist es notwendig, eine Referenz hinzuzufügen, welche Microsoft Excel 15.0 Objekt Larry heißt.

Dabei wird ein Worksheet Variable (**ws**) angelegt und mit der vorhandenen Funktion **Range** ist es möglich Werte in eine beliebige Zelle zu schreiben.

```
public Workbook createBill()
{
    Microsoft.Office.Interop.Excel.Application app = new Microsoft.Office.Interop.Excel.Application();
    app.Visible = true;
    app.WindowState = XlWindowState.xlMaximized;

    string path = System.IO.Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location) + "\\Excel\\RechnungZimmer.xlsx";
    Workbook wb = app.Workbooks.Open(path);
    Worksheet ws = wb.Worksheets[1];
    DateTime currentDate = DateTime.Now;
    ColumnData data = ReservierungData.SelectedItem as ColumnData;

    //Anpassung an Rechnungsvorlage
    ws.Range["A19"].Value = "=" + data.buchung.erwachseneBuchung + "+" + data.buchung.kinderBuchung;
    ws.Range["A20"].Value = data.buchung.erwachseneBuchung;
    ws.Range["I9"].Value = currentDate.ToShortDateString();
    ws.Range["E14:F14"].Value = data.buchung.vonBuchung;
    ws.Range["I14"].Value = data.buchung.bisBuchung;
    ws.Range["I19"].Value = data.buchung.preisBuchung;
    ws.Range["I20"].Value = "=" + data.buchung.erwachseneBuchung + "*1";
    ws.Range["I22"].Value = "=I19+I20";
    ws.Range["I25"].Value = "=I22-I23";
    ws.Range["I30"].Value = "=I20";
    ws.Range["D30:E30"].Value = "=I30";

    return wb;
}
```

Abb. 49 Code(Rechnung erstellen)

### 9.7.7 Delegates

Im Zuge dieser Diplomarbeit haben wir Delegates dafür benötigt um eine Referenz von einem Projekt zu einem anderen zu übergehen. Da bereits eine Referenz von der Schnittstellenschicht zur Datenschicht existiert, benötigten wir eine in beide Richtungen. Da Referenzen in beiden Richtungen nicht gesetzt werden können, wurde es mit dem „CallBack“ – Prinzip von Delegates übergangen.

Das „CallBack“ – Prinzip, auch Rückruffunktion genannt, bezeichnet in der Informatik eine Funktion, die einer anderen als Parameter übergeben und von dieser unter gewissen Bedingungen aufgerufen wird.(vgl. [WIKIPEDIA, Rückruffunktion])

Sie wurden vor der Klasse „EFRepository“ deklariert, die eine Buchung als Argumente und einen String für Fehlermeldungen als Rückgabeparameter besitzen. Danach wurde in der „EFRepository“ – Klasse ein Objekt von ihnen erstellt und diese Objekte wurden an der jeweiligen Stelle aufgerufen. Im Manger vom WPF-Projekt (Darstellungsschicht) werden die jeweiligen Methoden in die Liste der Methodenaufrufe vom Objekt hinzugefügt.

## 9.8 Risiken

Risiko	Risikohöhe	Eintrittswahrscheinlichkeit
Aufwand von Schnittstellenkonfiguration und Datenaustausch zu groß (WuBook)	5	4
Funktionsumfang zu groß	4	3
Qualitätsverlust	4	4

*Tabelle 5 Risiken*

## 9.8.1 Aufwand von Schnittstellenkonfiguration und Datenaustausch zu groß (WuBook)

### 9.8.1.1 Risikobeschreibung

Niemand weiß wie wir die Informationen von der externen Schnittstelle WuBook bekommen und somit wissen wir auch nicht ob wir diese verarbeiten können. Die Technologien, die Wubook verwendet, reichen über den Lehrinhalt der HTL-Perg hinaus.

### 9.8.1.2 Lösungsansatz

Wir kontaktierten Herrn Professor Obermüller und stellten ihm unsere offenen Fragen. Nebenbei kommunizierten wir auch mit WuBook für mögliche Tipps, die uns den Verbindungsaufbau zu ihnen erleichterten.

### 9.8.1.3 Gewünschte Auswirkung

Wir erhalten die nötigen Infos und können den Datenaustausch somit erfolgreich umsetzen.

### 9.8.1.4 Unerwünschte Auswirkung

Das nötige Verständnis wird nicht erreicht und wir erhalten somit keine Daten von der Schnittstelle.

## 9.8.2 Funktionsumfang zu groß

### 9.8.2.1 Risikobeschreibung

Es wird während der Entwicklungsphase klar, dass der Funktionsumfang zu groß ist und nicht alle Funktionen bis zum geplanten Projektende umsetzbar sind.

### 9.8.2.2 Lösungsansatz

Mit dem Auftraggeber die Prioritäten der einzelnen Funktionen besprechen und die wichtigsten zuerst umzusetzen.

### 9.8.2.3 Gewünschte Auswirkung

Es wird erkannt, dass einige Funktionen doch nicht so viel Zeit beanspruchen wie geplant und unsere Arbeit somit erfolgreich abgeschlossen werden kann.

### 9.8.2.4 Unerwünschte Auswirkung

Einige Funktionen können nicht fertiggestellt werden, da mehr Probleme auftreten als vorgesehen und der Kunde somit unzufrieden ist.

### 9.8.3 Qualitätsverlust

#### 9.8.3.1 Risikobeschreibung

Dieses Risiko ist teilweise von zu großem Funktionsumfang abhängig, so kann es sein das die Qualität des Gesamtprodukts leidet, wenn bemerkt wird, dass man nicht alle Funktionen im gewünschten Bereich meistern kann. Aufgrund des Zeitdrucks könnte es passieren, dass einige Funktionen zu wenig getestet werden und mögliche Fehler übersehen werden.

#### 9.8.3.2 Lösungsansatz

Einen strukturierten Zeitplan erstellen damit alle Meilensteine pünktlich erledigt werden können.

#### 9.8.3.3 Gewünschte Auswirkung

Alle Meilensteine werden zur gewünschten Zeit erfüllt und somit muss die Testphase nicht darunter leiden.

#### 9.8.3.4 Unerwünschte Auswirkung

Man hält sich nicht an die geplanten Zeitvorgaben der Meilensteine und hat somit nicht mehr genug Zeit zum Testen. Deshalb werden die Tests zu schnell und ungenau durchgeführt und Fehler übersehen.

# 10 Technologien

---

## 10.1 WPF

Windows Presentation Foundation (WPF) ist ein Grafik Framework und ein Teil des .Net Frameworks. Es dient zur Gestaltung von Oberflächen und zur Integration von Animation und Multimedia Komponenten. Es ist ein umfangreiches Modell, welches die Präsentation und Geschäftslogik trennt. So wird die grafische Oberfläche mit XAML (basiert auf XML) beschrieben und die Logik mit gewöhnlichen C# Code. WPF ist seit dem .NET-Framework 3.0 verfügbar.

Außerdem können Windows-Forms Steuerelemente in WPF verwendet werden und umgekehrt. WPF beinhaltet vordefinierte Steuerelemente wie Menüs, Listen usw. Die Steuerelemente haben ein Default Layout, dieses kann aber einfach mit der Style Property verändert werden. (vgl. [WIKIPEDIA])

## 10.2 XAML

XAML ist eine Beschreibungssprache für die GUI – Gestaltung (Oberflächengestaltung) und wurde von Microsoft entwickelt. XAML basiert auf XML, wie der Name schon verrät. (vgl. [WIKIPEDIA])

## 10.3 C# (C – Sharp)

C-Sharp ist eine robuste Programmiersprache die von vom Softwarehersteller Microsoft entwickelt wurde. C# ist eine objektorientierte Programmiersprache und die Anwendungen werden im .NET – Framework ausgeführt. Mit C-Sharp werden viele komplizierte Anweisungen von C++ vereinfacht. (vgl. [MICROSOFT CORPORATION])

## 10.4 GIT Hub

GIT ist im Allgemeinen ein Versionsverwaltungssystem. Ein Versionsverwaltungssystem ermöglicht, dass mehrere Entwickler auf dem gleichen Projekt arbeiten können ohne dabei sich Anweisungen oder Veränderungen sich gegenseitig zu überschreiben.

GIT Hub ist ein kostenloser File-Hosting Dienst und wird benutzt um die Entwicklung von einer Software zu erleichtern. (vgl. [YEEBASE MEDIA GMBH])

## 10.5 Dropbox

Dropbox ist ein Filehosting-Dienst von dem Unternehmen Dropbox Inc. Mit Dropbox kann man Daten online speichern und von jedem Standort mit Internetzugang zugreifen. Es dient hauptsächlich dazu, Daten einfach und schnell auszutauschen und sicher zu speichern. (vgl. [WIKIPEDIA])



Abb. 50 Logo: Dropbox (vgl. [WIKIPEDIA])

## 10.6 Visual Studio

Visual Studio ist eine Entwicklungsumgebung für verschieden Programmiersprachen, hauptsächlich für .NET-Anwendungen. Es lassen sich Webseiten sowie auch Webservices für das Internet oder auch Apps entwickeln. (vgl. [WIKIPEDIA])



Abb. 51: Logo Visual Studio (vgl. [Microsoft])

## 10.7 DBDesigner Fork

DBDesigner Fork ist ein visuelles Werkzeug für die Entwicklung eines Datenbankmodells. Diese Software ermöglicht es, Tabellen inklusive ihrer Attribute darzustellen. Auch die Beziehungen können somit einfach gesetzt werden. Somit hilft es, dass man keine Schlüssel falsch vergibt.

Auch ein einfacher Import sowie Export Funktion eines SQL Scripts ist implementiert.

## 10.8 ChannelManager (WuBook)

Wubook wurde 2011 gegründet und deren Standorte verbreiten sich weltweit. Es ist ein online Buchungstool und bietet viele Services. Um Buchungen von diversen Online – Portalen (booking.com, hrs, usw.) mit einer eigenen Applikation zu verwalten, kommt Wubook als Schnittstelle ins Spiel und tauscht via XML oder JSON Daten aus.

## 10.9 Booking.com

Booking.com ist ein Online Buchungsportal, welches es ermöglicht, Hotels auf der ganzen Welt zu buchen. Hotelbesitzer können ihr Hotel einfach anbieten. Dafür muss jedoch für jede Buchung eine Provision abgegeben werden.



Abb. 52: Logo Booking.com (vgl. [WIKIPEDIA])

## 10.10 Datenbank ADO.NET (Access Data Object)

### 10.10.1 ADO.NET Architektur

Ermöglicht den Zugriff auf eine relationale Datenbank. Klassen (Connection-Klassen) stellen die Verbindung zur Datenbank her.

#### Architektur:

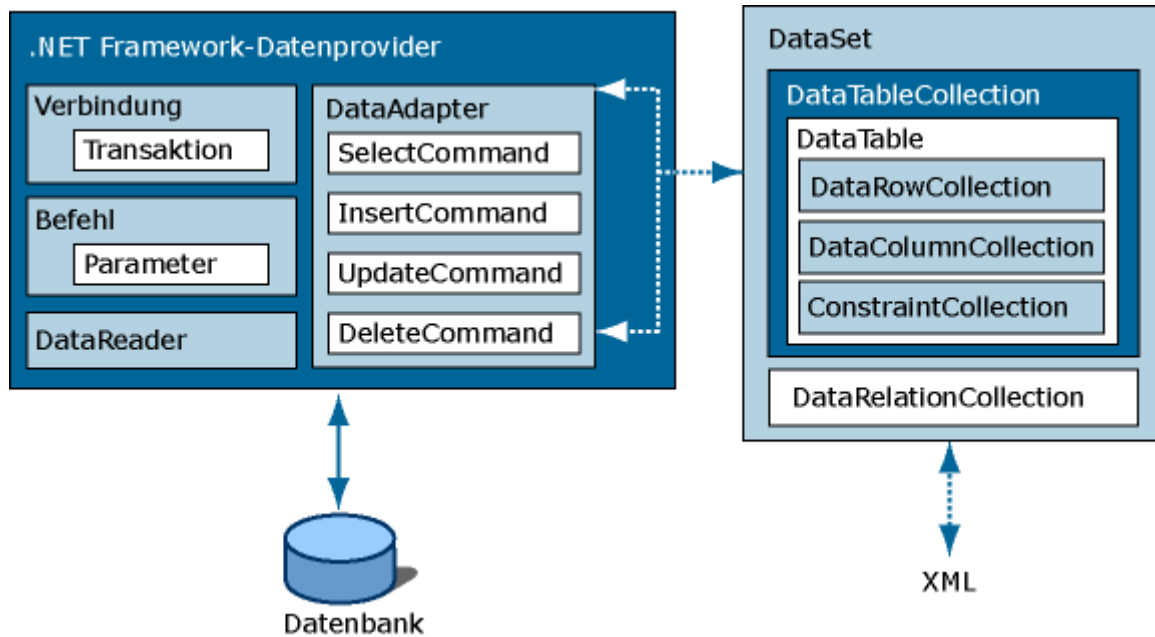


Abb. 53: ADO.NET Architektur (vgl. [Microsoft])

Die Architektur besteht aus zwei Bereichen: .NET Framework-Datenprovider und DataSet.

#### .NET Framework-Datenprovider:

Der Datenprovider ist für die Schnittstelle der Datenbankzuständig und enthält Informationen von der Datenbank.

#### DataSet:

Die Komponente DataSet stellt eine komplette Datengruppe einschließlich verknüpfter Tabellen, Einschränkungen und Beziehungen zwischen Tabellen dar. (vgl. [Microsoft ADO.NET-Datasets])

## 10.10.2 Entity Framework

Es ist ein Objekt-Relationaler Mapper (ORM) zur Abbildung von relationalen Datenbanktabelle auf .NET-Objektstrukturen.

### 10.10.2.1 Zusammenfassung

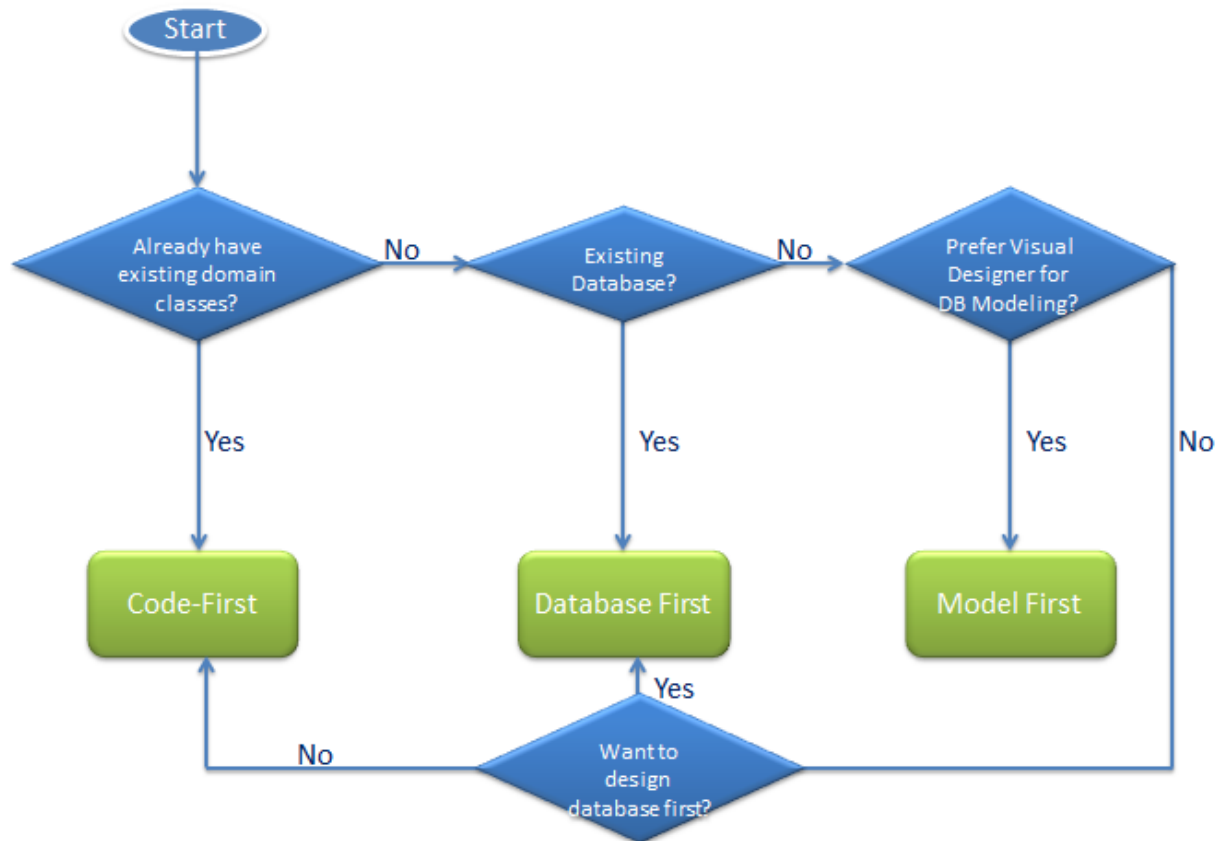


Abb. 54: Decision tree (vgl. [TUTORIAL, Entity Framework])

Wenn eine Applikation mit fertigen Entities existiert, wird **Code First** angewendet. Wenn eine Datenbank mit einem fertigen Datenbankmodell existiert, wird **Database First** verwendet. Im allerletzten Fall, also wenn nur ein Datenbankmodell von einem Designer vorhanden ist, wird **Model First** verwendet.

### 10.10.3 Microsoft SQL – Server:

„Microsoft SQL-Server ist ein relationales Datenbanksystem und wurde im Jahre 1989 von Microsoft entwickelt“.

Der SQL – Server bietet ein vielfältiges Angebot von Systemen und nutzt die SQL-Variante T-SQL. Die aktuelle Version ist SQL Server 2014. Da die Expressversion einfach und kostenlos zu erhalten ist benutzen wir den SQL Server 2014 Express. Die Express-Version enthält die vollständige Version von SQL Server 2014 und ist geeignet für kleiner Server oder Desktopanwendungen. (vgl. [WIKIPEDIA])

## Arten von Express – Versionen:

### 10.10.3.1 LocalDB

Diese Version ist die Basisversion und kann nur im Benutzermodus ausgeführt werden. Im Benutzermodus arbeiten bedeutet, dass die Prozesse nur mit virtuellen Speicher arbeiten. Der Vorteil von dieser Version ist die schnelle Installation ohne dabei aufwändige Konfigurationen durchzuführen.

### 10.10.3.2 Express

Diese Version enthält nur das SQL Server-Datenbankmodul und ist eher für Remoteverbindungen geeignet.

### 10.10.3.3 Express mit Tools

Diese Version ist ein Gesamtpaket mit allen Tools die für die Installation und Konfiguration von SQL Server benötigt werden. Sie liefert zusätzlich noch das Management Studio mit.

## 10.10.4 Datenbankverwaltungssysteme (DBVS)

Ein DBVS ist ein Programm für die Kontrollierung und Verwaltung bzw. Änderungen von Datenbestände. Das DBVS – System bietet die Funktion aufwändige SQL - Statements automatisch zu generieren und diese bei Bedarf wieder zu ändern.

Zugriffe können einfach und schnell mit einem Verwaltungssystem getätigt werden, wie zum Beispiel:

- DML- Anweisungen
  - Lesen (Select)
  - Ändern (Update)
  - Löschen (Delete)
- DDL – Anweisungen
  - wird verwendet um die Datenbankstruktur beschreiben, ändern oder zu löschen.
- DCL – Anweisungen
  - vergibt Berechtigungen und kann diese auch wieder entziehen

### 10.10.4.1 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio ist ein zuverlässiges, kostenloses Datenbankverwaltungssystem.

### 10.10.4.2 Visual Studio Server Explorer

Der Server Explorer von Visual Studio ist eine Serververwaltungskonsole und wird hauptsächlich für Verbindungen für Datenbanken oder Ähnliches benutzt. Mithilfe dieses Tools kann man einfach und schnell Datenbankänderungen vollziehen. Damit nicht bei jeder Änderung bei der Datenbank eine Remotedesktopverbindung oder sonstige Verbindungen wie zum Beispiel mit einer VMware getätigt werden müssen, kann man sich mit dem Server Explorer mit wenig Aufwand mit der Datenbank verbinden und Änderungen oder Inserts tätigen.

## 10.11 Language Integrated Query (LINQ)

LINQ bedeutet Language Integrated Query und filtert aus einer Eingabemenge gewünschte Daten heraus. LINQ bietet Funktionen wie Where, Select, GroupBy, Average, usw. (vgl. [MICROSOFT CORPORATION])

Es wird unterschieden zwischen semantischer (Query Syntax) und methodenbasierter (Method Syntax) Abfrage. In den folgenden Abbildungen sieht man den unterschied der Schreibweise, die Ausgabe der beiden Beispiele ist ident. Bei der Method Syntax wird mit Lambda Ausdrücken gearbeitet.

```
IEnumerable<int> numQuery2 = numbers.Where(num => num % 2 == 0).OrderBy(n => n);
```

*Abb. 55 Beispiel Method Syntax (vgl. [MICROSOFT CORPORATION])*

```
IEnumerable<int> numQuery1 =
    from num in numbers
    where num % 2 == 0
    orderby num
    select num;
```

*Abb. 56 Beispiel Method Syntax (vgl. [MICROSOFT CORPORATION])*

### 10.11.1 Lambda Ausdrücke

Der Ausdruck „num => num % 2 == 0“ wie oben im abgebildeten Beispiel wird als Lambda-Ausdruck bezeichnet. Dies ist eine praktische Möglichkeit, Code zu schreiben, der ansonsten in einer komplexeren Form als anonyme Methode, generischer Delegat oder Ausdrucksbaumstruktur geschrieben werden müsste. Das „=>“ wird in C# als Lambda-Operator bezeichnet, es heißt so viel wie „wechsle zu“.  
(vgl. [MICROSOFT CORPORATION])

## 10.12 Delegates

Ein Delegate, auch als Funktionszeiger bezeichnet ist ein Verweis auf eine Methode. Einfach gesagt leitet es einen Methodenaufruf an eine bestimmte Methode weiter.

Delegates werden grundsätzlich im Bereich von aufwändigen Berechnungen verwendet. Die Umsetzung erfolgt wie bei jeder Verwendung von Delegates, es wird ein Objekt von einem Delegate erstellen, der für überwachen und für Erfolgsmeldungen zuständig ist.

Wie bei Punkt 9.7.7 erklärt, benützen wir Delegates um Referenzen von Projects in beide Richtungen zu setzen.

## 10.13 XML-Technologie

### 10.13.1 XML – String:

Damit XML – Bäume so schnell und einfach wie möglich zum Erstellen sind bietet sich die Klasse XElement an (System.XML.Linq), welche LINQ integriert. Der Konstruktor kann wie folgt aussehen:

- XElement(XElement)
- XElement(XName)
- XElement(XName, Object)
- XElement(XName, Object[])

XName stellt den Namen von einem XML – Element oder Attributs dar.

Um XElement Objekte lokal auf der Festplatte zu speichern oder zu laden wird XmlDocument empfohlen.

#### **Verwendete Methoden (XElement):**

- Element()
  - Diese Methode ruft nur das erste untergeordnete Element auf, die dem „XName“ entspricht.
- Elements(XName)
  - Diese Methode ruft alle untergeordneten Elemente auf, die dem „XName“ entsprechen.
- Descendants(XName)
  - Gibt eine gefilterte Auflistung der Nachfolgeelemente für dieses Element in der Reihenfolge zurück. Es werden diejenigen Elemente zurückgegeben die dem „XName“ entsprechen. (vgl. [XELEMENT, Descendants-Methode])
- Load(Stream)
  - Erstellt mit dem „Stream“ eine neue XElement-Instanz

- Add(Object)
  - Fügt der XML-Struktur ein neues Object (XElement) hinzu.
- Parse()
  - Lädt eine beliebige XML-Struktur in ein XElement

### **Verwendete Methoden (XDocument):**

- Save(String)
  - Serialisiert dieses XDocument und speichert es als Datei in den entsprechenden Pfad
- Load(String)
  - Lädt eine XML-Datei vom entsprechenden Pfad und erstellt damit ein neues XDocument

### 10.13.2 Request & Response:

#### **Request (*deutsch*: Anfrage)**

Ein Request ist eine Anforderung oder Anfrage an einen Server. Server bieten meist Dienste an, wie zum Beispiel Bereitstellung von Daten.

#### **Response (*deutsch*: Antwort)**

Ein Response ist die Antwort des Servers auf die Anfrage vom Client. Diese Antworten werden mithilfe von verschiedenen Protokollen für die erfolgreiche Versendung unterstützt.

#### **Einbinden von System.NET über References:**

- WebRequest
- HttpRequest

#### **Einbinden von System.XML.Linq über References:**

- XElement
- XDocument

```

var req = (HttpWebRequest)WebRequest.Create("https://wubook.net/xrws/");
req.Method = "POST";

xmlFetchRoom = addLcode(xmlFetchRoom);

var bytes = Encoding.ASCII.GetBytes(xmlFetchRoom.ToString());

using (var stream = req.GetRequestStream())
{
    stream.Write(bytes, 0, bytes.Length);
}

using (var stream = new StreamReader(req.GetResponse().GetResponseStream()))
{
    responseRooms = XElement.Parse(stream.ReadToEnd());
    //Ausgabe von XML-Struktur
    Console.WriteLine(responseRooms.ToString());
}

```

Abb. 57: Request zum Server und Response vom Server

Die WebRequest-Klasse sendet eine Anforderung an einen bestimmten Server. Mit der „Create“ – Methode wird eine URI gesetzt. Eine URI ist eine Kennzeichnung mit dem alle Ressourcen oder Objekte die im globalen Web vorhanden sind identifiziert werden können (vgl. [ITWISSEN, URI]). Mit der Methode HTTP **POST** werden die Daten zur Identifikation mitgesendet. Die Variable „xmlFetchRoom“ wird mithilfe der Methode „addLcode(XElement xmlString)“ gesetzt. Die Methode „addLcode(XElement xmlString)“ fügt den richtigen Code für die Identifikation der vorgefertigten XML-Struktur („xmlFetchRoom“) hinzu.

Die Methode „GetRequestStream()“ gibt eine zum Senden von Daten zu verwendendes Stream-Objekt zurück. Mit der Methode „Write()“ wird ein Request zum Server gesendet.

Wenn der Request erfolgreich gesendet worden ist, sendet der Server einen Response mit den richtigen und gewünschten Daten oder eine Fehlermeldung.

Mithilfe der Methode „Parse()“ wird der Stream in ein XElement umgewandelt.

# 11 Qualitätssicherung

---

Um eine fehlerfreie Software ausliefern zu können, ist es nötig alle Funktionen zu jedem Änderungszeitpunkt zu testen. Dabei ist es notwendig alle möglichen Fälle genau durchzuarbeiten. Wenn man dies gewissenhaft erledigt, erspart man sich im weiteren Entwicklungsprozess mögliche Fehlersuchen und somit auch Zeit.

Durch regelmäßiges Testen stießen wir kaum auf größere Probleme, lediglich die Anpassung der Software an das veränderte Datenbankmodell führte zu einer längeren Fehlersuche.

## 11.1 Qualitätsmerkmale

Da unsere Software eine Arbeitserleichterung für die Rezeption darstellt, ist die Langlebigkeit dieses Produktes gegeben.

Die übersichtlich gestaltete Grafikmaske garantiert eine einfache Handhabung und Bedienung unseres Produkts.

Die gewissenhaft durchgeführten Testfälle gewährleisten die Korrektheit unseres Produkts.

# 12 Testen – Inbetriebnahme

---

## 12.1 Einleitung

Damit eine Software auch sinnvoll verwendet werden kann, zählt das Testen zu den wichtigsten Bestandteilen im Entwicklungsprozess. Deswegen versuchten wir bei unserer Diplomarbeit, direkt nach Fertigstellung einer Funktion, alle möglichen Fälle zu testen.

Das soll die Korrektheit unseres Produkts bereits vor Veröffentlichung gewährleisten und somit auch eine hohe Qualität für das Endprodukt.

## 12.2 Testfälle

Nummer	Testfall	Erwartetes Ergebnis	Erfolgreich?
1	Reservieren	Man trägt eine Buchung ein. Diese soll erfolgreich in die Datenbank gespeichert werden.	ja
2	Reservierung bearbeiten	Falls Änderungen auftreten, können Buchungen bearbeitet werden.	ja
3	Übersicht anzeigen	Es erscheint ein Übersichtsraster, wo freie und gebuchte Zimmer angezeigt werden.	ja
4	Rechnung erstellen	Man wählt eine Buchung für die man eine Rechnung benötigt aus und erhält ein ausgefülltes Excel-Sheet	ja
5	Zimmer zuordnen	Es erscheint eine Liste aller Buchungen von booking.com denen noch kein Zimmer zugeordnet ist. Durch eine Combobox kann man ein gewünschtes Zimmer zuordnen.	ja
6	Kundenverwalten	Man kann neue Kunden anlegen oder bestehende bearbeiten.	ja
7	Kunden E-Mail Adressen exportieren	Man kann die E-Mail Adressen an einen beliebigen Speicherort als CSV Datei speichern.	ja
8	Buchungsliste drucken	Eine Liste aller vorhandenen Buchungen in einem bestimmten Zeitraum kann erstellt und gedruckt werden.	ja
9	Zimmer sperren	Alle Zimmer werden für den gewählten Zeitraum gesperrt.	ja
10	Reservierungen anfordern	Neue Reservierungen von booking.com werden in die Datenbank gespeichert.	ja
11	Zimmertypen anfordern	Zimmertypen werden in die Datenbank gespeichert	ja

Tabelle 6 Testfälle

## 12.2.1 Reservieren

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Buttons „Reservieren“ oder Klick auf ein grünes Rechteck in der Übersichtsseite
3. Erwartetes Ereignis: Eingabemaske für neue Reservierung entweder mit neuen oder bestehenden Kunden.
4. Ergebnis: Eingabemaske erscheint und Reservierung wird korrekt in die Datenbank gespeichert

Abb. 58 Testfall: Zimmer reservieren/buchen

## 12.2.2 Reservierung bearbeiten

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Menüpunktes „Reservierung verwalten“
3. Filtern der Reservierungen
4. Gewünschte Buchung auswählen
5. Erwartetes Ereignis: Korrekte Reservierung soll geladen werden und kann bearbeitet werden
6. Ergebnis: Korrekte Reservierung kann bearbeitet werden und die Änderungen werden in die Datenbank gespeichert

**Filterfunktionen:**

Zimmer und Datum suche  
 Kunden und Datum suche

Zeitraum: von  bis  Kunde auswählen:  Suchen

idKunde	Name Kunde	Adresse	Telefonnummer	Piz- Ort	Datum von	Datum bis	Zimmer
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	13 Dec 2015	30 Dec 2015	Zimmer 5
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	22 Jan 2016	23 Jan 2016	Zimmer 3
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	21 Jan 2016	26 Jan 2016	Zimmer 5
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	20 Jan 2016	21 Jan 2016	Zimmer 13
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	31 Jan 2016	01 Feb 2016	Zimmer 9
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	19 Jan 2016	20 Jan 2016	Zimmer 14
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	21 Jan 2016	22 Jan 2016	Zimmer 23
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	27 Jan 2016	30 Jan 2016	Zimmer 8
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	04 Feb 2016	05 Feb 2016	Zimmer 7
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	03 Feb 2016	04 Feb 2016	Zimmer 1
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	10 Feb 2016	11 Feb 2016	Zimmer 3
3	Stefan Pilz	MusterStraße 1	00000	4362 Bad Kreuzen	13 Feb 2016	14 Feb 2016	Zimmer 7

Vorname:   
 Nachname:   
 Adresse:   
 Telefonnummer:   
 Ort:   
 Datum von:   
 Datum bis:   
 Zimmer:   
 Anzahl Erwachsene:   
 Anzahl Kinder:

Abb. 59 Testfall: Reservierung bearbeiten

### 12.2.3 Übersicht anzeigen

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Buttons „Übersicht“
3. Erwartetes Ereignis: Übersichtsraster soll angezeigt werden
4. Ergebnis: Übersichtsraster wird angezeigt

Zeitraum: von  bis  Drucken

	16.02.2016	17.02.2016	18.02.2016	19.02.2016	20.02.2016	21.02.2016	22.02.2016	23.02.2016	24.02.2016	25.02.2016	26.02.2016
Zimmer 1			Stefan Pilz Badste								
Zimmer 2	Stefan Pilz Bad										Josef Scheifer Neuer Kunde
Zimmer 3											
Zimmer 4									Fabian Radinger		
Zimmer 5											
Zimmer 6											
Zimmer 7									Simon Krottsmayer Nur Frühstück		Tobias Radinger Halbpension
Zimmer 8											
Zimmer 9											
Zimmer 10									Fabian Radinger Fest		
Zimmer 11											
Zimmer 12											
Zimmer 13											
Zimmer 14											
Zimmer 15											
Zimmer 16											
Zimmer 17											
Zimmer 18											

Abb. 60 Testfall: Übersichtsraster

## 12.2.4 Rechnung erstellen

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Buttons „Rechnung“
3. Filtern der Rechnungen
4. Gewünschte Reservierung auswählen
5. Erwartetes Ereignis: Ein Excel-File soll erstellt werden und die korrekten Daten enthalten
6. Ergebnis: Excel-File mit den richtigen Daten für die gewählte Buchung wird erstellt


			
Josef Leopold Schiefer, 4362 Bad Kreuzen 9 Tel. +43 / (0)7266 / 6212    info@gasthof-schiefer.at    www.gasthof-schiefer.at UID: ATU 6539 4815			
		Bad Kreuzen,	14.03.2016
		Rgnr.	376
	Aufenthalt vom	06.02.2016	bis 07.02.2016
RECHNUNG			
	2 Übernachtung Doppel Zimmer	€	72,00
	2 Ortstaxe pro Tag und erw. Person	á €	1,00 € 2,00
		<b>Zwischensumme</b>	€ 74,00
		abzügliche Anzahlung	€
		<b>Gesamtsumme brutto</b>	<b>€ 74,00</b>
Mehrwertssteueraufstellung:			
Steuer-	Netto	MWSt.	Brutto
satz			
0% Ortstaxe	€ 2,00 €	-	€ 2,00
13% Übernachtung	€ 63,72 €	€ 8,28 €	€ 72,00
Danke für Ihren Besuch			
Auf Wiedersehen in Bad Kreuzen			
Josef Schiefer und sein Team			

Abb. 61 Testfall: Automatisch generierte Rechnung

## 12.2.5 Zimmer zuordnen

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Menüpunktes „Zimmer zuordnen“
3. Eine Liste aller Reservierungen die keinen Zimmer zugeordnet sind erscheint
4. Gewünschtes Ereignis: Zimmer kann nach Auswahl der gewünschten Reservierung zugeordnet werden.
5. Ergebnis: Reservierung kann ausgewählt und das Zimmer zugeordnet werden, Änderung wird in die Datenbank übernommen.

**Zimmer zuordnen:**

Name Kunde	Telefonnummer	Plz - Ort	Adresse	Datum von	Datum bis	Zimmer
Stefan Pilz	00000	4392 Dorfstetten	MusterStraße 1	16 Feb 2016	18 Feb 2016	

Vorname:

Nachname:

Adresse:

Telefonnummer:

Ort:

Datum von:

Datum bis:

Zimmer:

Anzahl Erwachsene

Anzahl Kinder

- Zimmer 1
- Zimmer 2
- Zimmer 3
- Zimmer 4
- Zimmer 5
- Zimmer 6
- Zimmer 8
- Zimmer 9
- Zimmer 11
- Zimmer 12
- Zimmer 13
- Zimmer 14
- Zimmer 15
- Zimmer 16
- Zimmer 17
- Zimmer 18
- Zimmer 19
- Zimmer 20

Abb. 62 Testfall: Zimmer zuordnen

## 12.2.6 Kunden bearbeiten

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Menüpunktes „Übersicht Kunden“
3. Auswahl des Gewünschten Kunden
4. Gewünschtes Ereignis: Dialog Fenster soll erscheinen
5. Ergebnis: Dialog Fenster erscheint und Textfelder sind mit den vorhandenen Daten gefüllt

Kunden Dialog

### Kundenverwaltung

Vorname:

Nachname

Adresse

PLZ

Ort

E-Mail

Telefon

Abb. 63 Testfall: Dialog Fenster zum Kunden bearbeiten

## 12.2.7 Kunden E-Mail Adressen exportieren

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Menüpunktes „Übersicht Kunden“
3. Auswahl des Buttons „exportieren“
4. Gewünschtes Ereignis: E-Mail Adressen sollen in eine CSV-Datei gespeichert werden
5. Ergebnis: Speicher Dialog Fenster erscheint und die erstellte CSV-Datei kann an einem beliebigen Ort gespeichert werden.

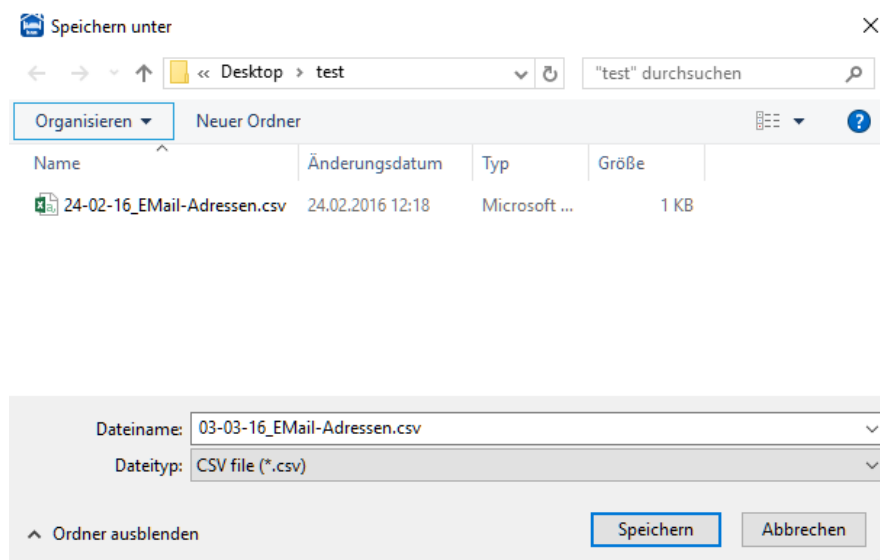


Abb. 64 Testfall: Speicher Dialog für CSV Datei

## 12.2.8 Buchungsliste drucken

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Buttons „Übersicht“
3. Auswahl des Zeitbereichs mittels zwei Datepicker (von & bis)
4. Auswahl des Buttons „Drucken“
5. Gewünschtes Ereignis: Liste aller Buchungen im gewählten Zeitraum soll erstellt werden und gedruckt werden können
6. Ergebnis: Druckvorschau der Liste wird angezeigt, diese kann auch gedruckt werden

Name	Aufenthalt von	Aufenthalt bis	Zimmer	Erw/Kind	
Stefan Pilz	31.01.2016	01.02.2016	9	3/0	(Halbpension)
Stefan Pilz	03.02.2016	04.02.2016	1	1/0	(nur Frühstück)
Fabian Radinger	04.02.2016	05.02.2016	1	2/0	(mit Haustier)
Florian Schimmer	04.02.2016	05.02.2016	2	3/1	(Wieder ein Test)
Stefan Pilz	04.02.2016	05.02.2016	7	2/0	(nur Frühstück)
Fabian Radinger	05.02.2016	07.02.2016	5	1/0	0
Simon Kroissmayr	06.02.2016	07.02.2016	12	2/0	0
Florian Schimmer	06.02.2016	07.02.2016	10	1/0	(nur Frühstück)
Simon Kroissmayr	07.02.2016	10.02.2016	1	2/0	(Halbpension)
Stefan Pilz	10.02.2016	11.02.2016	3	2/0	(Halbpension)
Stefan Pilz	13.02.2016	14.02.2016	7	1/1	(Nur Frühstück)
Sepp Aigner	15.02.2016	16.02.2016	2	2/0	(allergisch gegen Nüsse)
Stefan Pilz	15.02.2016	17.02.2016	22	2/0	0
Stefan Pilz	16.02.2016	18.02.2016		2/0	(U)
Stefan Pilz	18.02.2016	19.02.2016	1	1/0	(Halbpension)
Simon Kroissmayr	23.02.2016	26.02.2016	7	1/0	(Nur Frühstück)
Fabian Radinger	23.02.2016	26.02.2016	4	2/0	0
Fabian Radinger	24.02.2016	26.02.2016	10	2/2	0
Tobias Radinger	26.02.2016	27.02.2016	7	2/1	(Halbpension)
Josef Schiefer	26.02.2016	27.02.2016	2	2/0	(Neuer Kunde)

Abb. 65 Testfall: Druckvorschau für Buchungsliste

## 12.2.9 Zimmer sperren

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Buttons „Übersicht“
3. Auswahl des Buttons „Zimmer sperren“
4. Erwartetes Ereignis: Ein Dialog Fenster soll sich öffnen, in dem man den Zeitraum eintragen kann, anschließend sollen alle Zimmer im gewählten Zeitraum gesperrt sein
5. Ergebnis: Im Dialog Fenster wird der gewünschte Zeitraum eingetragen und der gewählte Zeitraum wird für alle Zimmer gesperrt, außerdem wird es im Übersichtstraster farblich gekennzeichnet.

Abb. 66 Testfall: Zimmer sperren

## 12.2.10 Reservierung anfordern

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Menüpunktes „Reservierung anfordern“
3. Erwartetes Ereignis: Neue Reservierungen sollen in der Datenbank gespeichert sein
4. Ergebnis: Neue Reservierungen sind in der Datenbank

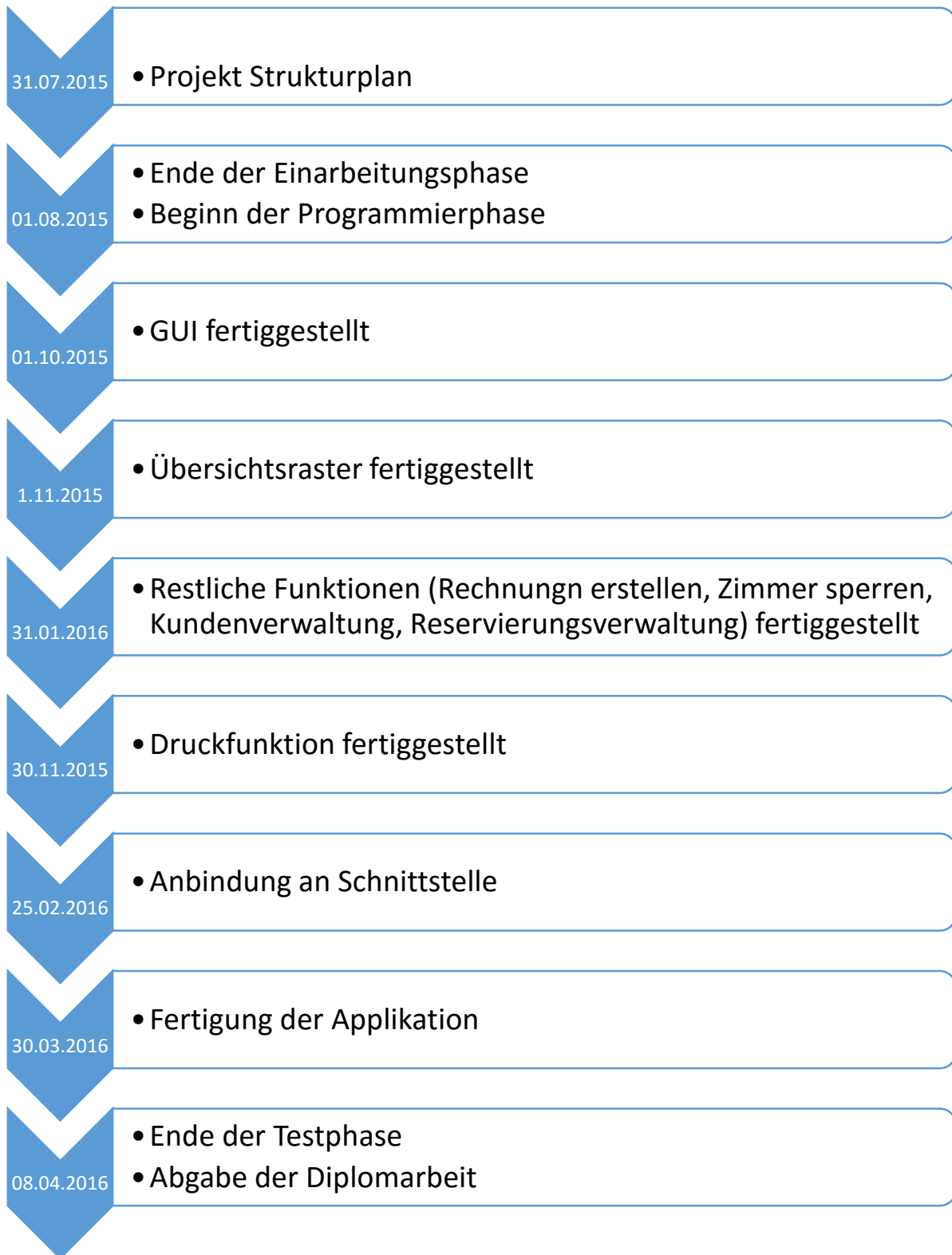
## 12.2.11 Zimmertypen anfordern

1. Starten der Desktop-Applikation „Hotel Room Manager“
2. Auswahl des Menüpunktes „Zimmer anfordern“
3. Erwartetes Ereignis: Aktuelle Zimmertypen sollen in die Datenbank gespeichert werden
4. Ergebnis: Neue Zimmertypen sind in die Datenbank übernommen worden

# 13 Auszug aus dem Projekthandbuch

---

## 13.1 Meilensteine



# 14 Installation

## 14.1 Anmeldung bei flexerasoftware.com:

Die Anmeldung bei „flexerasoftware“ ist Pflicht, um die Zusatzfunktion „InstallShield Limited Edition for Visual Studio“ zu downloaden. Für die Echtheit der Daten, erstellten wir eine neue Email für unsere Diplomarbeit (diplomarbeithrm@gmail.com). Nach der Anmeldung und dem Download wird eine Serial Nummer generiert und diese wird danach bei der Installation benötigt.

## 14.2 Installation:

Nach der Installation aktiviert es eine Zusatzfunktion in Visual Studio unter dem Menüpunkt „Other Project Types“ und diese muss jetzt nur mehr zum Installierenden Projekt hinzugefügt werden.

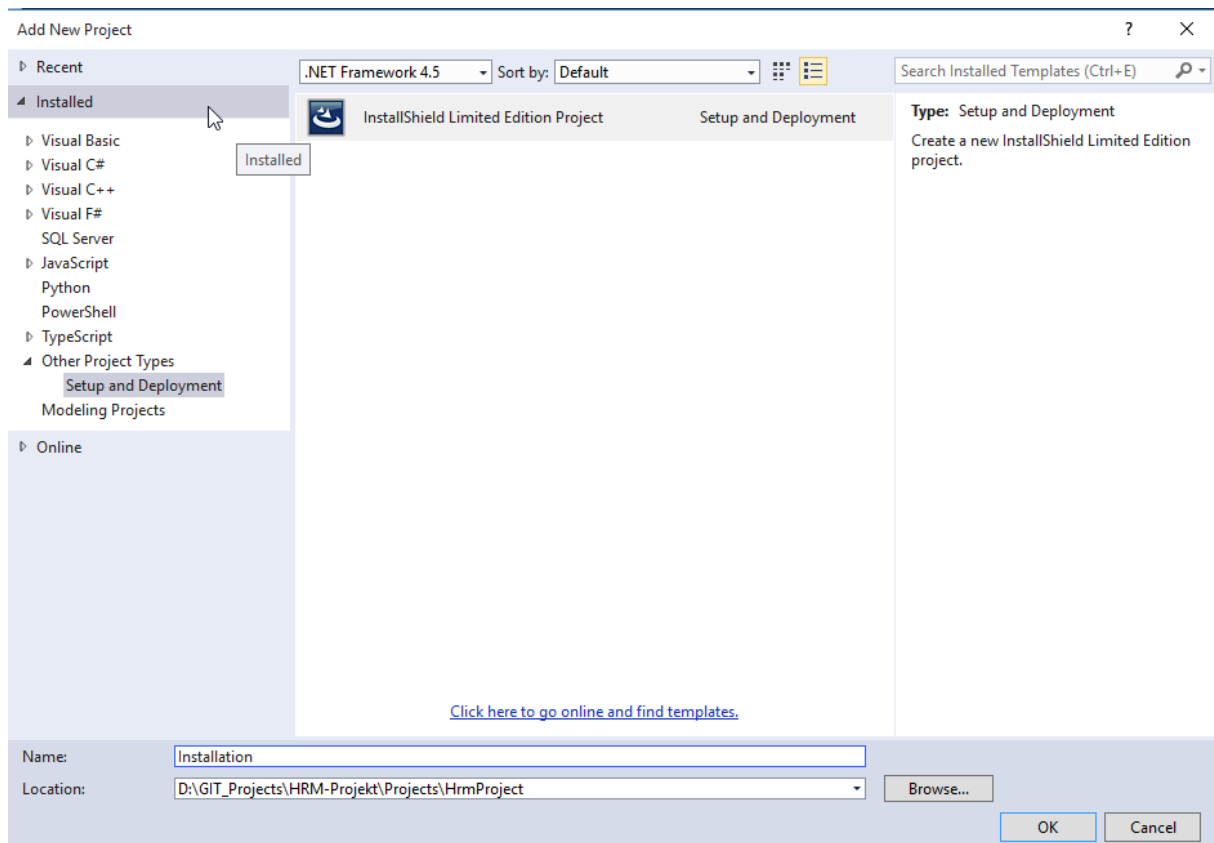


Abb. 67: Neues InstallShield Projekt

## 14.3 Verwendung des Tools

Der Reiter „Project Assistant“ führt das Erstellen von einer Installationsdatei aus. Auf der Hauptseite werden allgemeine Tipps vorgeführt. Rechts unten kann man mit einem Pfeil die einzelnen Schritte durchführen.

1. Auf der ersten Seite gibt man den Firmennamen, den Applikationsnamen und die Version an. Auch ein Icon kann ausgewählt werden.
2. Auf der zweiten Seite kann man die Mindestanforderung von einem Betriebssystem aussuchen. Auch die benötigten Programme für die Applikation können ausgewählt werden.
3. Auf der dritten Seite wird der Speicherort des Programms festgelegt
  - a. Hier können auch die Project Outputs hinzugefügt werden
  - b. In unserer Diplomarbeit haben wir von jedem einzelnen Project den „Primary Output“ hinzugefügt

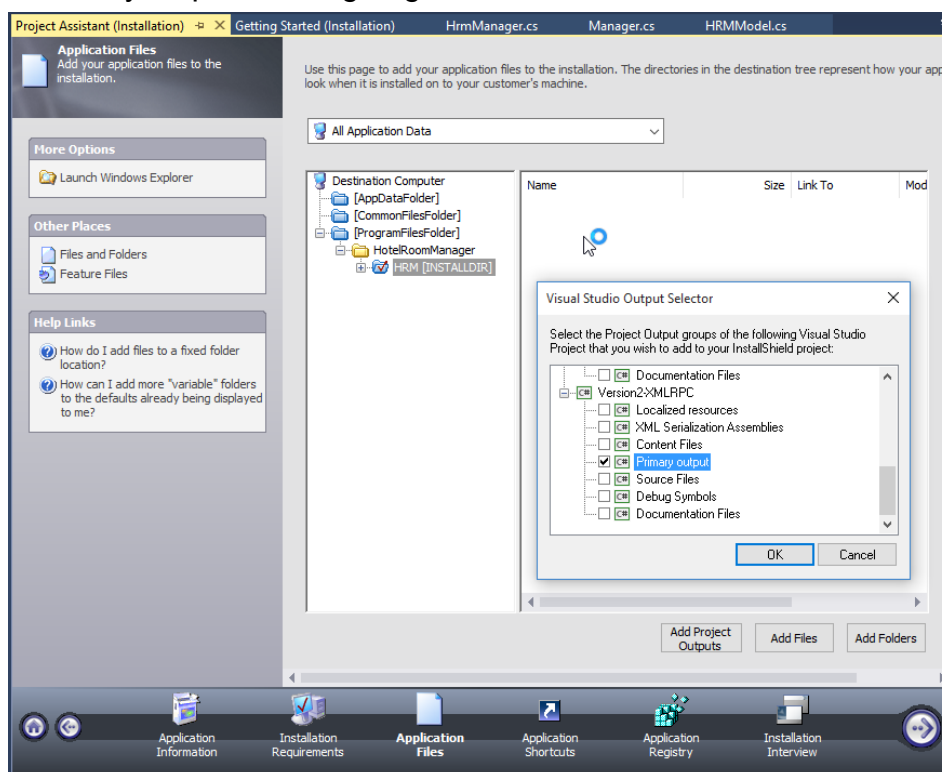


Abb. 68: Install Shield- Directory

4. Auf der vierten Seite können Shortcuts festgelegt werden
5. Auf der letzten Seite können noch Standardeinstellungen festgelegt werden wie zum Beispiel das Anzeigen von Lizenzvereinbarung.

## INSTALLATION

Unter dem Reiter „4 Customize the Setup Appearance“ können die Installations Fenster formatiert und designt werden.

Wenn alle Einstellungen festgelegt worden sind muss nur noch mehr eine Rechtsklick auf das Installationsprojekt tätigen werden und auf „Install“ klicken werden, dann wird die Applikation auf dem eigenen Laptop installiert. Um eine .exe auszuführen damit die Installation beginnt, speichert Visual Studio sie unter dem relativen Pfad „\Installation\Express\DVD-5\DiskImages\DISK1“.

# 15 Zusammenfassung

---

In diesem Kapitel wird die Diplomarbeit noch einmal kurz zusammengefasst. Es wird sowohl auf das Ergebnis, unsere persönlichen Erfahrungen, als auch auf die Verteilung der Funktionen eingegangen.

## 15.1 Ergebnis

Im Bearbeitungszeitraum dieser Diplomarbeit wurde ein Hilfstool für eine Rezeption eines Gasthofs entwickelt.

Konkret wurde eine WPF – Anwendung entwickelt. Somit wurde die grafische Oberfläche mit XAML programmiert. Der Aufbau unserer Applikation erfolgt über das „Model View Controller“ Prinzip. Unsere Anwendung greift mit Hilfe des Entity Frameworks auf die Datenbank zu, welche von uns neu konfiguriert werden musste, da der Gasthof noch keine Datenbank besaß. Entwickelt wurde das Produkt mit Visual Studio.

Der Benutzer muss sich beim Start unserer Anwendung anmelden, die Benutzerdaten sind ebenfalls in unserer Datenbank gespeichert. Wenn die Anmeldung erfolgreich war, verfügt der Benutzer über diverse Funktionen die ihm seine Rezeptionsarbeiten erleichtern.

Mithilfe einer gut detailliert durchgeführten Planungsphase, ist es uns gelungen, alle Funktionen die der Auftraggeber gewünscht hat zu implementieren.

## 15.2 Resümee

Diese Diplomarbeit ermöglichte uns, unser Wissen in diversen Bereichen zu erweitern. Im Zeitraum der Diplomarbeit mussten wir uns intensiv mit der Entwicklung von WPF-Anwendungen und den Datenaustausch via XML und JSON auseinandersetzen.

Zusätzlich konnten wir unsere Fähigkeiten in der Projektentwicklung erweitern. Uns wurde während der Diplomarbeit bewusst, wie wichtig es ist, Meilensteine und Deadlines festzulegen um somit an das gewünschte Ziel zu kommen. Ein weiterer wichtiger Punkt ist die Kommunikation im Team. Da unser Team nur aus zwei Mitgliedern besteht, fällt das Erarbeiten von Lösungen leichter.

Eine der größten Herausforderungen stellte die Kommunikation via XML oder JSON mit der Schnittstelle Wubook dar. Anfangs versuchten wir mit einem „Parser“, Daten direkt von der Webseite von booking.com zu erlangen. Da aber bei jeder Änderung der Webseite der „Parser“ neu programmiert werden müsste, ist dies keine langfristige Lösung. Daher ist eine Schnittstelle die einzige optimale Lösung um Daten auszutauschen.

Auch die Schnittstelle hat uns einige Probleme bereitet. Eines der größten Probleme war, dass unser Auftraggeber nie einen Hinweis bekommen hat, dass ab der Aktivierung der Schnittstelle keine Buchungen per Fax zugesendet werden. Somit bekam er keine Informationen mehr ob Zimmer gebucht worden sind und hat den Personen die telefonisch gebucht haben, bereits gebuchte Zimmer zugesichert.

Durch die Entwicklung dieser Diplomarbeit sammelten wir wichtige Erfahrungen für unseren späteren Berufsweg, da wir uns selbstständig und intensiv mit einem Thema auseinander setzen mussten.

## 15.2.1 Stefan Pilz

### 15.2.1.1 Kernfunktionen:

- GUI
- Kommunikation
  - Auftraggeber
  - Wubook
- Übersicht von Buchungen in einer Kalenderansicht
- Funktion für die Erstellung von Rechnungen
- Druckfunktion

## 15.2.2 Fabian Radinger

### 15.2.2.1 Kernfunktionen:

- Login
- Nebenfunktion und GUI
- Wubook Konfiguration
- Datenbank
  - Datenbank installieren
  - Datenbankmodell erstellen
  - Datenbankmodell installieren

## 15.3 Aufwandsverteilung

In dem folgenden Diagramm sieht man die Aufwandsverteilung. Diese beinhaltet die Einarbeitung, Programmierung, Testen, Dokumentation, Diplomschrift und die Vorbereitungen für die Präsentation.

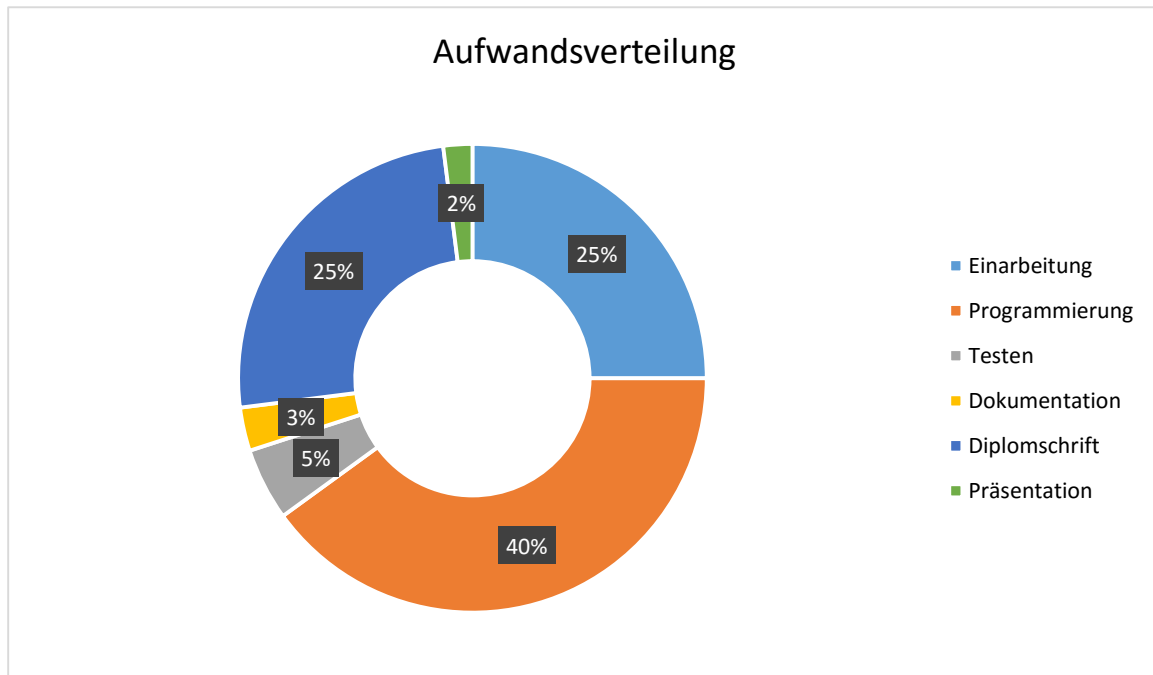


Abb. 69 Aufwandsverteilung

Der größte Aufwand war bei unserer Diplomarbeit, die Diplomschrift und die Programmierung der Software. Auch die Einarbeitung war sehr zeitintensiv, da wir zu Beginn nicht wussten, wie wir den Datenaustausch mit booking.com realisieren sollen.

Das Testen kostete uns erstaunlicherweise nicht so viel Zeit wie wir dachten, aber auch die Verfassung der Dokumentation und die Vorbereitung auf die P@bs Präsentation waren nicht sehr aufwendig.

# 16 Literatur und Quellen Verzeichnis:

---

## 16.1 Abbildungsverzeichnis

Abb. 1: Herr Pilz Stefan .....	12
Abb. 2: Herr Radinger Fabian	12
Abb. 3: Herr Pilz Stefan .....	12
Abb. 4: Herr Radinger Fabian .....	12
Abb. 5: HTL – Perg (vgl. [HTL Perg])	12
Abb. 6: Herr Radinger Fabian.....	12
Abb. 7 Projektumfeld, Hierarchie .....	15
Abb. 8: HTL – Perg (vgl. [HTL Perg]) .....	15
Abb. 9: Herr Prof. Mag. Rupert Obermüller (vgl. [LEHRER, HTL Perg]) .....	16
Abb. 10: Logo „Zur Zugbrücke“ (vgl. [SCHIEFER, Landgasthof]) .....	16
Abb. 11: Herr Josef Schiefer (vgl. [SCHIEFER, Landgasthof]) .....	16
Abb. 12: Gant Project Plan .....	18
Abb. 13 Projektstrukturplan .....	20
Abb. 14 Konfigurationsplan.....	21
Abb. 15 Use-Case Gesamtsystem .....	24
Abb. 16: Screenshot von tdocs.Wubook (vgl. [DOC, Wubook Wired Authentication]) .....	25
Abb. 17: Screenshot von XML-String (acquire_token) .....	26
Abb. 18: Screenshot von tdocs.Wubook (vgl. [DOC, Wubook Wired Fetching reservations] ) .....	27
Abb. 19: Screenshot von tdocs.Wubook (vgl. [DOC, Wubook Wired Fetching reservations]) .....	28
Abb. 20 XML Struktur .....	30
Abb. 21: XML-Struktur.....	31
Abb. 22 Datenbankmodell.....	32
Abb. 23: Code First from database in VisualStudio .....	35
Abb. 24: Code First from database in VisualStudio – Auswahl von Datenbank .	35
Abb. 25 GUI Aufbau der Desktop Applikation.....	36
Abb. 26 Use-Case Funktionalitäten .....	37
Abb. 27 Login .....	38
Abb. 28: Kundenverwaltung.....	38
Abb. 29 Übersicht der Kunden und Export der Mail Adressen.....	39
Abb. 30: Buchen von Zimmern .....	39
Abb. 31: Übersicht von Buchungen in Kalenderansicht .....	40
Abb. 32 Druckvorschau .....	41
Abb. 33 Druckdialog .....	41

Abb. 34 Dialog Fenster: Zimmer sperren.....	42
Abb. 35: Bearbeiten von Reservierungen .....	42
Abb. 36 Zimmer zuordnen.....	43
Abb. 37: Erzeugen von Rechnungen .....	44
Abb. 38: Ergebnis von der Erstellung der Rechnung .....	44
Abb. 39 Menüpunkt Schnittstelle .....	45
Abb. 40: App.xaml Startup Event.....	46
Abb. 41: App.xaml.cs Splash Screen .....	46
Abb. 42: Code (Variablendeklaration für Rectangles) .....	47
Abb. 43: Code (Erstellung von Rectangles) .....	48
Abb. 44: Code (Format für Überschriften) .....	49
Abb. 45 Code (füllen der Datumsliste).....	49
Abb. 46: Code (Aktualisiert die Übersicht).....	50
Abb. 47 Code (Export CSV) .....	50
Abb. 48 Code(Daten für Zimmer sperren) .....	51
Abb. 49 Code(Rechnung erstellen) .....	51
Abb. 50 Logo: Dropbox (vgl. [WIKIPEDIA]).....	56
Abb. 51: Logo Visual Studio (vgl. [Microsoft]).....	56
Abb. 52: Logo Booking.com (vgl. [WIKIPEDIA]).....	56
Abb. 53: ADO.NET Architektur (vgl. [Microsoft]) .....	57
Abb. 54: Decision tree (vgl. [TUTORIAL, Entity Framework]) .....	58
Abb. 55 Beispiel Method Syntax (vgl. [MICROSOFT CORPORATION]) .....	60
Abb. 56 Beispiel Method Syntax (vgl. [MICROSOFT CORPORATION]) .....	60
Abb. 57: Request zum Server und Response vom Server .....	63
Abb. 58 Testfall: Zimmer reservieren/buchen .....	67
Abb. 59 Testfall: Reservierung bearbeiten .....	68
Abb. 60 Testfall: Übersichtsraster .....	68
Abb. 61 Testfall: Automatisch generierte Rechnung .....	69
Abb. 62 Testfall: Zimmer zuordnen .....	70
Abb. 63 Testfall: Dialog Fenster zum Kunden bearbeiten .....	71
Abb. 64 Testfall: Speicher Dialog für CSV Datei .....	72
Abb. 65 Testfall: Druckvorschau für Buchungsliste.....	72
Abb. 66 Testfall: Zimmer sperren .....	73
Abb. 67: Neues InsaalShield Projekt.....	75
Abb. 68: Install Shield- Directory .....	76
Abb. 69 Aufwandsverteilung .....	81

## 16.2 Tabellenverzeichnis

Tabelle 1 Kontaktdaten HTL Perg .....	15
Tabelle 2 Kontaktdaten Rupert Obermüller .....	16
Tabelle 3 Kontaktdaten Gasthof Schiefer .....	16
Tabelle 4 IVM – Matrix .....	19
Tabelle 5 Risiken .....	52
Tabelle 6 Testfälle .....	66

## 16.3 Literaturverzeichnis

DOC, Wubook Wired Fetching reservations. *Fetching reservations*. [online]. Available from World Wide Web: <<http://tdocs.wubook.net/wired/fetch.html>>

DOC, Wubook Wired Authentication. *Wubook Doc*. [online]. Available from World Wide Web: <<http://tdocs.wubook.net/wired/auth.html>>

HTL Perg. [online]. Available from World Wide Web: <<http://www.htl-perg.ac.at/>>

ITWISSEN, URI. *itwissen*. [online]. Available from World Wide Web: <<http://www.itwissen.info/definition/lexikon/uniform-resource-identifier-URI.html>>

LEHRER, HTL Perg. *HTL-Perg Lehrer*. [online]. Available from World Wide Web: <<https://www.htl-perg.ac.at/organisation/lehrer>>

Microsoft. [online]. Available from World Wide Web: <[https://msdn.microsoft.com/de-de/library/aa719511\(v=vs.71\).aspx](https://msdn.microsoft.com/de-de/library/aa719511(v=vs.71).aspx)>

Microsoft ADO.NET-Datasets. [online]. Available from World Wide Web: <[https://msdn.microsoft.com/de-de/library/zb0sdh0b\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/zb0sdh0b(v=vs.110).aspx)>

MICROSOFT CORPORATION. *Introduction to the C# Language and the .NET Framework*. [online]. Available from World Wide Web: <<https://msdn.microsoft.com/de-de/library/z1zx9t92.aspx>>

MICROSOFT CORPORATION. *Microsoft SQL Server 2014 Express*. [online]. Available from World Wide Web: <<https://www.microsoft.com/de-at/download/details.aspx?id=42299>>

## LITERATUR UND QUELLEN VERZEICHNIS:

MICROSOFT CORPORATION. *Query Syntax and Method Syntax in LINQ (C#)*. [online]. Available from World Wide Web: <<https://msdn.microsoft.com/de-de/library/bb397947.aspx>>

SCHIEFER, Landgasthof. [online]. Available from World Wide Web: <<http://www.gasthof-schiefer.at/>>

TOWER, Jonathan. *jtower.com/*. [online]. Available from World Wide Web: <<http://jtower.com/blog/undocking-parts-of-team-explorer-in-visual-studio>>

TUTORIAL, Entity Framework. [online]. Available from World Wide Web: <<http://www.entityframeworktutorial.net/>>

WIKIPEDIA. *Booking.com*. [online]. Available from World Wide Web: <[https://upload.wikimedia.org/wikipedia/en/1/12/Booking.com\\_logo\\_blue.jpg](https://upload.wikimedia.org/wikipedia/en/1/12/Booking.com_logo_blue.jpg)>

WIKIPEDIA. *Dropbox*. [online]. Available from World Wide Web: <<https://de.wikipedia.org/wiki/Dropbox>>

WIKIPEDIA. *Extensible Application Markup Language*. [online]. Available from World Wide Web: <[https://de.wikipedia.org/wiki/Extensible\\_Application\\_Markup\\_Language](https://de.wikipedia.org/wiki/Extensible_Application_Markup_Language)>

WIKIPEDIA. *GitHub*. [online]. Available from World Wide Web: <<https://de.wikipedia.org/wiki/GitHub>>

WIKIPEDIA. *Microsoft SQL Server*. [online]. Available from World Wide Web: <[https://de.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://de.wikipedia.org/wiki/Microsoft_SQL_Server)>

WIKIPEDIA. *Microsoft SQL Server*. [online]. Available from World Wide Web: <[https://de.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://de.wikipedia.org/wiki/Microsoft_SQL_Server)>

WIKIPEDIA. *Visual Studio*. [online]. Available from World Wide Web: <[https://de.wikipedia.org/wiki/Visual\\_Studio](https://de.wikipedia.org/wiki/Visual_Studio)>

WIKIPEDIA, Rückruffunktion. *Wikipedia Rückruffunktion*. [online]. Available from World Wide Web: <<https://de.wikipedia.org/wiki/R%C3%BCckrufffunktion>>

WIKIPEDIA. *Windows Presentation Foundation*. [online]. Available from World Wide Web: <[https://de.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](https://de.wikipedia.org/wiki/Windows_Presentation_Foundation)>

XELEMENT, Descendants-Methode. [online]. Available from World Wide Web: <[https://msdn.microsoft.com/de-de/library/bb353813\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/bb353813(v=vs.110).aspx)>

YEEBASE MEDIA GMBH. *Was ist eigentlich dieses GitHub?* [online]. Available from World Wide Web: <<http://t3n.de/news/eigentlich-github-472886/>>

## 16.4 Abkürzungsverzeichnis

C#	C-Sharp
SQL	Structured Query Language
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
Xlsx	Excel Spreadsheet
XmlRpc	Extensible Markup Language Remote Procedure Call
JSON	JavaScript Object Notation
ORM	Object-Relational Mapping
ADO.NET	ActiveX Data Object.NET
T-SQL	Transact- Structured Query Language
DML	Data Manipulation Language
DCL	Data Control Language
DDL	Data Definition Language
URI	Uniform Resource Identifier

## 17 Im Anhang

---

- Eigentliches Projekt
- Pflichtenheft PDF
- Stundenliste