

EU Meetings

Erfassung und Analyse von Flug- und Hotelkosten
für Meetings im EU-Raum

Frühlingssemester 2009

16. Februar - 12. Juni 2009

Bachelorarbeit von:

Jonpaul Feuerstein

Fabius Bernet

Betreuer:

Prof. Dr. Peter Heinzmann, cnlab AG

Korreferent:

Prof. Stefan Keller, HSR

Experte:

Dr. Thomas Siegenthaler, CSI Consulting AG

Projektpartner:

Action COST298 - Participation in the Broadband
Society

Hochschule für Technik Rapperswil (HSR)

1 Aufgabenstellung

Bei der Organisation von Treffen zu europäischen Forschungsprojekten steht immer wieder zur Diskussion, welche Städte hinsichtlich Flug- und Übernachtungskosten am Günstigsten seien. Im Rahmen der vorliegenden Bachelor-Arbeit soll untersucht werden, wie stark die Flug- und Übernachtungskosten von der Wahl des Tagungsortes abhängen und welche Faktoren die Kosten am stärksten beeinflussen.

Zur Untersuchung des Einflusses verschiedener Faktoren sollen die Flug- und Übernachtungskosten automatisiert aus Web-Buchungsplattformen zusammengetragen, in einer Datenbank abgelegt und analysiert werden. Basierend auf diesen Daten ist der Einfluss von Faktoren wie beispielsweise Tagungstermin, Tagungsdauer, Wochentage/Wochenendtage und Buchungszeitpunkt für Teilnehmende und Tagungsorte des COST298 Projektes aufzuzeigen.

Es ist eine Webanwendung zu realisieren, mit welcher die Flug- und Übernachtungskosten für ein bestimmtes Tagungsdatum, für eine vorzugebende Teilnehmerzusammensetzung und für bis zu drei Tagungsorte durch automatisierte Abfragen auf Buchungsplattformen zusammengetragen werden. Die Resultate sollen dem Nutzer direkt via Web oder per Mail ausgeliefert werden.

2 Erklärung der Studenten¹

Die vorliegende Arbeit basiert auf Ideen, Arbeitsleistungen, Hilfestellungen und Beiträgen gemäss folgender Aufstellung:

Gegenstand, Leistung	Person	Funktion
	Jonpaul Feuerstein	Autor der Arbeit
	Fabius Bernet	Autor der Arbeit
Idee, Aufgabenstellung, allgemeines Pflichtenheft, Betreuung während der Arbeit	Prof. Dr. P. Heinzmann	Verantwortlicher Professor

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit gemäss obiger Zusammenstellung selber und ohne weitere fremde Hilfe durchgeführt habe,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.

Rapperswil, den.....

.....

Student

Rapperswil, den.....

.....

Student

¹ Diese Erklärung basiert auf der Muster-Erklärung in den *Richtlinien der HSR zur Durchführung von Projekt-, Studien-, Diplom- oder Bachelorarbeiten* vom 16. Februar 2009.

3 Inhaltsverzeichnis

1	Aufgabenstellung	2
2	Erklärung der Studenten	3
3	Inhaltsverzeichnis.....	4
4	Abstract	6
5	Management Summary	7
5.1	Ausgangslage	7
5.2	Ergebnisse	8
5.3	Ausblick	9
6	Technischer Bericht.....	10
6.1	Einführung.....	10
6.1.1	Problemstellung, Vision	10
6.1.2	Risiken der Arbeit	10
6.1.3	Ziele und Unterziele.....	11
6.1.4	Vorgehen, Aufbau der Arbeit	11
6.2	„Stand der Technik“	12
6.3	Umsetzungskonzept.....	13
6.3.1	Grobe Beschreibung des eigenen Lösungskonzepts	13
6.3.2	Was musste man selbst tun?	14
6.4	Resultate, Bewertung und Ausblick	15
6.4.1	Zielerreichung	15
6.4.2	Ausblick: Weiterentwicklung.....	15
6.4.3	Persönlicher Berichte	16
6.4.4	Dank.....	16
7	Analyse der Flug- und Hotelkosten	17
7.1	Anforderungen.....	17
7.1.1	Flugdaten	17
7.1.2	Hoteldaten	19
7.2	Vorgehen.....	21
7.2.1	Auswertung in Excel ohne Data-Warehouse und Cube	22
7.2.2	Auswertung mit Data-Warehouse und Cube	23
7.3	Ergebnisse	23
7.3.1	Flugdaten	24
7.3.2	Hoteldaten	30
7.3.3	Zusammenfassung	33
8	Software-Projekt Dokumentation.....	34
8.1	Technologien.....	34
8.1.1	C#	34
8.1.2	MS SQL Server 2005	34
8.1.3	Google WebDriver	35

8.2	Anforderungsspezifikation.....	36
8.2.1	Anforderung an die Arbeit.....	36
8.2.2	Use Cases (Success Scenario / Success Diagram)	41
8.2.3	System-Sequenzdiagramme	42
8.2.4	Weitere Funktionen.....	43
8.3	Analyse (Business Modell)	44
8.3.1	Domain Modell	44
8.3.2	Objektkatalog	45
8.4	Design (Entwurf)	46
8.4.1	Architektur.....	46
8.4.2	Package- und Klassendiagramme (konzeptionell).....	47
8.4.3	Datenspeicherung.....	55
8.4.4	Sequenzdiagramme	59
8.4.5	Prozesse und Threads.....	61
8.5	Implementation (Entwicklung) und Test	62
8.5.1	Implementation: Erläuterungen wichtiger konkreter Klassen	62
8.5.2	Datenbank	71
8.6	Resultate und Weiterentwicklung	77
8.6.1	Resultate.....	77
8.6.2	Möglichkeiten der Weiterentwicklung.....	77
8.7	Projektmanagement	78
8.7.1	Prototypen, Releases, Meilensteine.....	78
8.7.2	Team, Rollen und Verantwortlichkeiten.....	78
8.7.3	Aufwandschätzung, Zeitplan, Projektplan.....	78
8.7.4	Risiken.....	87
8.7.5	Prozessmodell.....	87
8.8	Projektmonitoring.....	88
8.8.1	Soll-Ist-Zeit-Vergleich.....	88
8.9	Softwaredokumentation.....	90
8.9.1	Tutorial Webapplikation	90
9	Schlussfolgerungen	95
10	Abbildungsverzeichnis	96
11	Glossar	98
12	Literaturverzeichnis	100
13	Anhang.....	101
13.1	Zugriff Webapplikation.....	101
13.2	CD-ROM.....	101

4 Abstract

Diese Bachelorarbeit befasst sich mit dem Verhalten von Flug- und Hotelkosten, welche bei Meetings mit Teilnehmern aus verschiedenen Ländern im EU-Raum entstehen.

Einerseits sollen durch gezielte Analysen von gesammelten Flug- und Hoteldaten die wichtigsten Faktoren gesucht und erkannt werden, die sich direkt auf die entsprechenden Kosten auswirken.

Andererseits soll im Rahmen dieser Arbeit eine Webapplikation entwickelt werden. Diese soll dem Benutzer ermöglichen, Echtzeit-Abfragen für ein geplantes Meeting zu erstellen. Als Resultat erhält der Benutzer die jeweiligen Flug- und Hotelkostendetails zu seiner Anfrage und kann so z.B. die Kosten für verschiedene Meeting-Destinationen miteinander vergleichen oder die Resultate einfach als eine Aufwandsschätzung verwenden.

Um möglichst aussagekräftige Ergebnisse über die, bei der Preisbildung für Flüge und Hotels eine Rolle spielenden Faktoren, zu ermitteln, muss eine ausreichend grosse Datenmenge für die Analyse vorhanden sein. Nur so können eventuelle, allgemein gültigen Tendenzen erkannt und entsprechende Aussagen gemacht werden. Bei diesen Untersuchungen soll festgestellt werden, ob und wie stark sich verschiedenen Aspekte (Buchungszeitpunkt, Aufenthaltsdauer, Frühbuchung/Last Minute, usw.) auf die Flug- und Hotelpreise auswirken.

Die zu entwickelnde Webapplikation soll dem Benutzer möglichst einfach und komfortabel die Möglichkeit geben, Flug- und Hotelkosten für ein bestimmtes Meeting zu ermitteln. Anhand des Meeting-Ortes, den Herkunftsorten der Teilnehmer sowie deren Anzahl werden dann die Kosten berechnet und dem Benutzer zugänglich gemacht.

Um die Faktoren, welche die Preisbildung beeinflussen, zu ermitteln wurde eine Anwendung entwickelt, welche täglich automatische Abfragen an eine Flug- und Hotelbuchungsplattform sendet. Die Resultate dieser Abfragen wurden für die Auswertung in einer Datenbank persistiert. Mit Hilfe von gezielten Abfragen auf der vorhandenen Datenmenge wurden Diagramme erzeugt, welche das Preisverhalten veranschaulichen und die Einflüsse von bestimmten Faktoren auf den Preis aufzeigen. Des Weiteren wurden auch Data-Mining Algorithmen benutzt, um eventuelle komplexere Zusammenhänge zu ermitteln.

Bei der Webapplikation wird für die Kostenberechnung einer Benutzeranfrage ebenfalls die vorhandene Kernapplikation benutzt, welche die Preise in Echtzeit von einer online Buchungsplattform bezieht. Alle getätigten Abfragen und deren Resultate sind für den Benutzer jederzeit wieder abrufbar.

5 Management Summary

5.1 Ausgangslage

Bei der Organisation von Treffen zu europäischen Forschungsprojekten steht immer wieder zur Diskussion, welche Städte hinsichtlich Flug- und Übernachtungskosten am Günstigsten seien. Im Rahmen der vorliegenden Bachelor-Arbeit soll untersucht werden, wie stark die Flug- und Übernachtungskosten von der Wahl des Tagungsortes abhängen und welche anderen Faktoren die Kosten am stärksten beeinflussen.

Eine automatisierte Lösung, welche die Gesamtkosten berechnet existiert unseres Wissens noch nicht.

Des weitern ist meist auch nicht ganz klar ersichtlich welche Faktoren (z.B. Aufenthaltsdauer, Wochentage, Buchungszeitpunkt, Frühbuchung / Last Minute) sich am meisten auf das Preisverhalten von Flug- und Hotelkosten auswirken.

Bei unseren Recherchen zu diesem Thema sind wir auf ein paar durchaus interessante Publikationen gestossen, welche sich mit dem Verhalten von Flugpreisen befassen. Aus diesen Arbeiten konnten wir Faktoren erkennen, welche sich auf die Preise auswirken und auf die wir uns bei unserer Analyse konzentrieren sollten.

Am Ender der Projektarbeit sollen zwei Hauptziele erreicht werden.

Erstes Ziel ist es, das Verhalten von Flug- und Hotelpreisen zu analysieren. Dafür sollen automatisiert Daten gesammelt und in eine Datenbank abgelegt werden. Die gesammelten Daten sollen untersucht werden, um preisebeeinflussende Faktoren zu erkennen. Die resultierenden Erkenntnisse sollen anhand von Diagrammen veranschaulicht werden.

Das zweite Ziel ist das Bereitstellen einer Webapplikation, welche es einem Benutzer ermöglicht automatisiert eine Echtzeit-Kostenanfrage für ein geplantes Meeting mit Teilnehmern aus verschiedenen Orten zu erstellen. Die Anwendung soll die erforderlichen Preisinformationen automatisch zusammentragen und für den Benutzer aufbereiten.

Die obengenannten Zielsetzungen sollten nach Abschluss der Arbeit erreicht worden sein, damit das Projekt als erfolgreich betrachtet werden kann.

Des Weiteren wäre es wünschenswert wenn die erstellte Webapplikation von den Verantwortlichen des COST298 Projektes als Hilfestellung benutzt werden kann, um möglichst kostengünstige Tagungen zu planen.

Da zu Beginn der Arbeit noch keine genaue Aufgabenstellung vorhanden war, sondern nur eine ungefähre Idee, musste zuerst definiert werden, was das Endprodukt der Arbeit sein soll.

Die Grundidee war, dass die Resultate der Arbeit von den Verantwortlichen des COST298 Projektes als eine Entscheidungshilfe für das Organisieren und Durchführen ihrer Meetings verwendet werden können.

5.2 Ergebnisse

Das Gesamtergebnis dieser Arbeit besteht aus drei verschiedenen Produkten. Sämtliche Funktionalität um die Flug- und Hoteldaten zusammenzutragen und in der Datenbank zu speichern musste selbst implementiert werden.

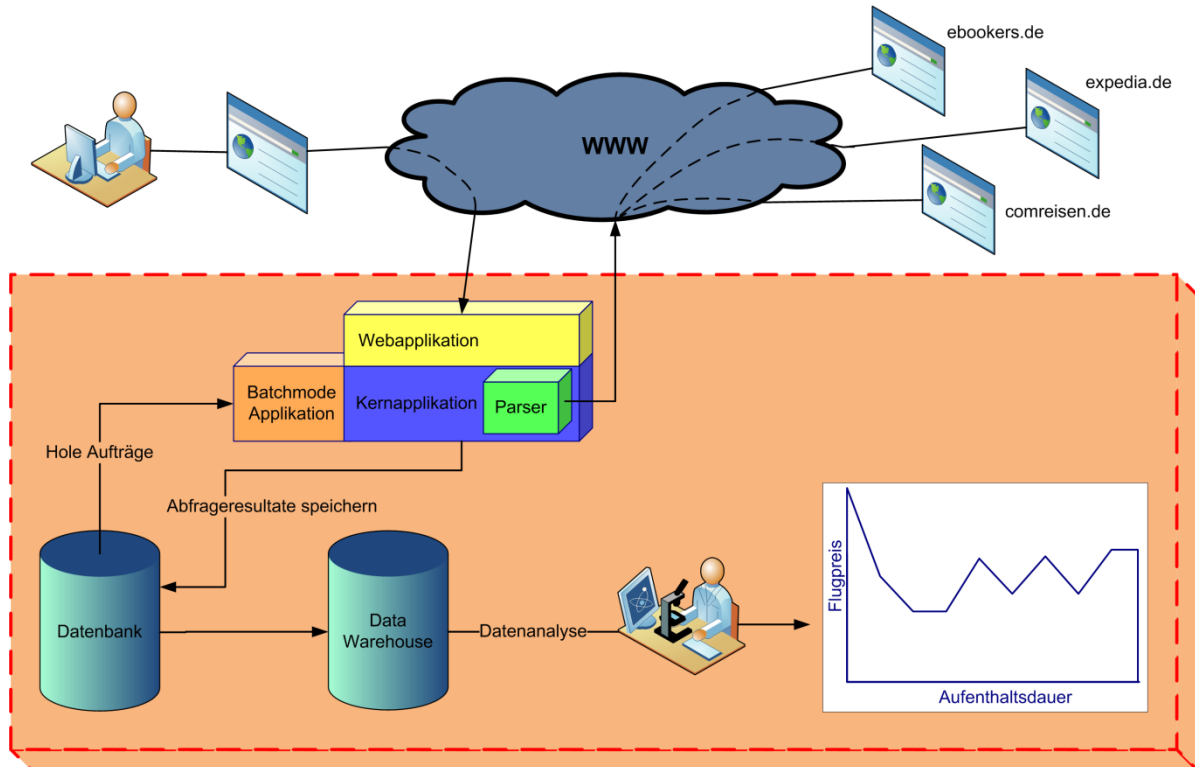


Abbildung 1: Übersicht der erarbeiteten Funktionalitäten

Kernapplikation

Abbildung 1 zeigt unter anderem eine Übersicht der Funktionsweise der Kernapplikation und der Batchmode-Applikation.

Die Batchmode-Applikation dient lediglich dazu, die bereits vordefinierten Abfrage-Tasks in der Datenbank auszulesen wenn diese fällig sind und an die Kernapplikation zu übergeben. So können automatisch sehr viele Flug- und Hoteldaten für die Analyse gesammelt werden.

Die Kernapplikation sendet die Anfragen an eine online Buchungsplattform und extrahiert die benötigten Flug- und Hoteldaten mit Hilfe eines Parser aus der Antwortseite. Die Resultate werden dann in der Datenbank persistiert.

Webapplikation

Die Webapplikation ermöglicht einem Benutzer eine Abfrage für ein Meeting zu erstellen und die daraus resultierenden Kosten auszuwerten.

Aus der Abbildung 1 ist ersichtlich, dass die Webapplikation für das Durchführen der Abfragen auf die Funktionen der Kernapplikation zurückgreift. Die Resultate werden können dann vom Benutzer ebenfalls mittels der Webapplikation eingesehen werden.

Analyse von Flug- und Hoteldaten

Die gesammelten Flug- und Hoteldaten wurden so aufbereitet, dass die Auswertungen möglichst Effizient durchgeführt werden konnten. Dies wurde durch die Realisation eines Data-Warehouse erreicht. Mit Hilfe verschiedener Tools wurden Diagramme und Analysen erstellt.

Die beiden Hauptziele, das Analysieren und Auswerten von Flug- und Hoteldaten und deren Preisverhalten sowie das erstellen der Webapplikation für Benutzeranfragen wurden bis zum Ende des Projekts erreicht.

Der Richtwert für den zeitlichen Aufwand dieser Arbeit beträgt ca. 720 Stunden. Aufgrund unserer Projektplanung gingen wir von einem Aufwand von ca. 720 Stunden aus. Die Anzahl der tatsächlich benötigten zeitlichen Ressourcen betrug am Projektende ca. 740 Stunden.

5.3 Ausblick

Der grösste Lerneffekt ergab sich für uns durch den Einsatz der von uns gewählten Technologien. Da wir bis anhin wenig Erfahrung im .NET Umfeld (C#, ASP.NET) hatten, mussten wir uns dort zuerst einiges Wissen aneignen.

Durch das Verwenden des MS SQL Servers und den dazugehörigen Tools konnten ebenfalls nützliche, neue Erfahrungen gesammelt werden.

Da die Flug- und Hoteldaten aus einer Webseite herausgelesen werden, muss die Applikation allenfalls angepasst werden, wenn sich die Struktur der Resultate-Seite ändert.

Das Layout und die Benutzerfreundlichkeit der Webapplikation könnte sicherlich noch verbessert werden. Um dies noch während der Projektphase umzusetzen fehlten uns am Ende schlicht und einfach die dafür nötigen zeitlichen Ressourcen.

Als Negatives gibt es vielleicht zu sagen, dass am Anfang der Arbeit ziemlich viel Zeit benötigt wurde, um die genauen Ziele und den Umfang der Arbeit zu definieren. Diese Zeit hätten wir am Ende der Arbeit gut gebrauchen können.

Des Weiteren war die Abschätzung des zeitlichen Aufwands für das erstellen der Webapplikation im Nachhinein viel zu optimistisch ausgefallen.

6 Technischer Bericht

6.1 Einführung

6.1.1 Problemstellung, Vision

Bei der Organisation von Treffen zu europäischen Forschungsprojekten steht immer wieder zur Diskussion, welche Städte hinsichtlich Flug- und Übernachtungskosten am Günstigsten seien. Im Rahmen der vorliegenden Bachelor-Arbeit soll untersucht werden, wie stark die Flug- und Übernachtungskosten von der Wahl des Tagungsortes abhängen und welche anderen Faktoren die Kosten am stärksten beeinflussen.

Um die verschiedenen Gesamtkosten für ein geplantes Meeting für unterschiedliche Durchführungsorte zu ermitteln, ist für den Meeting-Veranstalter ein grosser zeitlicher Aufwand nötig. Es bleibt ihm nicht viel Anderes übrig, als auf einer Buchungsplattform die Kosten der Flüge und für die Übernachtungen sämtlicher Teilnehmer manuell abzufragen und zusammenzutragen. Der Aufwand steigert sich natürlich mit der Anzahl der Meeting-Teilnehmer aus verschiedenen Orten und den in Frage kommenden Meeting-Durchführungsdestinationen. Bei einer grösseren Anzahl von verschiedenen Teilnehmern ist dieses manuelle Zusammentragen der Kosten sehr zeitintensiv und einfach nur noch mühsam.

Eine automatisierte Lösung, welche die Gesamtkosten berechnet existiert unseres Wissens noch nicht oder ist nicht öffentlich und kostenlos nutzbar.

Eine andere Variante wäre, die Planung des Meetings durch ein Reisebüro oder einen Reiseveranstalter erfolgen zu lassen. Diese Möglichkeit hat jedoch den Nachteil, dass die Gesamtkosten wahrscheinlich erheblich gesteigert werden. Denn Flug- und Hotelkosten werden mit grosser Wahrscheinlichkeit höher ausfallen als bei Online-Buchungen. Zudem kommt natürlich noch eine Marge des Reisebüros oder des Veranstalters dazu, was die Kosten nochmals ansteigen lässt.

Des weitern ist meist auch nicht ganz klar ersichtlich welche Faktoren (z.B. Aufenthaltsdauer, Wochentage, Buchungszeitpunkt, Frühbuchung / Last Minute) sich am meisten auf das Preisverhalten von Flug- und Hotelkosten auswirken. Sollte es klar erkennbare Tendenzen geben, welche die Preise direkt beeinflussen, wäre es schön diese zu kennen, damit man dieses Wissen bei der Planung der Meetings eventuell nutzen kann um die Kosten zu optimieren.

6.1.2 Risiken der Arbeit

Als Grundlage für unsere Arbeit werden natürlich aktuelle und korrekte Flug- und Hotelpreise benötigt. Damit stellt dieser Punkt auch das grösste Risiko in unserem Projekt dar. Woher bekommen wir die benötigten Daten? Dies war Anfangs die wichtigste Frage die wir uns stellen mussten. Denn mit dem Vorhandensein oder Nicht-Vorhandensein der Daten steht oder fällt unser Projekt.

Ein weiteres Risiko entstand bei der Wahl der zu verwendenden Technologien. Da wir uns entschlossen die Anwendung mit .NET und MS SQL Server zu realisieren, obwohl uns klar war, dass wir damit nicht allzu grosse Erfahrung hatten und uns in viele Sachen zuerst noch „einlesen“ und „einarbeiten“ werden müssen. Das Risiko hierbei war vor allem der zeitliche Aufwand der hierfür benötigt werden wird und nicht klar abzuschätzen war.

6.1.3 Ziele und Unterziele

Am Ende der Projektarbeit sollen zwei Hauptziele erreicht werden.

Erstes Ziel ist es, das Verhalten von Flug- und Hotelpreisen zu analysieren. Dafür sollen automatisiert Daten gesammelt und in eine Datenbank abgelegt werden. Bei den Untersuchungen der Daten soll herausgefunden werden, ob und wenn wie stark sich gewisse Faktoren bei einer Flug- oder Hotelbuchung auf den Preis auswirken. Es sollen diejenigen Faktoren gesucht werden, bei denen eine eindeutige Tendenz auszumachen ist, damit eine möglichst generelle Aussage aufgrund unserer Daten möglich wird. Dafür muss allerdings zuerst herausgefunden werden, ob überhaupt solche Faktoren existieren die sich eindeutig auf die Preise niederschlagen. Die resultierenden Erkenntnisse sollen anhand von Diagrammen veranschaulicht werden.

Das zweite Ziel ist das Bereitstellen einer Webapplikation, welche es einem Benutzer ermöglicht automatisiert eine Echtzeit-Kostenanfrage für ein geplantes Meeting mit Teilnehmern aus verschiedenen Orten zu erstellen. Die Anwendung soll die erforderlichen Preisinformationen automatisch zusammentragen und für den Benutzer aufbereiten. Es soll möglich sein eine Abfrage nur für die Flugkosten, nur für die Hotelkosten oder auch eine kombinierte Abfrage zu erstellen. Aufgrund der Resultate soll der Anwender dabei unterstützt werden die Kosten für verschiedene Meeting-Destinationen miteinander zu vergleichen oder er kann die Ergebnisse als eine Kostenabschätzung verwenden.

Die obengenannten Zielsetzungen sollten nach Abschluss der Arbeit erreicht worden sein, damit das Projekt als erfolgreich betrachtet werden kann.

Des Weiteren wäre es wünschenswert wenn die erstellte Webapplikation von den Verantwortlichen des COST298 Projektes als Hilfestellung benutzt werden kann, um möglichst kostengünstige Tagungen zu planen.

6.1.4 Vorgehen, Aufbau der Arbeit

Da zu Beginn der Arbeit noch keine genaue Aufgabenstellung vorhanden war, sondern nur eine ungefähre Idee, wurde zuerst relativ viel Zeit gebraucht um zu definieren, was als Resultat der Arbeit überhaupt heraussehen soll.

Die Grundidee war, dass die Resultate der Arbeit von den Verantwortlichen des COST298 Projektes als eine Entscheidungshilfe für das Organisieren und Durchführen ihrer Meetings verwendet werden können.

Nun wurden Möglichkeiten gesucht wie dies am Besten umzusetzen wäre. Schnell kam man zum Schluss, dass ein Tool, welches total selbständig, lediglich durch Angabe gewisser Meeting-Rahmenbedingungen, versucht den Besten Ort und Durchführungszeitraum zu finden, so nicht realisiert werden kann. Dies wäre mit den vorgegebenen zeitlichen Ressourcen dieser Arbeit nicht zu realisieren.

Nach dieser Erkenntnis kristallisierten sich zwei verschiedene Punkte heraus, die man umsetzen konnte und welche der Grundidee gerecht wurden. Man entschied sich dafür, einerseits Flug- und Hotelpreise zu analysieren und deren Verhalten zu ergründen, andererseits eine Webapplikation zu entwickeln, welche dem Benutzer ermöglicht die Kosten für ein Meeting und dessen Teilnehmer zu berechnen.

Um herauszufinden, ob sich überhaupt irgendwelche einflussreiche Faktoren bei der Preisbildung von Flug- und Hotelpreisen finden lassen, wurden als erstes mit Hilfe des Google WebDriver halbautomatische Abfragen an ein Buchungsportal gesendet und die Resultate von Hand in einem Excel-Sheet ausgewertet. Aufgrund der Resultate wurde entschieden, dass durch eine genauere Analyse durchaus interessante Resultate herauskommen könnten und dass man diesen Weg weiterverfolgen möchte.

Nun stellte sich die Frage, woher man am besten die Flug- und Hoteldaten beziehen sollte. Es standen Anfangs grundsätzlich zwei verschiedene Varianten im Raum. Einerseits das Beschaffen der

benötigten Informationen über ein Global Distribution System (z.B. Sabre, Amadeus, Galileo) oder durch das Parsen der Resultate von online Buchungsplattformen. Schnell musste man sich von der ersten Variante verabschieden, da die Kosten für den Zugriff auf ein Global Distribution System beträchtlich sind und den finanziellen Rahmen der Arbeit bei weitem sprengen würde. So wurde entschieden, dass die benötigten Daten von einer online Buchungsplattform ausgelesen werden sollen.

Bei der Evaluation des zu verwendenden Portals entschied man sich für ebbookers.de. Dies vor allem aus zwei Gründen. Einerseits sind dort sowohl Flug- als auch Hotelabfragen möglich. Zweitens werden die Preise dort in Euro angegeben, was für uns wünschenswert war, da uns damit Währungsumrechnungen erspart blieben.

So wurde nun eine die Kernapplikation mit C# entwickelt. Diese arbeitet automatisch die auszuführenden, vordefinierten Abfragen ab und speichert die Resultate in der Datenbank. Mit Hilfe dieses Batchbetriebes wurden hatten wir sehr schnell viele Flug- und Hoteldaten für unsere Analyse zur Verfügung.

Um die Daten besser und effizienter analysieren zu können, wurde ein Data-Warehouse aufgebaut und die Daten dorthin migriert. Anschliessend wurden mit Hilfe, zuerst von Excel später dann von MS SQL Server Reporting Services, Diagramme erstellt, welche die Kostenentwicklung anhand verschiedener Faktoren aufzeigen. Es wurden auch verschiedene Data-Mining Algorithmen des MS SQL Server Analysis Services angewandt, um komplexere Zusammenhänge zu erörtern.

Gleichzeitig wurde auch die erforderliche Webapplikation mit ASP.NET entwickelt.

6.2 „Stand der Technik“

Bei unseren Recherchen zu diesem Thema sind wir auf ein paar durchaus interessante Publikationen gestossen, welche sich mit dem Verhalten von Flugpreisen befassen. Meist wurden aber nur die Preise von bestimmten Airlines analysiert. Allerdings wurden in diesen Arbeiten keine generellen Aussagen gemacht welche genauen Faktoren wie stark auf die Preisbildung einen Einfluss haben.

Mit Hilfe einiger gefundenen Berichte, konnten wir jedoch erkennen, welche Faktoren vermutlich eine Rolle spielen bei der Preisentwicklung. Damit wurde uns gezeigt auf welche Einflüsse wir uns bei unserer Auswertung besonders konzentrieren sollten, so dass eventuelle generelle Aussagen möglich werden.

Bei der Suche nach einem Tool, welches die Meeting-Kosten automatisch in Echtzeit berechnet, wurden wir allerdings, wie bereits erwähnt, nicht fündig.

6.3 Umsetzungskonzept

6.3.1 Grobe Beschreibung des eigenen Lösungskonzepts

Das Gesamtergebnis dieser Arbeit besteht aus drei verschiedenen Produkten.

6.3.1.1 Kernapplikation

Die erstellte Kernapplikation, sammelt automatisiert die Daten für Flüge und Hotels und speichert die Resultate in der Datenbank.

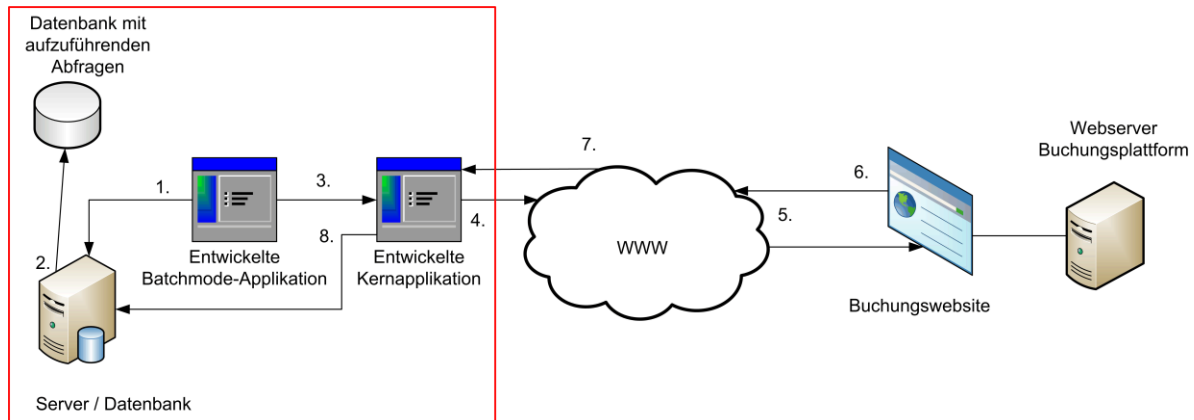


Abbildung 2: Übersicht der entwickelten Kernapplikation

Abbildung 2 zeigt eine Übersicht der Funktionsweise der Kernapplikation und der Batchmode-Applikation.

Die Batchmode-Applikation dient lediglich dazu, die bereits vordefinierten Abfrage-Tasks in der Datenbank auszulesen wenn diese fällig sind und an die Kernapplikation zu übergeben, welche dann ihrerseits die Abfragen durchführt und die Resultate persistiert. Die Kernapplikation greift dabei auf das online Buchungsportal zu und sendet die entsprechenden Flug- oder Hotelanfragen. Die Resultate werden dann mit Hilfe eines Parsers aus der HTML-Antwortseite der Buchungsplattform extrahiert und in der Datenbank gespeichert.

Die Anfragen an die Buchungsplattform werden dabei parallelisiert abgesetzt. Dadurch kann die Wartezeit bei grösseren Abfragen teils massiv verkürzt werden. Denn die Antwortzeit (Zeit nach Absenden der Anfrage bis die Resultate verfügbar sind) ist aus Performance-Sicht ganz klar der Flaschenhals der Anwendung. Wenn man bedenkt, dass für eine einzelne Meeting-Abfrage mit einem Meeting-Durchführungsort pro Teilnehmer mit unterschiedlichem Herkunftsort eine Abfrage nötig ist, sieht man schnell, dass mit dieser Parallelisierung bei grösseren Abfragen viel Wartezeit eingespart werden kann. Die Antwortzeit einer einzigen Abfrage beträgt bei ebookers.de zwischen 10 und 20 Sekunden. Durch das parallele Abfragen kann eine Wartezeit für ca. 10 Abfragen um die 50 Sekunden erreicht werden. Auch für bei mehr als 10 Abfragen steigt die Wartezeit nur noch leicht an.

6.3.1.2 Webapplikation

Die Webapplikation ermöglicht einem Benutzer eine Abfrage für ein Meeting zu erstellen und die daraus resultierenden Kosten auszuwerten.

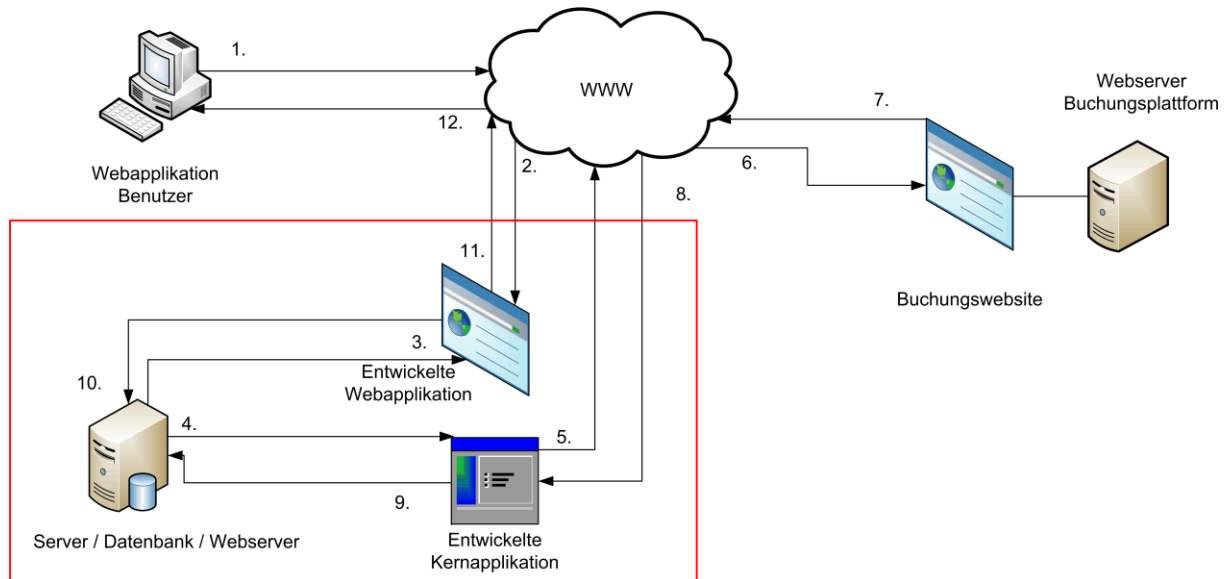


Abbildung 3: Übersicht der entwickelten Webapplikation

Abbildung 3 zeigt eine Übersicht der Funktionalität der Webapplikation. Aus der Figur ist ersichtlich, dass die Webapplikation für das Durchführen der Abfragen auf die Funktionen der Kernapplikation zurückgreift.

Der Benutzer loggt sich bei der Webapplikation ein. Danach hat er Zugriff auf all seine bisher getätigten Abfragen und deren Resultate. Ebenso kann er eine neue Meeting-Abfrage erstellen und absenden. Diese wird sofort durch die Kernapplikation bearbeitet und sobald alle Resultate verfügbar sind, kann sich der Benutzer diese anzeigen lassen.

Bei der erstellten Abfrage kann der Benutzer das Meeting-Durchführungsdatum, bis zu drei verschiedene Meeting-Orte und beliebig viele Teilnehmer mit verschiedenen Herkunftsorten erfassen. Zusätzlich kann er angeben, ob nur Flüge, nur Hotels oder eine kombinierte Abfrage durchgeführt werden soll.

6.3.1.3 Analyse von Flug- und Hoteldaten

Um die gesammelten Flug- und Hoteldaten möglichst effizient zu untersuchen und die entscheidenden Faktoren die bei der Preisbildung eine Rolle spielen zu identifizieren wurde ein Data-Warehouse aufgebaut und die vorhandenen Daten in dieses migriert.

Mit Hilfe der MS SQL Analysis Services und Reporting Services wurden gezielte Abfragen erstellt und die Ergebnisse in Diagrammen veranschaulicht. Ebenfalls wurden verschiedene Data-Mining Algorithmen auf die Datenmenge angewandt um allenfalls versteckte, komplexere Zusammenhänge zwischen den verschiedenen Faktoren ausfindig zu machen.

6.3.2 Was musste man selbst tun?

Sämtliche Funktionalität um die Flug- und Hoteldaten zusammenzutragen und in der Datenbank zu speichern musste selbst implementiert werden. Einzig für das Parsen der HTML Resultate-Webseite der Flug- und Hotelergebnisse wurde eine externe Bibliothek verwendet um den Vorgang effizienter und einfacher zu gestalten.

Bei der Webapplikation wurden keine externen Bibliotheken verwendet. Allerdings wurde hier natürlich zum Teil auf die vorhandenen Controls von ASP.NET zurückgegriffen, welche das Framework zu Verfügung stellt.

6.4 Resultate, Bewertung und Ausblick

6.4.1 Zielerreichung

Die beiden Hauptziele, das Analysieren und Auswerten von Flug- und Hoteldaten und deren Preisverhalten sowie das erstellen der Webapplikation für Benutzeranfragen wurden bis zum Ende des Projekts erreicht.

Für die Analyse der Flug- und Hoteldaten wurden verschiedene Diagramme erstellt, welche die Preisentwicklung anhand einflussreicher Faktoren veranschaulichen.

Ob die Webapplikation tatsächlich auch produktiv benutzt werden wird, lässt sich im Moment noch nicht sagen. Die Webapplikation hat durchaus noch Weiterentwicklungspotential. Das Design und das Layout sowie die Benutzerfreundlichkeit sind durchaus noch verbesserungsfähig. Um dies noch während des Projektes zu realisieren fehlten uns die dafür nötigen zeitlichen Reserven.

6.4.2 Ausblick: Weiterentwicklung

Da die Flug- und Hoteldatenbeschaffung anhand eines Parser aus einer HTML-Seite erfolgt, liegt hier natürlich ein Risiko vor. Es kann nicht dynamisch auf allfällige Veränderungen oder Neuerungen der HTML Struktur der Resultate-Seiten reagiert werden. Die Parser Klassen müssen unumgänglich manuell angepasst werden falls sich Bezeichnungen, Layout oder Namen innerhalb der Zielseite ändern.

Das Software-Design wurde so konzipiert, dass das Einbinden von weiteren Parser-Klassen (z.B. um Resultate von anderen online Buchungsplattformen zu erhalten möglichst simpel und effizient umsetzbar ist. Das Erweitern mit neuen Parser für andere Buchungsportale würde durchaus Sinn machen. Denn so wäre es möglich die verschiedenen Resultate der Buchungsplattformen für identische Abfragen miteinander zu vergleichen. Dies würde wahrscheinlich durchaus zu interessanten Resultaten führen.

Das Layout und die Benutzerfreundlichkeit der Webapplikation könnte sicherlich noch verbessert werden. Um dies noch während der Projektphase umzusetzen fehlten uns am Ende schlicht und einfach die dafür nötigen zeitlichen Ressourcen.

Als Negatives gibt es vielleicht zu sagen, dass am Anfang der Arbeit ziemlich viel Zeit benötigt wurde, um die genauen Ziele und den Umfang der Arbeit zu definieren. Diese Zeit hätten wir am Ender der Arbeit gut gebrauchen können. Wir würden bei einem nächsten Projekt versuchen, die genauen Ziele und die Anforderungen möglichst früh zu definieren um genug Reserven für die Implementation und Umsetzung zur Verfügung zu haben.

Des Weiteren war die Abschätzung des zeitlichen Aufwands für das erstellen der Webapplikation im Nachhinein viel zu optimistisch ausgefallen. Hier musste deutlich mehr aufgewendet werden als Anfangs geplant war.

6.4.3 Persönlicher Berichte

6.4.3.1 Jonpaul Feuerstein



Die Bachelorarbeit war für mich nach der Studienarbeit das zweite grosse Softwareprojekt, obwohl wir nicht nur programmiert haben sondern auch viel Zeit zur Auswertung der Daten verwendet haben. Dieses Projekt parallel zu den Vorlesungen durchzuführen empfand ich als nicht optimal. Wenn man neben der Arbeit auch noch Vorlesungen und Übungen besuchen muss vergisst man gerne das eine oder das andere. Auch hat man noch nicht alles durchgenommen und wenn man jetzt das in der Arbeit braucht muss man es lernen bevor man es im Unterricht durchführen würde.

Die Zusammenarbeit im Team und mit den Betreuern war gut.

Mit der Arbeit bin ich zufrieden, obwohl ich einige Dinge im Nachhinein anders machen würde. Viele Details sieht man erst wenn man schon daran arbeitet oder auch erst nachdem man es mit jemand anderem bespricht. Darum ist es wichtig, dass alle Ergebnisse mindestens von jemandem anderen angeschaut werden.

Jetzt bin ich froh, dass die Arbeit vorbei ist und freue mich schon auf die nächste Herausforderung. Aber erst kommen jetzt die Ferien und dann die Prüfungen. Danach geht's dann (hoffentlich) in die Arbeitswelt.

6.4.3.2 Fabius Bernet



Für mich war diese Bachelorarbeit nach dem SE2-Projekt und der Studienarbeit erst das dritte grössere Softwareprojekt. Trotzdem konnte ich, was die Aufwandsschätzung angeht, bei der Projektplanung viel von den Erfahrungen aus den vorherigen Arbeiten profitieren. Auch konnte der Umfang der Dokumentation in etwa abgeschätzt werden. So konnte bei diesem Projekt die Zeitplanung einigermaßen eingehalten werden.

Die Zusammenarbeit im Team war gut. Ich denke wir konnten beide unser Wissen einbringen und so gegenseitig von einander profitieren.

Die Betreuung durch die Betreuer war sehr professionell und hilfreich. Die erhaltenen Anregungen waren meist sehr konstruktiv und unterstützten uns in der Projektabwicklung.

Mit dem Resultat der Arbeit bin ich eigentlich zufrieden. Schlussendlich hätte ich zwar gerne die Webapplikation noch optimiert, doch dafür war keine Zeit mehr. Einziger Wermutstropfen aus meiner Sicht ist, dass am Anfang des Projekts viel Zeit gebraucht wurde um die Anforderungen genau festzulegen. Diese Zeit fehlte uns zum Teil bei der Implementation und der Datenanalyse.

Natürlich gibt es auch ein paar Dinge, die ich nun anders lösen würde. Aber es ist ja sicherlich auch Sinn dieser Arbeit, dass Erfahrungen gesammelt werden können.

Diese Bachelorarbeit bildet ja (hoffentlich) nun einen Schlusspunkt meiner Ausbildung an der HSR. Ich denke, dass ich dank den gemachten Erfahrungen und des gelernten Know-How's während dieses Studiums bestens gerüstet bin für neue Herausforderungen.

6.4.4 Dank

Wir möchten uns herzlich bei unserem Betreuer Dr. Prof. Peter Heinzmann bedanken. Wir wurden von Herrn Heinzmann während unserer Arbeit bestens unterstützt und bekamen immer wieder hilfreiche Tipps und Anregungen die uns bei der Umsetzung weitergeholfen haben.

Des Weiteren möchten wir uns noch bei Prof. Hansjörg Huser sowie Manuel Bauer bedanken, welche uns eine grosse Hilfe waren im Umgang mit den Tools des MS SQL Servers.

7 Analyse der Flug- und Hotelkosten

7.1 Anforderungen

In diesem Kapitel sind die Ergebnisse der mehrheitlich manuell durchgeführten Forschungen zur Preisentwicklung beschrieben. Es ging darum, herauszufinden, welche Faktoren überhaupt einen Einfluss auf die Preisbildung haben könnten. Diese sollen erkannt werden, damit bei der späteren Analyse das Hauptaugenmerk auf diese Einflüsse gelegt werden konnte.

Die Diagramme die hier dargestellt sind basieren nicht auf reellen Resultaten und dienen nur zur Veranschaulichung möglicher Auswertungen.

7.1.1 Flugdaten

Anhand der vorhandenen Daten in der Datenbank soll das Verhalten der Flugpreise analysiert werden. Dabei sollen Tendenzen der Preise sowie die Parameter welche den Preis beeinflussen ersichtlich sein (Preisentwicklung, Kosten Frühbuchung, Kosten Last Minute, usw.) Die Daten für die Auswertung stammen dabei vor allem von den bereits durchgeführten Batch-Abfragen. Aber auch die vorhandenen Benutzerabfragen werden miteinbezogen falls diese passend sind.

Für die Auswertung der Flugpreise sollen Charts erstellt werden, welche die Tendenzen und Entwicklungen der Preise veranschaulichen.

Mögliche unabhängige Variablen, welche untersucht und verglichen werden sollen sind:

- Meeting Destination
- Meeting Date (Season)
- Meeting Duration
- Meeting Week Day (including weekends or only working days)
- Flight departure time of day
- Booking time "days before departure" (Anzahl Tage vor dem Meeting)
- Buchungsplattform

Zur Auswertung dieser Faktoren sind folgende Charts denkbar:

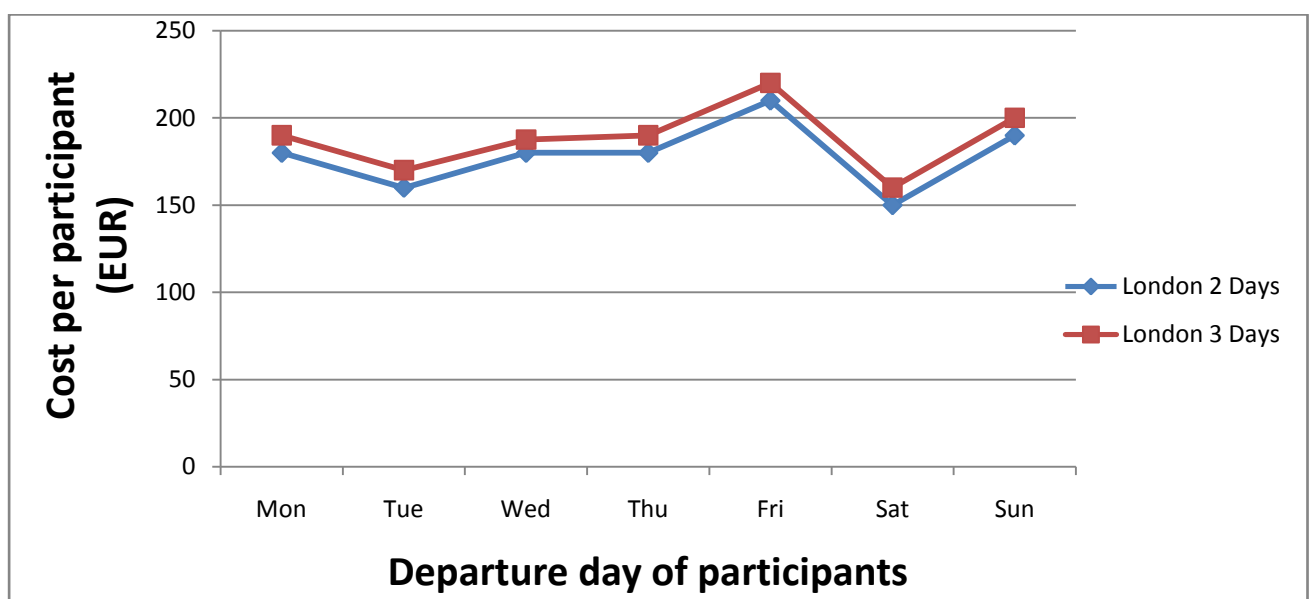


Abbildung 4: Mögliche Auswertung „Einfluss Wochentag“

Abbildung 4 zeigt die unterschiedlichen Flugkosten von in Abhängigkeit des Abflugwochentages.

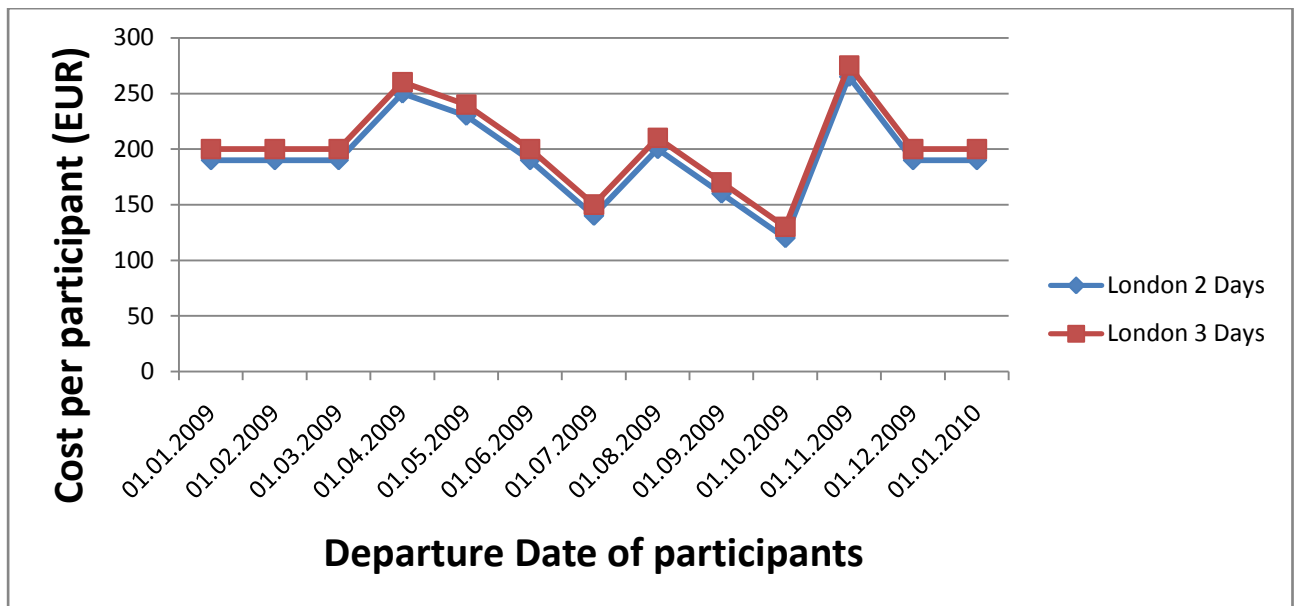


Abbildung 5: Mögliche Auswertung „Einfluss Durchführungsdatum (saisonal)“

Abbildung 5Abbildung 5 zeigt die saisonale Flugpreisentwicklung.

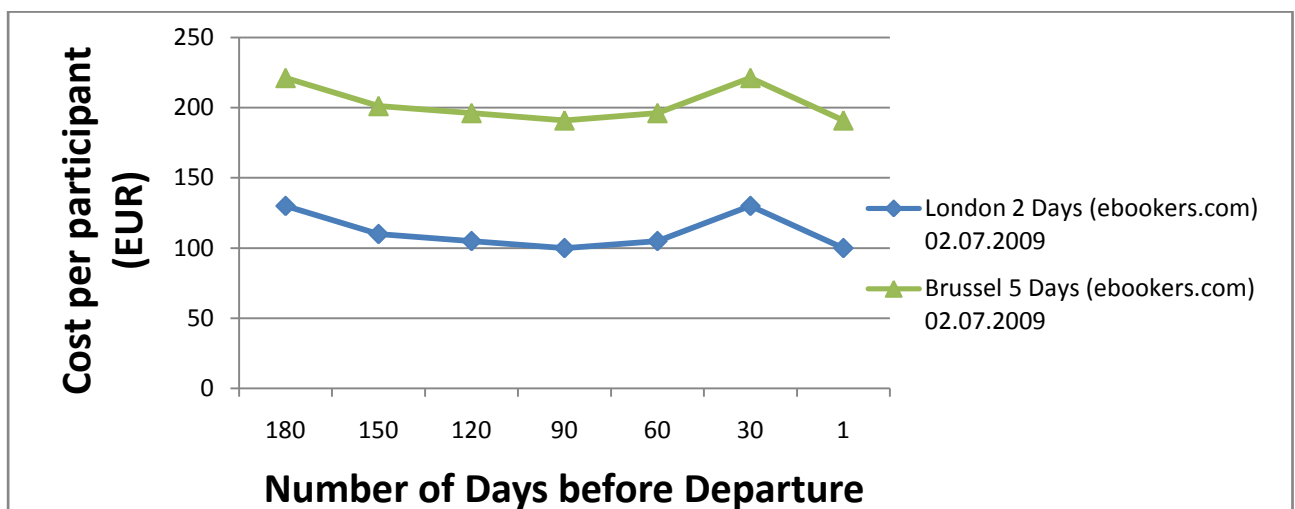


Abbildung 6: Mögliche Auswertung „Einfluss der Zeitspanne zwischen Buchung und Reise“

Abbildung 6 zeigt die Preisentwicklung in Abhängigkeit zum Buchungszeitpunkt. Das heisst, wie lange vor dem Flug, das Ticket gebucht wurde.

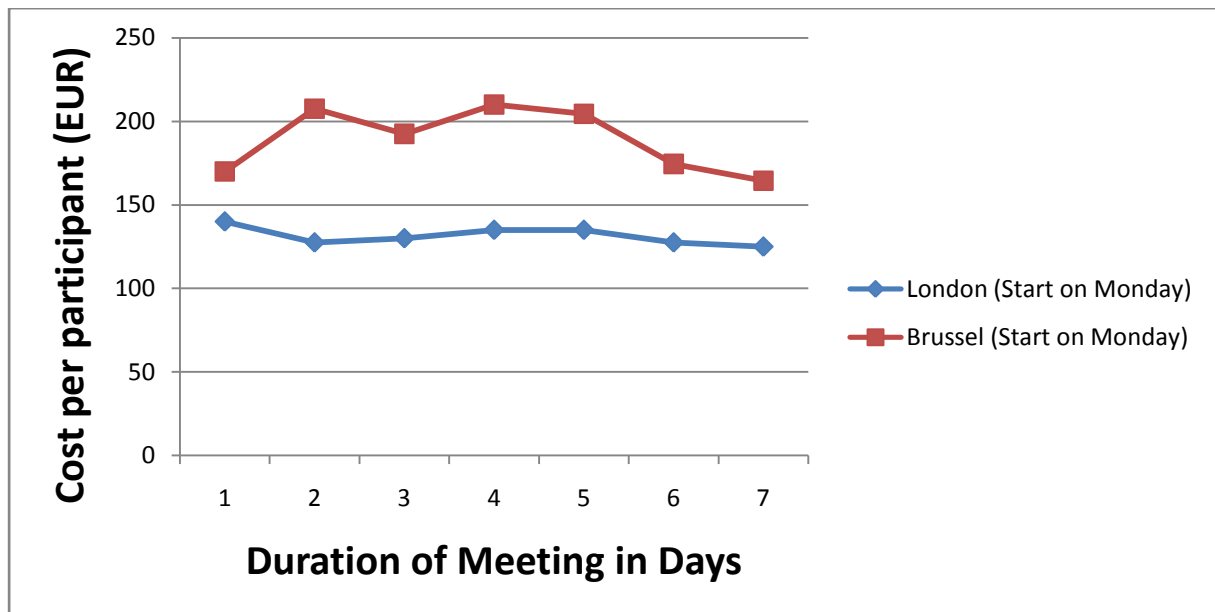


Abbildung 7: Mögliche Auswertung: „Einfluss Meeting-Dauer“

Abbildung 7 zeigt das Preisverhalten in Abhängigkeit der Aufenthaltsdauer am Destinationort.

7.1.2 Hoteldaten

Anhand der vorhandenen Daten in der Datenbank soll das Verhalten der Hotelpreise analysiert werden. Dabei sollen Tendenzen der Preise ersichtlich sein (Preisentwicklung, Kosten Frühbuchung, Kosten Last Minute, usw.) Die Daten für die Auswertung stammen dabei von den bereits durchgeführten Abfragen.

Für die Auswertung der Hotelpreise einer bestimmten Destination sollen Charts erstellt werden, welche das entsprechende Hotelangebot repräsentieren und veranschaulichen.

Als unabhängigen Variablen, welche untersucht und verglichen werden sollen sind folgende Einflüsse denkbar:

- Meeting place
- Meeting date (season)
- Meeting duration (duration of stay)
- Meeting week day (including weekends or only working days)
- Booking time “days before stay” (Anzahl Tage vor dem Meeting)
- Buchungsplattform

Zur Auswertung dieser Faktoren sind folgende Charts denkbar:

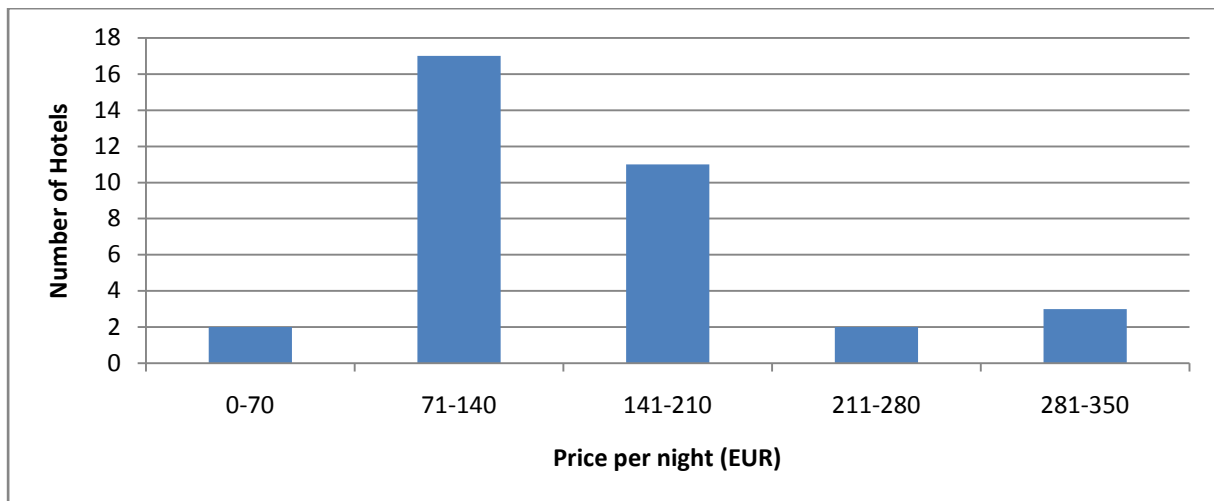


Abbildung 8: Mögliche Auswertung „Anzahl Hotels in verschiedenen Preissegmenten“

Abbildung 8 zeigt die Anzahl Hotels in einem bestimmten Preissegment einer Destination.
(Histogramm)

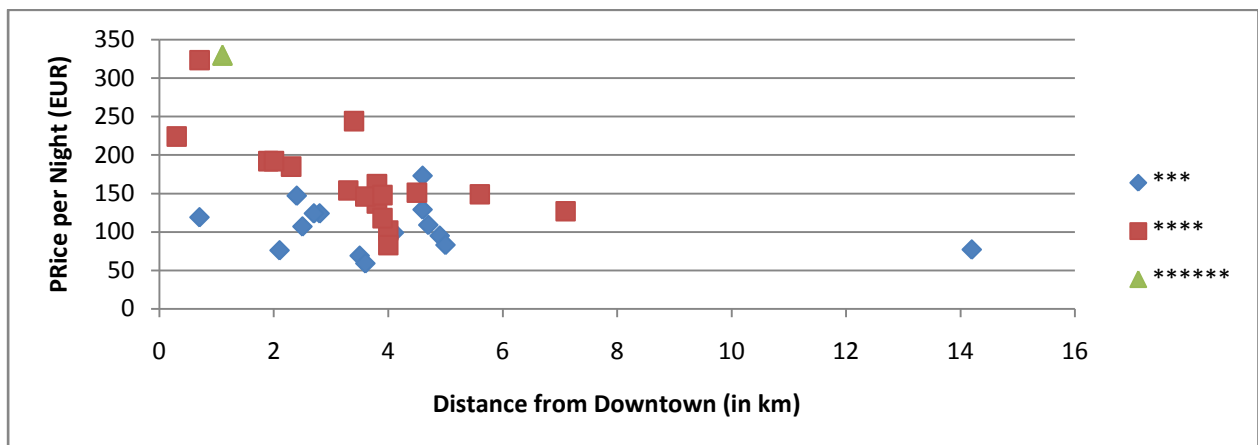


Abbildung 9: Mögliche Auswertung „Übersicht Hoteldetails“

Abbildung 9 zeigt das Hotelangebot einer Destination in Abhängigkeit des Preises pro Nacht, der Distanz des Hotels zum Stadtzentrum und dessen Komfortklasse.

7.2 Vorgehen

Um die Analysen durchzuführen, gibt es verschiedene Möglichkeiten.

Einerseits kann man die Daten mittels SQL-Abfragen aus der Datenbank herauslesen und exportieren. Die exportierten Daten können anschliessend zum Beispiel mit Excel aufbereitet werden. Diagramme erzeugt werden. Mit dieser Methode haben wir anfangs die ersten Graphen erstellt. Es ist sehr zeitaufwendig die Daten in die richtige Form für die Diagrammerzeugung zu bekommen. Dafür kann danach die recht einfache Bedienung von Excel genutzt werden.

Komfortabler ist es, die Daten zuerst in ein Data-Warehouse zu integrieren und einen sogenannten „Cube“ zu erstellen. D.h. die zu analysierenden Daten werden nach bestimmten Dimensionen geordnet. Daraus resultiert bildlich gesehen ein mehrdimensionaler „Datenwürfel“. Die Dimensionen können als Filter eingesetzt werden.



Abbildung 10: Veranschaulichung eines Data-Cube für die Datenanalyse

Es der Cube erstellt worden, kann mit den MS SQL Analysis Services den ganzen Cube als Pivot Tabelle direkt in Excel verwenden und damit beliebige Tabellen und Diagramme erstellen.

Die letzte und auch eleganteste Methode ist, mit den MS SQL Reporting Services direkt auf den erstellten Cube zuzugreifen um Auswertungen zu erstellen. Der Vorteil dieser Methode liegt darin, dass die Daten immer direkt von der Datenbank gelesen werden und das Ganze sehr schnell ist. Der Nachteil gegenüber dem Erstellen der Diagramme in Excel ist, das unübersichtliche Anpassen und Formatieren der Graphen.

In den Nächsten Kapiteln werden diese drei Methoden näher beschrieben.

7.2.1 Auswertung in Excel ohne Data-Warehouse und Cube

Am einfachsten geht der Export der Daten aus der Datenbank mit Hilfe der MS SQL Integration Services. Damit kann man verschiedene Tabellen laden und die entsprechenden Abfragen mehr oder weniger nur noch zusammenklicken. Die Resultate der Abfragen können direkt in Excel gespeichert werden.

name	duration	totalprice
amsterdam	0	99151.68
amsterdam	1	94829.44
amsterdam	2	76513.28
amsterdam	3	45554.24
amsterdam	4	45612.48
amsterdam	5	47306.88
amsterdam	6	38046.88
amsterdam	7	37854.88
athen	0	67296.96
athen	1	65728.96
athen	2	68115.36
athen	3	48562.74
athen	4	50433.92

Abbildung 11: Unbearbeitete Export-Tabelle

Leider wird bei dieser Art von Export nur eine Tabelle in der Form wie in Abbildung 11 erstellt. Eine solche Tabelle ist nur begrenzt zum erstellen eines Diagramms in Excel geeignet. Darum muss man erst einmal die ganzen Daten von Hand in eine Form bringen, in welcher Excel die Daten in einem Diagramm darstellen kann. Eine Möglichkeit dies zu tun, wäre die Verwendung eines Makros welches dies erledigt. Darauf haben wir aber verzichtet da wir das ganze nur einige wenige Male gemacht haben.

	moskau	zürich	berlin
0	65952.32	139548.8	36959.2
1	85627.2	113729.9	54562.88
2	83098.88	66962.24	43369.28
3	63053.01	47641.12	26628.18
4	64060.32	49082.88	29390.24
5	65722.24	49266.24	29303.36
6	64728.64	47794.24	33162.4
7	64248.48	47265.92	29296.96

Abbildung 12: Bearbeitete Export-Tabelle

Die Daten sollten nach dem Aufbereiten ähnlich wie in Abbildung 12 aussehen. Sobald die Daten in dieser Form sind kann man recht einfach Diagramme erstellen.

7.2.2 Auswertung mit Data-Warehouse und Cube

Um die Daten mit dieser Variante auszuwerten, müssen die Daten von der Hauptdatenbank (relationale, produktive Datenbank) in ein Data-Warehouse migriert werden. Dies wird mit Hilfe der MS SQL Integration Services realisiert. Wie dies detailliert geschieht, wird in Kapitel 8.5.2.3.1 erklärt.

Sobald die Daten im Data-Warehouse vorhanden sind, kann man mit den MS SQL Analysis Services einen Cube erstellen. Dies kann mit Hilfe eines Assistenten recht einfach gemacht werden. Beim Erstellen muss man zwischen Facts und Dimension Tabellen unterscheiden. Facts-Tabellen enthalten die zu analysierende Daten, bei uns sind das die Reise- und die Hoteltabelle. Die Anderen Tabellen sind die Dimensionstabellen. Die Beziehung zwischen Dimensionstabelle und Facts-Tabelle muss immer n:1 sein.

Die Dimensionen dienen zur Filterung der Fact-Daten. Man kann auch manuell zusätzliche Dimensionen hinzufügen, ohne dafür eine Tabelle zu haben. Auch den Fact-Tables können manuell zusätzliche Messgrößen(Werte) hinzugefügt werden. Sobald man einen Cube erstellt hat, kann man im Cube-Browser die Daten anschauen. Auch kann man den Cube in Excel importieren, was im nächsten Abschnitt erklärt wird.

7.2.2.1 Excel mit Cube

Das Einbinden eines Cube ist recht einfach. Unter Daten->Externe Daten ->Analysis Services muss der gewünschte Cube selektiert werden. Danach hat man eine Pivot Tabelle zur Verfügung. In dieser Tabelle kann man recht einfach die gewünschten Spalten und Zeilen zusammenstellen, die anzuzeigenden Daten auswählen und bei Bedarf die Daten noch filtern. Die Diagramme kann man direkt aus der Pivot Tabelle erzeugen. Wird dies so gemacht, sollte die Tabelle aber nicht mehr gross verändert werden, da das Diagramm bei abweichenden Spalten- oder Zeilenanzahlen nicht automatisch angepasst wird.

7.2.2.2 Data-Mining

Mit den Analysis Services können auch verschiedene Data Mining Algorithmen gestartet werden. Für das ermitteln der wichtigsten Faktoren haben wir auf den Decision Tree Algorithmus gesetzt welcher die Wichtigkeit der Faktoren anzeigt. Bei einer grossen Datenmenge kann die Berechnung jedoch ziemlich viel Zeit in Anspruch nehmen

7.2.2.3 SQL Reporting Services

Sobald ein Cube gebildet worden ist, kann man die MS SQL Reporting Services verwenden, um die Daten auszuwerten.

Das Arbeiten mit den Reporting Services ist recht einfach. Man kann eine Reporttabelle oder eine Matrix mit einem Wizard erstellen.

Auch Diagramme können erstellt werden. Als erstes muss man die anzuzeigenden Daten auswählen. Dabei können allenfalls Filter verwendet werden. Sind die gewünschten Daten ausgewählt, werden diese einem Diagramm zugewiesen. Es stehen verschiedene Arten von Diagrammen zur Auswahl.

7.3 Ergebnisse

Nachfolgend werden die Ergebnisse der Analyse der gesammelten Daten dargestellt. Alle Preise sind immer in Euro angegeben und beziehen sich auf eine gesamte Flugreise. D.h. mit Hin- sowie Rückflug. Die Ergebnisse beruhen immer nur auf unseren gesammelten Daten.

7.3.1 Flugdaten

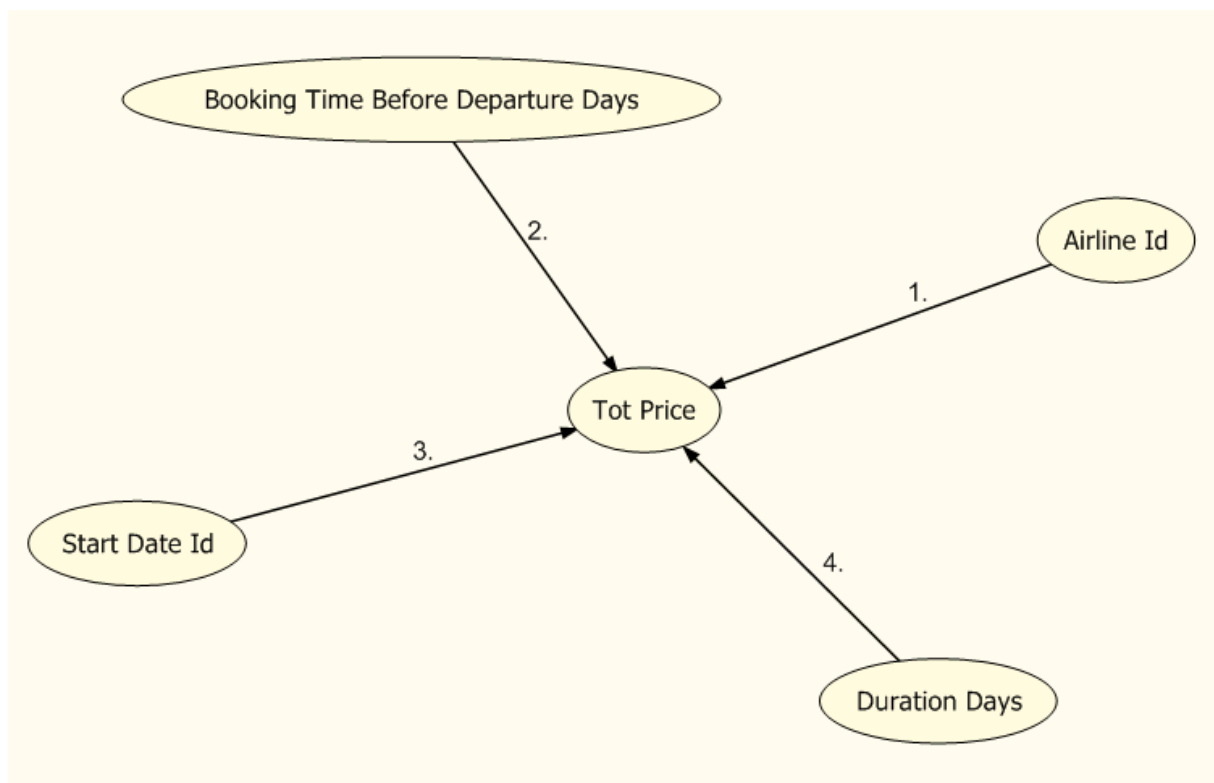


Abbildung 13: Einfluss von Faktoren auf die Flugkosten

Mit einem Data-Mining Algorithmus wurden die Faktoren mit einem direkten Einfluss auf den Flugpreis gesucht. Wie in Abbildung 13 zu sehen ist, hat die Wahl der Airline den grössten Einfluss auf den Preis, was nicht weiter verwundert. Danach folgen in dieser Reihenfolge die Zeitspanne zwischen Buchungs- und Reisedatum, das Abreisedatum und die Aufenthaltsdauer. Diese Ergebnisse entstanden bei der Verwendung des Microsoft Decision Tree Algorithmus. Den Einfluss der Airline werden wir nicht näher anschauen, da es nicht darum geht, die billigste Airline zu finden. Als nächstes schauen wir die anderen drei Faktoren genauer an.

7.3.1.1 Aufenthaltsdauer

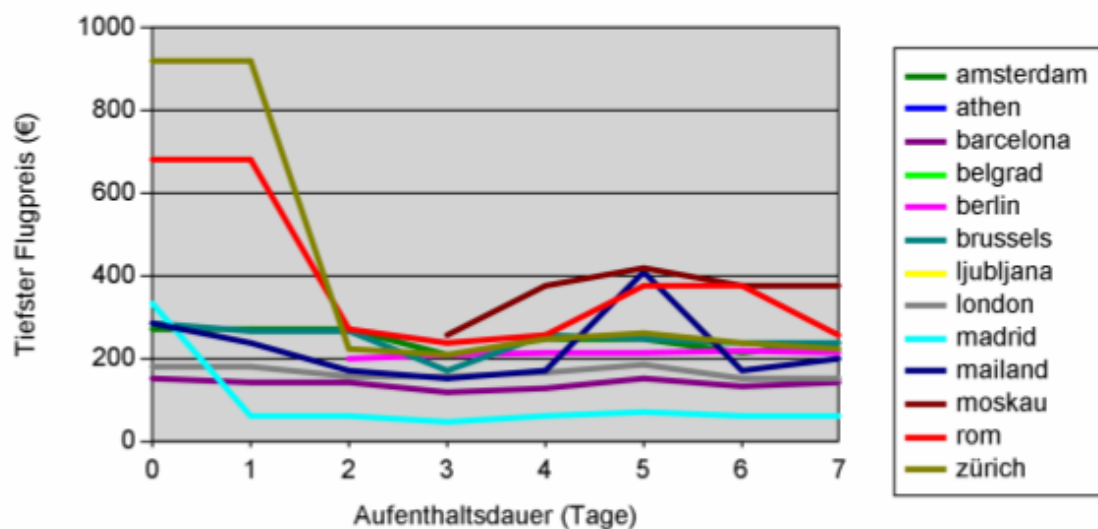


Abbildung 14: Startort Lissabon - Tiefste Flugreisekosten in Abhängigkeit der Aufenthaltsdauer

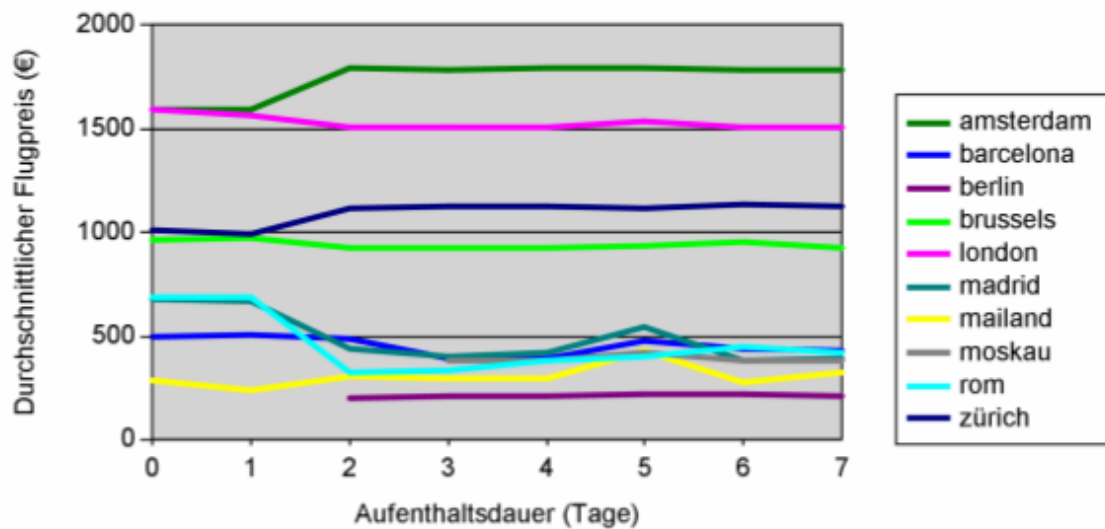


Abbildung 15: Startort Lissabon - Durchschn. Flugreisekosten in Abhängigkeit der Aufenthaltsdauer

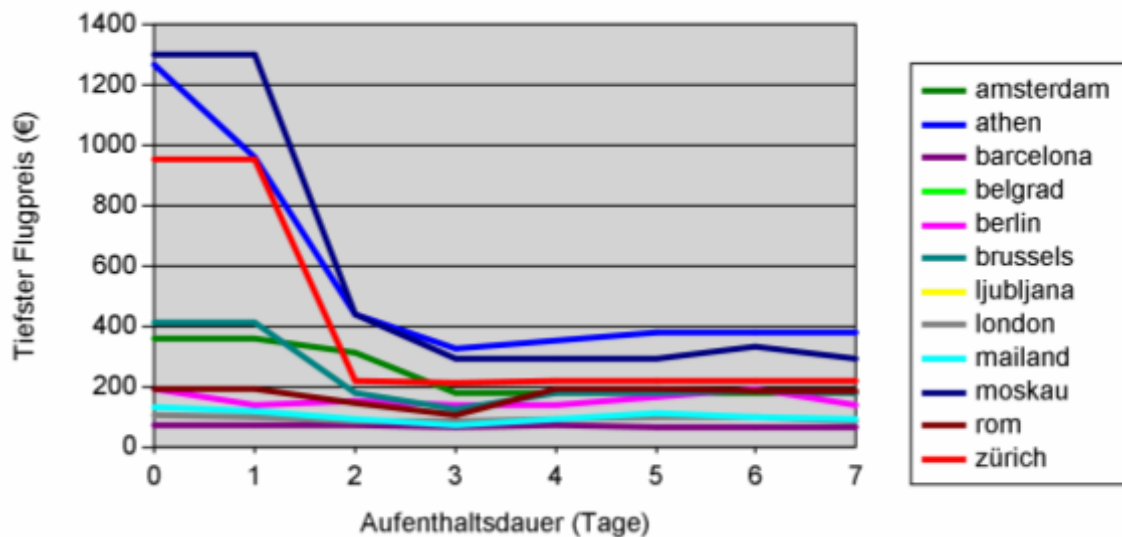


Abbildung 16: Startort Madrid - Tiefste Flugreisekosten in Abhängigkeit der Aufenthaltsdauer

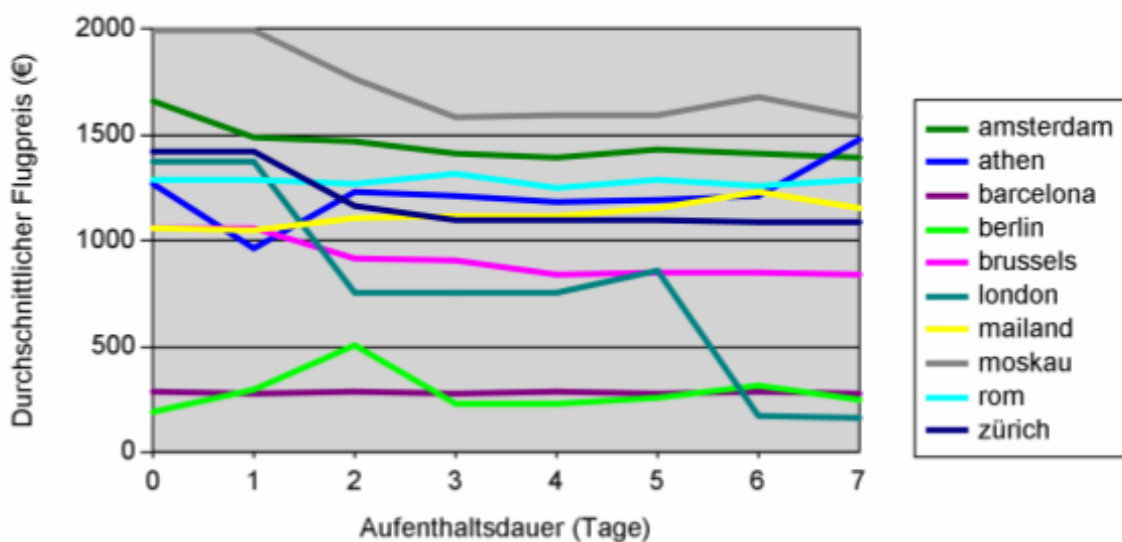


Abbildung 17: Startort Madrid - Durchschn. Flugreisekosten in Abhängigkeit der Aufenthaltsdauer

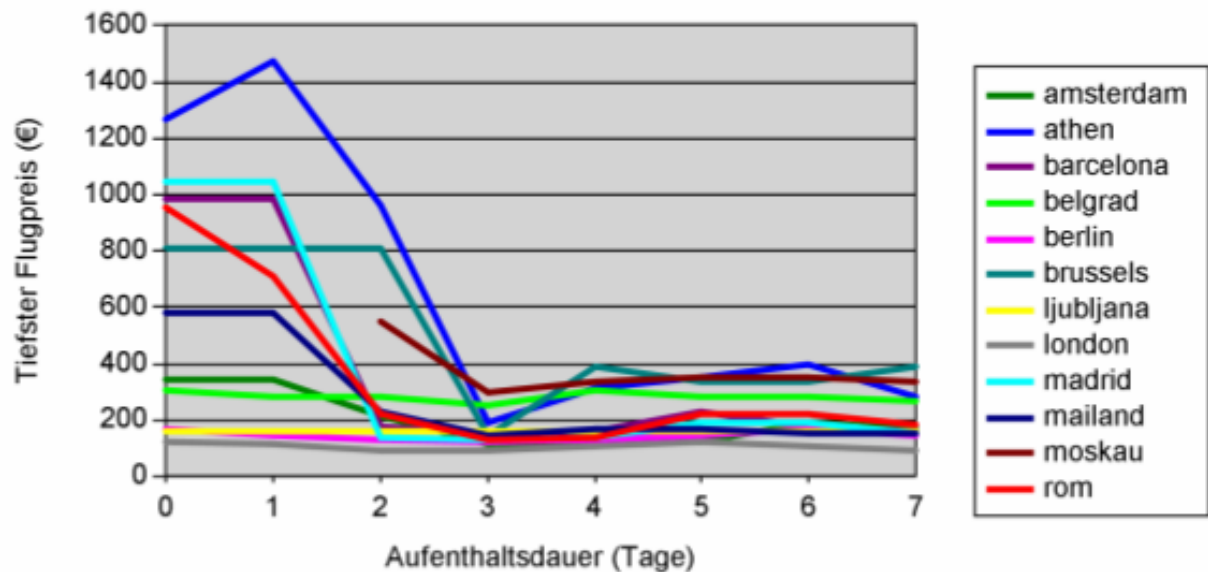


Abbildung 18: Startort Zürich - Tiefste Flugreisekosten in Abhängigkeit der Aufenthaltsdauer

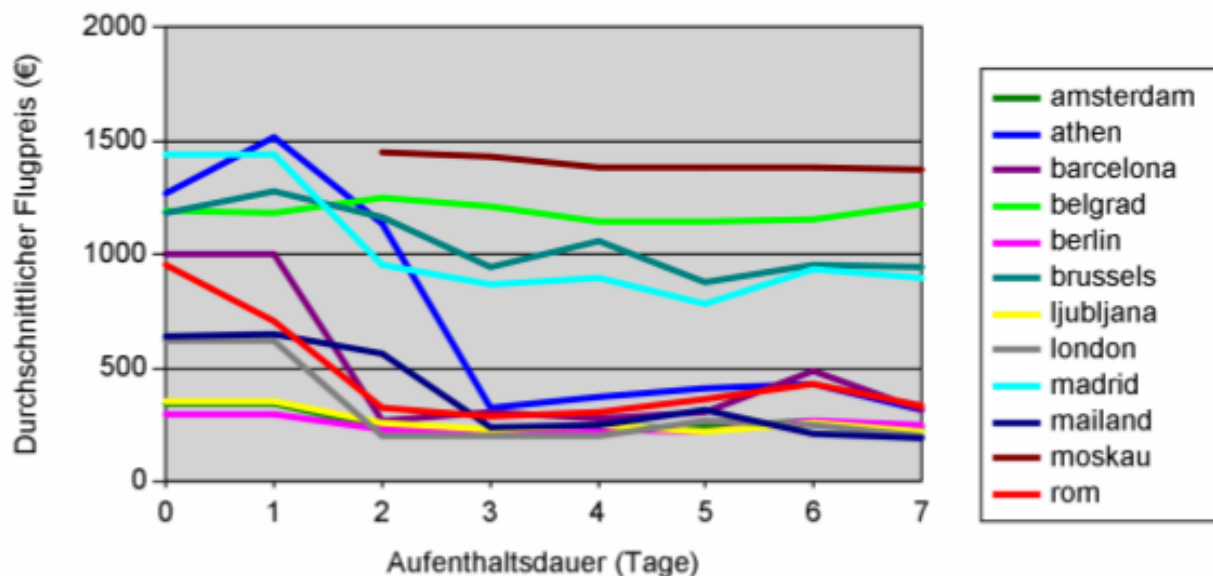


Abbildung 19: Startort Zürich - Durchschn. Flugreisekosten in Abhängigkeit der Aufenthaltsdauer

Wie man in den Diagrammen erkennen kann, sind die Kosten bei sehr kurzer Aufenthaltsdauer um einiges höher als wenn man mehr als zwei Nächte bleibt. Sobald die Aufenthaltsdauer länger als 3 Tage beträgt, stabilisieren sich die Preise.

Im Weiteren lässt sich erkennen, dass auf den kürzeren Flugstrecken die Preise auch bei kurzer Aufenthaltsdauer eher stabil sind.

Die sehr teuren Preise bei kurzer Aufenthaltsdauer entstehen vermutlich daraus, dass das Angebot an Rückflügen gleichen Tag sehr knapp ist. Diese wenige Flüge sind meistens auch teurer.

Bei gewissen Destinationen sind sogar gar keine Rückflug-Angebote am gleichen Tag gefunden worden.

Für die weiteren Startorte sieht das Preisverhalten meist sehr ähnlich aus.

7.3.1.2 Abflugdatum (Saisonal)

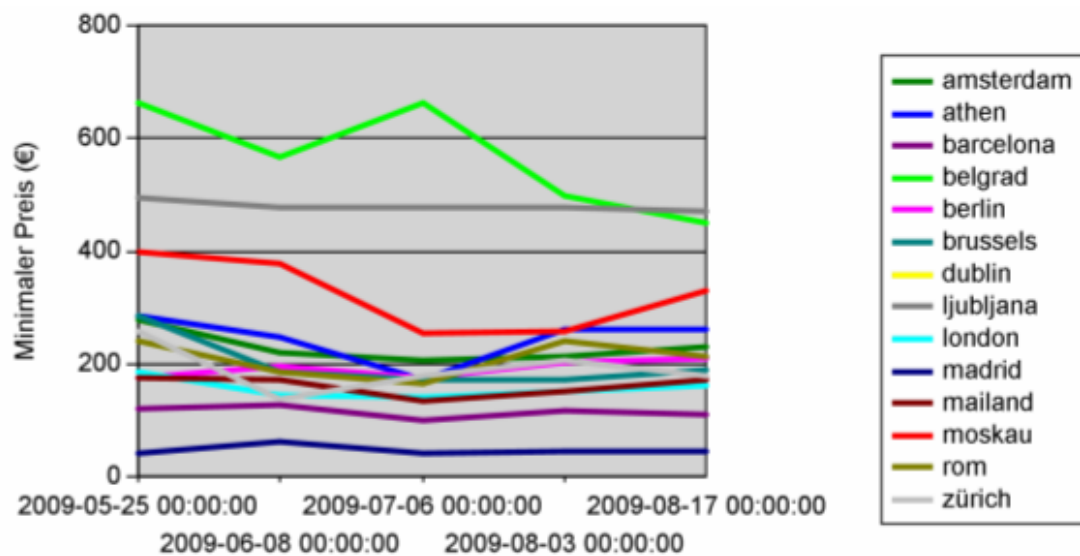


Abbildung 20: Startort Lissabon - Saisonale Entwicklung der Flugreisekosten

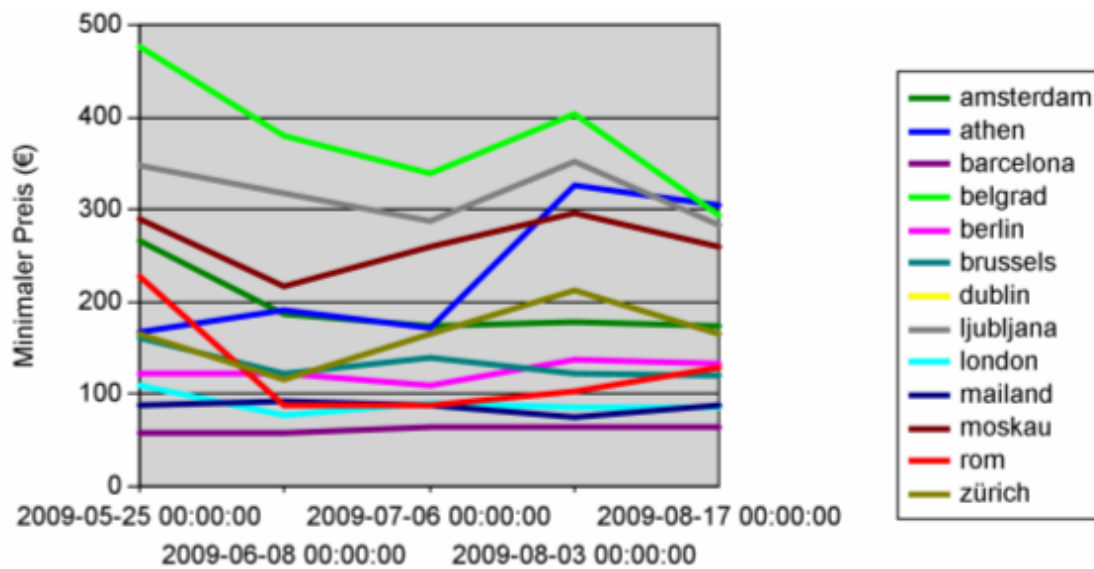


Abbildung 21: Startort Madrid - Saisonale Entwicklung der Flugreisekosten

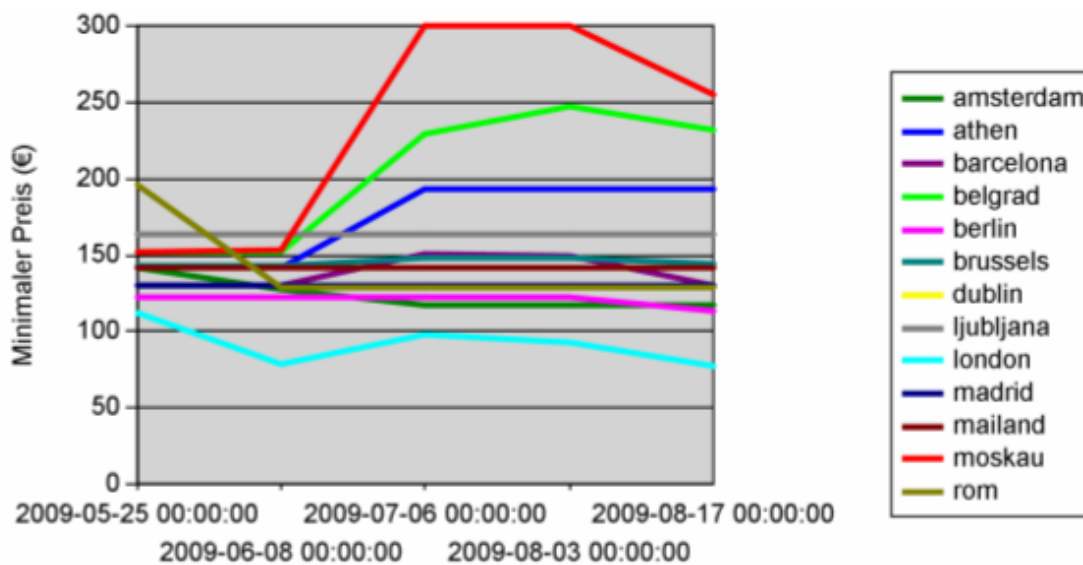


Abbildung 22: Startort Zürich - Saisonale Entwicklung der Flugreisekosten

Bei der Analyse der Preise sind wir auf ein recht erstaunliches Ergebnis gekommen. Die Preise sind ausser bei kurzfristigen Reisen meistens konstant oder ändern sich in einem bestimmten wiederkehrenden Rhythmus. Zum Beispiel gibt es Flüge von Billig-Airlines, welche nur alle zwei Tage stattfinden.

Dieses Phänomen kann auch durch unsere Methode der Datensammlung entstanden sein. Wir konnten wegen dem engen Zeitplan nicht die Daten über ein ganzes Jahr sammeln. Vielleicht würde sich da ein anderes Bild zeigen.

7.3.1.3 Buchungszeitpunkt vor Abflug

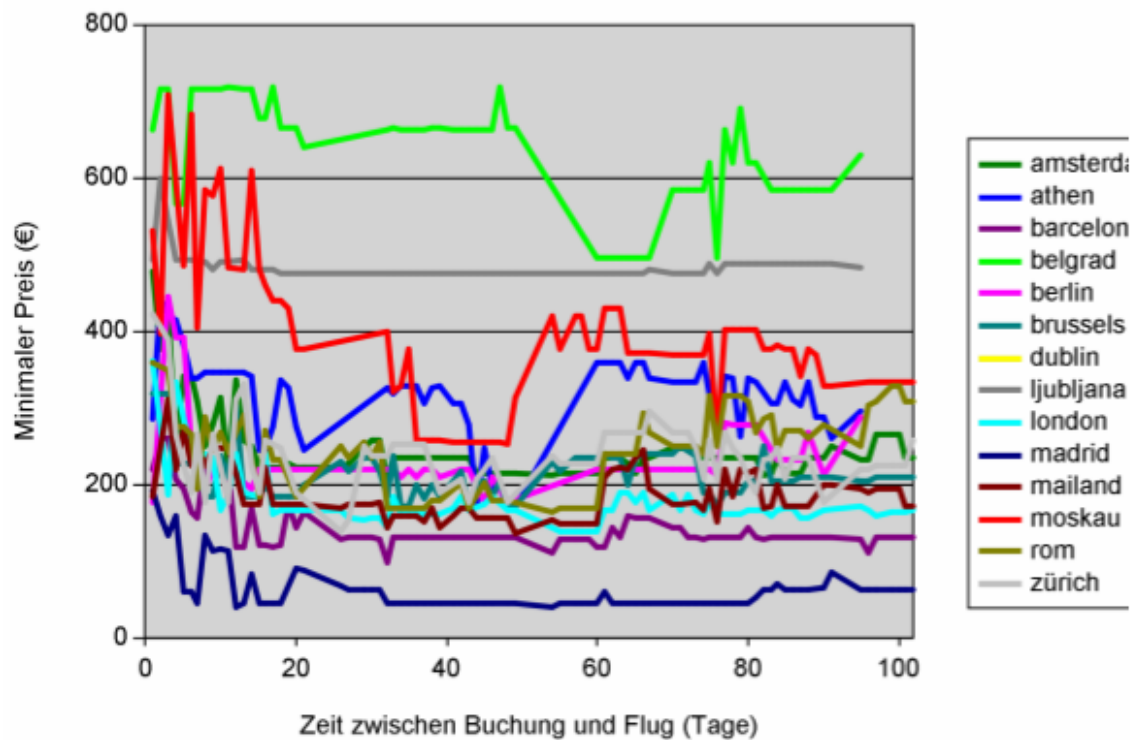


Abbildung 23: Startort Lissabon - Einfluss der Zeitspanne zwischen Buchung und Reise auf die Flugreisekosten

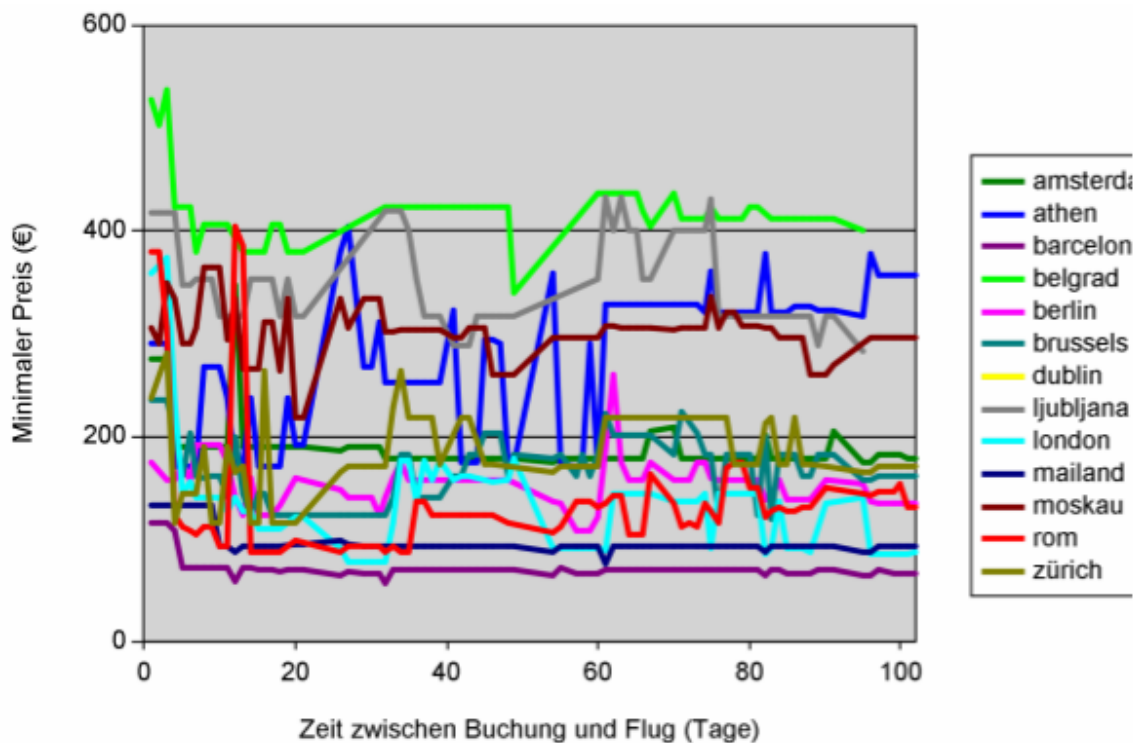


Abbildung 24: Startort Madrid - Einfluss der Zeitspanne zwischen Buchung und Reise auf die Flugreisekosten

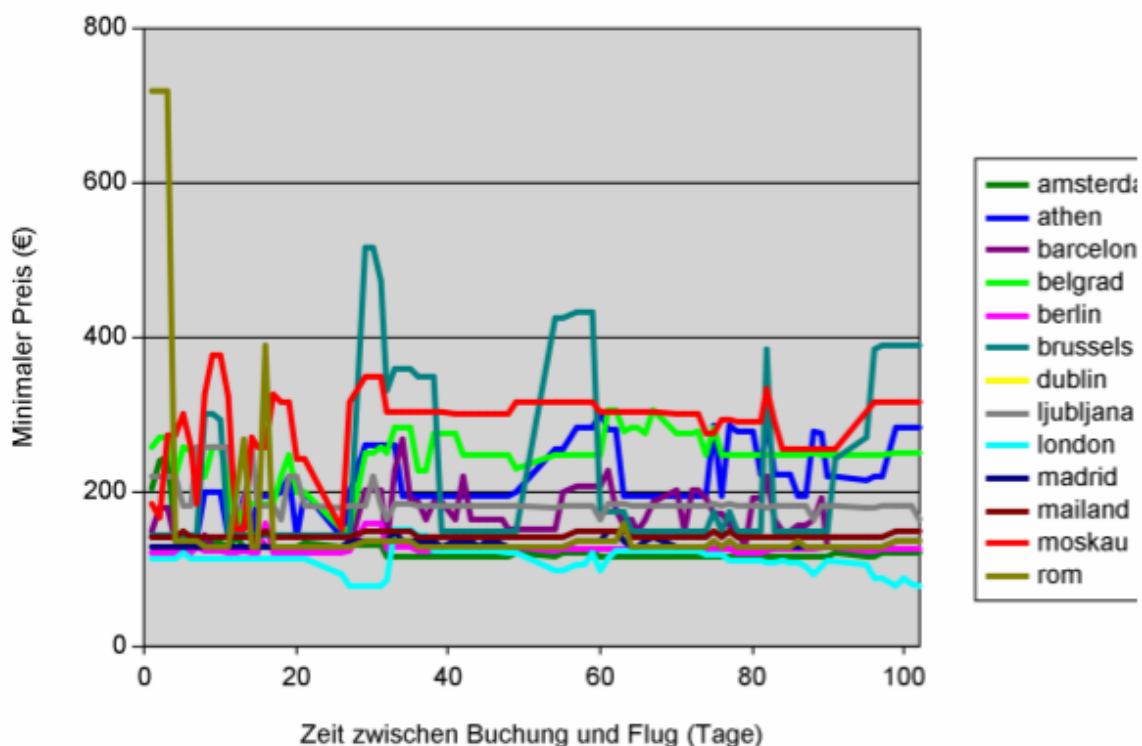


Abbildung 25: Startort Zürich - Einfluss der Zeitspanne zwischen Buchung und Reise auf die Flugreisekosten

Wie man aus den Diagrammen erkennen kann, sind die Preise vor allem im Bereich unter 10 Tagen sehr variabel.

Je weiter im Voraus die Buchung vorgenommen wurde, umso mehr stabilisierten sich die Preise. Grundsätzlich kann aber anhand der Graphen gesagt werden, dass nach diesen Auswertungen die Preise bei Last-Minute Buchungen eher ein bisschen höher angesiedelt sind.

7.3.2 Hoteldaten

7.3.2.1 Aufenthaltsdauer

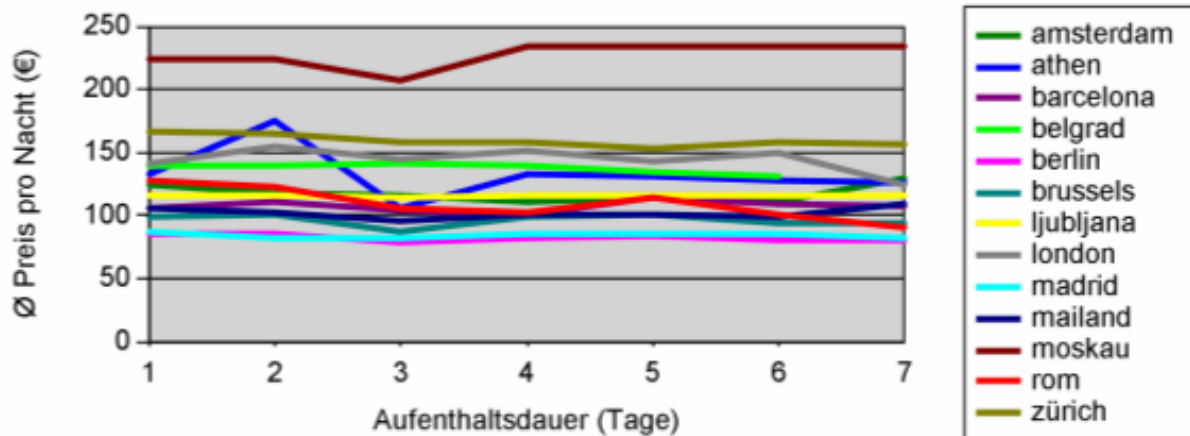


Abbildung 26: 3-Sterne Hotels - Einfluss der Aufenthaltsdauer auf die Hotelkosten

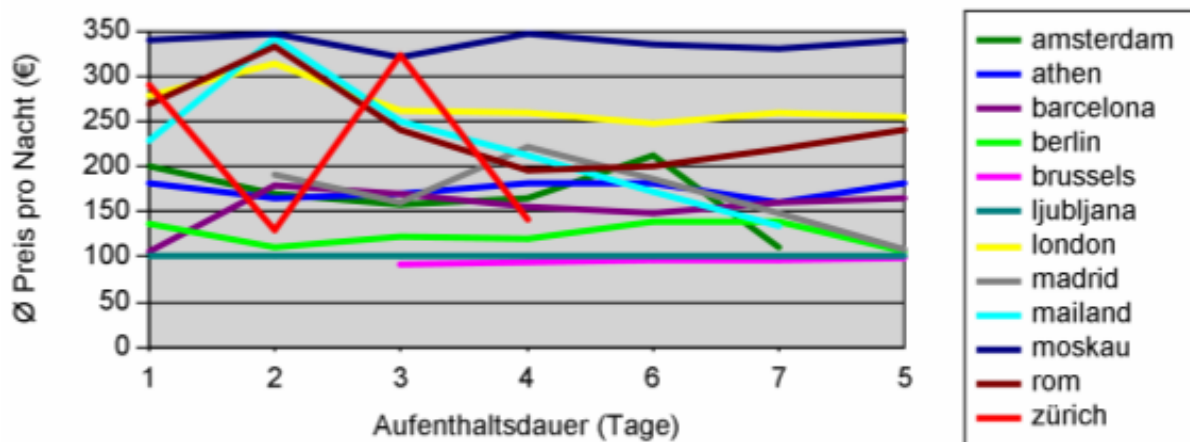


Abbildung 27: 5-Sterne Hotels - Einfluss der Aufenthaltsdauer auf die Hotelkosten

Wie man in den Diagrammen sehen kann sind die Hotelkosten pro Nacht bis auf einigen Ausnahmen sehr stabil. Bei den 3-Sterne Hotels etwas mehr als bei der 5-Sterne Hotels. Dies lässt sich eventuell mit der Anzahl der Resultate erklären. Den es gibt meist nur wenige 5-Sterne Hotels an einem Ort im Gegensatz zu den 3-Sterne Hotels.

7.3.2.2 Datum des Aufenthaltes (Saisonal)

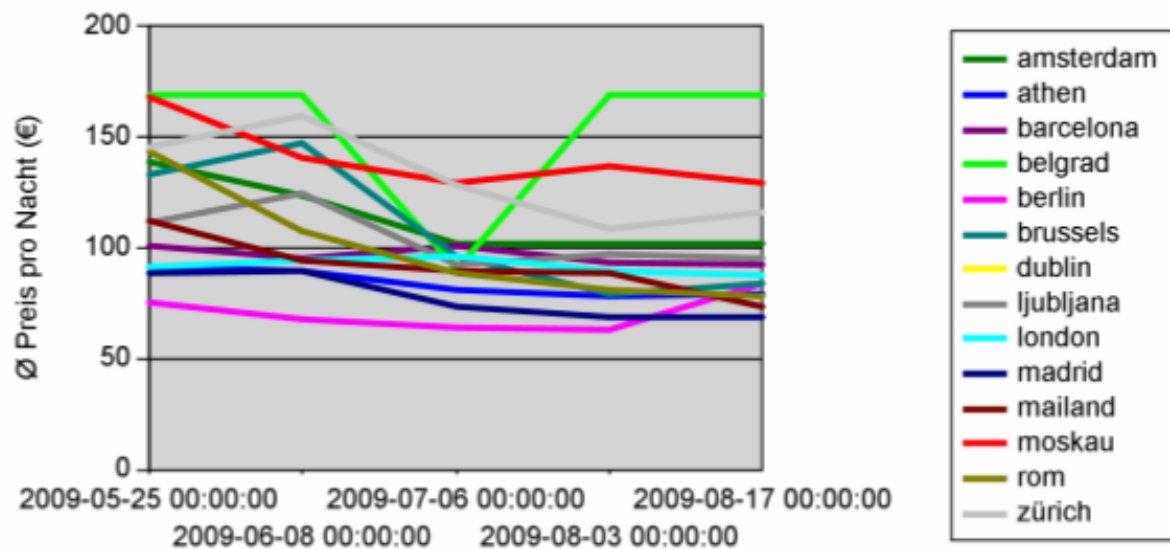


Abbildung 28: 3-Sterne Hotels - Einfluss des Aufenthaltsdatums auf die Hotelkosten

Die Hotelpreise variieren je nach Aufenthaltszeitpunkt ein wenig. Wobei sich dabei keine bestimmte Regel erkennen lässt. Die Preise verändern sich je nach Saison und die kann je nach Destination unterschiedlich sein.

7.3.2.3 Buchungszeitpunkt vor Aufenthalt

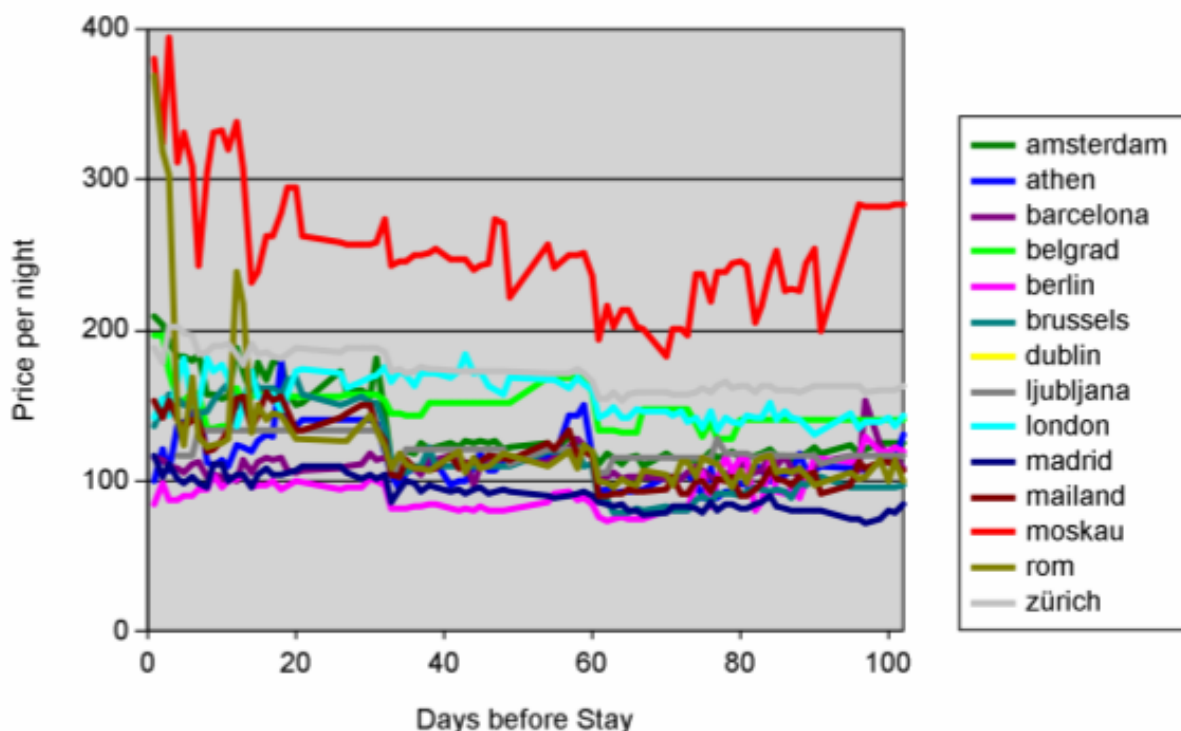


Abbildung 29: 3-Sterne Hotels - Einfluss der Zeitspanne zwischen Buchung und Aufenthalt auf die Hotelkosten

Wie man hier sehen kann verhalten sich die Hotelpreise ähnlich wie die Flugpreise. Wobei hier die Preise kurz bei Last-Minute Buchungen meist nicht viel tiefer sind aber dafür richtig stark ansteigen können. Darum lohnt es sich kaum zu warten bis ein Hotel günstiger wird, denn die Preise sind meistens fix vorgegeben.

7.3.2.4 Hotelangebot

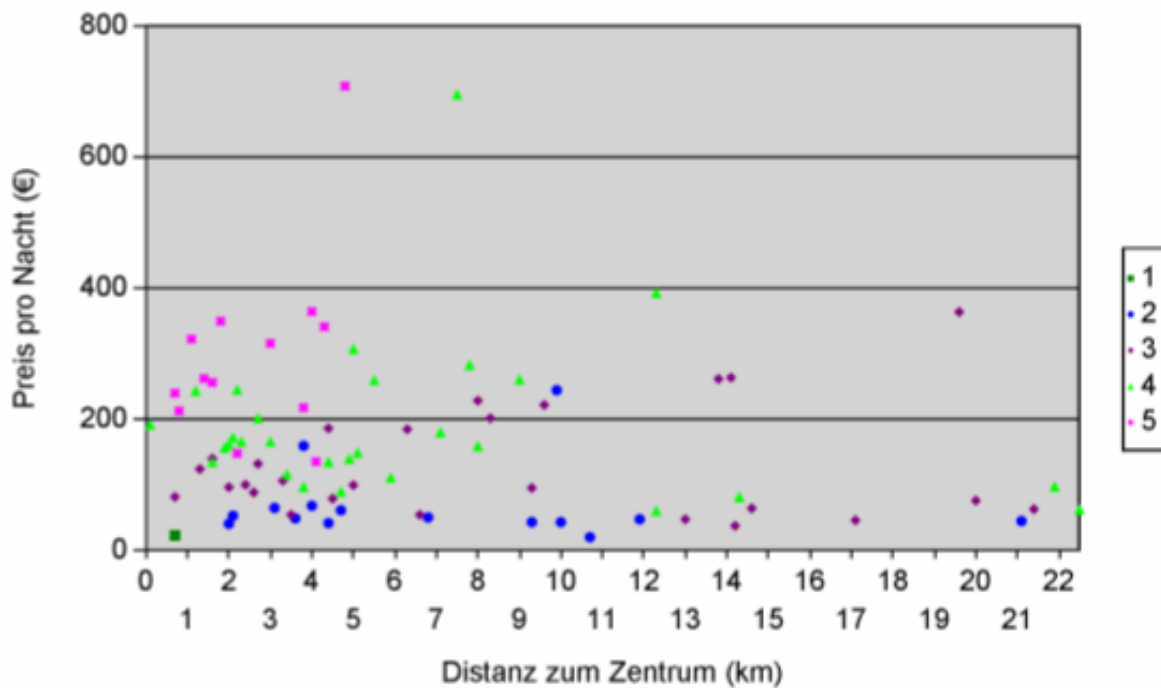


Abbildung 30: London - Übersicht des Hotelangebotes

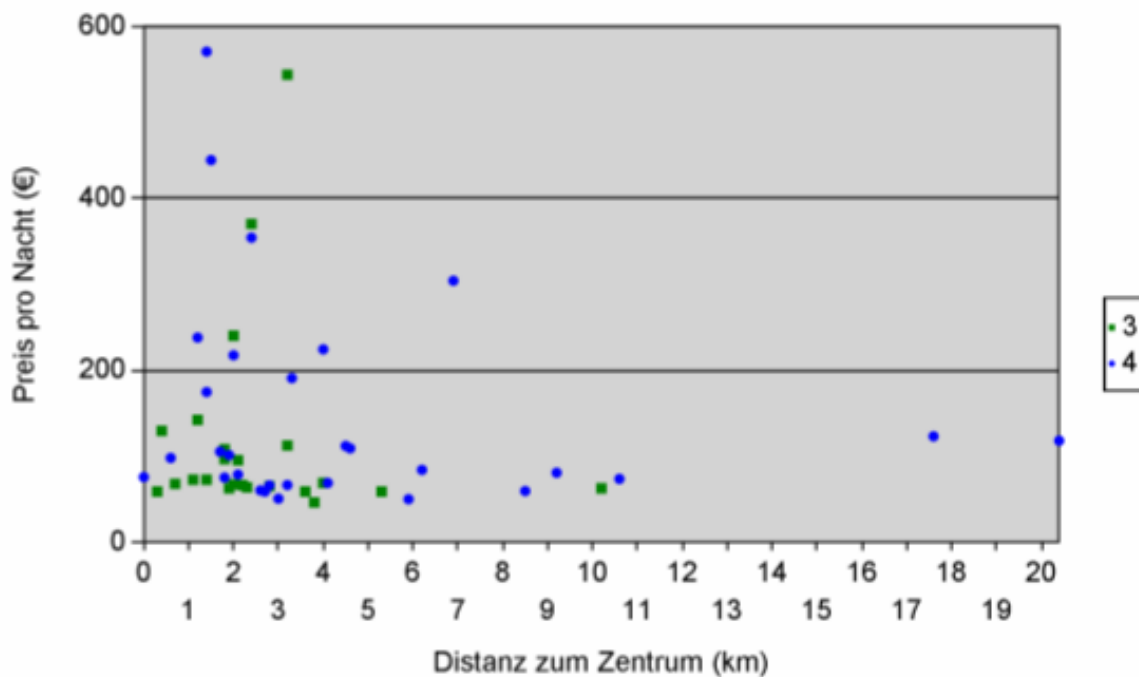


Abbildung 31: Madrid - Übersicht des Hotelangebotes

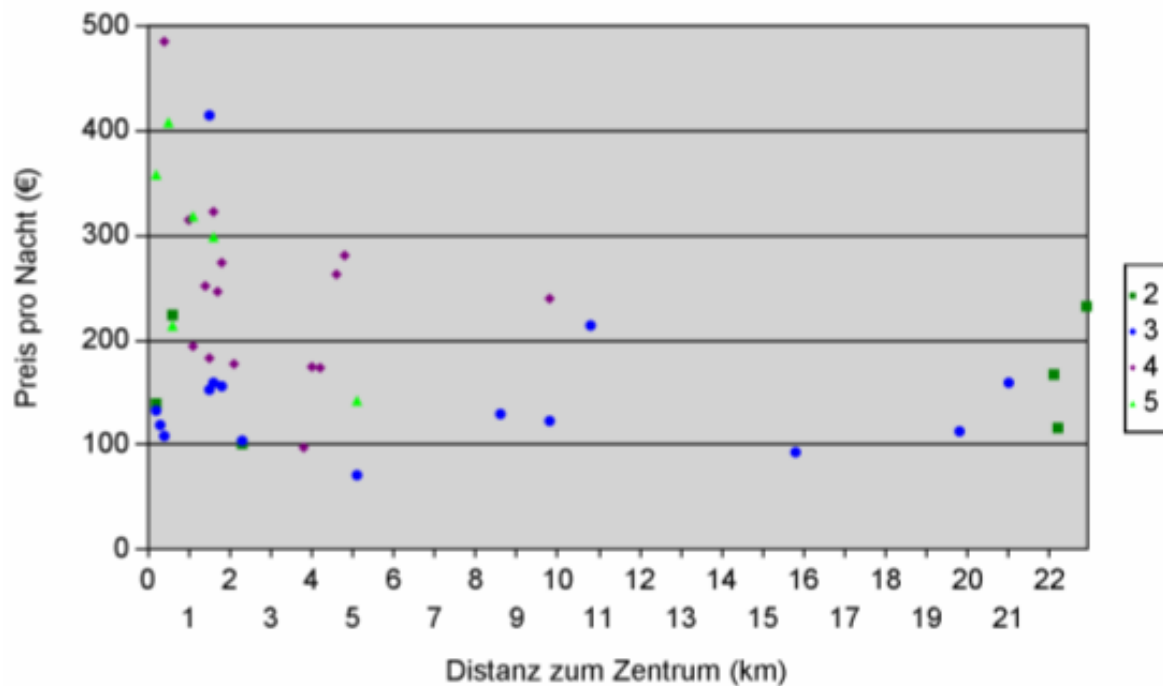


Abbildung 32: Zürich - Übersicht des Hotelangebotes

In diesen Diagrammen sind die Unterschiede im Hotelangebot der verschiedenen Destinationen zu erkennen.

7.3.3 Zusammenfassung

Die Preise werden von verschiedenen Faktoren beeinflusst.

Sobald aber der Aufenthalt länger als 3 Tage dauert und man nicht in den nächsten 2 Wochen reist sind die Preise relativ konstant.

Die Preise weisen meist auch saisonale Schwankungen auf, was aber meistens nicht zu umgehen ist, da der Zeitraum einer Reise meistens vorgegeben ist.

Beim Buchen gibt es kein Patentrezept wie man an einen möglichst billigen Flug bekommt. Zu empfehlen ist, dass man früh bucht da man ja nicht wirklich flexibel sonstwohin gehen kann, wenn das Flugzeug voll ist.

Bei den Hotelkosten spielt die Dauer des Aufenthalts nur eine minimale Rolle. Die Preise verhalten sich meist linear mit zur Aufenthaltsdauer. Auch bei den Hotelpreisen gibt es saisonale Unterschiede. Ein Hotelzimmer ist in der Nebensaison billiger als in der Hauptsaison. Zu langes Warten vor der Hotelbuchung lohnt sich meistens auch nicht, weil sich die Preise nicht gross verändern. Man riskiert höchstens, dass man kein Hotel mehr findet, weil alle ausgebucht sind.

Anhand unserer Daten könnte man zusammenfassend sagen:

Am besten plant man ein Meeting möglichst früh und mit einer Dauer von mindestens drei Tagen.

Diese Aussage bezieht sich nur auf die Gesamtkosten-Aufenthaltsdauer Verhältnis.

8 Software-Projekt Dokumentation

8.1 Technologien

8.1.1 C#

Bei der Wahl der zu verwenden Technologie für die zu erarbeitenden Applikationen entschieden wir uns für C#. Dies vor allem aus den folgenden drei Gründen.

Erstens wollten wir unsere bis anhin eher spärlichen Erfahrungen im .NET Umfeld mit dem Gebrauch von C# erweitern. Der zweite Grund war, dass wir uns für MS SQL Server als Datenbank-Managementssystem entschieden und die C# Umgebung viele Features bereitstellt für den effizienten Gebrauch dieser Datenbanklösung (z.B. LINQ to SQL). Auch in Anbetracht der zu entwickelnden Webapplikation hatten wir durch diese Wahl mit ASP.NET eine mächtige Technologie zur Verfügung.

8.1.1.1 Verwendete externe Bibliotheken

8.1.1.1.1 HTML Agility Pack

Um Flug- und Hoteldaten zu sammeln mussten HTML-Seiten ausgelesen werden. Dies kann schnell zu Problemen führen da die Seiten meist nicht aus wohlgeformten XML bestehen. Mit der Open Source Bibliothek HTML Agility Pack wird dieses Problem umgangen. Es vereinfacht das Parsen einer Webseite wesentlich. Auf die verschiedenen Elemente der HTML-Seite kann mit Hilfe von XPath zugegriffen werden.

8.1.1.1.2 Thread Pool Helper

Um die Fluganfragen parallel durchzuführen wurde der von C# bereitgestellte Thread Pool benutzt. Dies hat den Vorteil, dass die zu parallelisierenden Funktionen lediglich dem Thread Pool übergeben werden müssen und der Entwickler sich danach nicht mehr um die Verwaltung der Threads kümmern muss.

Um den Umgang mit dem Thread Pool noch einfacher zu gestalten sind wir auf eine kleine Wrapper-Klasse gestossen, welche den Gebrauch des Thread Pools nochmals erleichtert. Diese Klasse kapselt die Funktionalität des Thread Pools und wurde bei unserer Applikation eingesetzt.

8.1.2 MS SQL Server 2005

Dank der Wahl von MS SQL Server als Datenbanksystem hatten wir eine Vielzahl von zusätzlichen Werkzeugen für die Datenanalyse und das Reporting zur Verfügung.

Bei der Auswertung der Flug- und Hoteldaten konnten wir, nach einiger Einarbeitungszeit, enorm von den mitgelieferten Analysis Services profitieren, die uns das ganze erheblich erleichtert haben. Auch für das Erstellen der Diagramme konnten wir auf die in der Umgebung integrierten Reporting Services zurückgreifen.

8.1.3 Google WebDriver

Ganz am Anfang des Projekts wurden Testweise mit dem Google WebDriver Anfragen an ein online Buchungsportal gesendet und die Resultate in einem Excel-Sheet ausgewertet. Dies wurde gemacht, um zu sehen, ob sich überhaupt Tendenzen erkennen lassen und eine genauere Analyse dieser Daten überhaupt Sinn machen würde.

Mit Hilfe des Google WebDrivers können Web Applikationen automatisiert getestet werden. Die API basiert auf Java.

Kernfunktionalität des WebDrivers ist es, verschieden Requests automatisiert ablaufen zu lassen. So können Features einer Webapplikation auf eine einfache Art und Weise getestet werden.

Es gibt grundsätzlich zwei verschiedene Varianten wie dies realisiert werden kann.

HTMLUnitDriver:

- Requests werden im Hintergrund ausgeführt
- Es wird kein Browser gestartet

„Browser Driver“ (FirefoxDriver oder InternetExplorerDriver):

- Browser wird geöffnet und die erstellten Requests werden automatisch im Browser ausgeführt

Um die Funktionalität des WebDrivers zu testen haben wir eine kleine „Anwendung“ realisiert welche auf www.ebookers.ch eine Anfrage für Flugverbindungen automatisiert abarbeitet. Dabei werden die Formularfelder der Flugsuche automatisch durch den WebDriver ausgefüllt und die Anfrage abgeschickt. Dies funktionierte soweit ohne Probleme. Allerdings ist die Wartezeit bis die Resultate verfügbar sind relativ gross (ca. 10 bis 20 Sekunden pro Anfrage).

Der Zugriff auf Formularfelder oder Blockelemente der HTML Seite erfolgt über den ID Tag, den Namen Tag oder eine XPath Abfrage.

8.2 Anforderungsspezifikation

8.2.1 Anforderung an die Arbeit

8.2.1.1 Allgemeine Beschreibung (Overall Description)

8.2.1.1.1 Produkt Perspektive (Product Perspective)

Ziel der Bachelorarbeit ist es, Daten über die Gesamtkosten (Flug- und Hotelpreise) bei Meetings von Teilnehmenden aus verschiedenen Ländern in Abhängigkeit vom Meeting-Ort, vom Meeting-Zeitpunkt, von der Meeting-Dauer und vom Buchungszeitpunkt zu sammeln und auswerten zu können.

Anhand von Meetings der EU Action COST298 sollen typische Kostentrends und Optimierungsfaktoren aufgezeigt werden. Um detaillierte Informationen zu Flugpreisen, Flugzeiten, Hotels etc. zu erhalten, werden Anfragen an Buchungsplattformen gesendet und in einer lokalen Datenbank abgelegt.

Zur **Evaluation der Kostentrends** sollen Anfragen für eine bestimmte Teilnehmergruppe ausgelöst werden können, wobei mehrere Parameter (Meeting-Ort, Zeit zwischen Buchungszeitpunkt und Meeting, Meeting-Dauer, Meeting-Wochentage) automatisch variiert werden. Die Resultate sind in Grafiken zu visualisieren.

Zur **Bestimmung der Kosten eines bestimmten Meetings** ist eine Webanwendung für Anfragen mit wählbarer Teilnehmerzusammensetzung und Meeting-Ort für ein bestimmtes Meeting-Datum und Meeting-Ende zu erstellen. Die Resultate sollen dem Nutzer direkt via Web oder per Mail ausgeliefert werden.

8.2.1.1.2 Produkt Funktion (Product Functions)

Um die Ziele der Arbeit zu erreichen werden grundsätzlich zwei verschiedene Anwendungen benötigt.

Die erste Applikation (Kernapplikation) sendet automatisch Flug- und Hotelanfragen an eine bestimmte Internetbuchungsplattform. Die Resultate diese Anfrage werden anschliessend „geparst“ und die entsprechenden Resultate in einer Datenbank abgelegt. Diese Funktionalität dient zur allgemeinen Evaluation von Kostentrends bei der Flug- und Hotelbuchung. Daraus sollen die Parameter erkannt werden, welche sich auf die Preisentwicklung einer Buchung auswirken.

Die zweite Anwendung soll einem interessierten Meeting-Veranstalter die Möglichkeit geben, die jeweiligen finanziellen Aufwände möglicher Meeting-Destinationen und/oder Meeting-Durchführungsdaten miteinander zu vergleichen. Dabei sendet er eine oder mehrere Meeting-Anfragen über eine Webapplikation an das System. Eine Anfrage beinhaltet den jeweiligen Meeting-Ort, das Meeting-Durchführungsdatum sowie die verschiedenen Heimatorte der Teilnehmer und deren Anzahl. Für die Berechnung der Resultate wird dabei die Kernfunktionalität der Ersten Applikation verwendet.

Die Kernapplikation zur Evaluation von Flug- und Hotelpreisen soll grundsätzlich folgende Funktionalität bieten:

- Erstellen und ausführen von Flugkostenabfragen anhand eines Meetings mit Meeting-Ort, Meeting-Durchführungsdatum und Teilnehmerinformationen. Die Resultate werden in der Datenbank für die Auswertung gespeichert.
- Erstellen und ausführen von Hotelkostenabfragen anhand eines Meetings mit Meeting-Ort, Meeting-Durchführungsdatum und Teilnehmerinformationen. Die Resultate werden in der Datenbank für die Auswertung gespeichert.
- Erstellen und ausführen von kombinierten Flug- und Hotelkostenabfragen anhand eines Meetings mit Meeting-Ort, Meeting-Durchführungsdatum und Teilnehmerinformationen. Die Resultate werden in der Datenbank für die Auswertung gespeichert.

- Erstellen von „Batch“-Abfragen für bestimmte Meetings. Diese Anfragen werden automatisch und zeitgesteuert vom System ausgeführt und die Resultate in der Datenbank gespeichert. Anhand dieser systematisch Abgefragten Daten sollen die Kostentrends analysiert werden.

Die Webapplikation soll grundsätzlich die folgenden Funktionalitäten bieten:

- Registrierte Benutzer sollen Flug- oder Hotelkostenabfragen (oder auch kombinierte Anfragen) für verschiedene Meetings über ein Webinterface an das System senden können.
- Die Kosten für die Meetings werden von der Kernapplikation abgefragt und die berechneten Resultate werden dem Benutzer zur Verfügung gestellt. Die Resultatdaten werden in der Datenbank gespeichert.

8.2.1.1.3 Benutzer Charakteristik (User Characteristics)

Die eine Zielgruppe dieser Anwendung sind Personen, die sich für die Kosten bei Meetings mit Teilnehmenden aus verschiedenen Ländern und den Einfluss der Wahl des Meeting-Ortes, der Meeting-Dauer, des Meeting-Zeitpunktes etc. interessieren. Also für Leute, welche sich für die Auswirkungen von verschiedenen Parameter auf die Preisentwicklung von Flug- und Hotelpreisen interessieren.

Die andere Zielgruppe sind Verantwortliche, welche Meetings mit internationalen Teilnehmern planen wollen. Diese können mit Hilfe der Webapplikation die verschiedenen finanziellen Aufwände für verschiedene Meeting-Möglichkeiten vergleichen.

8.2.1.1.4 Einschränkungen (Constraints)

Es sollen keine direkten Buchungen von Hotels oder Flügen über die Applikation selbst durchgeführt werden könne.

Um die Kosten zu berechnen, werden Informationen aus Fluginformationssystemen und/oder Buchungsplattformen benötigt. Es wird davon ausgegangen, dass die erhaltenen Daten aktuell und korrekt sind.

8.2.1.1.5 Abhängigkeiten (Dependencies)

Um die Software realisieren zu können, sind wir stark auf den Zugang zu den Flug- und Hoteldaten angewiesen. Die Daten für unsere Anwendung werden aus verschiedenen Internetbuchungsplattformen zusammengesammelt.

8.2.1.2 Spezifische Anforderungen (Specific Requirements)

8.2.1.2.1 Funktionale Anforderungen zur Webanwendung

8.2.1.2.1.1 Benutzerverwaltung

Benutzer der Applikation sollen sich mit einem Benutzernamen und einem Passwort authentisieren.

Ein registrierter und authentifizierter Benutzer kann folgende Funktionalität nutzen:

- Kann Anfragen zur Berechnung der Flugkosten erstellen und durchführen
- Kann Anfragen zur Berechnung der Hotelkosten erstellen und durchführen
- Kann kombinierte Anfragen zur Berechnung der Hotel- und er Flugkosten erstellen und durchführen
- Erhält Zugriff auf die Resultate seiner Abfragen

8.2.1.2.1.2 Flugkostenabfrage erstellen

Ein Benutzer kann die Flugkosten anhand von einer variablen Anzahl von Teilnehmerherkunftsorten und der Anzahl der Teilnehmer dieses Herkunftsortes, bis zu drei verschiedenen Meeting-Destinationen und des Meeting-Durchführungsdatum berechnen lassen. Die Resultate der Abfrage werden für den Benutzer in der Datenbank gespeichert.

8.2.1.2.1.3 **Hotelkostenabfrage erstellen**

Ein Benutzer kann die Hotelkosten anhand von einer variablen Anzahl von Teilnehmerherkunftsorten und der Anzahl der Teilnehmer dieses Herkunftsortes, der Meeting-Destination und des Meeting-Durchführungsdatum berechnen lassen. Die Resultate der Abfrage werden für den Benutzer in der Datenbank gespeichert.

8.2.1.2.1.4 **Kombinierte Hotel- und Flugkostenabfrage erstellen**

Ein Benutzer kann die Hotel und die Flugkosten anhand von Teilnehmerherkunftsorten und der Anzahl der Teilnehmer dieses Herkunftsortes, bis zu drei verschiedenen Meeting-Destinationen und des Meeting-Durchführungsdatum berechnen lassen. Die Resultate der Abfrage werden für den Benutzer in der Datenbank gespeichert.

8.2.1.2.1.5 **Resultate der Flug-, Hotelkostenabfrage oder einer kombinierten Abfrage anzeigen**

Sobald eine Benutzeranfrage bearbeitet wurde und die Resultate zur Verfügung stehen, sich bei der Benutzer bei der Webapplikation anmelden und dort die Resultate betrachten.

Die **Resultate der Flugkostenabfrage** enthalten folgende Informationen:

- Gesamtflykosten für alle Teilnehmer eines Meetings (dabei wird jeweils die günstigste Flugreise eines Teilnehmerherkunftsortes zum Meeting und zurück mit der Anzahl der Teilnehmer dieses Ortes multipliziert)
- Meeting-Durchführungsdatum
- Meeting-Destination
- Auflistung der Anzahl Teilnehmer und Ihrer Herkunftsorte
- Informationen (Details zu den Flügen) zu allen verschiedenen Flugreisen der Teilnehmer und der Anzahl der Teilnehmer die diese Reise benötigen

Die **Resultate der Hotelkostenabfrage** enthalten folgende Informationen:

- Hotelkosten (Für gefundene Hotels der Komfortklassen ***,****,*****, werden jeweils folgende Informationen ausgegeben)
 - o Anzahl der gefunden Hotels
 - o Tiefster Preis pro Nacht
 - o Höchster Preis pro Nacht
 - o Durchschnittlicher Preis pro Nacht
 - o Tiefste Gesamtkosten pro Teilnehmer für die Meeting-Dauer
 - o Höchste Gesamtkosten pro Teilnehmer für die Meeting-Dauer
 - o Durchschnittliche Gesamtkosten pro Teilnehmer für die Meeting-Dauer
 - o Tiefste Gesamtkosten für alle Teilnehmer für die Meeting-Dauer
 - o Höchste Gesamtkosten für alle Teilnehmer für die Meeting-Dauer
 - o Durchschnittliche Gesamtkosten für alle Teilnehmer für die Meeting-Dauer
- Meeting-Durchführungsdatum
- Meeting-Destination
- Auflistung der Anzahl Teilnehmer

Die **Resultate der kombinierten Abfrage** enthalten folgende Informationen:

- Alle Resultate der Flugkostenanfrage
- Alle Resultate der Hotelkostenabfrage
- Tiefste Gesamtkosten (Flüge und günstigstes Hotel pro Komfortklasse)
- Höchste Gesamtkosten (Flüge und teuerstes Hotel pro Komfortklasse)
- Durchschnittliche Gesamtkosten (Flüge und durchschnittliche Hotelkosten pro Komfortklasse)

8.2.1.2.2 **Funktionale Anforderungen zur Kernapplikation**

8.2.1.2.2.1 **Flugkostenabfrage ausführen**

Für ein bestimmtes Meeting (mit Teilnehmerheimatorten und der jeweiligen Anzahl der Teilnehmer dieses Ortes, Meeting-Destination, Meeting-Durchführungsdatum) soll eine Abfrage an eine oder an

mehrere verschiedene Buchungsplattformen gesendet werden. Es soll möglich sein, nur Direktflüge zu beachten. Die benötigten Daten werden aus der Antwortseite geparkt und in die Datenbank gespeichert. Es werden alle in Frage kommenden Resultate gespeichert. D.h. alle Flugresultate welche die Buchungsplattform liefert, werden auch in die Datenbank gespeichert.

Pro Flugreise werden die folgenden Daten „herausgeparkt“ und persistiert:

- Gesamtpreis
- Für alle Flüge der Hin- und der Rückreise
 - o Airline Name
 - o Flug Nr.
 - o Abflugs-Flughafen
 - o Ankunfts-Flughafen
 - o Abflugs-Destination
 - o Ankunfts-Destination
 - o Abflugs-Daten und Zeiten
 - o Ankunft-Daten und Zeiten

8.2.1.2.2.2 **Hotelkostenabfrage ausführen**

Für eine bestimmtes Meeting (mit Anzahl Teilnehmer, Meeting-Destination, Meeting-Durchführungsdatum) soll eine Abfrage an eine oder an mehrere verschiedene Buchungsplattformen gesendet werden. Die benötigten Daten werden aus der Antwortseite geparkt und in die Datenbank gespeichert. Es werden alle in Frage kommenden Resultate gespeichert. D.h. alle Hotelresultate welche die Buchungsplattform liefert, werden auch in die Datenbank gespeichert.

Pro Hotel werden die folgenden Daten „herausgeparkt“ und persistiert:

- Gesamtpreis
- Ort
- Anzahl Sterne
- Distanz zum Stadtzentrum (falls vorhanden)

8.2.1.2.2.3 **Kombinierte Hotel- und Flugkostenabfrage ausführen**

Für eine bestimmtes Meeting (mit Teilnehmerheimatorten und der jeweiligen Anzahl der Teilnehmer dieses Ortes, Meeting-Destination, Meeting-Durchführungsdatum) soll jeweils eine Hotelkosten- sowie auch eine Flugkostenabfrage ausgeführt und die Resultate in der Datenbank gespeichert werden.

8.2.1.2.3 **Prioritäten**

Funktionale Anforderung Kernapplikation	Priorität
Flugkostenabfrage	1
Hotelkostenabfrage	1
Kombinierte Abfrage	1

Funktionale Anforderung Webapplikation	Priorität
Benutzerverwaltung	2
Manuelle Flugkostenabfrage	2
Manuelle Hotelkostenabfrage	2
Kombinierte manuelle Abfrage	3
Resultate der Abfrage per Mail versenden	3
Resultat der Abfrage in Webapplikation	2

Funktionen mit Prioritäten 1 – 2 sollten am Projektende implementiert und funktionstüchtig sein. Funktionen mit tieferen Prioritäten sind als optionale Features geplant, welche je nach Zeitreserven implementiert werden können. (Priorität 1 = muss, 2 = soll, 3 = „nice to have“)

8.2.1.2.4 Bedienbarkeit (Usability)

8.2.1.2.5 Zuverlässigkeit (Reliability)

Für das Berechnen der Meeting-Kosten ist die Applikation auf die Daten der externen Informationssysteme angewiesen. Für Fehler aufgrund von Unzuverlässigkeit dieser externen Dienste ist die Applikation nicht verantwortlich.

Falls die gewünschte Internetbuchungsplattform nicht erreichbar ist, wird dies dem Benutzer mit einer Fehlermeldung angezeigt.

Falls die Buchungsplattform wegen technischer Probleme zurzeit keine Resultate liefern kann, wird die Abfrage wiederholt. Nach dem dritten erfolglosen Versuch wird die Anfrage beendet und mit der nächsten fortgefahren.

8.2.1.2.6 Leistung (Performance)

Der Flaschenhals bei der Abfrage der Flug und Hoteldaten befindet sich bei den jeweiligen Buchungsplattformen. Bis die Resultate für eine Fluganfrage geliefert werden dauert es bei ebookers.de z.B. typischerweise etwa 20 Sekunden. Für die Resultate der Hotelabfrage braucht ebookers.de um die 10 Sekunden. D. h., dass bei einer Fluganfrage für ein Meeting mit Teilnehmern aus 10 verschiedenen Herkunftsorten und den dazu nötigen 10 Abfragen bereits $10 \cdot 20s = 200s$ (3min 20s) gebraucht werden, nur um die Resultate zu erhalten. Selbstverständlich nimmt das Parsen der Resultate und das Schreiben in die Datenbank auch noch zusätzliche Zeit in Anspruch, doch diese Zeit ist im Gegensatz zur Abfragewartezeit zu vernachlässigen.

Um die Bearbeitungszeit der Flugkostenabfragen für ein Meeting zu verkürzen, sollen die verschiedenen Abfragen parallel gestartet und geparkt werden. Das Schreiben der Resultate in die Datenbank kann danach allerdings wieder seriell implementiert werden.

8.2.1.2.7 Wartbarkeit (Supportability)

8.2.1.2.7.1 Erweiterbarkeit

Das Software-Design muss künftigen Erweiterungen offen stehen, ohne die Frage nach einem Redesign aufzuwerfen. Vor allem soll es möglich sein, problemlos neue Parser-Klassen weiterer online Buchungsplattformen hinzuzufügen.

8.2.1.2.7.2 Analysierbarkeit

Das Softwaredesign muss nach erlernten Software Engineering-Methoden aufgebaut sein. Damit sind die vermittelten Modelle und Verfahren der Module SE1-2 der HSR gemeint. Auf Kommentare im Programmcode soll möglichst verzichtet werden. Es sollen in erster Linie aussagekräftige Verkabeln-, Klassen- und Funktionsbezeichnungen gewählt werden.

8.2.1.2.7.3 Modifizierbarkeit

Das Design muss modular und generisch aufgebaut sein. Dies vereinfacht nachträgliche Erweiterungen oder Anpassungen am Programmcode. Somit muss eine funktionale Gliederung (Packages) und klar definierte Schnittstellen realisiert werden.

8.2.1.2.7.4 Prüfbarkeit

Die Webapplikation wird mittels manuellen Benutzertests auf korrekte Funktionalität geprüft. Dabei werden alle verfügbaren Funktionalitäten durch einen Benutzer durchgeführt und getestet.

8.2.1.2.8 Schnittstellen (Interfaces)

8.2.1.2.8.1 *Benutzerschnittstelle (User Interface)*

Die Kernapplikation besitzt kein User Interface in dem Sinne. Sie nimmt auch nicht direkt User Inputs entgegen.

Der Benutzer erstellt seine Anfragen und erhält die entsprechenden Resultate über die Webapplikation. Beim Erstellen des User Interfaces wird auf das Einhalten von bekannten User Interface Design Standards wie sie in den User Interface Modulen 1 und 2 an der HSR vermittelt werden geachtet. So soll der Anwender die Applikation möglichst intuitiv bedienen können und ohne grosse Einarbeitungszeit mit der Applikation arbeiten können.

8.2.1.2.8.2 *Datenbankschnittstelle (Database Interface)*

Die zu persistierenden Daten werden mittels LINQ to SQL in einer MS SQL gespeichert. Dank der Wahl einer MS SQL Datenbank stehen verschiedene Möglichkeiten für die Analyse und das Reporting der Daten zur Verfügung (Analysis Services, Reporting Services).

8.2.1.2.8.3 *Kommunikationsschnittstelle (Communication Interface)*

Um an die benötigten Informationen zur Berechnung der Kosten zu kommen, werden Zugriffe auf Buchungsplattformen benötigt. D.h. um die Abfragen durchzuführen muss eine Verbindung zum Internet bestehen.

8.2.2 Use Cases (Success Scenario / Success Diagram)

Diese Use Cases beziehen sich auf die Webapplikation.

Use Case	Beschreibung	Priorität
UC1: Benutzer anmelden	Ein Benutzer kann sich mit seinem Benutzernamen und Passwort bei der Applikation anmelden und authentisieren.	2
UC2: Benutzer abmelden	Ein Benutzer kann sich von der Applikation abmelden.	2
UC3: Teilnehmer Orte verwalten	Der Benutzer kann neue Teilnehmer Orte erfassen, bestehende bearbeiten oder löschen/deaktivieren.	3
UC4: Flugkosten Abfragen	Der Benutzer kann eine Flugkostenabfrage erstellen. Diese besteht aus den verschiedenen Teilnehmern, einer gewählten Destination und dem Hinflug- sowie dem Rückflugzeitpunkt. Die Resultate der Abfrage werden in der Datenbank gespeichert.	1
UC5: Hotelkosten Abfragen	Der Benutzer kann eine Hotelkostenabfrage erstellen. Diese besteht aus den verschiedenen Teilnehmern, einer gewählten Destination und dem Anreise- sowie dem Rückreisezeitpunkt. Die Resultate der Abfrage werden in der Datenbank gespeichert.	1
UC6: Kombinierte Kosten Abfragen		1
UC7: Flugkosten Resultate anzeigen	Der Benutzer kann eine statistische Auswertung der Flugpreise erzeugen.	2
UC8: Hotelkosten Resultate anzeigen	Der Benutzer kann eine statistische Auswertung der Hotelpreise erzeugen.	2
UC9: Kombinierte Resultate anzeigen	Der Benutzer kann eine statistische Auswertung der Flug- und Hotelpreise erzeugen.	2

Funktionen mit Prioritäten 1 – 2 sollten am Projektende implementiert und funktionstüchtig sein. Funktionen mit tieferen Prioritäten sind als optionale Features geplant, welche je nach Zeitreserven implementiert werden können. (Priorität 1 = muss, 2 = soll, 3 = „nice to have“)

8.2.3 System-Sequenzdiagramme

8.2.3.1 Batchmode-Applikation und Kernapplikation

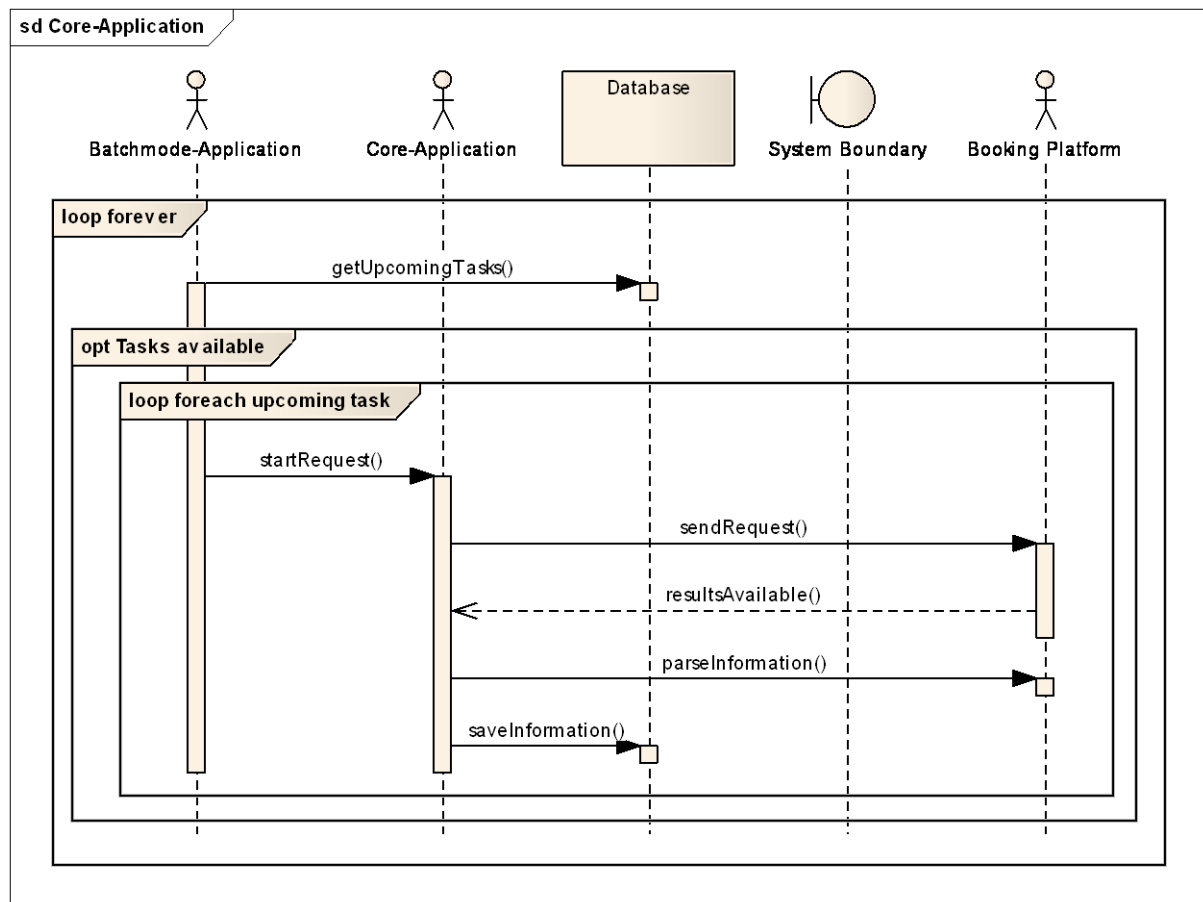


Abbildung 33: Systemsequenz Diagramm Kernapplikation

In Abbildung 33 ist das Systemsequenz Diagramm der Kernapplikation und der Batchmodeapplikation dargestellt. Daraus ist der grobe Ablauf ersichtlich.

Die Batchmode-Applikation überprüft regelmässig ob sich in der Datenbank Abfrage-Aufträge befinden, welche zum jetzigen Zeitpunkt fällig sind.

Sobald einer oder mehrere Abfrage-Aufträge gefunden wurden, kommt die Kernapplikation ins Spiel. Die Batchmode-Applikation übergibt nun die gefundenen Abfrage-Aufträge an die Kernapplikation. Diese sendet die Anfrage an das entsprechende online Buchungsportal. Sobald die Resultate auf der Buchungsplattform vorliegen werden die gebrauchten Informationen aus der Seite herausgelesen und in der Datenbank persistiert.

8.2.3.2 Webapplikation

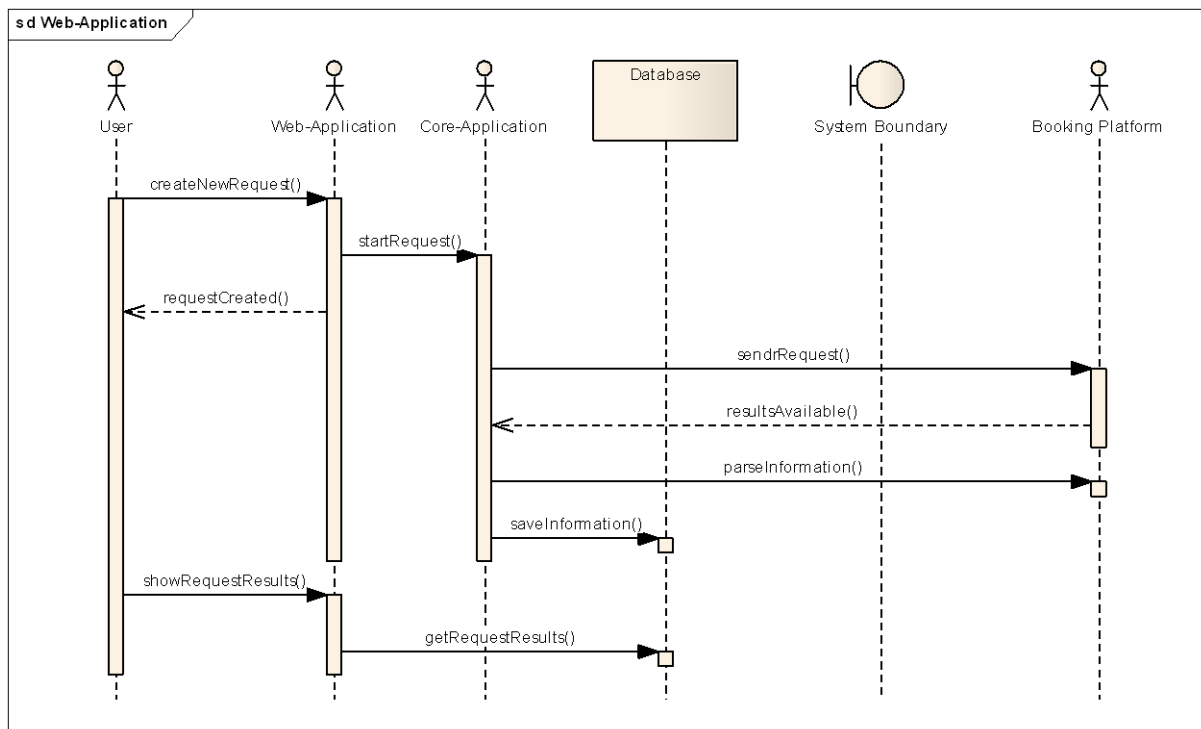


Abbildung 34: Systemsequenz Diagramm Webapplikation

Abbildung 34 zeigt den groben Ablauf einer Benutzeranfrage über die Webapplikation.

Der Benutzer erstellt eine Abfrage über die Webapplikation. Diese leitet die Anfrage an die Kernapplikation weiter, welche dann die online Buchungsplattform aufruft und die benötigten Daten aus den Resultaten herausliest und in der Datenbank ablegt.

Wenn der Benutzer Resultate bereits ausgeführter Abfragen ansehen möchte, holt die Webapplikation die gewünschten Daten aus der Datenbank, bereitet diese auf und zeigt sie dem Benutzer an.

8.2.4 Weitere Funktionen

Um die zeitgesteuerten Abfragen für die automatische Flug- und Hoteldatensammlung, welche zur Analyse der Daten dienen, auszuführen wird eine Batchmode-Verarbeitung benötigt.

Dabei werden auszuführende Abfragen mit ihrem Ausführungszeitpunkt definiert und in eine Datenbank gespeichert. Eine kleine Batchmode-Applikation prüft regelmässig ob sich auszuführende Abfragen in der Datenbank befinden und startet diese falls vorhanden. Dabei werden die Abfragen an die Kernapplikation übergeben, welche die eigentliche Ausführung übernimmt

8.3 Analyse (Business Modell)

8.3.1 Domain Modell

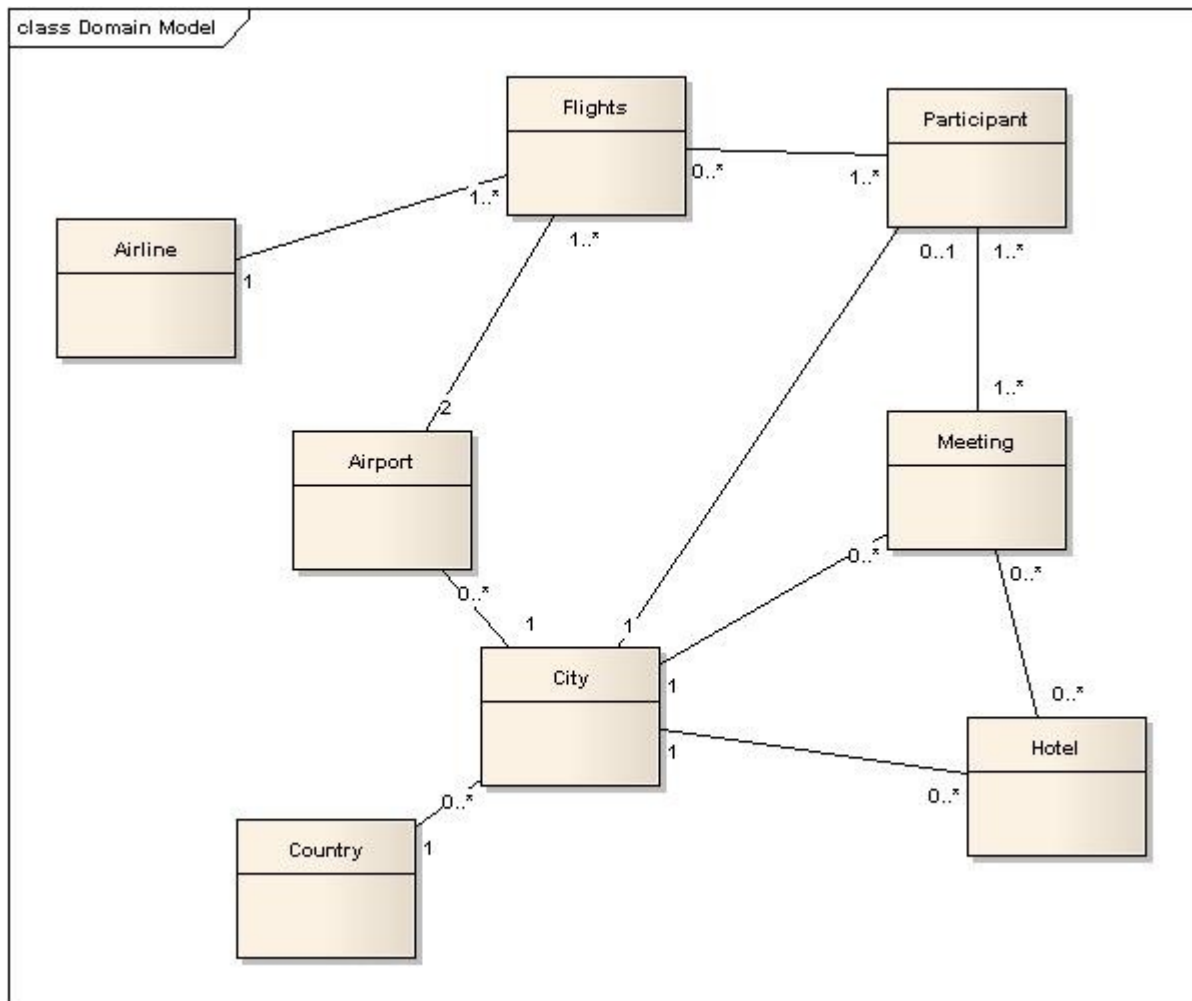


Abbildung 35: Domainmodel

8.3.2 Objektkatalog

Hier werden die verschiedenen Klassen des Domainmodels beschrieben.

8.3.2.1 Airline

Airline die einen Flug durchführt.

8.3.2.2 Airport

Flughäfen von denen die Teilnehmer starten oder landen. Ein Flughafen gehört immer zu einer Stadt.

8.3.2.3 City

Stadt in der ein Meeting stattfindet oder von der die Teilnehmer kommen.

8.3.2.4 Country

Eine Stadt gehört immer zu einem Land.

8.3.2.5 Flights

Flüge haben einen Abflug- sowie einen Ankunftsflughafen.

8.3.2.6 Hotel

Ein Hotel befindet sich in einer bestimmten Stadt.

8.3.2.7 Meeting

Ein Meeting findet in einer bestimmten Stadt statt und hat eine Anzahl von Teilnehmern.

8.3.2.8 Participant

Meeting Teilnehmer benötigen Flüge um an den Meeting-Ort zu gelangen.

8.4 Design (Entwurf)

8.4.1 Architektur

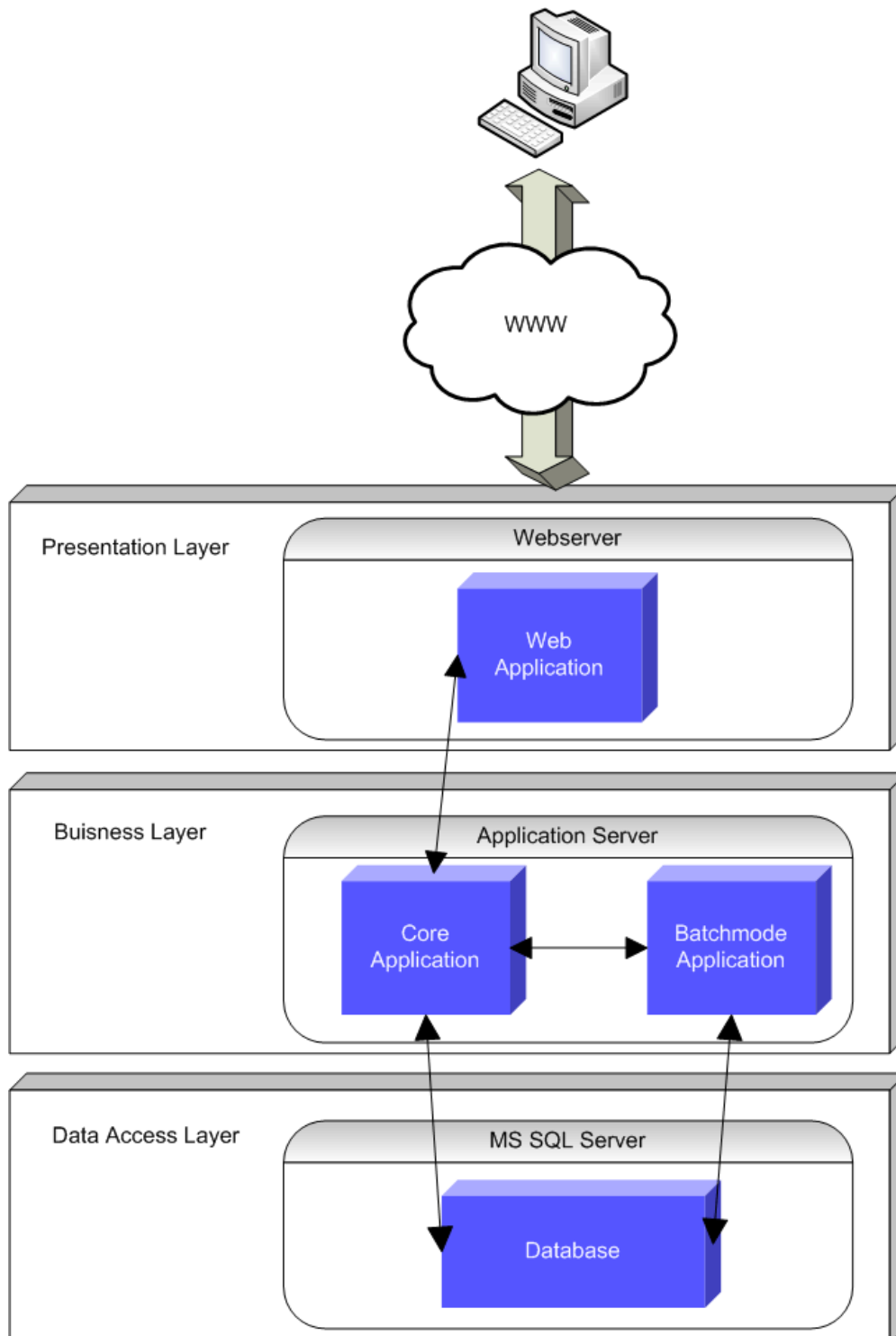


Abbildung 36: Übersicht der logischen Architektur

8.4.2 Package- und Klassendiagramme (konzeptionell)

8.4.2.1 Presentation Layer

Als Benutzerschnittstelle (GUI) dient die Webapplikation. Sie nimmt die Benutzerinputs entgegen und leitet diese an die Kernapplikation im Business Layer weiter, welche die Benutzerbefehle bearbeitet und die Resultate in der Datenbank speichert.

Da die Webapplikation und das GUI mit ASP.NET erstellt werden, wird hier auf ein Klassendiagramm verzichtet. Dies deshalb, da die #C Klassen hauptsächlich im Hintergrund automatisch durch die .NET Runtime generiert werden.

Die Webapplikation soll dem User folgende Möglichkeiten bieten:

- Registrieren eines neuen Benutzers um auf die Funktionalität der Applikation zugreifen zu können
- Login anhand Benutzername und Passwort
- Logout
- Erstellen einer neuen Meeting-Abfrage mit, Meeting-Durchführungsdatum, Meeting-Destination und einer beliebigen Anzahl Teilnehmer aus beliebig vielen Herkunftsorten.
- Übersicht der bereits durchgeführten Abfragen anzeigen
- Resultate der bereits ausgeführten Abfragen anzeigen

Für jede der obengenannten Funktionalitäten wird je eine ASPX-Seite verwendet.

8.4.2.2 Business Layer

Der Business Layer enthält die eigentliche Funktionalität der gesamten Anwendung. Die in diesem Layer angesiedelten Klassen und Objekte kann man grob gesagt in zwei Gruppen unterteilen.

Erstens die Problem Domain Klassen, welche hauptsächlich nur Funktionalität beinhalten. Sie steuern den Ablauf, führen Berechnungen durch, sammeln Daten usw.

Die zweite Gruppe sind die Daten Objekte, welche mehr oder weniger nur für die Datenhaltung gebraucht werden. Die Meisten werden dazu verwendet, Informationen während der Laufzeit zu halten, die anschliessend in der Datenbank persistiert werden.

8.4.2.2.1 Problem Domain

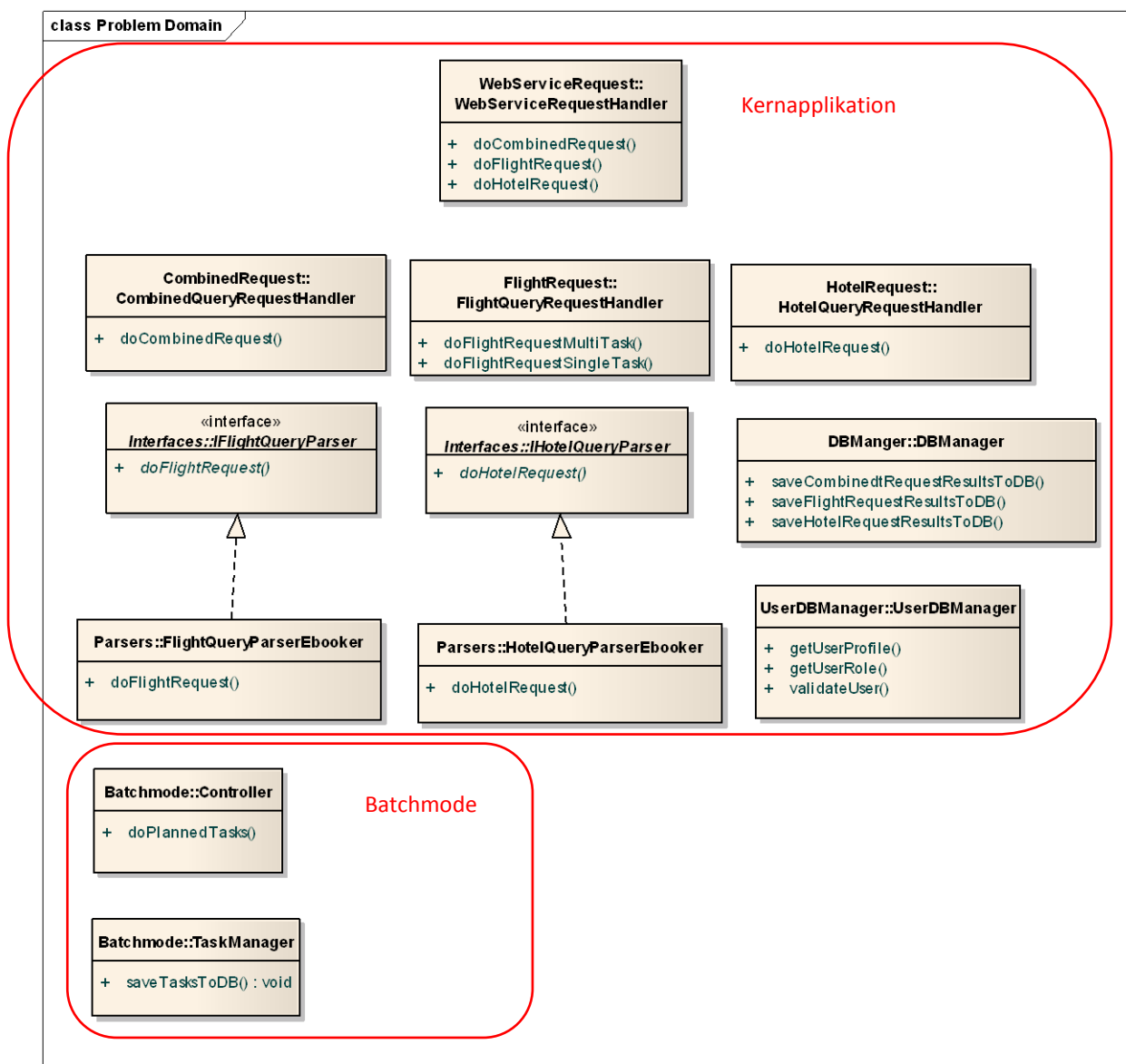


Abbildung 37: Konzeptionelles Klassendiagramm Problem Domain

In Abbildung 37 sind die Klassen der Problem Domain dargestellt. Es handelt sich hierbei um eine konzeptionelle Ansicht der Klassen, d.h. ohne spezifische Details. Die einzelnen Klassen und deren Funktion werden hier kurz beschrieben.

8.4.2.2.1.1 **WebServiceRequestHandler**

Der WebServiceRequestHandler nimmt die Benutzer Requests aus der Webapplikation entgegen. Je nach dem um was es sich für einen Request handelt (Flug-, Hotel- oder kombinierte Abfrage) leitet er die Anfrage an den dafür verantwortlichen RequestHandler weiter. Dieser ist dann für den weiteren Ablauf verantwortlich.

8.4.2.2.1.2 **FlightQueryRequestHandler**

Der FlightQueryRequestHandler nimmt Flugafragen entgegen und ist verantwortlich für die korrekte Ausführung des Requests.

Anhand des BookingPlatform Objekts im Request, welches den zu benutzenden Parser enthält, wird der Parser gestartet. Der Parser sammelt die Informationen von der entsprechenden Buchungsplattform zusammen und liefert die Resultate zurück an den Handler.

Beim Ausführen des Parsings gibt es beim FlightRequest zwei verschiedene Varianten. Single-Task und Multi-Task sind verfügbar. Beim Single-Task werden die Anfragen an die Buchungsplattform sequentiell abgearbeitet. D.h. jede nötige Abfrage für jeden Teilnehmer mit einem anderen Herkunftsort wird erst gestartet, wenn die Resultate der vorherigen ausgelesen worden sind.

Beim Multi-Task Modus, werden die Aufrufe an das Buchungsportal und das Auslesen der Ergebnisse parallel ausgeführt. D.h. alle benötigten Abfragen für alle Teilnehmer werden gleichzeitig gestartet. Dadurch kann die Wartezeit bis das Resultat einer einzelnen Anfrage verfügbar ist besser genutzt werden und die Applikation ist nicht blockiert bis ein einzelnes Ergebnis bereit ist.

Sobald alle Resultate an den Handler zurückgegeben wurden, sorgt dieser dafür, dass die Ergebnisse in der Datenbank gespeichert werden.

8.4.2.2.1.3 **HotelQueryRequestHandler**

Der HotelQueryRequestHandler nimmt Hotelanfragen entgegen und ist für deren korrekte Durchführung verantwortlich.

Anhand des BookingPlatform Objekts im Request, welches den zu benutzenden Parser enthält, wird der Parser gestartet. Der Parser sammelt die Informationen von der entsprechenden Buchungsplattform zusammen und liefert die Resultate zurück an den Handler.

Sobald alle Resultate an den Handler zurückgegeben wurden, sorgt dieser dafür, dass die Ergebnisse in der Datenbank gespeichert werden.

8.4.2.2.1.4 **CombinedQueryRequestHandler**

Der CombinedQueryRequestHandler nimmt kombinierte Anfragen (Flug und Hotel) entgegen und ist für deren korrekte Ausführung verantwortlich.

Die Ausführung der jeweiligen Flight- und HotelRequests delegiert er dabei an den FlightQueryRequestHandler bzw. HotelQueryRequestHandler.

Sobald der Handler die Resultate von den jeweiligen Handler zurückerhalten hat, werden diese in der Datenbank gespeichert.

8.4.2.2.1.5 **IFlightQueryParser**

Das Interface IFlightQueryParser sollte von allen Parsern implementiert werden, welche für das beschaffen von Flugdaten aus den Abfragen auf einem Buchungsportal benutzt werden.

8.4.2.2.1.6 **FlightQueryParserEbooker**

Die FlightQueryParserEbooker Klasse enthält die Funktionalität um eine Flugdatenabfrage auf ebookers.de durchzuführen. Sie sendet die entsprechende Anfrage (Anreisedatum, Rückreisedatum, Abreiseort, Zielort) an das Buchungsportal und liest die Resultate aus der Antwortseite aus. Die FlightQueryParserEbooker Klasse implementiert das Interface IFlightQueryParser.

8.4.2.2.1.7 **IHotelQueryParser**

Das Interface IHotelQueryParser sollte von allen Parseern implementiert werden, welche für das beschaffen von Hoteldaten aus den Abfragen auf einem Buchungsportal benutzt werden.

8.4.2.2.1.8 **HotelQueryParserEbooker**

Die HotelQueryParserEbooker Klasse enthält die Funktionalität um eine Hoteldatenabfrage auf ebookers.de durchzuführen. Sie sendet die entsprechende Anfrage (Anreisedatum, Rückreisedatum, Abreiseort, Zielort) an das Buchungsportal und liest die Resultate aus der Antwortseite aus. Die HotelQueryParserEbooker Klasse implementiert das Interface IHotelQueryParser.

8.4.2.2.1.9 **DBManger**

Der DBManager ist für die Zugriffe auf die Datenbank verantwortlich. Er erhält die Resultate einer Abfrage von einem Handler (Combined-, Flight- oder Hotel-Handler) und speichert diese in der Datenbank.

8.4.2.2.1.10 **UserDBManager**

Der UserDBManager ist hauptsächlich für das Erstellen und Überprüfen von Benutzern zuständig. Er stellt Methoden für die Erstellung eines neuen Benutzers in der Datenbank sowie deren Validierung (Benutzername und Passwort bei der Anmeldung) bereit. Allgemein formuliert kapselt er sämtliche Zugriffe auf die Datenbank die mit der Benutzerverwaltung einhergehen.

8.4.2.2.1.11 **Controller**

Die Controller Klasse wird für die Batchmode-Applikation verwendet. Der Controller überprüft dabei die in der Task-Datenbank gespeicherten Anfragen auf ihre Fälligkeit und startet diese falls der Ausführungszeitpunkt erreicht wird. Der die auszuführende Abfrage wird dabei vom Controller an den dafür zuständigen Kontroller delegiert der für die korrekte Ausführung verantwortlich ist.

8.4.2.2.1.12 **TaskManager**

Der TaskManager bietet Methoden an, um Abfragen die automatisch ausgeführt werden sollen in der Task-Datenbank zu speichern.

8.4.2.2.2 Data Objects

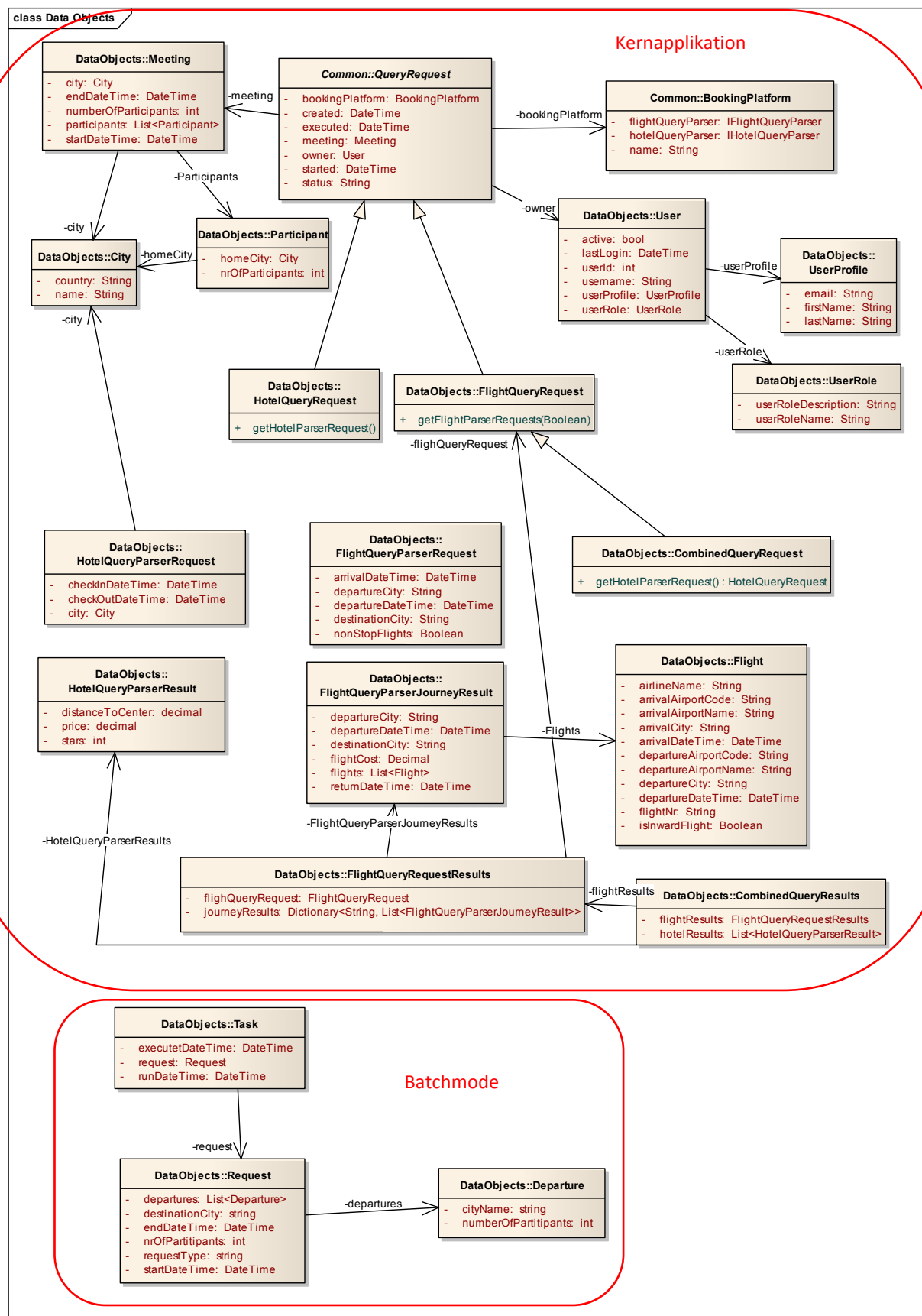


Abbildung 38: Konzeptionelles Klassendiagramm Data Objects

In Abbildung 38 Abbildung 37 sind die Data Objects Klassen dargestellt. Es handelt sich hierbei um eine konzeptionelle Ansicht der Klassen, d.h. ohne spezifische Details. Die einzelnen Klassen und deren Funktion werden hier kurz beschrieben.

8.4.2.2.2.1 **Task**

Die Task Klasse enthält die Informationen zu einem automatisch auszuführenden Task durch die Batchmode-Applikation. Sie enthält den gewünschten Ausführungszeitpunkt sowie den Zeitpunkt an welchem die Abfrage schlussendlich abgeschlossen wurde.

8.4.2.2.2.2 **Departure**

Die Departure Klasse enthält die Informationen zu den Teilnehmern eines Meetings für eine geplante Abfrage durch die Batchmode-Applikation.

8.4.2.2.2.3 **Request**

Die Request Klasse enthält die Informationen über den Durchführungsort sowie den Durchführungszeitpunkt des Meetings, für welches eine automatische Abfrage durch die Batchmode-Applikation durchgeführt werden soll.

8.4.2.2.2.4 **Meeting**

Die Meeting Klasse enthält die Informationen zum Durchführungsort und Durchführungsdatum des Treffens sowie Angaben zu den teilnehmenden Personen.

8.4.2.2.2.5 **City**

Die City Klasse enthält Informationen zu einem Ort. Wichtigste Eigenschaft ist der City-Name.

8.4.2.2.2.6 **Participant**

Die Participant Klasse abstrahiert einen Meeting Teilnehmer Ort. D.h. sie enthält Informationen über den Abreiseort der Teilnehmer und die Anzahl Teilnehmer aus diesem Ort.

8.4.2.2.2.7 **User**

Das User Objekt enthält Informationen über den Anwendungsbenutzer. Jeder User hat eine Rolle und ein User Profil.

8.4.2.2.2.8 **UserRole**

Die User Role Klasse besitzt Informationen über die zugewiesene Benutzerrolle eines Benutzers. Anhand dieser Rolle können verschiedene Berechtigungen für die jeweiligen Rollen implementiert werden.

8.4.2.2.2.9 **UserProfile**

Das UserProfile Objekt enthält erweiterte Informationen eines bestimmten Benutzers.

8.4.2.2.2.10 **BookingPlatform**

Das BookingPlatform Objekt enthält Informationen über die, bei der Abfrage zu benutzenden Parser (Flug- und Hotelparser). Anhand der enthaltenen Parser startet der jeweilige QueryRequestHandler den Parse-Vorgang.

Dank der Verwendung eines Interfaces können die hier verschiedene implantierte Parser angegeben werden, die sich in Ihrem spezifischen Verhalten unterscheiden. D.h. für verschiedene Buchungsplattformen auf denen Abfragen durchgeführt werden sollen, werden unterschiedliche Parser mit differentem Verhalten benötigt.

8.4.2.2.2.11 **QueryRequest**

Die QueryRequest klasse ist eine abstrakte Klasse, welche die gemeinsamen Eigenschafften aller möglichen Arten von Requests (Flug-, Hotel- oder Kombierter Request) sammelt. Die entsprechenden spezialisierten QueryRequest Klassen erben von dieser Mutterklasse.

Die Klasse enthält ein BookingPlatform Objekt mit den zu verwendenden Parser für die Anfrage. Im Weiteren besitzt sie ein paar Timestamp-Informationen (Erstellungszeitpunkt, Zeitpunkt des Starts der Ausführung, Zeitpunkt der Beendigung der Ausführung(Resultate verfügbar)). Zusätzlich besitzt die Klasse ein Meeting Objekt, welches alle Informationen die auszuführende Abfrage enthält (Anreisedatum, Rückreisedatum, Durchführungsort und Teilnehmerinformationen). Im Status Feld ist der aktuelle Status des Requests in Textform vermerkt. Jedes QueryRequest Objekt enthält auch ein User Objekt, welches die Informationen zum Benutzer, der die Anfrage erstellt hat, beinhaltet.

8.4.2.2.2.12 **FlightQueryRequest**

Das FlightQueryRequest Objekt beinhaltet die benötigten Informationen für eine Flugdatenabfrage. Diese Klasse spezialisiert die abstrakte QueryRequest Klasse.

Zusätzlich zu den geerbten Eigenschaft und Methoden besitzt die Klasse eine Methode die aus den Request Informationen neben dem Flug-Parser-Request auch einen Hotel-Parser-Request generiert. Dieser werden von den Parser Klassen verwendet und beinhalten nur noch die wirklich benötigten Informationen um die Abfragen auf dem entsprechenden Buchungsportal durchzuführen.

8.4.2.2.2.13 **CombinedQueryRequest**

Das HotelQueryRequest Objekt beinhaltet die benötigten Informationen für eine Hoteldatenabfrage. Diese Klasse spezialisiert FlightQueryRequest Klasse.

Zusätzlich zu den geerbten Eigenschaft besitzt die Klasse eine Methode die aus den Request Informationen einen Hotel-Parser-Request generiert. Dieser wird von einer Parser Klasse verwendet und beinhaltet nur noch die wirklich benötigten Informationen um die Abfrage auf dem entsprechenden Buchungsportal durchzuführen.

8.4.2.2.2.14 **HotelQueryRequest**

Das HotelQueryRequest Objekt beinhaltet die benötigten Informationen für eine Hoteldatenabfrage. Diese Klasse spezialisiert die abstrakte QueryRequest Klasse.

Zusätzlich zu den geerbten Eigenschaft besitzt die Klasse eine Methode die aus den Request Informationen einen Hotel-Parser-Request generiert. Dieser wird von einer Parser Klasse verwendet und beinhaltet nur noch die wirklich benötigten Informationen um die Abfrage auf dem entsprechenden Buchungsportal durchzuführen.

8.4.2.2.2.15 **HotelQueryParserRequest**

Das HotelQueryParserRequest Objekt beinhaltet die benötigten Informationen für die Parser-Klasse um eine spezifische Hotelabfrage durchzuführen. Dies sind Anreisedatum, Abreisedatum und die Destination.

8.4.2.2.2.16 **FlightQueryParserRequest**

Das FlightQueryParserRequest Objekt beinhaltet die benötigten Informationen für die Parser-Klasse um eine spezifische Flugdatenabfrage durchzuführen. Dies sind Anreisedatum, Abreisedatum, die Destination, Abflugort und die Eigenschaft ob nur Direktflüge oder alle verfügbaren Flüge gesucht werden sollen.

8.4.2.2.2.17 **HotelQueryParserResult**

Das HotelQueryParserResult Objekt hält die Resultat Daten eines Hotels, welches bei einer Anfrage gefunden wurde. Es beinhaltet den Hotelpreis pro Person für die abgefragte Aufenthaltsdauer, die Komfortklasse sowie die Lage des Hotels.

8.4.2.2.2.18 **FlightQueryJourneyParserResult**

Das FlightQueryJourneyParserResult Objekt hält die Resultat Daten einer Flugdatenabfrage. Es stellt dabei eine spezifische gefundene Flugreise dar. Zu dieser wird der Preis vermerkt sowie eine Liste der einzelnen Flüge mit den genauen Informationen dazu.

8.4.2.2.19 **Flight**

Das Flight Objekt abstrahiert einen einzigen Flug mit seinen dazugehörigen Informationen. Eine Reise besteht aus mindestens zwei Flügen (Anreise, Rückreise). Die Eigenschaft isInwardFlight zeigt dabei an, ob es sich um einen Flug bei der Anreise oder der Rückreise handelt.

8.4.2.2.20 **FlightQueryRequestResults**

Diese Klasse ist das Resultatobjekt eines gesamten Flug-Requests. Es enthält alle gefundenen Flugreisen für alle Teilnehmer. Zusätzlich enthält es auch noch das das FlightQueryRequest Objekt mit den Details der Anfrage.

8.4.2.2.21 **CombinedQueryRequestResults**

Diese Klasse ist das Resultatobjekt einer kombinierten Abfrage. Es enthält ein FlightQueryRequestResults Objekt sowie eine alle gefundenen Hotels als Liste von HotelQueryParserResult Objekten.

8.4.3 Datenspeicherung

Die zu persistierenden Daten werden in einer relationalen Datenbank gespeichert. Es werden zwei verschiedene Datenbanken benötigt. Einerseits die Hauptdatenbank, welche die gesammelten Flug- und Hoteldaten sowie die Benutzerinformationen speichert. Andererseits die Task-Datenbank, die für die Speicherung der automatisch ausgeführten Abfrageaufträge persistiert, welche für die Datensammlung für die Analyse gebraucht werden.

8.4.3.1 Hauptdatenbank

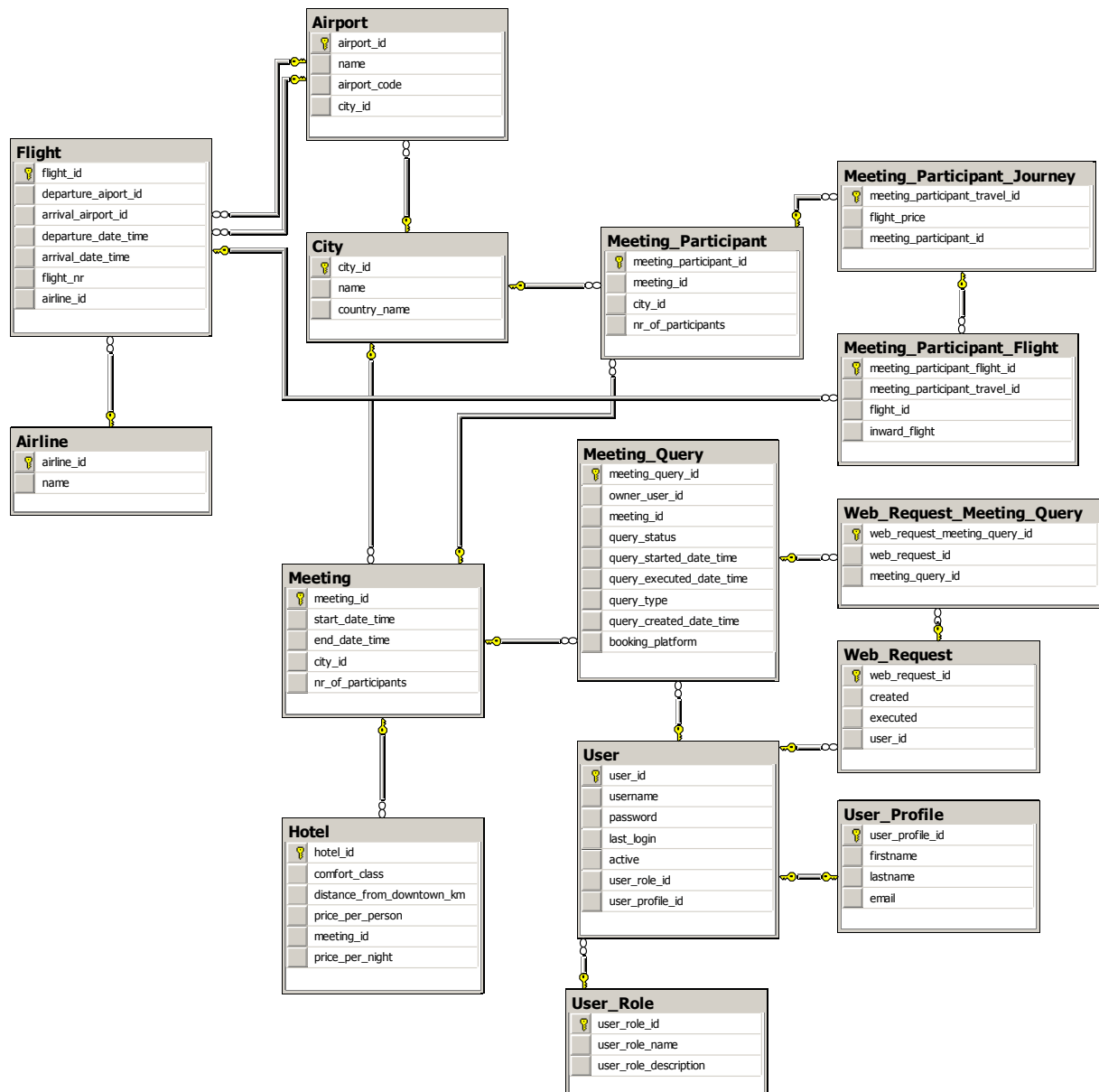


Abbildung 39: Datenbankmodel Hauptdatenbank

In Abbildung 39 sind die Tabellen der Hauptdatenbank und ihre Beziehungen zu einander dargestellt. Es handelt sich hierbei um ein konzeptionelles Model, d.h. spezifische Details werden hier weggelassen. Die einzelnen Tabellen und deren Funktion werden hier kurz beschrieben.

8.4.3.1.1 Web_Request

In dieser Tabelle werden Informationen über einen ausgeführten Request eines Benutzers über die Webapplikation gespeichert. Sie besitzt eine Fremdschlüsselbeziehung zu dem Benutzer in der User Tabelle, welcher die Abfrage erstellt hat.

8.4.3.1.2 User

In dieser Tabelle werden die Informationen über einen vorhandenen User gespeichert. Sie besitzt eine Fremdschlüsselbeziehung zu dem Benutzer-Rollen Eintrag in der Use_Role Tabelle sowie zu dem Benutzer Profil Eintrag in der User_Profile Tabelle.

8.4.3.1.3 User_Role

In dieser Tabelle werden die verschiedenen Benutzer-Rollen gespeichert. Anhand der Benutzer-Rolle können den Benutzern verschiedene Zugriffsrechte zugeteilt werden.

8.4.3.1.4 User_Profile

In dieser Tabelle werden erweiterte Benutzerinformationen abgelegt.

8.4.3.1.5 Meeting_Query

In dieser Tabelle werden die einzelnen Meeting Anfragen abgelegt. Sie besitzt eine Fremdschlüsselbeziehung zu dem Benutzer Eintrag in der User Tabelle. Damit ist ersichtlich, von welchem Benutzer die Anfrage erstellt wurde. Die zweite Fremdschlüsselbeziehung verknüpft den dazugehörigen Eintrag in der Meeting Tabelle mit der Anfrage.

Im weitem enthält die Tabelle eine Spalte mit dem Anfrage Typ. Daraus ist ersichtlich, um was für eine Abfrage es sich genau handelt (Flug-, Hotel- oder eine kombinierte Abfrage). Die für die Abfrage benutzte Buchungsplattform wird im entsprechenden Feld ebenfalls vermerkt.

8.4.3.1.6 Web_Request_Meeting_Query

Bei dieser Tabelle handelt es sich um eine reine Mapping-Tabelle, welche einen Eintrag der Web_Request Tabelle mit dem dazugehörigen Eintrag der Meeting_Query Tabelle verknüpft.

8.4.3.1.7 Meeting

In der Meeting Tabelle werden die Informationen zu einem spezifischen Meeting (Startdatum, Enddatum, Durchführungsort) gespeichert. Ein Eintrag in der Meeting Tabelle wird durch einen Fremdschlüssel mit dem dazugehörigen Ort Eintrag in der City Tabelle verknüpft.

8.4.3.1.8 Hotel

In der Hotel Tabelle werden alle benötigten Informationen zu einem Hotel gespeichert, welches bei einer Abfrage gefunden wurde. Über einen Fremdschlüssel wird der Hotel Eintrag mit dem dazugehörigen Meeting Eintrag in der Meeting Tabelle verknüpft.

8.4.3.1.9 Meeting_Participant

In dieser Tabelle werden die Teilnehmer eines Meetings gespeichert. Für jeden Teilnehmer mit einem unterschiedlichen Herkunftsort ist Eintrag vorhanden. Dieser wird mit einem Fremdschlüssel mit dem entsprechenden Herkunftsort in der City Tabelle verknüpft. Zusätzlich wird die Anzahl der Teilnehmer aus diesem Ort abgelegt. Über einen weiteren Fremdschlüssel wird der Teilnehmer mit dem Meeting Eintrag in der Meeting Tabelle verknüpft.

8.4.3.1.10 City

In dieser Tabelle werden alle verwendeten Orte abgelegt (Meeting-Durchführungsorte, Teilnehmer-Herkunftsorte und Orte der Flughäfen)

8.4.3.1.11 Meeting_Participant_Journey

In dieser Tabelle werden die Informationen zu einer Flugreise eines bestimmten Teilnehmers gespeichert. Mit einem Fremdschlüssel wird der jeweilige Eintrag mit dem Teilnehmer Eintrag in der Meeting_Participant Tabelle verknüpft.

8.4.3.1.12 Meeting_Participant_Flight

Bei dieser Tabelle handelt es sich um eine Verknüpfungs-Tabelle. Sie verbindet die einzelnen Flüge aus der Flight Tabelle mit der dazugehörigen Flugreise aus der Meeting_Participant_Journey Tabelle. Denn eine Reise kann aus mehreren Einzelflügen bestehen.

8.4.3.1.13 Flight

In dieser Tabelle werden die einzelnen Flüge und deren genaue Informationen abgelegt. Über einen Fremdschlüssel wird der Flug mit der dazugehörige Airline aus der Airline Tabelle verknüpft. Mit weiteren Fremdschlüsseln werden die Abflug- und Destinationsflughäfen mit dem Flugeintrag verknüpft.

8.4.3.1.14 Airline

Hier werden alle die Airlines gespeichert, welche einen Flug durchführen gespeichert.

8.4.3.1.15 Airport

Diese Tabelle enthält alle verwendeten Flughäfen und deren Informationen.

8.4.3.2 Task-Datenbank (für Batchmode)

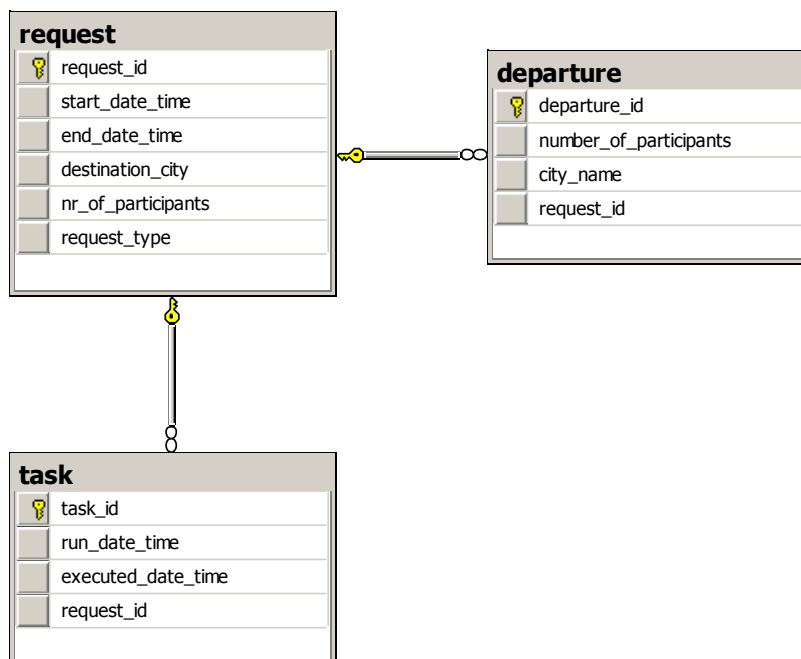


Abbildung 40: Datenbankmodel Task-Datenbank

In Abbildung 40 sind die Tabellen der Task-Datenbank und ihre Beziehungen zu einander dargestellt. Es handelt sich hierbei um ein konzeptionelles Model, d.h. spezifische Details werden hier weggelassen. Die einzelnen Tabellen und deren Funktion werden hier kurz beschrieben.

8.4.3.2.1 Departure

Die Departure Tabelle enthält die Informationen über die Teilnehmer und deren Herkunftsorte. Mit einem Fremdschlüssel wird er jeweilige Eintrag mit dem dazugehörigen Request Eintrag in der Request Tabelle verknüpft.

8.4.3.2.2 Task

Diese Tabelle enthält die Informationen über die Ausführung der Abfrage. Es kann angegeben werden, wann die Abfrage ausgeführt werden soll. Nachdem die Ausführung erfolgt ist, wird der Zeitpunkt der Fertigstellung der Anfrage hinzugefügt. Mit einem Fremdschlüssel werden die entsprechenden Requests, die zu Task gehören referenziert.

8.4.3.2.3 Request

Die Request Tabelle enthält die Informationen über das Durchführungsdatum und den Zielort der Reise, für welche die Abfrage durchgeführt werden soll. Hier wird auch angegeben, um was für eine Anfrage es sich dabei handelt (Flug-, Hotel oder kombinierte Abfrage).

8.4.3.3 Data Warehouse

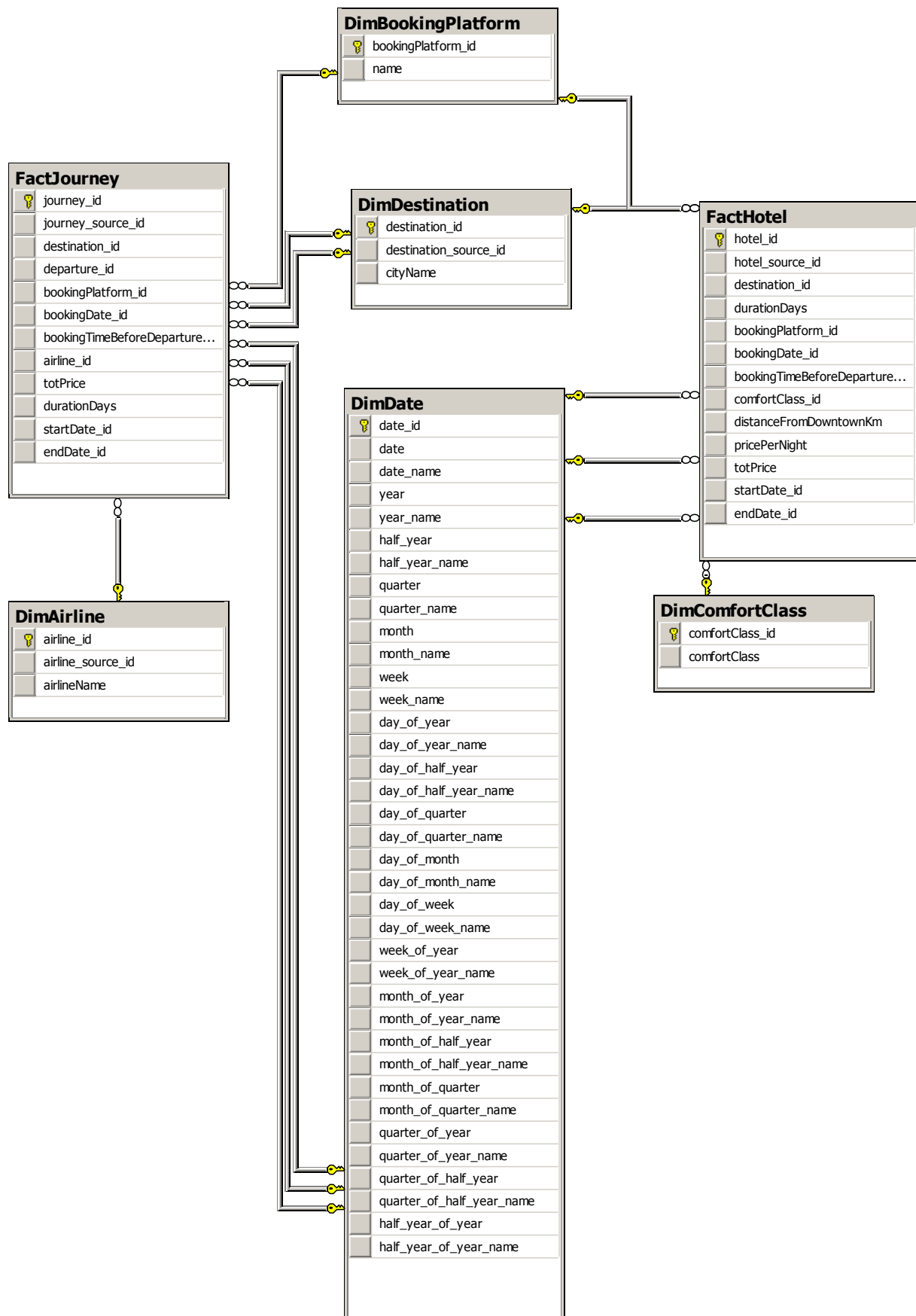


Abbildung 41: Datenbankmodel Data Warehouse

In Abbildung 41 ist das Model des Data-Warehouses zu sehen. Dieses wird verwendet um die gesammelten Daten effizienter auszuwerten und analysieren zu können.

Bei der Architektur des Data-Warehouses wird Redundanz bewusst in Kauf genommen, dafür können Abfragen auf dieser Art Struktur viel einfacher und zielführender durchgeführt werden. Die Daten werden aus einer „Produktiven-Datenbank“ (sind meist möglichst redundanzfrei) in das Data-Warehouse migriert.

Ein Data-Warehouse besteht in der Regel aus Facts-Tables und sogenannten Dimension-Tables. Die Fact-Tables enthalten dabei die eigentlichen Informationen. Die Dimension-Tables enthalten spezielle Eigenschaften die zu einem Fact-Table Eintrag gehören. Die Dimension-Tables können darum als eine Art Filter für die Fact-Table Einträge angeschaut werden.

Die vorhandenen Fact- und Dimension-Tables werden nun kurz beschrieben.

8.4.3.3.1 FactJourney

Dies ist die Faktentabelle für die Flugreisen. Sie enthält alle notwendigen Daten einer spezifischen Flugreise. Über Fremdschlüssel werden die Dimensionen Airline (durchführende Fluggesellschaft), die Dimension BookingPlatform (von welcher Buchungsplattform die Resultate stammen), die Dimension Destination (für Abflug- und Zielort) sowie die Dimension Date (für Anreise, Rückreisezeitpunkt und Buchungszeitpunkt (also Zeitpunkt der Abfrage auf dem Buchungsportal)) mit dem Fact Eintrag verknüpft.

8.4.3.3.2 FactHotel

Dies ist die Faktentabelle für die Hotels. Sie enthält alle notwendigen Daten eines spezifischen Hotelangebots. Über Fremdschlüssel werden die Dimension BookingPlatform (von welcher Buchungsplattform die Resultate stammen), die Dimension Destination (wo sich das Hotel befindet) sowie die Dimension Date (für Anreise, Rückreisezeitpunkt und Buchungszeitpunkt (also Zeitpunkt der Abfrage auf dem Buchungsportal)) mit dem Fact Eintrag verknüpft.

8.4.3.3.3 DimBookingPlatform

In dieser Dimensions-Tabelle sind die, für die Abfragen verwendeten Buchungsplattformen vorhanden.

8.4.3.3.4 DimDestination

In dieser Dimensions-Tabelle sind die vorkommenden Orte vorhanden.

8.4.3.3.5 DimDate

In dieser Dimensions-Tabelle sind die vorkommenden Datums- und Zeitinformationen vorhanden.

8.4.3.3.6 DimAirline

In dieser Dimensions-Tabelle sind die vorkommenden Fluggesellschaften vorhanden.

8.4.3.3.7 DimComfortClass

In dieser Dimensions-Tabelle sind die vorkommenden Hotel-Komfortklassen vorhanden.

8.4.4 Sequenzdiagramme

Um die Übersichtlichkeit zu erhöhen wurden bei den folgenden Diagrammen die Auslösung der Aktion sowie der genaue Ablauf bei der Speicherung der Daten weggelassen.

Als aktionsauslösender Akteur kann bei allen entweder der Benutzer über die Webapplikation oder aber die Batchmode-Applikation, welche gespeicherte Aufträge startet, fungieren.

8.4.4.1 Flugdaten Abfrage

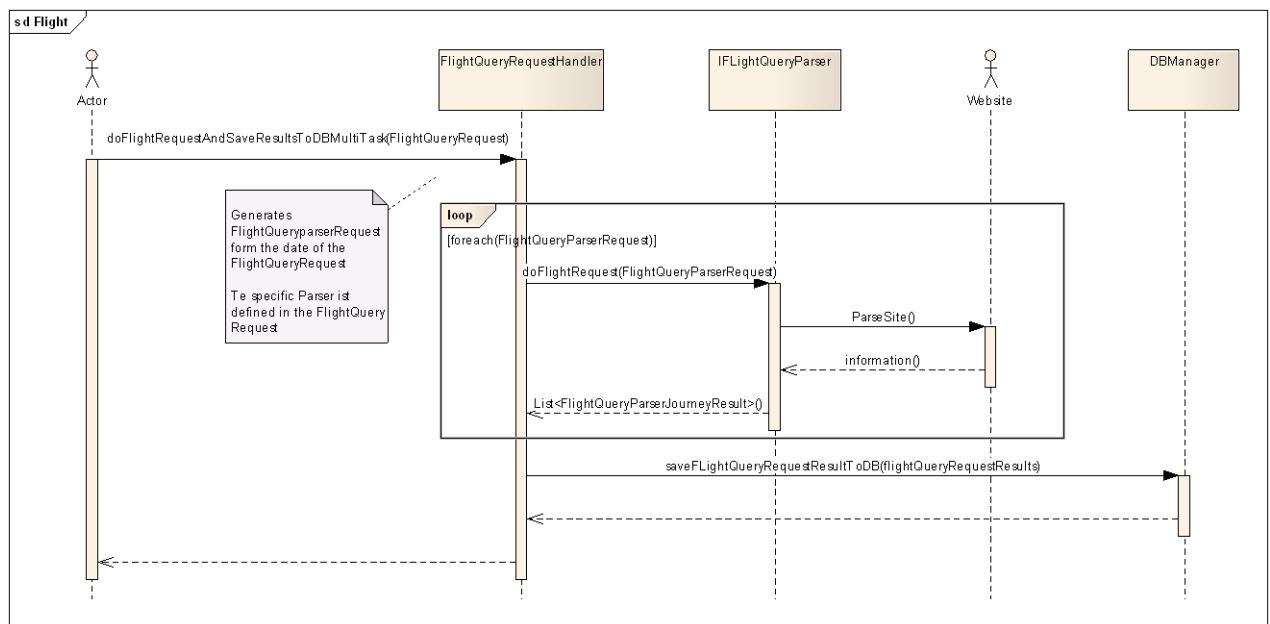


Abbildung 42: Sequenzdiagramm Flugdaten Abfrage

Abbildung 42 zeigt den Ablauf einer Flugdatenabfrage.

8.4.4.2 Hoteldaten Abfrage

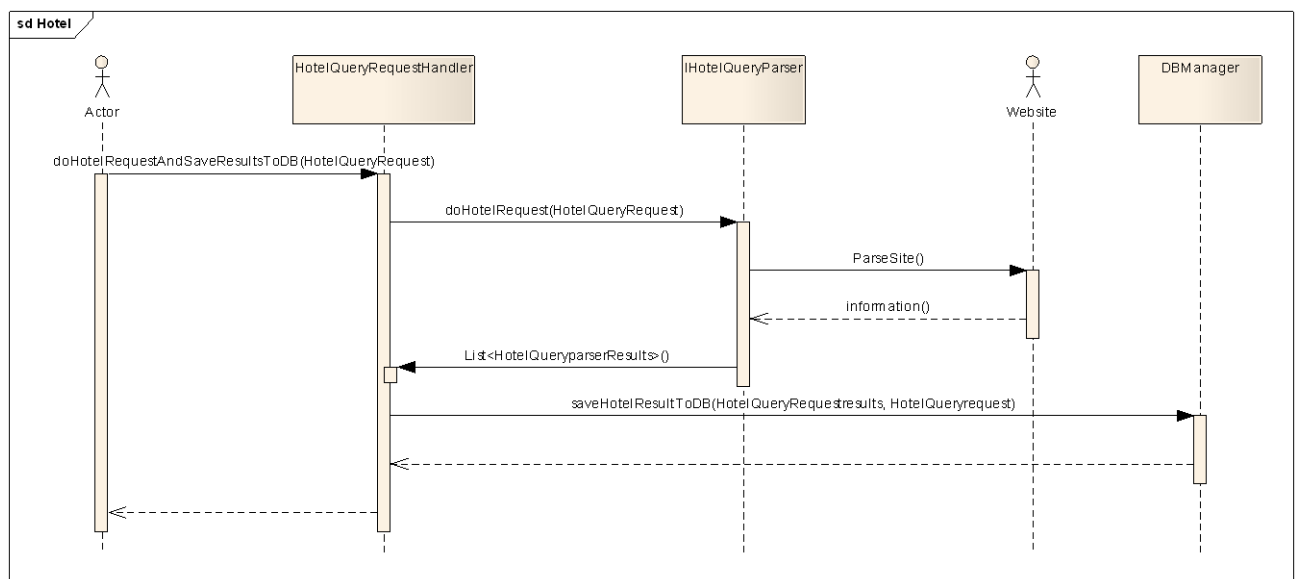


Abbildung 43: Sequenzdiagramm Hoteldaten Abfrage

Abbildung 43 zeigt den Ablauf einer Hoteldatenabfrage.

8.4.4.3 Kombinierte Abfrage

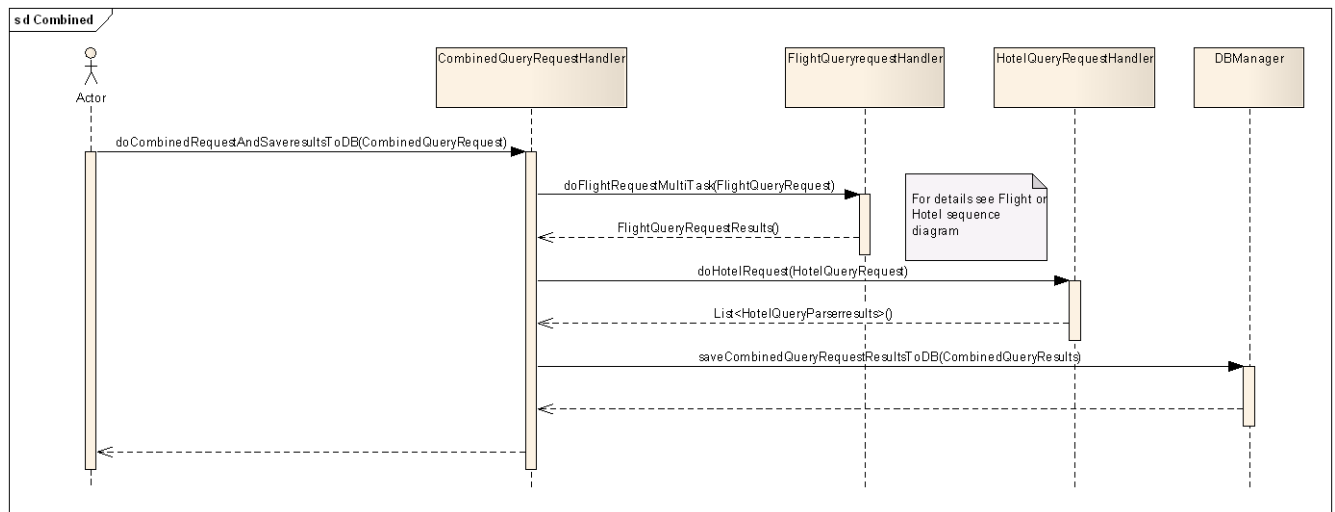


Abbildung 44: Sequenzdiagramm kombinierte Abfrage

Abbildung 44 zeigt den Ablauf einer kombinierten Abfrage.

8.4.5 Prozesse und Threads

Bei der Flugkostenabfrage wurde, wie bereits früher erwähnt, eine Multi-Threading Lösung erarbeitet. Damit kann die Wartezeit bis das Resultat einer einzelnen Flugabfrage von der Buchungsplattform zurückgegeben wird durch das Ausführen weiterer Abfragen genutzt werden. Die Parallelisierung wurde mit Hilfe eines Thread-Pools umgesetzt. Der benutzte Thread-Pool wird durch die .NET Umgebung verwaltet.

Die auszuführenden Fluganfragen werden an den Thread-Pool übergeben und werden von diesem parallel ausgeführt. Sobald alle Resultate aller Abfragen an den Handler zurückgegeben wurden werden diese dann in die Datenbank gespeichert.

Auch bei der Ausführung der Abfragen über die Webapplikation wurde Threading eingesetzt. Sobald der User eine Abfrage startet wird diese in einem separaten Thread von der Kernapplikation ausgeführt. So bleibt die Webapplikation für die Ausführungsdauer der Abfrage nicht blockiert.

8.5 Implementation (Entwicklung) und Test

8.5.1 Implementation: Erläuterungen wichtiger konkreter Klassen

In diesem Kapitel werden die implementierten Klassen mit ihren Eigenschaften und Methoden erklärt. Da die Konzepte bereits im Kapitel 8.4.2 ausführlich beschrieben wurden, werden hier nur noch die Klassendiagramme der konkreten Klassen aufgeführt und nur falls notwendig weitere Erklärungen gemacht.

8.5.1.1 Gesamtübersicht

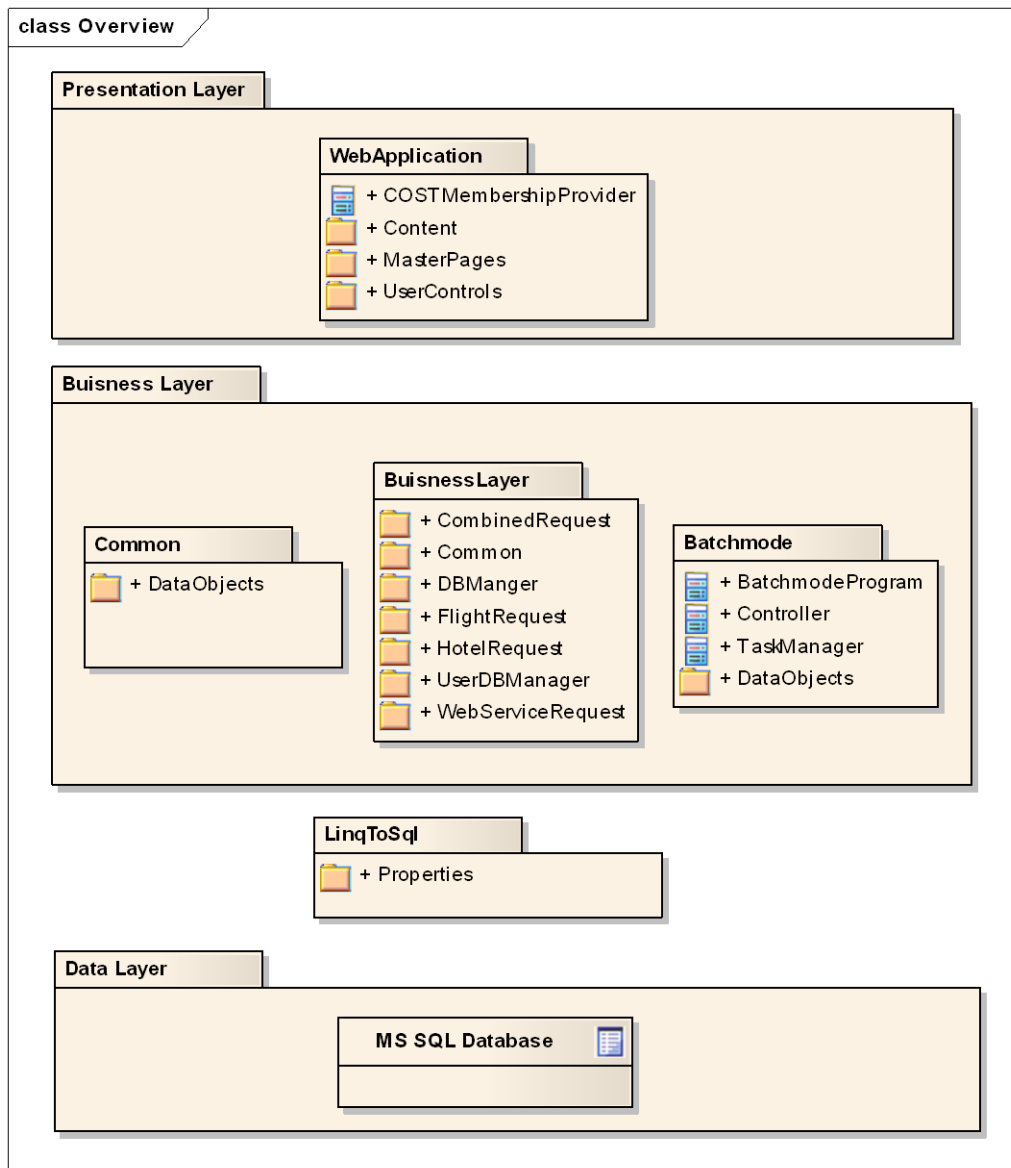


Abbildung 45: Übersicht Packages

Abbildung 45 zeigt eine Übersicht wie die Klassen nach Schichten und Paketen geordnet und organisiert sind.

8.5.1.2 Batchmode

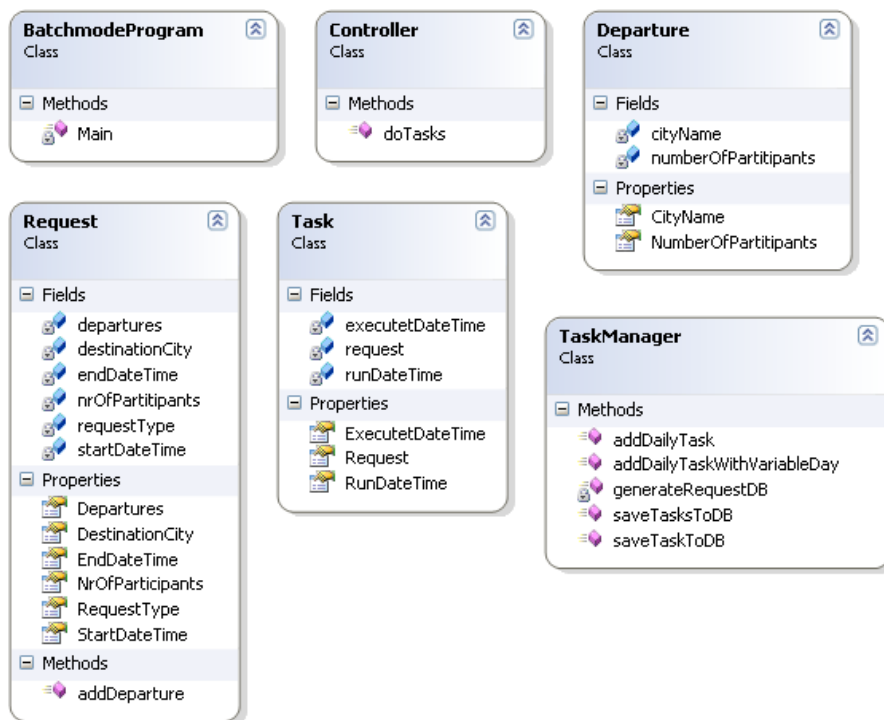


Abbildung 46: Klassendiagramm Batchmode

8.5.1.3 Business Layer

8.5.1.3.1 Request

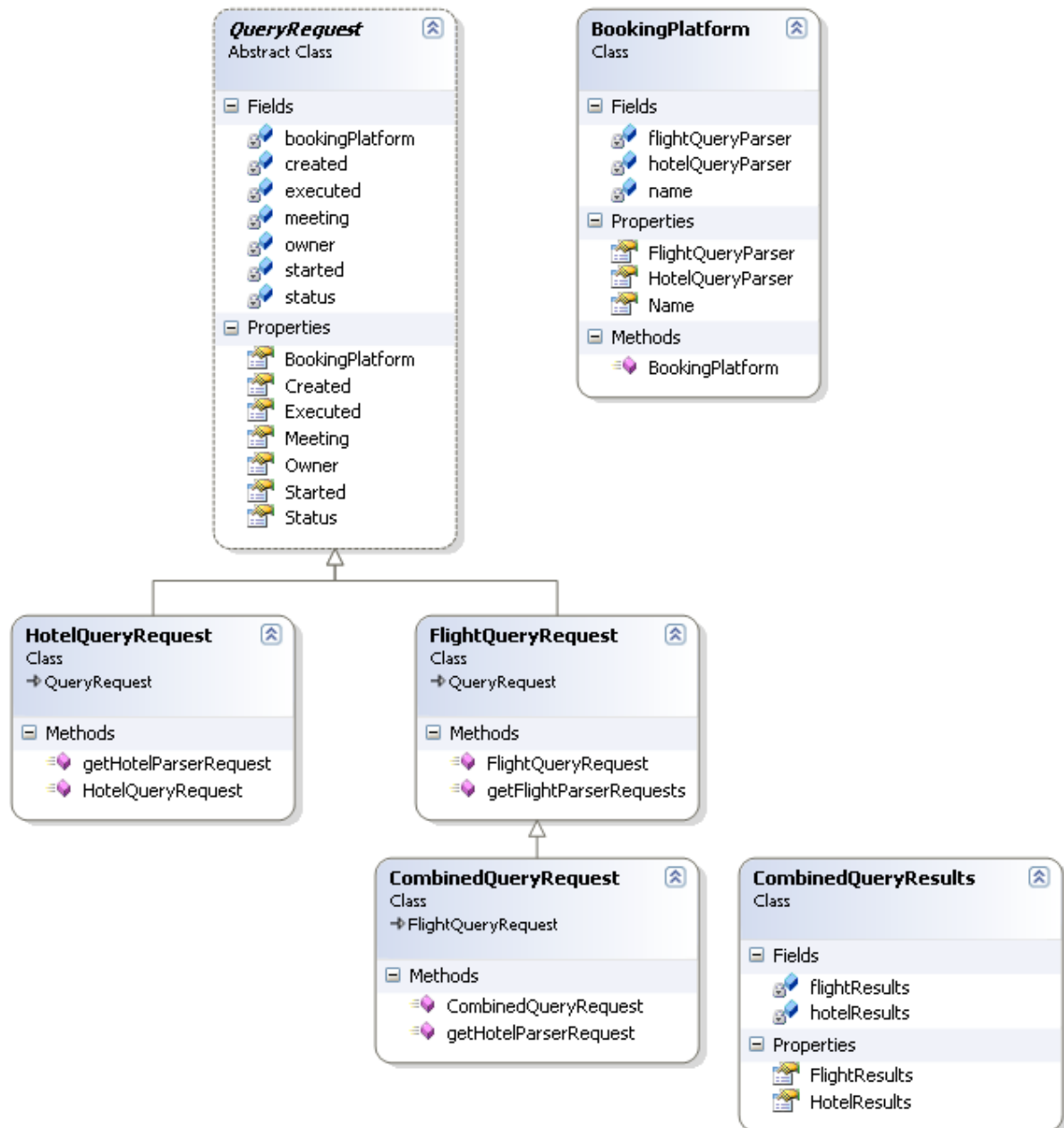


Abbildung 47: Klassendiagramm Request

8.5.1.3.2 Flight Parser

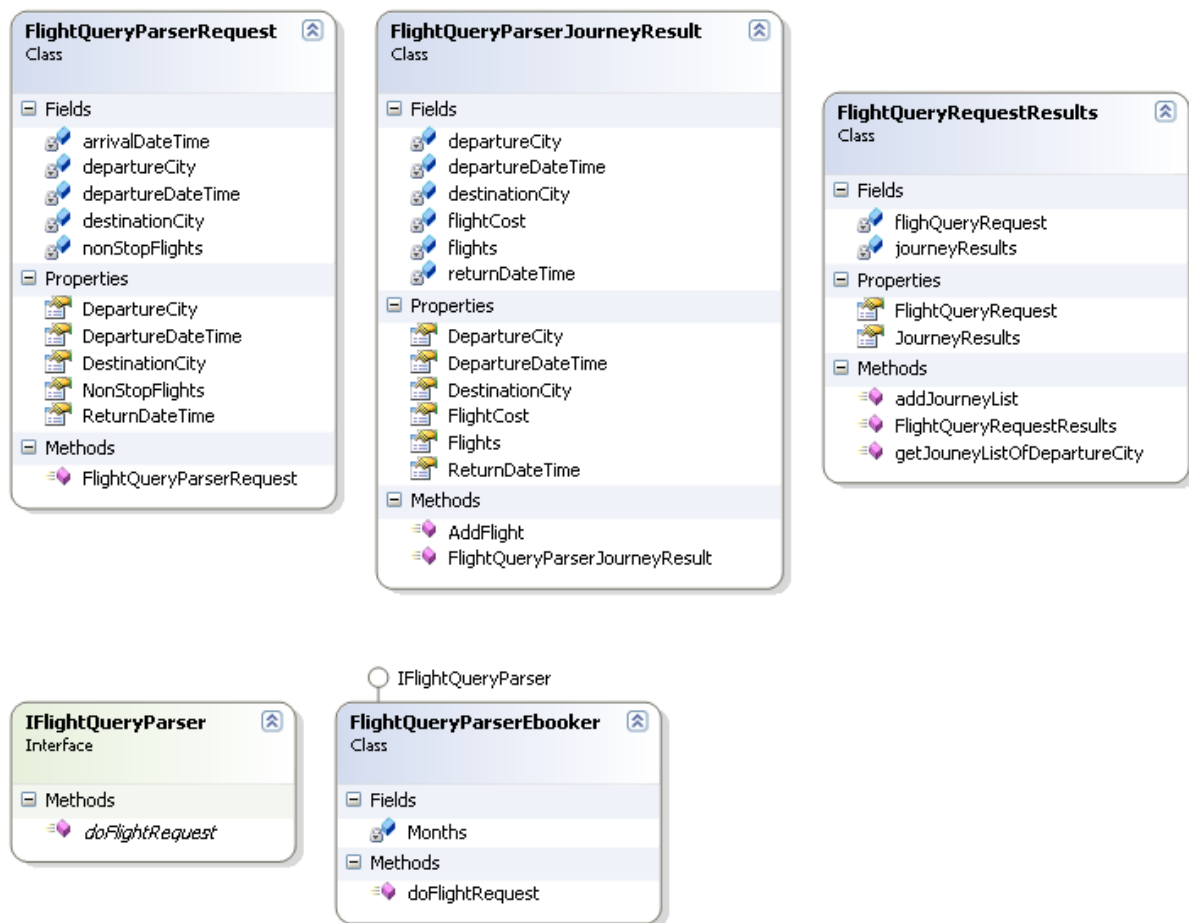


Abbildung 48: Klassendiagramm Flight Parser

8.5.1.3.3 Hotel Parser

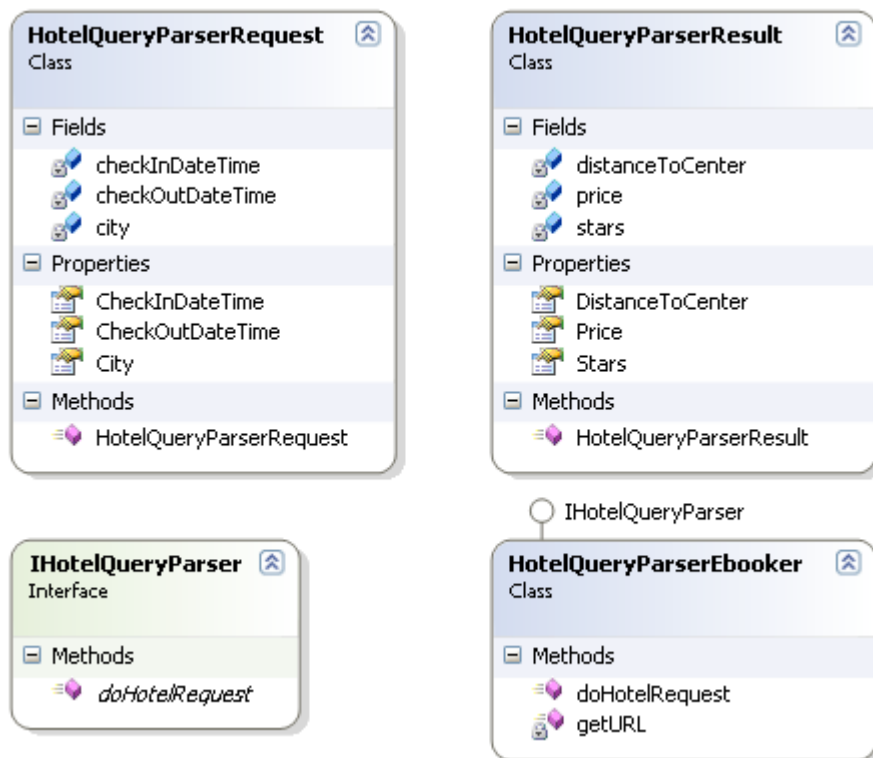


Abbildung 49: Klassendiagramm Hotel Parser

8.5.1.3.4 Handler

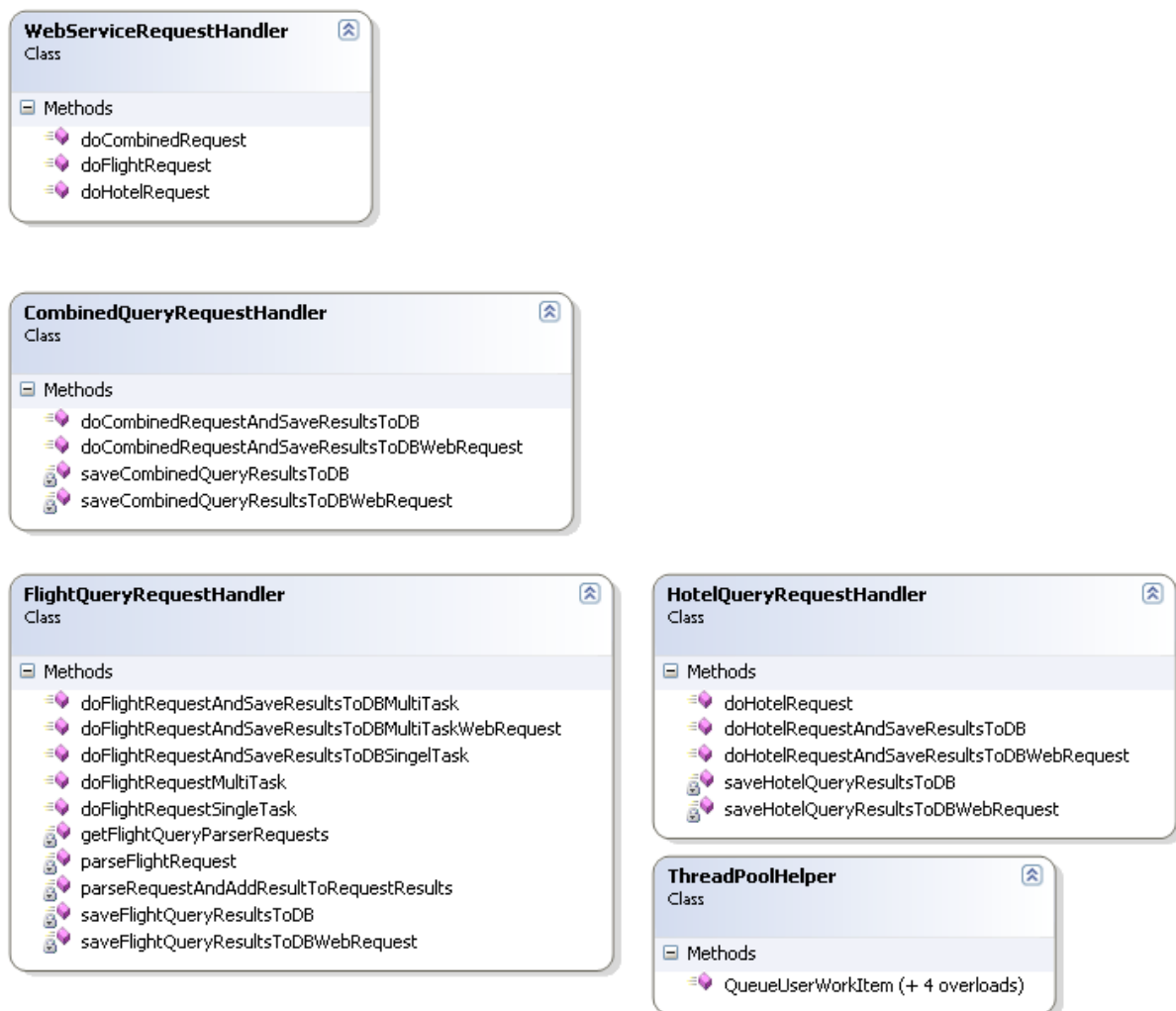


Abbildung 50: Klassendiagramm Parser

Die ThreadPoolHelper Klasse ist eine Hilfsklasse für die Benutzung des von C# zu Verfügung gestellten ThreadPools. Sie vereinfacht den Umgang mit dem ThreadPool wesentlich.

8.5.1.4 Common

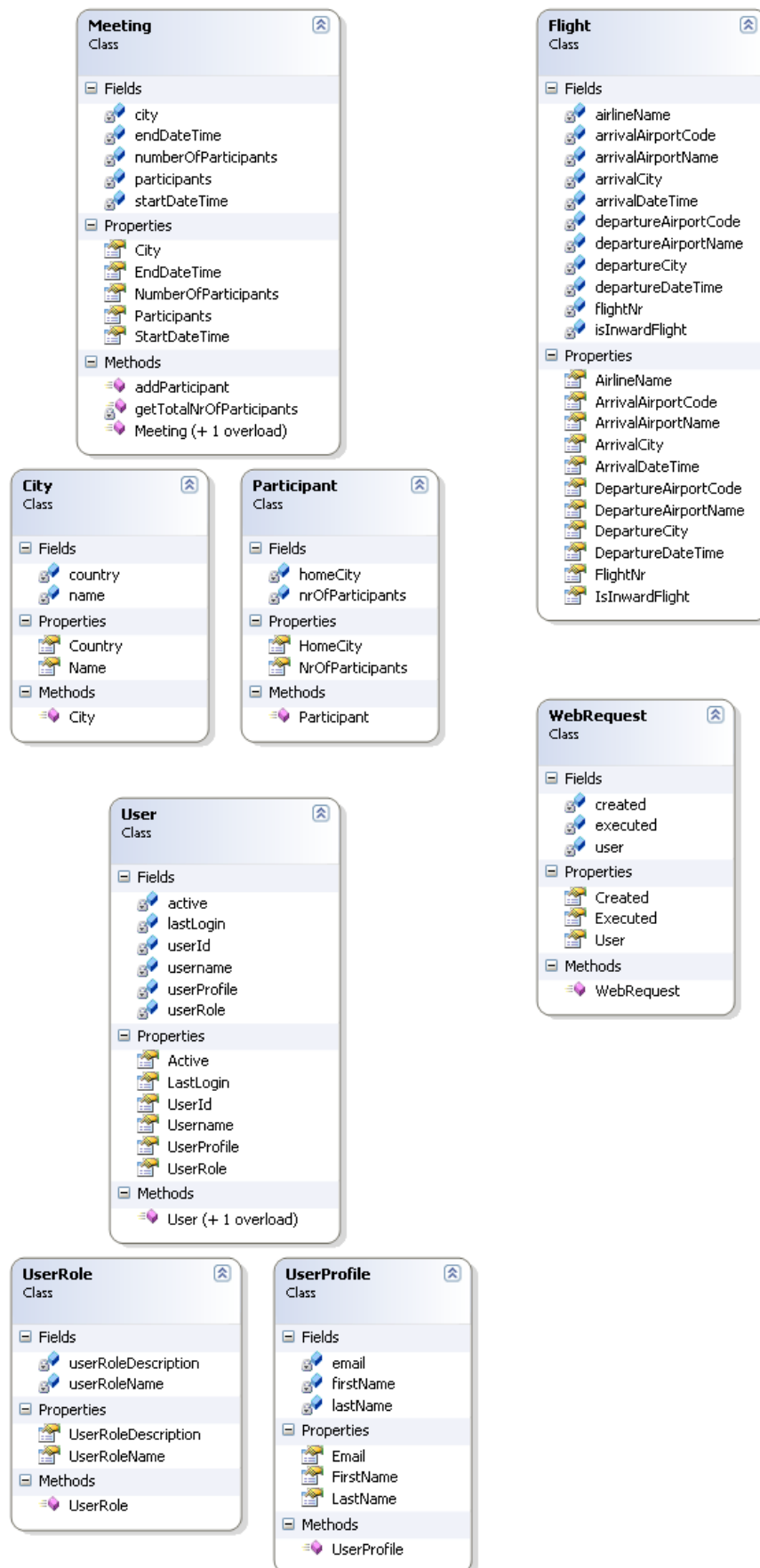


Abbildung 51: Klassendiagramm Common

8.5.1.5 LingToSql

Die LinqToSql Klassen werden von der .NET Umgebung automatisch anhand der verwendeten Datenbank generiert. Dabei bilden sie die Tabellen auf Objekte ab.

Mit Hilfe von LinqToSql und dieser generierten Klassen wird das Verwalten der Daten in der Datenbank stark vereinfacht.

8.5.1.5.1 Datenklassen der Hauptdatenbank

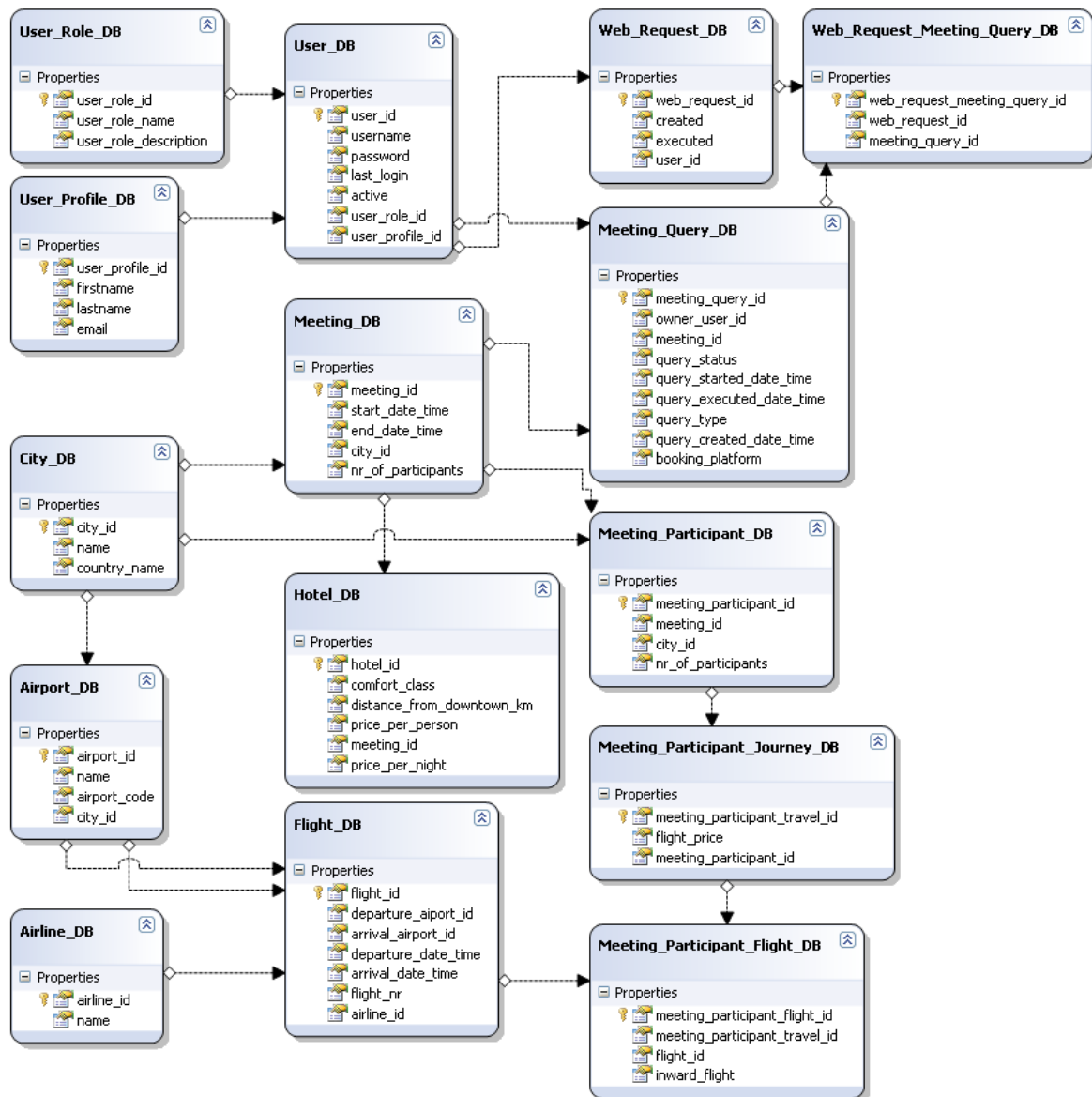


Abbildung 52: Klassendiagramm LinqToSql Klassen Hauptdatenbank

8.5.1.5.2 Datenklassen für Task-Datenbank (für Batchmode)

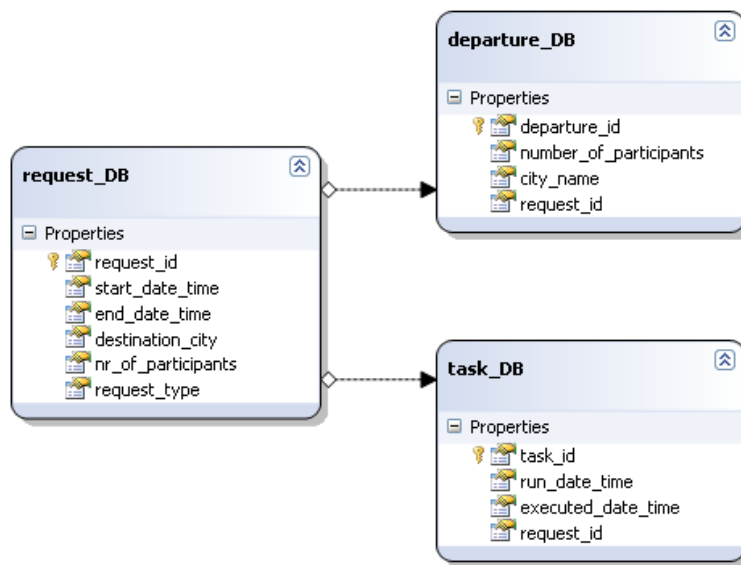


Abbildung 53: Klassendiagramm LinqToSql Klassen Task-Datenbank

8.5.2 Datenbank

Da die Konzepte der Datenbanken bereits in Kapitel 8.4.3 ausführlich beschrieben wurden, wird hier auf eine detaillierte Ausführung verzichtet. In den Diagrammen sind die jeweiligen Datentypen der Felder zu sehen und es ist zu erkennen, ob die Felder Null-Werte zulassen.

8.5.2.1 Hauptdatenbank

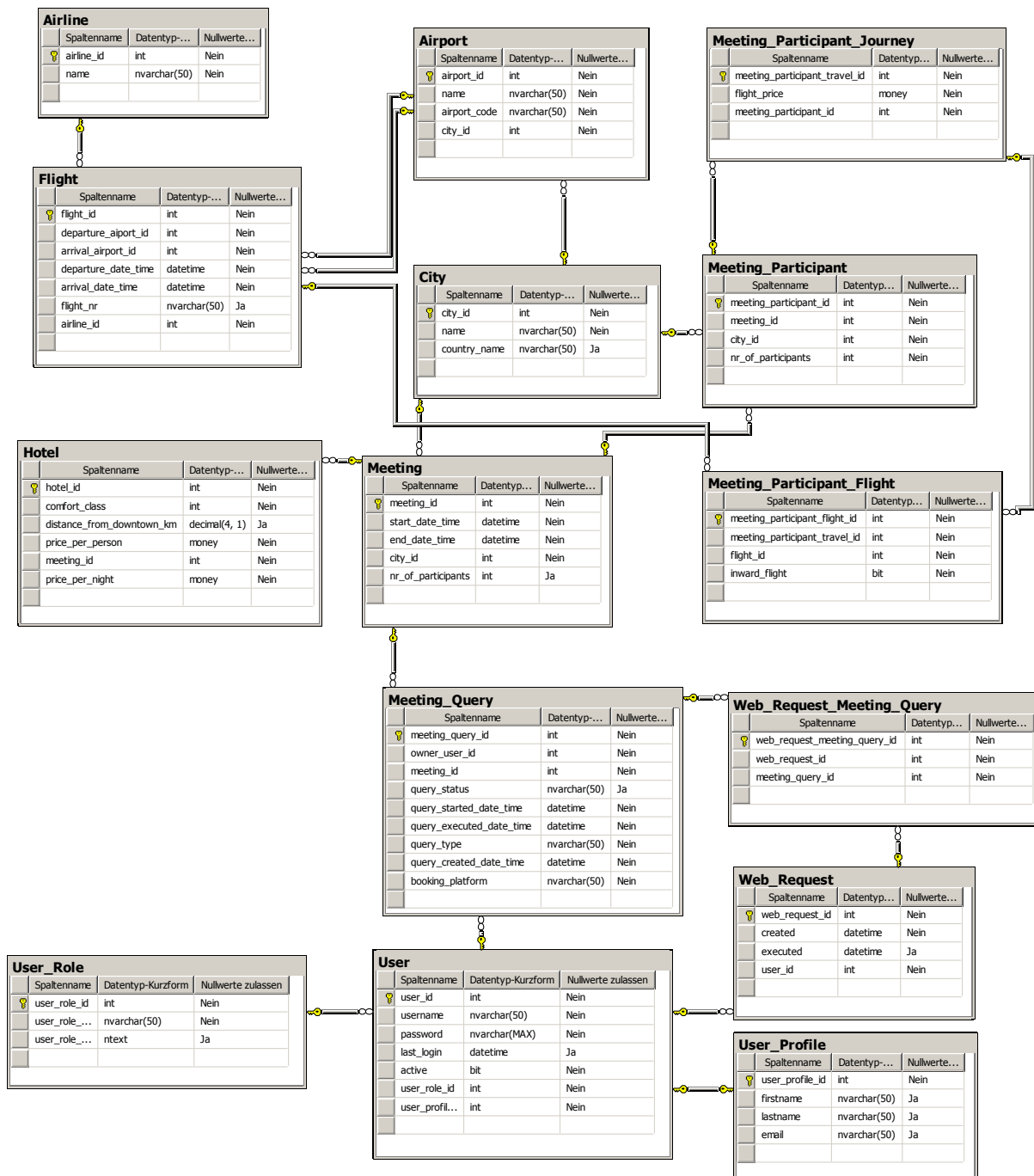


Abbildung 54: Datenbankdiagramm Hauptdatenbank

8.5.2.2 Task-Datenbank (für Batchmode)

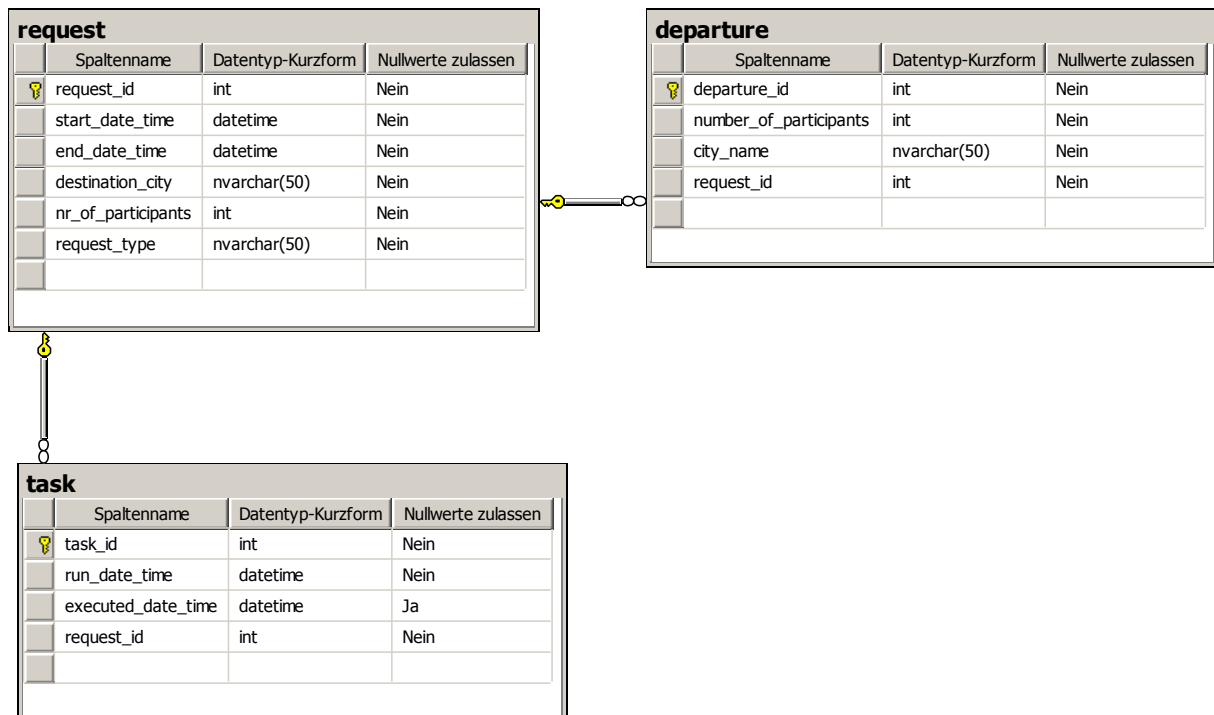


Abbildung 55: Datenbankdiagramm Task-Datenbank

8.5.2.3 Data-Warehouse

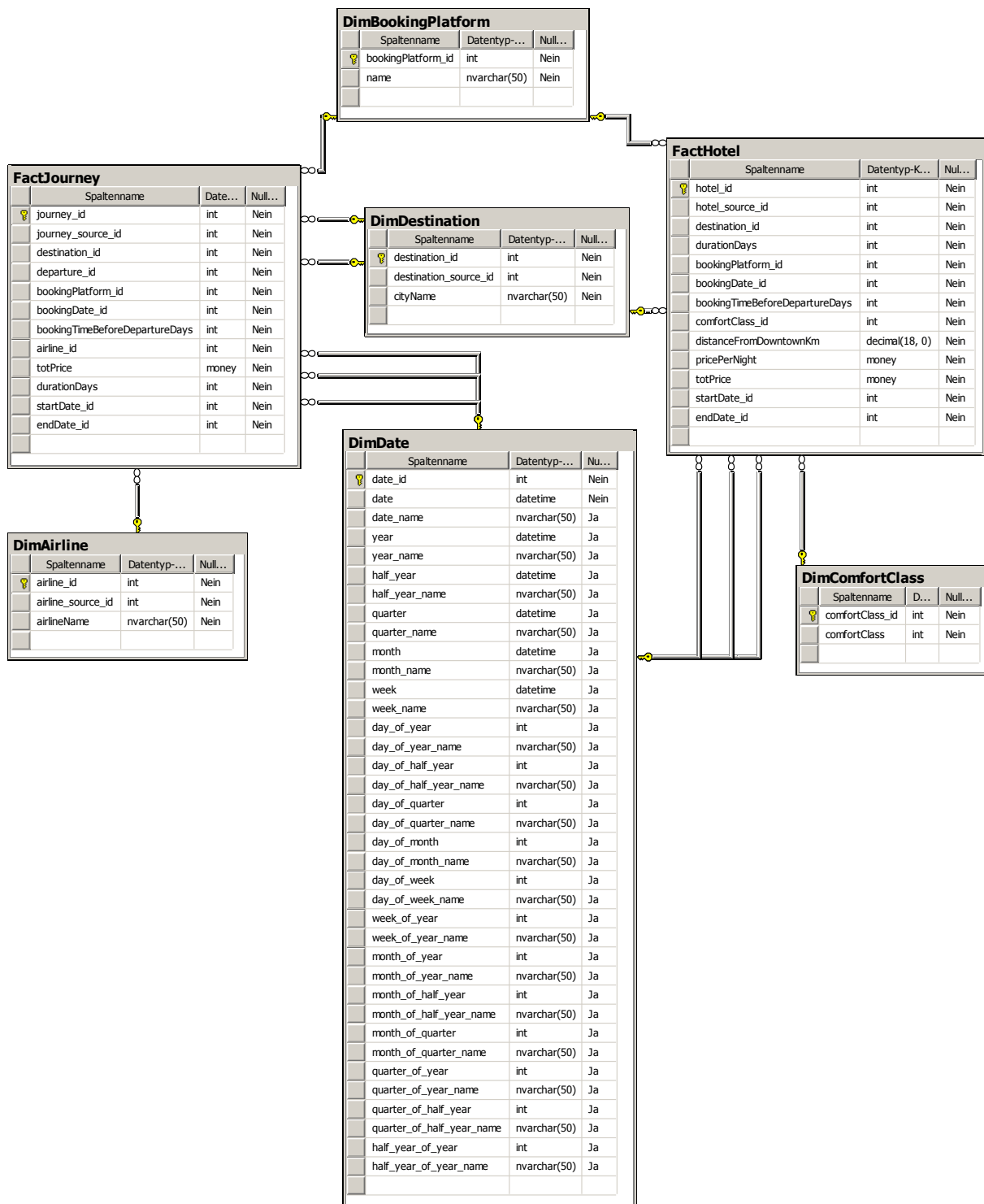


Abbildung 56: Datenbankdiagramm Data-Warehouse

8.5.2.3.1 Datenintegration

Für die Analyse der Daten wurden diese von der Hauptdatenbank in das Data-Warehouse importiert. Dies wurde mit Hilfe der MS SQL Server Integration Services durchgeführt.

Hier wird im Groben beschrieben wie dies mit Hilfe der Integration Services umgesetzt wurde.

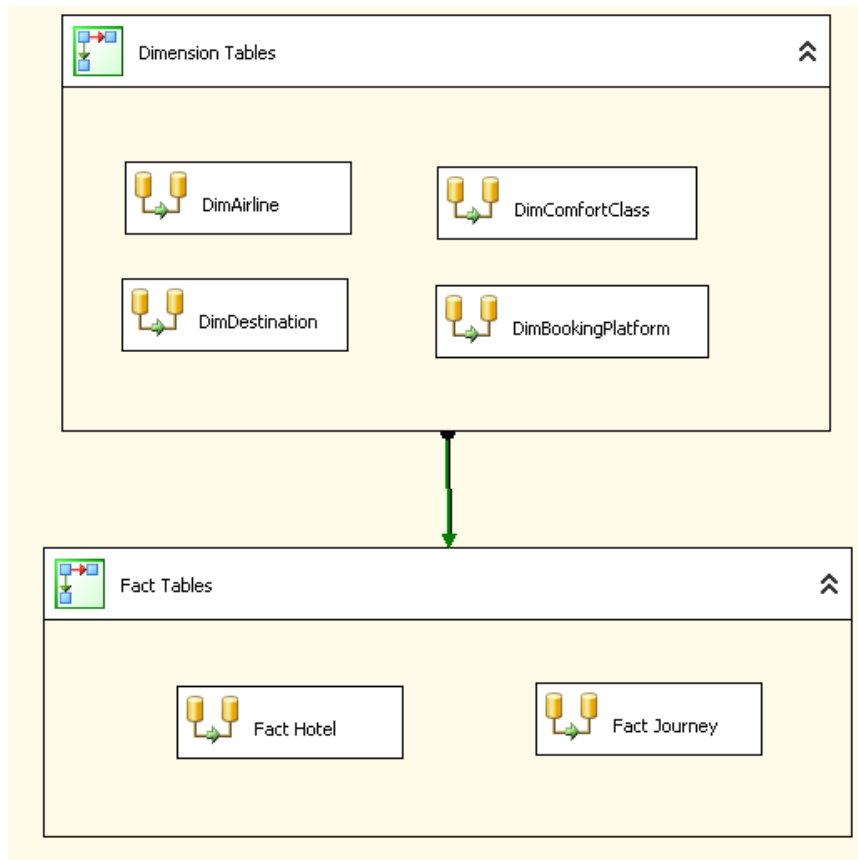


Abbildung 57: Ablaufsteuerung der Datenintegration

In Abbildung 56 ist eine Übersicht über die auszuführenden Aktionen und deren zeitlicher Ablauf zu sehen.

Zuerst werden die Daten für die Dimension-Tables aus der Hauptdatenbank geholt und in das Data-Warehouse importiert. Anschliessend werden die Fact-Tables mit den Daten aus der Hauptdatenbank gefüllt.

Der genaue Ablauf der Datenintegration einer Dimension-Table wird hier anhand der DimComfortClass Tabelle illustriert.

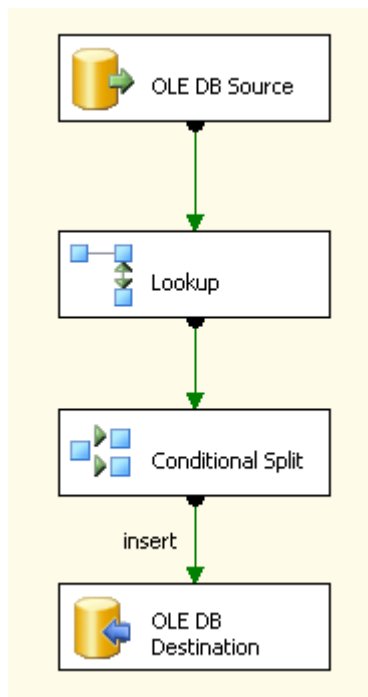


Abbildung 58: Datenfluss beim Füllen einer Dimension-Table

Der genaue Ablauf der Datenintegration einer Dimension-Table wird hier anhand der DimComfortClass Tabelle illustriert (Abbildung 58).

- Zuerst werden die vorhandenen Komfortklassen in der Hauptdatenbank mittels einer SQL Abfrage selektiert
- Anschliessend wird geschaut, ob die jeweilige Komfortklasse schon in der Dimension-Table vorhanden ist
- Falls dies nicht der Fall ist wird die Komfortklasse der Dimension-Table hinzugefügt

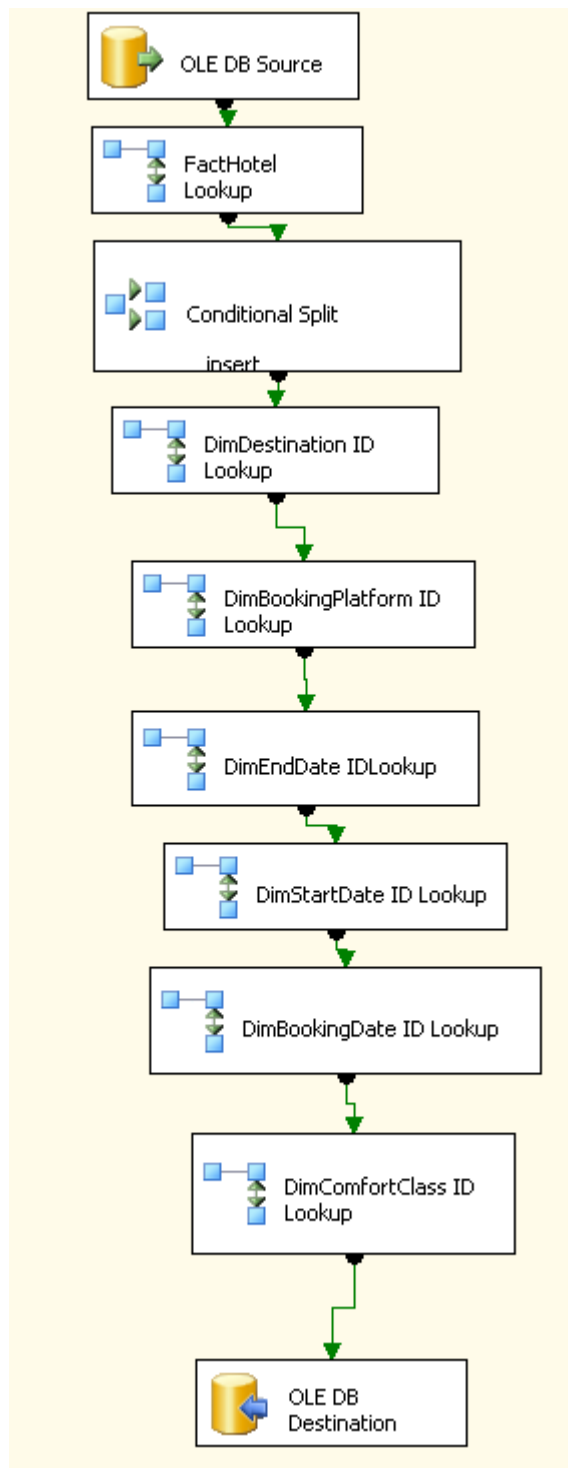


Abbildung 59: Datenfluss beim Füllen einer Fact-Table

Der genaue Ablauf der Datenintegration einer Fact-Table wird hier anhand der FactHotel Tabelle illustriert (Abbildung 59).

- Zuerst werden die vorhandenen Hoteldaten in der Hauptdatenbank mittels einer SQL Abfrage selektiert
- Anschliessen wird geschaut, ob die jeweiligen Hoteldaten schon in der Fact-Table vorhanden sind
- Falls dies nicht der Fall ist, werden die dazugehörigen Dimension-Table Einträge in den Dimension Tabellen gesucht und die Hoteldaten mit den gefundenen Verknüpfungen zu den Dimensionen in die Hotel-Fact Table gespeichert

8.6 Resultate und Weiterentwicklung

8.6.1 Resultate

Wir haben eine Webapplikation entwickelt, welche voll funktionsfähig ist. Man kann damit gut für verschiedene Meetings-Varianten die Kosten anzeigen lassen und diese vergleichen. Die Applikation berücksichtigt auch, dass mehrere Teilnehmer aus demselben Herkunftsort stammen können. Man kann je nach Bedarf nur die Flug-, nur die Hotel- oder die kombinierten Kosten berechnen lassen.

Für die Analyse haben wir ein Data-Warehouse eingerichtet und die Routinen für das Migrieren der Daten erstellt. Die Daten werden mit den Integration Services des Microsoft SQL Servers ins Data-Warehouse transferiert. Diese Services bieten ein recht effizientes Auslesen der Daten aus der Stammdatenbank und können direkt in das Data-Warehouse übernommen werden.

Damit man die Daten möglichst effizient analysieren kann, haben wir auf Basis der Daten im Datawarehouse einen Cube erstellt. In diesem Cube haben wir zwei Facts-Tabellen erstellt. Die Flugreise- und die Hotelreiseinformationen Tabelle. Damit man die Facts besser auswerten kann haben wir verschiedene Dimensionen hinzugefügt. Einige Dimensionen sind auch nur virtuelle Dimensionen, welche nur auf berechnete Daten basieren. Die Daten im Cube haben wir mit verschiedener Data Mining Verfahren analysiert.

Für die Analyse haben wir einige Reporte erstellt. Diese zeigen verschiedene Szenarien, die im Kapitel 0 veranschaulicht wurden.

8.6.2 Möglichkeiten der Weiterentwicklung

Unsere Arbeit kann an verschiedenen Orten weiterentwickelt werden.

Zum einen kann man die Benutzeroberfläche der Webanwendung überarbeiten und anschaulicher sowie benutzerfreundlicher gestalten.

Weiter kann man neue Parser für andere Buchungsplattformen implementieren oder die Applikation direkt über einen API mit einem Buchungssystem verbinden. Vor allem bei diesem Punkt ist noch viel Optimierungspotential vorhanden. Man könnte einiges an Geschwindigkeit herausholen, da das Warten auf die Resultate der Buchungswebsite und das anschließende Parsen viel Zeit in Anspruch nimmt. Zusätzlich ist dieses Verfahren auch Anfällig auf eine sich ändernde Seitenstruktur der Buchungsplattform.

Auch die Analyse der Daten kann man weiterführen und für bessere Resultate mehr Daten über eine längere Zeitspanne sammeln.

8.7 Projektmanagement

8.7.1 Prototypen, Releases, Meilensteine

8.7.1.1 Meilensteine

Datum	Meilenstein	Beschreibung
29.03.09	MS1	Planung / Requirements abgeschlossen und dem Betreuer zum Review abgegeben.
19.04.09	MS2	Analyse und Software Design erarbeitet und können dem Betreuer zum Review übergeben werden. Automatisierte Abfragen von Flug- und Hotelpreisen implementiert. Daten werden in Datenbank gespeichert. Funktionalität kann dem Betreuer demonstriert werden.
10.05.09	MS3	Flug- und Hotelpreisabfragen funktionieren vollständig. Die ersten Auswertungen der Hotel- und Flugkosten erstellt. Erarbeitete Auswertungen werden dem Betreuer demonstriert.
07.06.09	MS4	Benutzerverwaltung ist implementiert. Webapplikation erstellt und kann dem Betreuer demonstriert werden. Feature Freeze.
12.06.09	MS5	Applikation ist fertiggestellt und vollständig benutzbar. Bugs wurden behoben. Projektende, Abgabe der Arbeit.

8.7.2 Team, Rollen und Verantwortlichkeiten

Die Studienarbeit wird in Zweierteams durchgeführt. Der Aufwand pro Person für die 12 ETCS Punkte beträgt ca. 360h.

8.7.2.1 Organisationsstruktur (Organizational Structure)

Fabius Bernet	Projektmitarbeiter
Jonpaul Feuerstein	Projektmitarbeiter

8.7.2.2 Externe Schnittstellen (external Interfaces)

Prof. Dr. Peter Heinzmann	Betreuer
---------------------------	----------

8.7.2.3 Besprechungen (Meetings)

Die Besprechungen finden nach Möglichkeit jeweils am Mittwochnachmittag statt. Diese Besprechungen können bei Bedarf verschoben oder abgesagt werden.

8.7.3 Aufwandschätzung, Zeitplan, Projektplan

8.7.3.1 Projekt Kostenvoranschlag (Project Estimates)

Pro Person wird mit einem Aufwand von ca. 360h während dem Projektzeitraum vom 16.02.2009 bis zum 12.06.2009 gerechnet.

Dem Betreuer wird von der HSR ein Aufwand von 50h vergütet. Experte und Korreferent dürften zusammen etwa 10h für diese Arbeit aufwenden. Das ergibt einen Gesamtaufwand von ca. 780 Personenstunden.

8.7.3.2 Projektplan (Project plan)

8.7.3.2.1 Zeitplan (Phase Plan)

Siehe „Projektplan.xlsx „

8.7.3.3 Arbeitspakete (Work Package)

Die verschiedenen Arbeitspakete werden wie folgt dargestellt.

{Kat. Nr.}	{Kategorienname}		{Aufwand für Kat.}
{AP Nr.}	{Arbeitspaketname}	{Verantwortlicher}	{Aufwand für AP}
Beschreibung	{Beschreibung des Arbeitspakets}		
Abhängigkeiten	{Aufzählung der Arbeitspakete, von welchen das spezifische Arbeitspaket abhängig ist}		
Risiken/Probleme	{Aufzählung der Risikofaktoren aus der Risikoanalyse, welche dieses spezifische Arbeitspaket tangieren könnten}		

8.7.3.3.1 Projektmanagement

1	Projektmanagement		22 Std.
1.1	Projektzeitplan erstellen	Team	8 Std.
Beschreibung	Projektzeitplan erstellen. Enthält Aufwandschätzung aller Arbeitspakete sowie deren Ist-Aufwand		
Abhängigkeiten	-		
Risiken/Probleme	R01		
1.2	Projektplan Dokument erstellen	Team	6 Std.
Beschreibung	Projektplan Dokument erstellen. Projektorganisation festlegen und Iterationen / Meilensteine planen.		
Abhängigkeiten	-		
Risiken/Probleme	R01		
1.3	SVN installieren	Team	2 Std.
Beschreibung	SVN Repository einrichten und initialisieren.		
Abhängigkeiten	-		
Risiken/Probleme	R01		
1.4	IDE einrichten	Team	6 Std.
Beschreibung	Die IDE installieren und an unsere Bedürfnisse anpassen		
Abhängigkeiten	-		
Risiken/Probleme	R01		

8.7.3.3.2 Requirements

2	Requirements		64Std.
2.1	Ideen für Features Sammeln	Team	12 Std.
Beschreibung	Abklären was die Applikation machen soll und was (noch) nicht.		
Abhängigkeiten	-		
Risiken/Probleme	R01		
2.2	Anforderungsspezifikationen definieren	Team	20 Std.
Beschreibung	Anforderungsspezifikationen definieren und entsprechendes Dokument erstellen		
Abhängigkeiten	[2.1] Ideen für Features Sammeln		
Risiken/Probleme	R01		
2.3	Passende Daten API evaluieren	Team	10 Std.
Beschreibung	Die verschiedenen API die Flugdaten anbieten vergleichen und eine oder zwei mögliche aussuchen.		
Abhängigkeiten			
Risiken/Probleme	R01		
2.4	Technologien studieren und Evaluieren	Team	12 Std.
Beschreibung	Die geeignetste Technologie für unsere Applikation evaluieren.		
Abhängigkeiten	[2.3] Passende Daten API evaluieren		
Risiken/Probleme	R01		
2.5	Testfälle spezifizieren	Team	10 Std.
Beschreibung	Testfälle spezifizieren mit denen man die Funktionalität der Applikation getestet werden kann.		
Abhängigkeiten			
Risiken/Probleme	R01		

8.7.3.3 Analyse

3	Analyse		65 Std.
3.1	Externes Design	Team	10 Std.
Beschreibung	Das Erscheinungsbild der Applikation aufzeichnen.		
Abhängigkeiten	[2.2] Anforderungsspezifikationen definieren		
Risiken/Probleme	R01		
3.2	Domainmodel erstellen	Team	15 Std.
Beschreibung	Domainmodel der Analyseapplikation erstellen.		
Abhängigkeiten	-		
Risiken/Probleme	R01		
3.3	System-Sequenzdiagramme erstellen	Team	10 Std.
Beschreibung	Erstellen der wichtigsten Systemsequenzdiagramme.		
Abhängigkeiten	[3.2] Domainmodel erstellen		
Risiken/Probleme	R01		
3.4	DB Struktur erstellen	Team	10 Std.
Beschreibung	Die Struktur der Datenbank definieren.		
Abhängigkeiten	[3.2] Domainmodel erstellen		
Risiken/Probleme	R01		
3.5	Dokumentation	Team	20 Std.
Beschreibung	Erstellen der Dokumentation der Analyse.		
Abhängigkeiten	-		
Risiken/Probleme	R01		

8.7.3.3.4 Design

4	Design		32 Std.
4.1	Klassendiagramm entwerfen	Team	10 Std.
Beschreibung	Das Klassendiagramm entwerfen.		
Abhängigkeiten	[3.2] Domainmodel erstellen [3.3] System-Sequenzdiagramme erstellen		
Risiken/Probleme	R01		
4.2	Logische Architektur entwerfen	Team	14Std.
Beschreibung	Die Logische Architektur entwerfen. (Schichten Model)		
Abhängigkeiten	[3.2] Domainmodel erstellen [3.3] System-Sequenzdiagramme erstellen [3.4] DB Struktur erstellen		
Risiken/Probleme	R01		
4.3	DB modellieren	Team	8 Std.
Beschreibung	Die Tabellen mit allen Attributen definieren.		
Abhängigkeiten	[3.4] DB Struktur erstellen		
Risiken/Probleme	R01		

8.7.3.3.5 Implementation

5	Implementation		311 Std.
5.1	Webserver einrichten	Team	5 Std.
Beschreibung		Einen Webserver für die Applikation einrichten..	
Abhängigkeiten		-	
Risiken/Probleme		R01	
5.2	Benutzerverwaltung implementieren	fbernet	10 Std.
Beschreibung		Benutzerverwaltung implementieren	
Abhängigkeiten		[4.1] Klassendiagramm entwerfen [4.2] Logische Architektur entwerfen [4.3] DB modellieren	
Risiken/Probleme		R01, R03, R04	
5.3	Kernapplikation implementieren	Team	20 Std.
Beschreibung		Die Kernapplikation implementieren	
Abhängigkeiten		[4.1] Klassendiagramm entwerfen [4.2] Logische Architektur entwerfen [4.3] DB modellieren	
Risiken/Probleme		R01, R03, R04	
5.4	Batchmode-Applikation implementieren	jfeuerst	6 Std.
Beschreibung		Die Batchmode-Applikation implementieren	
Abhängigkeiten		[4.1] Klassendiagramm entwerfen [4.2] Logische Architektur entwerfen [4.3] DB modellieren	
Risiken/Probleme		R0, R03, R04	
5.5	Flugkostenabfrage implementieren	fbernet	30 Std.
Beschreibung		Abfrage für die Flugkosten implementieren.	
Abhängigkeiten		[4.1] Klassendiagramm entwerfen [4.2] Logische Architektur entwerfen [4.3] DB modellieren	
Risiken/Probleme		R01, R02, R03, R04,R05	
5.6	Hotelkostenabfrage implementieren	jfeuerst	30 Std.
Beschreibung		Abfrage für die Hotelkosten implementieren.	
Abhängigkeiten		[4.1] Klassendiagramm entwerfen [4.2] Logische Architektur entwerfen [4.3] DB modellieren	

Risiken/Probleme		R01, R02, R03, R04,R05	
5.7	Datenanalyse Flugkosten	Team	93 Std.
Beschreibung		Die Statistik und Reporting Funktionen für die Flugkosten implementieren	
Abhängigkeiten		[5.5] Flugkostenabfrage implementieren	
Risiken/Probleme		R01, R03, R04	
5.8	Statistik Hotelkosten implementieren	Team	62 Std.
Beschreibung		Die Statistik und Reporting Funktionen für die Hotelkosten implementieren	
Abhängigkeiten		[5.6] Hotelkostenabfrage implementieren	
Risiken/Probleme		R01, R03, R04	
5.9	Webapplikation implementieren	Team	27 Std.
Beschreibung		Eine Webseite für den Zugriff auf die Applikation erstellen.	
Abhängigkeiten		-	
Risiken/Probleme		R01, R03, R04	
5.10	DB erstellen	Team	11 Std.
Beschreibung		Die Datenbank erstellen und einrichten	
Abhängigkeiten		[4.3] DB modellieren	
Risiken/Probleme		R01, R03, R04	
5.11	Prototyp Google Web Driver	Team	10 Std.
Beschreibung		Ein lauffähiges Testprogramm mit Google WebDriver	
Abhängigkeiten		-	
Risiken/Probleme		R01, R03, R04	
5.12	Prototyp Google Web Driver	Team	14 Std.
Beschreibung		Ein lauffähiges Testprogramm mit Google WebDriver	
Abhängigkeiten		-	
Risiken/Probleme		R01, R03, R04	

8.7.3.3.6 Test

6	Test		16 Std.
6.1	Testcases durchführen	Team	16 Std.
Beschreibung	Die Applikation anhand der spezifizierten Tests testen. Tests durch Betreuer und Externe.		
Abhängigkeiten	[2.3] Testfälle spezifizieren [5.6] Webinterface implementieren		
Risiken/Probleme	R01		

8.7.3.3.7 Dokumentation

7	Dokumentation		114 Std.
7.1	Benutzeranleitung	Team	24 Std.
Beschreibung	Erstellen einer Benutzeranleitung.		
Abhängigkeiten	[5.3] Packet Loss Analyse implementieren		
Risiken/Probleme	R01		
7.2	Technische Dokumentation	Team	90 Std.
Beschreibung	Erstellen der Dokumentation zur Studienarbeit.		
Abhängigkeiten	-		
Risiken/Probleme	R01		

8.7.3.3.8 Studium Technologien

8	Studium Technologien		62 Std.
8.1	Google Webdriver	Team	16 Std.
Beschreibung	Informationen zu Google Webdriver sammeln. Funktionalität testen und beschreiben.		
Abhängigkeiten	-		
Risiken/Probleme	R01		
8.2	API Fluginformationen	Team	22 Std.
Beschreibung	Analysieren der API der Fluginformationenanbieter		
Abhängigkeiten	-		
Risiken/Probleme	R01		
8.2	.NET & MS SQL	Team	24 Std.
Beschreibung	Informationen zu .NET und MS SQL sammeln.		
Abhängigkeiten	-		
Risiken/Probleme	R01		

8.7.3.3.9 Sitzungen

9	Sitzungen		30 Std.
9.1	Meeting mit Betreuer	Team	24 Std.
Beschreibung	Meeting mit Betreuer des Projekts.		
Abhängigkeiten	-		
Risiken/Probleme	R01		
9.2	Protokoll schreiben	Team	6 Std.
Beschreibung	Protokolle der Meetings mit den Betreuern schreiben.		
Abhängigkeiten	-		
Risiken/Probleme	R01		

8.7.3.4 Infrastruktur (Infrastructure)

Hardware:

- SVN-Server der HSR
- private Notebooks der einzelnen Teammitglieder
- Für jeden Mitarbeiter einen Computer (wird von der HSR bereitgestellt).
- Webserver mit IIS (virtueller Server bereitgestellt durch HSR)

Software:

- Visual Studio 2008
- MS SQL Server 2005
- Tortoise SVN
- Adobe Acrobat
- Microsoft Office
- Enterprise Architect

8.7.3.5 Qualitätsmassnahmen (Quality Management)

8.7.3.5.1 Reviews mit Betreuer

Die regelmässigen Reviews mit den Verantwortlichen und den Betreuern sollen ebenfalls dazu beitragen, dass das Resultat der Studienarbeit möglichst zufriedenstellend sein wird.

8.7.3.5.2 Software

8.7.3.5.3 Versionenmanagement

Der gesamte Entwicklungsprozess wird mit einem Versionsmanagements-System (SVN) durchgeführt. So sind alle Teammitglieder immer auf dem neusten Stand und allenfalls können Änderungen rückgängig gemacht werden.

8.7.3.5.4 Code Reviews

Allfällige Unschönheiten oder Fehler können diskutiert und allenfalls behoben werden. Dies ermöglicht eine fortlaufende Verbesserung des Coding Styles.

8.7.4 Risiken

Risk ID	Risiko	Auswirkung	Massnahme	Kosten der Massnahmen in h	Max. Schaden in h	Wahrscheinlichkeit des Eintreffens	Gewichteter Schaden in h	Priorität
R01	Ausfall von Teammitgliedern wegen Krankheit	Verzögerungen durch Abhängigkeiten	Keine statische Aufgabenzuteilung von Anfang an	2	20	25%	5	4
R02	Kein Zugriff auf Fluginformationssysteme	Ausweichen auf das Auslesen von Webseiten	Mehrere Systeme anschauen	6	30	90%	27	1
R03	Fehlerhafte Planung, zu enger Terminplan	Gewisse Funktionale Anforderungen können nicht erfüllt werden	Gewisse funktionale Anforderungen als optional deklarieren	1	30	30%	9	5
R04	Anforderungen ändern sich	- Planung muss angepasst werden - Aufwand ändert sich	Präzise Anforderungsspezifikation	6	20	10%	2	3
R05	Parameter der Buchungsplattform ändern sich	Parameter müssen im Code angepasst werden	Parameter in Config File auslagern, so müssen sie nur einmal geändert werden	6	20	30%	6	2
	Total Kosten in Arbeitspaketen enthalten			21				
	Total Rückstellungen						49	

Die Massnahmen dienen dazu das Risiko zu vermindern. Die Wahrscheinlichkeiten des Auftretens sind Schätzungen. Sie geben das Restrisiko an, welches trotz dem Ergreifen der entsprechenden Massnahme übrig bleibt.

Das grösste Risiko sehen wir darin, dass wir keinen Zugriff auf ein Fluginformationssystem bekommen werden. Denn diese werden uns kaum kostenlosen Zugang gewähren und die Preise sind viel zu hoch.

Darum werden wir auf das Auslesen von Webseiten von Buchungsplattformen zurückgreifen müssen, um die benötigten Informationen zu erhalten.

Dieses Vorgehen wird uns etwas Zeit kosten, da das Parsen aus den Webseiten nicht gerade die Entwicklerfreundlichste Variante darstellt. Zudem ergibt sich bei diesem Vorgehen ein weiteres Risiko. Sollten die Buchungsplattformbetreiber ihr Seitenlayout umstellen, müssen in unserer Applikation diese Anpassungen ebenfalls vorgenommen werden.

Um die Risiken abzudecken, werden Reserven eingeplant. Die letzte Projektwoche wird zu diesem Zweck als Puffer reserviert. Also sollten die eigentlichen Arbeiten bereits eine Woche vor dem Abgabetermin erledigt sein.

8.7.5 Prozessmodell

Bei der Software Entwicklung wurde nach dem RUP Prozessmodell vorgegangen, wie es an der HSR anhand der Module SE1, SE2 vermittelt wurde.

8.8 Projektmonitoring

In diesem Kapitel werden das Zeitmanagement und die Projektplanung interpretiert. Es wird vor allem auf Abweichungen zwischen Soll und Ist, sowie deren Gründe eingegangen.

8.8.1 Soll-Ist-Zeit-Vergleich

8.8.1.1 Projektphasen

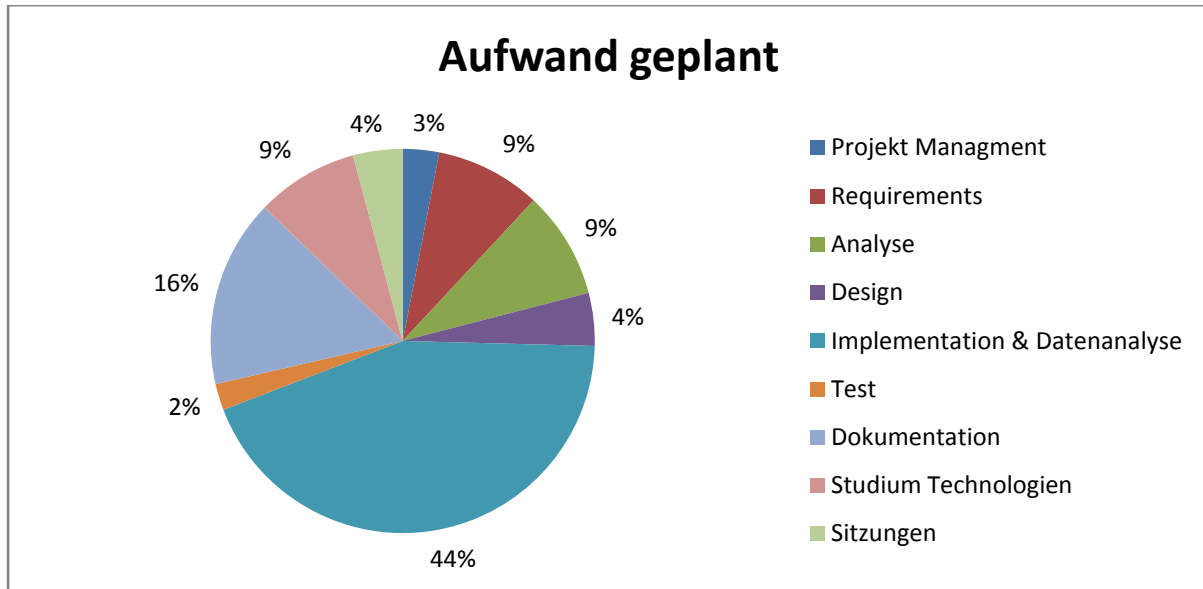


Abbildung 60: Geplanter Aufwand

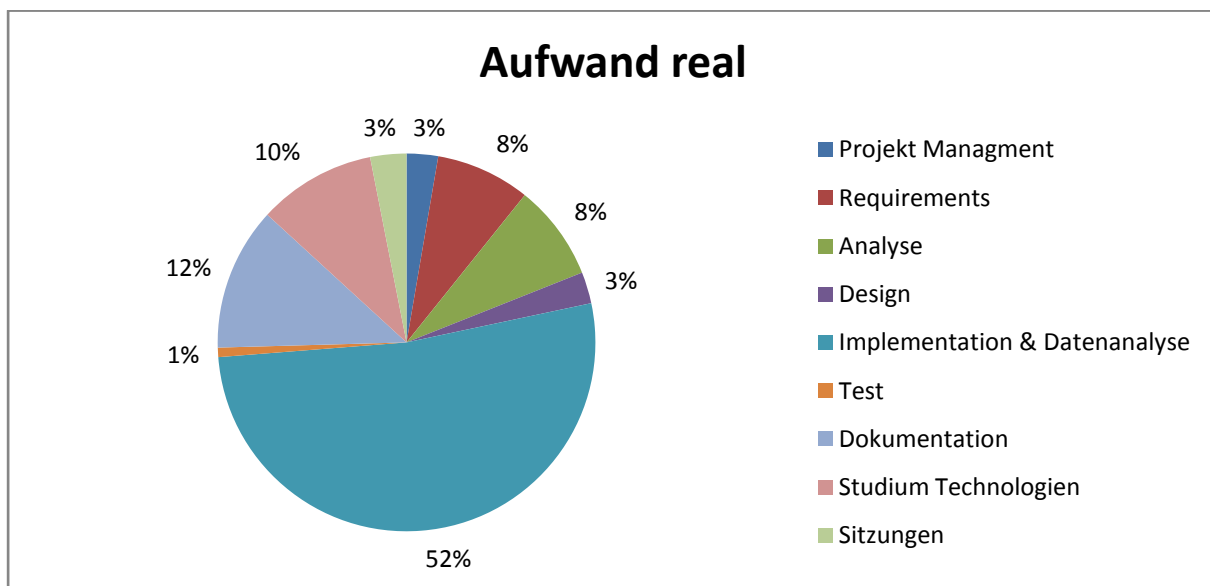


Abbildung 61: Realer Aufwand

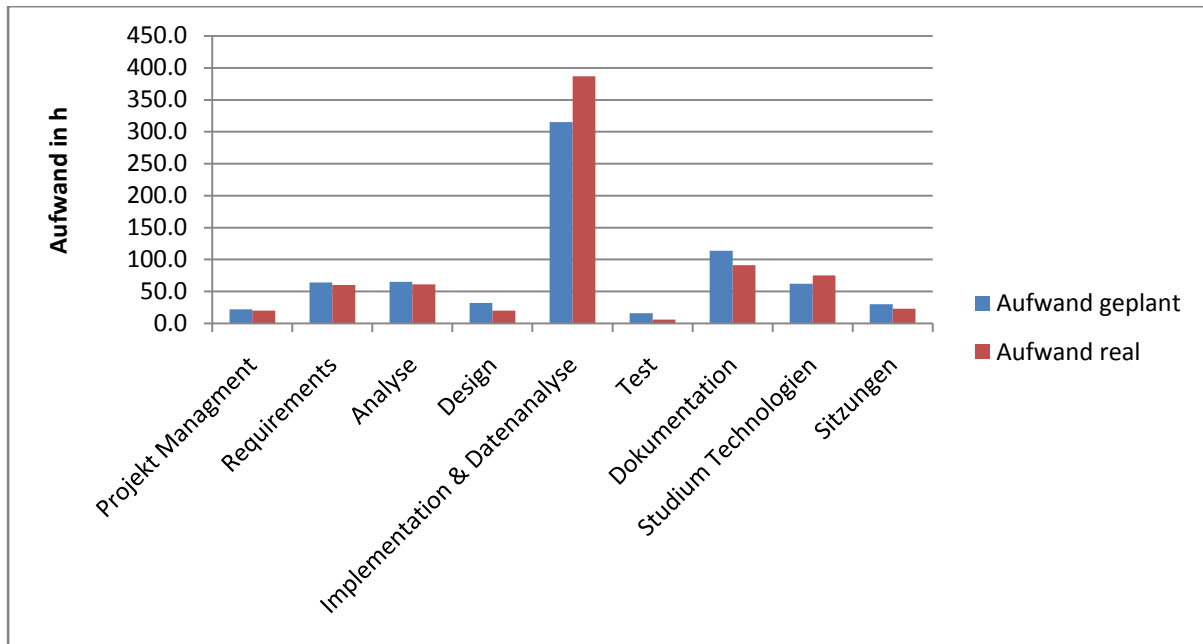


Abbildung 62: Aufwand nach Stunden (Ist/Soll)

Als erstes fällt einem beim Betrachten der Diagramme auf, dass die Implementation mit Abstand am meisten Zeit benötigte. Im Weiteren kann gesagt werden, dass die Aufwandsschätzungen eigentlich ziemlich nahe an die realen Aufwände herankommen.

Die grössten Abweichungen entstanden bei der Implementation und der Dokumentation. Während der Projektarbeit zeigte sich, dass die Schätzungen für die Implementation zu optimistisch ausfielen und in Wirklichkeit mehr Zeit dafür aufgewendet werden musste. Der Hauptgrund dafür ist, dass am Anfang des Projektes die genaue Aufgabenstellung noch nicht klar war und zuerst noch definiert werden musste, was genau gemacht werden soll. Damit ergaben sich immer wieder kleine Änderungen nach dem die Aufwandschätzung bereits gemacht wurde.

Obwohl wir bei der Implementation mehr Zeit brauchten als geplant, hätten wir schlussendlich auch noch mehr Zeit brauchen können um die Webapplikation zu optimieren.

8.8.1.2 Zeitmanagement

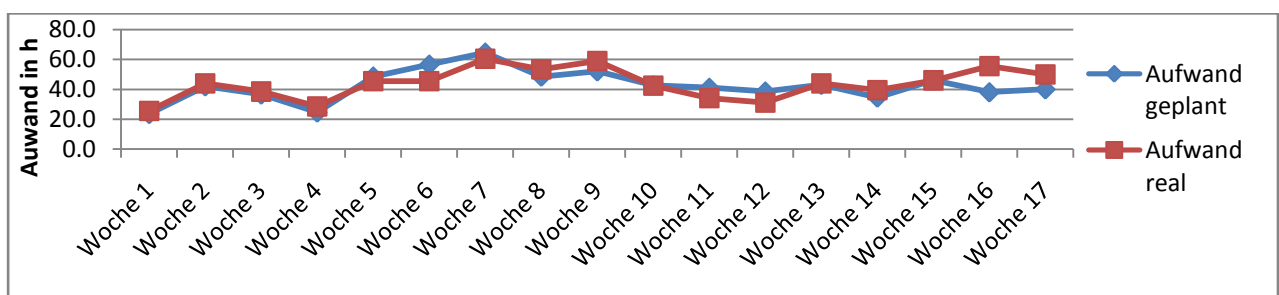


Abbildung 63: Zeitmanagement Ist/Soll

Bei der Planung gingen wir von einem Aufwand von ca. 21 Stunden pro Woche und Person aus, das heisst also ca. 42 Stunden pro Woche und ca. 720 Stunden für das ganze Projekt. Wie man an der Kurve im Diagramm erkennen kann, wurden am Anfang des Projekts die Wochenvorgaben meist nicht erreicht. Das lag daran, dass in dieser Phase die Requirements definiert werden mussten, welche mit den Betreuern beim wöchentlichen Meeting zuerst besprochen und angepasst werden mussten, bevor mit der Implementation begonnen werden konnte. Dieser Umstand wurde aber auch in etwa so eingeplant. Sonst konnte die Planung eigentlich gut eingehalten werden. Ausser am Schluss des Projektes musste mehr investiert werden als geplant um die Dokumentation zu vervollständigen und die Webapplikation fertig zu stellen.

8.9 Softwaredokumentation

8.9.1 Tutorial Webapplikation

Für die Webapplikation wurde eine DynDNS Domain eingerichtet. Sie ist mindestens bis zur Präsentation der Arbeit am 21.08.2009 über <http://eu-meetings.dnsalias.com> benutzbar.

An dieser Stelle werden die Funktionen der der Webapplikation kurz erklärt.

8.9.1.1 Login

EU-Meetings

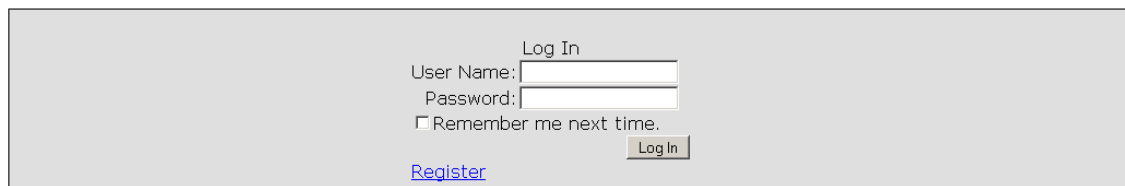


Abbildung 64: Webapplikation – Login

Beim Aufrufen der Domain wird das Login-Fenster geladen, falls man noch nicht angemeldet ist.

Hat man bereits einen Useraccount kann man sich hier anmelden. Um einen neuen Benutzer anzulegen klickt man auf „Register“ um auf die Registrierungsseite zu gelangen.

Für Demonstrationszwecke kann der Demo-Benutzer benutzt werden.

Username: demo

Password: demo

8.9.1.2 Neuen Benutzer registrieren

EU-Meetings



Abbildung 65: Webapplikation - Registrierung eines neuen Benutzers

Hat man noch keinen Useraccount um sich anzumelden, kann man einen neuen Benutzer registrieren.

Die verlangten Eingaben sind zu machen. Sobald alle Eingaben erfasst wurden, wird mit einem Klick auf „Create User“ der neue Benutzer angelegt. Ab sofort kann man sich mit den gerade erstellten Benutzerinformationen beim Login-Fenster anmelden.

8.9.1.3 Home

EU-Meetings

Logged in as: demo [Logout](#)

Home | [My Requests](#) | [New Request](#) |

Welcome

- See the results of your Requests in the "My Requests" section
- Create a new Request in the "New Request" section

Abbildung 66: Webapplikation – Home

Nach dem erfolgreichen Anmelden kommt man auf die Home-Seite. Diese ist hier in der Abbildung 66 noch ziemlich leer. Hier könnten noch Willkommens-Meldungen und weitere Informationen über die Anwendung erstellt werden.

Über das Hauptmenü kann zu den verschiedenen Funktionen der Webapplikation navigiert werden. Unter „My Requests“ erhält man die Übersicht über die eigenen, bereits ausgeführten Abfragen. Unter „New Request“ kann eine neue Flug- und/oder Hoteldatenabfrage erstellt werden.

8.9.1.4 Abfragen Übersicht

EU-Meetings

Logged in as: demo [Logout](#)

Home | [My Requests](#) | [New Request](#) |

My Requests

Flight Requests

Request created	Request executed	Meeting Destination	Meeting start date	Meeting end date	Nr. of participants	
11.06.2009 14:19:28	11.06.2009 14:25:27	madrid	04.09.2009	10.09.2009	12	Flights
11.06.2009 14:19:29	11.06.2009 14:25:40	rom	04.09.2009	10.09.2009	12	Flights
11.06.2009 14:19:29	11.06.2009 14:25:52	lissabon	04.09.2009	10.09.2009	12	Flights
11.06.2009 14:28:48	11.06.2009 14:33:43	madrid	24.12.2009	30.12.2009	10	Flights
11.06.2009 14:28:48	11.06.2009 14:33:55	kopenhagen	24.12.2009	30.12.2009	10	Flights
11.06.2009 14:28:47	11.06.2009 14:33:41	amsterdam	24.12.2009	30.12.2009	10	Flights

Hotel Requests

Request created	Request executed	Meeting Destination	Meeting start date	Meeting end date	Nr. of participants	
11.06.2009 14:25:53	11.06.2009 14:26:27	dublin	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:52	11.06.2009 14:26:30	dublin	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:49	11.06.2009 14:26:32	dublin	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:53	11.06.2009 14:26:34	berlin	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:52	11.06.2009 14:26:39	berlin	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:49	11.06.2009 14:26:39	berlin	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:49	11.06.2009 14:26:42	amsterdam	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:52	11.06.2009 14:26:45	amsterdam	10.12.2009	17.12.2009	14	Hotels
11.06.2009 14:25:53	11.06.2009 14:26:50	amsterdam	10.12.2009	17.12.2009	14	Hotels

Combined Requests

Request created	Request executed	Meeting Destination	Meeting start date	Meeting end date	Nr. of participants			
11.06.2009 14:14:28	11.06.2009 14:17:52	barcelona	20.08.2009	25.08.2009	9	Flights	Hotels	Combined
11.06.2009 14:14:28	11.06.2009 14:18:09	london	20.08.2009	25.08.2009	9	Flights	Hotels	Combined
11.06.2009 14:14:28	11.06.2009 14:25:26	paris	20.08.2009	25.08.2009	9	Flights	Hotels	Combined

Abbildung 67: Webapplikation - Anfragen-Übersicht

Die bereits ausgeführten Abfragen sind in drei Kategorien aufgeteilt.

Unter „Flight Requests“ sind die Abfragen, bei denen nur die Flugdaten abgefragt wurden aufgelistet.

Unter „Hotel Requests“ sind die Abfragen, bei denen nur die Hoteldaten abgefragt wurden aufgelistet.

Unter „Combined Requets“ sind die Abfragen, bei denen sowohl Flug- als auch Hoteldaten abgefragt wurden aufgelistet.

Alle Spalten der Tabellen können mit einem Klick auf die gewünschte Spalten-Überschrift sortiert werden.

Über die Links in der hintersten Kolonne können die Resultate der Abfrage angezeigt werden.

Bei den kombinierten Abfragen können die Flug- und Hotelresultate jeweils einzeln oder in einer kombinierten Resultate-Ansicht geöffnet werden.

8.9.1.5 Neue Anfrage

Logged in as: demo [Logout](#)

[Home](#) | [My Requests](#) | [New Request](#) |

New Request

Request details

Choose desired Request Type

☒ Flight
☐ Hotel

Choose desired Booking Platform

☒ Ebookers.de

Meeting details

Choose desired start date and end-date

Meeting Start Date (dd.mm.yyyy) Meeting End Date (dd.mm.yyyy)

Choose desired destination-cities

Meeting City #1

Meeting City #2

Meeting City #3

Participant details

Choose Nr of Participants Cities (1-100)

Participant City #1 Nr. of Participants

Participant City #2 Nr. of Participants

Participant City #3 Nr. of Participants

Participant City #4 Nr. of Participants

Abbildung 68: Webapplikation - Neue Anfrage

Hier wird ein Formular für das Erstellen einer neuen Abfrage angezeigt.

Man wählt den gewünschten Abfrage-Typ (Flug, Hotel oder eine kombinierte Abfrage) und die Buchungsplattform, welche für die Berechnung der Preise verwendet werden soll.

Danach gibt man die Informationen zum Meeting ein. Das Start- und Enddatum des Meetings, sowie den Durchführungsort. Hier können bis zu drei Meeting-Durchführungsorte gewählt werden. Damit ist ein vergleichen der Resultate möglich.

Als nächstes muss die Anzahl der verschiedenen Herkunfts-Städte der Meeting-Teilnehmer angegeben werden. Daraufhin wird die Anzahl der Detailfelder für die genaueren Angaben zu den Teilnehmern angepasst.

Pro Teilnehmer-Herkunftsland kann anschliessend noch die genaue Anzahl der Teilnehmer aus diesem Ort angegeben werden, falls es sich um mehr als einen handelt.

Mit einem Klick auf „Start request“ wird die Anfrage dem Server übermittelt und so fern möglich, sofort ausgeführt. Sobald die Resultate berechnet worden sind, werden sie in der „My Request“ Sektion in der entsprechenden Kategorie aufgelistet.

8.9.1.6 Flugdaten-Resultate

EU-Meetings

Logged in as: demo Logout			
Home My Requests New Request			
Flight Details			
Meeting city:	barcelona		
Meeting start date:	20.08.2009		
Meeting end date:	25.08.2009		
Meeting duration days:	5		
Total Nr of participants:	9		
Total flight costs:	1'531.54 €		
Participant hometown	Cheapest flight price	Nr. of prticipants	Total flight costs
amsterdam	143.00 €	2	286.00 €
zürich	192.63 €	4	770.52 €
berlin	158.34 €	3	475.02 €

Abbildung 69: Webapplikation - Flugdaten-Resultate

Unter dieser Ansicht werden die Flugkosten-Resultate einer Flugdaten- oder einer kombinierten Abfrage angezeigt.

8.9.1.7 Hoteldaten-Resultate

EU-Meetings

Logged in as: demo Logout		
Home My Requests New Request		
Hotel Details		
Meeting city: barcelona		
Meeting start date: 20.08.2009		
Meeting end date: 25.08.2009		
Meeting duration days: 5		
Nr of participants: 9		
3 Star Hotels	4 Star Hotels	5 Star Hotels
Nr of hotels: 28	Nr of hotels: 30	Nr of hotels: 4
Price per night per participant:	Price per night per participant:	Price per night per participant:
Min 55.05 €	Min 48.20 €	Min 114.99 €
Max 182.50 €	Max 190.08 €	Max 168.77 €
Avg 91.96 €	Avg 106.43 €	Avg 142.55 €
Total costs per participant:	Total costs per participant:	Total costs per participant:
Min 275.24 €	Min 241.00 €	Min 574.95 €
Max 912.50 €	Max 950.40 €	Max 843.83 €
Avg 459.78 €	Avg 532.13 €	Avg 712.76 €
Total cost for all participants:	Total cost for all participants:	Total cost for all participants:
Min 2'477.16 €	Min 2'169.00 €	Min 5'174.55 €
Max 8'212.50 €	Max 8'553.60 €	Max 7'594.47 €
Avg 4'137.99 €	Avg 4'789.13 €	Avg 6'414.86 €

Abbildung 70: Webapplikation - Hoteldaten-Resultate

Unter dieser Ansicht werden die Hotelkosten-Resultate einer Hoteldaten- oder einer kombinierten Abfrage angezeigt.

8.9.1.8 Kombinierte-Abfrage Resultate

EU-Meetings

Logged in as: demo [Logout](#)

[Home](#) | [My Requests](#) | [New Request](#) |

Flight Details

Meeting city:	barcelona
Meeting start date:	20.08.2009
Meeting end date:	25.08.2009
Meeting duration days:	5
Total Nr of participants:	9
Total flight costs:	1'531.54 €

Participant hometown	Cheapest flight price	Nr. of prticipants	Total flight costs
amsterdam	143.00 €	2	286.00 €
zürich	192.63 €	4	770.52 €
berlin	158.34 €	3	475.02 €

Hotel Details

Meeting city:	barcelona
Meeting start date:	20.08.2009
Meeting end date:	25.08.2009
Meeting duration days:	5
Nr of participants:	9

3 Star Hotels

Nr of hotels:	28
Price per night per participant:	
Min	55.05 €
Max	182.50 €
Avg	91.96 €
Total costs per participant:	
Min	275.24 €
Max	912.50 €
Avg	459.78 €
Total cost for all participants:	
Min	2'477.16 €
Max	8'212.50 €
Avg	4'137.99 €

4 Star Hotels

Nr of hotels:	30
Price per night per participant:	
Min	48.20 €
Max	190.08 €
Avg	106.43 €
Total costs per participant:	
Min	241.00 €
Max	950.40 €
Avg	532.13 €
Total cost for all participants:	
Min	2'169.00 €
Max	8'553.60 €
Avg	4'789.13 €

5 Star Hotels

Nr of hotels:	4
Price per night per participant:	
Min	114.99 €
Max	168.77 €
Avg	142.55 €
Total costs per participant:	
Min	574.95 €
Max	843.83 €
Avg	712.76 €
Total cost for all participants:	
Min	5'174.55 €
Max	7'594.47 €
Avg	6'414.86 €

Unter dieser Ansicht werden die Flugkosten-Resultate und die Hotelkosten-Resultate einer kombinierten Abfrage angezeigt.

9 Schlussfolgerungen

Aus unserer Sicht wurden die Zielsetzungen der Arbeit mehrheitlich erreicht.

Die Webapplikation zur Berechnung der Meeting-Kosten wurde implementiert und ist funktionstüchtig. Aus zeitlichen Gründen konnte allerdings nicht mehr viel am Layout und der Bedienungsfreundlichkeit der Website gefeilt werden. Hier könnte man aber noch viel herausholen und die erarbeitete Lösung besser präsentieren und die Bedienung für den Benutzer noch komfortabler gestalten. Auch auf Sicherheitsaspekte wurde nicht das grösste Augenmerk gelegt, da die Applikation nicht besonders Sicherheitskritisch ist.

Als Fazit, die geforderte Funktionalität wurde implementiert und ist lauffähig aber zum Teil noch verbesserungsfähig.

Die Analyse der Flug- und Hoteldaten auf Preisbeeinflussende Faktoren wurde durchgeführt und die Resultate ausgewertet. Um die Analyse durchzuführen wurden automatisiert grosse Datenmengen mit Flug- und Hotelinformationen gesammelt.

Bei der Untersuchung der Daten stellte sich jedoch teilweise heraus, dass die Zeitspanne für welche die Resultate gesammelt wurden und/oder die Anzahl der Resultate für einige Auswertungen zu gering war. Denn mit der Datensammlung wurde nach dem ersten Drittel der gesamten Projektdauer begonnen.

Durch diesen Umstand war es schlicht unmöglich, z.B. eine aussagekräftige Analyse über das saisonale Preisverhalten (während eines ganzen Jahres) durchzuführen. Trotzdem wurden versucht, mit den vorhandenen Daten die interessantesten Auswertungen durchzuführen und die Ergebnisse zu beschreiben. Auch wenn die ganz bahnbrechenden neuen Erkenntnisse ausblieben, konnten trotzdem die eine oder andere interessante Verhaltensweise bei den Preisen erkannt werden.

Die Datensammlung durch das Parsen von Resultatseiten der Buchungsplattformen ist sicherlich nicht die Wunschvariante. Aber der Zugang zu einem Web-Service eines Buchungsinformationssystems ist finanziell für dieses Projekt nicht zu tragen. Beim Parsen bleibt halt das Risiko, dass sich die Struktur der Zielseite ändert und die Parser keine Resultate mehr liefern bis sie angepasst werden. Dieser Fall ist schliesslich auch noch eingetreten. Zwei Tage vor dem Abgabetermin hat Ebookers.de die Struktur ihrer Hotel-Resultatseite verändert und unser Parser fand natürlich keine Ergebnisse mehr. Der Parser musste nochmals angepasst werden.

Die Resultate dieser Arbeit sind nicht wirklich spektakulär, aber die Arbeit die dahintersteckt ist nicht zu unterschätzen. Nur schon der Aufbau des Data-Warehouses für die Analyse der Daten sowie die Untersuchungen mit den MS SQL Analysis Services waren sehr zeitintensiv. Ebenso muss beachtet werden, dass mit der eigentlichen Implementation und den Analysen erst etwa nach einem Drittel der gesamten Projektdauer begonnen werden konnte. Zuerst mussten die genauen Anforderungen erfasst werden und durch manuelle Forschung die Faktoren gesucht werden, welche bei der Analyse genauer angeschaut werden sollen.

Aus unserer Sicht haben wir die Projektziele erreicht.

10Abbildungsverzeichnis

Abbildung 1: Übersicht der erarbeiteten Funktionalitäten.....	8
Abbildung 2: Übersicht der entwickelten Kernapplikation	13
Abbildung 3: Übersicht der entwickelten Webapplikation	14
Abbildung 4: Mögliche Auswertung „Einfluss Wochentag“	17
Abbildung 5: Mögliche Auswertung „Einfluss Durchführungsdatum (saisonal)“	18
Abbildung 6: Mögliche Auswertung „Einfluss der Zeitspanne zwischen Buchung und Reise“	18
Abbildung 7: Mögliche Auswertung: „Einfluss Meeting-Dauer“	19
Abbildung 8: Mögliche Auswertung „Anzahl Hotels in verschiedenen Preissegmenten“	20
Abbildung 9: Mögliche Auswertung „Übersicht Hoteldetails“	20
Abbildung 10: Veranschaulichung eines Data-Cube für die Datenanalyse	21
Abbildung 11: Unbearbeitete Export-Tabelle	22
Abbildung 12: Bearbeitete Export-Tabelle	22
Abbildung 13: Einfluss von Faktoren auf die Flugkosten	24
Abbildung 14: Startort Lissabon - Tiefste Flugreisekosten in Abhängigkeit der Aufenthaltsdauer	24
Abbildung 15: Startort Lissabon - Durchschn. Flugreisekosten in Abhängigkeit der Aufenthaltsdauer	25
Abbildung 16: Startort Madrid - Tiefste Flugreisekosten in Abhängigkeit der Aufenthaltsdauer	25
Abbildung 17: Startort Madrid - Durchschn. Flugreisekosten in Abhängigkeit der Aufenthaltsdauer	25
Abbildung 18: Startort Zürich - Tiefste Flugreisekosten in Abhängigkeit der Aufenthaltsdauer	26
Abbildung 19: Startort Zürich - Durchschn. Flugreisekosten in Abhängigkeit der Aufenthaltsdauer	26
Abbildung 20: Startort Lissabon - Saisonale Entwicklung der Flugreisekosten.....	27
Abbildung 21: Startort Madrid - Saisonale Entwicklung der Flugreisekosten.....	27
Abbildung 22: Startort Zürich - Saisonale Entwicklung der Flugreisekosten	27
Abbildung 23: Startort Lissabon - Einfluss der Zeitspanne zwischen Buchung und Reise auf die Flugreisekosten	28
Abbildung 24: Startort Madrid - Einfluss der Zeitspanne zwischen Buchung und Reise auf die Flugreisekosten	29
Abbildung 25: Startort Zürich - Einfluss der Zeitspanne zwischen Buchung und Reise auf die Flugreisekosten	29
Abbildung 26: 3-Sterne Hotels - Einfluss der Aufenthaltsdauer auf die Hotelkosten.....	30
Abbildung 27: 5-Sterne Hotels - Einfluss der Aufenthaltsdauer auf die Hotelkosten.....	30
Abbildung 28: 3-Sterne Hotels - Einfluss des Aufenthaltsdatums auf die Hotelkosten.....	31
Abbildung 29: 3-Sterne Hotels - Einfluss der Zeitspanne zwischen Buchung und Aufenthalt auf die Hotelkosten	31
Abbildung 30: London - Übersicht des Hotelangebotes	32
Abbildung 31: Madrid - Übersicht des Hotelangebotes.....	32
Abbildung 32: Zürich - Übersicht des Hotelangebotes.....	33
Abbildung 33: Systemsequenz Diagramm Kernapplikation	42
Abbildung 34: Systemsequenz Diagramm Webapplikation	43
Abbildung 35: Domainmodel	44
Abbildung 36: Übersicht der logischen Architektur	46
Abbildung 37: Konzeptionelles Klassendiagramm Problem Domain	48
Abbildung 38: Konzeptionelles Klassendiagramm Data Objects.....	51
Abbildung 39: Datenbankmodel Hauptdatenbank	55
Abbildung 40: Datenbankmodel Task-Datenbank.....	57
Abbildung 41: Datenbankmodel Data Warehouse	58
Abbildung 42: Sequenzdiagramm Flugdaten Abfrage.....	60
Abbildung 43: Sequenzdiagramm Hoteldaten Abfrage.....	60
Abbildung 44: Sequenzdiagramm kombinierte Abfrage	61
Abbildung 45: Übersicht Packages	62
Abbildung 46: Klassendiagramm Batchmode	63

Abbildung 47: Klassendiagramm Request.....	64
Abbildung 48: Klassendiagramm Flight Parser	65
Abbildung 49: Klassendiagramm Hotel Parser	66
Abbildung 50: Klassendiagramm Parser.....	67
Abbildung 51: Klassendiagramm Common	68
Abbildung 52: Klassendiagramm LinqToSql Klassen Hauptdatenbank	69
Abbildung 53: Klassendiagramm LinqToSql Klassen Task-Datenbank	70
Abbildung 54: Datenbankdiagramm Hauptdatenbank	71
Abbildung 55: Datenbankdiagramm Task-Datenbank	72
Abbildung 56: Datenbankdiagramm Data-Warehouse	73
Abbildung 57: Ablaufsteuerung der Datenintegration.....	74
Abbildung 58: Datenfluss beim Füllen einer Dimension-Table	75
Abbildung 59: Datenfluss beim Füllen einer Fact-Table	76
Abbildung 60: Geplanter Aufwand.....	88
Abbildung 61: Realer Aufwand.....	88
Abbildung 62: Aufwand nach Stunden (Ist/Soll)	89
Abbildung 63: Zeitmanagement Ist/Soll.....	89
Abbildung 64: Webapplikation – Login	90
Abbildung 65: Webapplikation - Registrierung eines neuen Benutzers	90
Abbildung 66: Webapplikation – Home	91
Abbildung 67: Webapplikation - Anfragen-Übersicht	91
Abbildung 68: Webapplikation - Neue Anfrage	92
Abbildung 69: Webapplikation - Flugdaten-Resultate	93
Abbildung 70: Webapplikation - Hoteldaten-Resultate	93

11 Glossar

Begriff	Bedeutung
API	API (für engl. application programming interface, deutsch: „Schnittstelle zur Anwendungsprogrammierung“)
ASP.NET	ASP.NET (Active Server Pages .NET) ist eine serverseitige Technologie von Microsoft zum Erstellen von Webanwendungen auf Basis des Microsoft-.NET-Frameworks.
Batch-Applikation	Entwickelte Applikation, welche automatisch nach auszuführenden Abfrage-Aufträgen sucht und diese der Kernapplikation zur Ausführung übergibt.
Batchbetrieb	sequentielle Kommandoabarbeitung
C#	C# (lies engl. c sharp) ist eine vom Softwarehersteller Microsoft im Rahmen seiner .NET-Strategie entwickelte Programmiersprache. C# ist bei ECMA und ISO als Standard registriert.
Data-Mining	Unter Data Mining (englisch für „Datenschürfen“) versteht man die systematische Anwendung von Methoden, die meist statistisch-mathematisch begründet sind, auf einen Datenbestand mit dem Ziel der Mustererkennung.
Data-Warehouse	Ein Data-Warehouse bzw. Datenlager ist eine zentrale Datensammlung (meist eine Datenbank), deren Inhalt sich aus Daten unterschiedlicher Quellen zusammensetzt. Die Daten werden von den Datenquellen in das Data-Warehouse geladen und dort vor allem für die Datenanalyse und zur betriebswirtschaftlichen Entscheidungshilfe in Unternehmen langfristig gespeichert.
Global Distribution Systems	Computerreservierungssysteme (CRS), international auch Global Distribution Systems (GDS) genannt, bündeln in grossen Rechenzentren Informationen über Preise, Verfügbarkeiten und Buchungsmöglichkeiten von Pauschalreisen, Flügen, Hotels, Mietwagen, Fähren, Kreuzfahrten, Bahnen, Bussen und anderen Produkten.
Kernapplikation	Entwickelte Applikation, welche die Hotel- und Fluganfragen an ein online Buchungsportal sendet und die Resultate in einer Datenbank speichert.

LINQ	LINQ (Abkürzung für Language INtegrated Query) ist eine Komponente von Microsofts .NET-Framework zur Abfrage von Datenquellen wie Datenbanken oder XML-Dateien.
MS SQL Server	Der Microsoft SQL Server (abgekürzt MS-SQL-Server) ist ein relationales Datenbankmanagementsystem auf der Microsoft-Windows-Plattform.
.NET	.NET (sprich dotnet) ist eine von Microsoft entwickelte Software-Plattform.
SQL	SQL (das Kürzel für Structured Query Language), ist eine Datenbanksprache zur Definition, Abfrage und Manipulation von Daten in relationalen Datenbanken.
XPath	Die XML Path Language (XPath) ist eine vom W3-Konsortium entwickelte Abfragesprache, um Teile eines XML-Dokumentes zu adressieren.

12Literaturverzeichnis

- [1] „Pricing Strategies of low-costs airlines: the Ryanair case“, AirlineBookingsUniBergamo.pdf
- [2] Projekt-Website HTML Agility Pack , <http://www.codeplex.com/htmlagilitypack>
- [3] Informationen zu XPath, http://www.zvon.org/xxl/XPathTutorial/General_ger/examples.html
- [4] C# Thread Pool Helper, <http://thevalerios.net/matt/2008/06/threadpoolqueueuserworkitem-with-multiple-arguments/>
- [5] Lernprogramme für den Umgang mit den MS SQL Server Tools, [http://msdn.microsoft.com/de-de/library/ms169620\(SQL.90\).aspx](http://msdn.microsoft.com/de-de/library/ms169620(SQL.90).aspx)
- [6] Informationen über die MS SQL Data-Mining Algorithmen, [http://msdn.microsoft.com/de-de/library/ms175595\(SQL.90\).aspx](http://msdn.microsoft.com/de-de/library/ms175595(SQL.90).aspx)
- [7] Gespräch vom 08.04.2009 mit Prof. Hansjörg Huser, INS an der HSR
- [8] ZhaoHui Tang, Jamie MacLennan, “Data Mining with SQL Server 2005“, Wiley Publishing Inc., 2005
- [9] Gespräch vom 12.05.2009 mit Manuel Bauer, INS an der HSR
- [10] Informationen zu ASP.NET, <http://www.asp101.com>
- [11] Domain für Webapplikation, <http://www.dyndns.com/>
- [12] Informationen zu den Flughäfen Europas, <http://www.flugplandaten.de/html/europa/europa.htm>

13Anhang

13.1 Zugriff Webapplikation

Für die Webapplikation wurde eine DynDNS Domain eingerichtet. Sie ist mindestens bis zur Präsentation der Arbeit am 21.08.2009 über <http://eu-meetings.dnsalias.com> benutzbar.

Es kann ein neuer eigener Benutzer registriert werden oder der folgende User benutzt werden.

Benutzername: demo

Passwort: demo

Der Webserver befindet sich auf einem virtuellen Server (vServer) der HSR. Dieser wird nach Ablauf der Bachelorarbeit gelöscht.

13.2 CD-ROM

Es wird eine CD-ROM zu der Arbeit erstellt.

Diese enthält den Source-Code sowie die verwendeten Dokumente und Bilder.