

# Pflichtenheft

*Version 1.3*



**Projekt:** Reisekosten-Abrechnungsmodul

**Auftraggeber:** **Günter Brackly**  
Fachhochschule Kaiserslautern  
Fachbereich Informatik und Mikrosystemtechnik  
Standort Zweibrücken  
Amerikastr. 1  
Büro: Gebäude H Raum H246  
Tel.: 06332 - 914 332  
e-mail: guenter.brackly@fh-kl.de

**Auftragnehmer:** **Team Confidentia**  
Andreas Hoff  
Jens Mertz  
Julian Thomé  
Matthias Juchmes  
Marco Hack

## Dokumentenhistorie

Version	Datum	Status	Autor	Geprüft	Kommentar
1.0	13.03.2008	Vorläufig	Hoff		Dokument erstellt
1.1	20.03.2008	Vorläufig	Hoff		Abgrenzungskriterien um Private Reisen ergänzt, Internen Meilenstein hinzugefügt
1.1	23.03.2008	Freigegeben	Hoff	Mertz	Überprüft
1.2	14.04.2008	Vorläufig	Hoff		Indikatoren überarbeitet
1.2	14.04.2008	Freigegeben	Hoff	Mertz	Überprüft
1.3	13.06.2008	Vorläufig	Hoff		Abgleich mit finalem Projektstand
1.3	16.06.2008	Freigegeben	Hoff	Mertz	Überprüft

## Inhaltsverzeichnis

1 Vorbemerkung.....	4
2 Zielbestimmung.....	5
2.1 Muss-Kriterien.....	5
2.2 Kann-Kriterien.....	5
2.3 Abgrenzungskriterien.....	6
3 Vorgehen.....	7
3.1 Vorgehensmodell.....	7
3.1.1 Vertikaler Prototyp.....	7
3.1.2 Core.....	8
3.1.3 Release Candidate.....	8
3.1.4 Final.....	8
3.2 Meilensteine.....	8
3.2.1 Meilenstein „Projektplan und Pflichtenheft“ (26.03.2008).....	9
3.2.2 Meilenstein „Systemdesign / Vertikaler Prototyp“ (16.04.2008).....	9
3.2.3 Meilenstein „Core“ (05.05.2008).....	10
3.2.4 Interner Meilenstein (21.05.2008).....	10
3.2.5 Meilenstein „Release Candidate“ (04.06.2008).....	10
3.2.6 Meilenstein „Übergabe und Abnahme“ (18.06.2008).....	10
3.3 Work Breakdown Structure.....	12
3.4 Fortschrittskontrolle: Indikatoren.....	13
4 Produkteinsatz.....	14
4.1 Anwendungsbereich und Produktumgebung.....	14
4.1.1 FH-Info.....	14
4.2 Zielgruppen.....	14
4.2.1 Zukünftige Benutzer.....	14
4.2.2 Zukünftige Entwickler.....	15
4.3 Workflow.....	16
5 Technologie und Architektur.....	17
6 Produktfunktionen.....	18
7 Benutzeroberfläche.....	20
7.1 Menüstruktur.....	20
7.2 Layout der einzelnen Seiten.....	20
8 Fachliches Klassendiagramm.....	21
9 Dokumentation.....	22
9.1 Entwickler-Dokumentation.....	22
9.2 Quellcode-Dokumentation.....	22
9.3 Installationsanleitung.....	22
9.4 Online-Hilfe.....	22
10 Qualitätsanforderungen.....	23
11 Referenzierte Dokumente.....	25

## 1 Vorbemerkung

Das Projekt „Reisekosten-Abrechnungsmodul“ hat aus Sicht des Confidentia-Teams viele Ansprechpartner mit unterschiedlichen Vorkenntnissen und Hintergründen. Daher führen wir an dieser Stelle kurz auf, welche Abschnitte unserer Meinung nach für welchen Ansprechpartner interessant sind.

### *Kapitel 2: Zielbestimmung*

Die Zielbestimmung beschreibt einleitend die Projektziele, sie ist für jeden interessant.

### *Kapitel 3: Vorgehen*

Dieses Kapitel beschreibt das Vorgehensmodell, die einzelnen Entwicklungsstadien und Meilensteine, sowie die Work Breakdown Structure und die Fortschrittskontrolle. Es ist daher insbesondere für den Auftraggeber und das Project Office interessant.

### *Kapitel 4: Produkteinsatz*

An dieser Stelle wird der Anwendungsbereich, die Benutzergruppen sowie der grundsätzliche Ablauf (Workflow) erläutert. Dieses Kapitel ist somit interessant für unsere Ansprechpartner des FH-Info-Systems sowie die Ansprechpartner der Verwaltung.

### *Kapitel 5: Technologie und Architektur*

Dieses Kapitel beschreibt die eingesetzte Technologie und Architektur der Software.

### *Kapitel 6: Produktfunktionen*

An dieser Stelle werden die Produktfunktionen in Form von Use Case Diagrammen dargestellt. Dieser Abschnitt ist daher eher interessant für diejenigen, die mit der Beschreibungssprache UML vertraut sind.

### *Kapitel 7: Benutzeroberfläche*

An dieser Stelle wird die vorgeschlagene Menüstruktur beschrieben. Die Vorschläge für die einzelnen Seiten wurde in das Dokument [Objektorientierte Analyse] ausgelagert. Die Benutzeroberfläche ist interessant für alle zukünftigen Benutzer der Software.

### *Kapitel 8: Fachliches Klassendiagramm*

Hier werden die Produktdaten anhand eines fachlichen Klassendiagramms beschrieben. Das Kapitel ist daher interessant für diejenigen, die mit der Beschreibungssprache UML vertraut sind – insbesondere die Ansprechpartner des FH-Info-Systems, da auf bereits vorhandene Klassen von FH-Info zugegriffen wird.

### *Kapitel 9: Dokumentation*

Dieses Kapitel zeigt auf, welche Arten von Dokumentation erstellt werden und ist daher für den Auftraggeber und die Ansprechpartner des FH-Info-Systems interessant.

### *Kapitel 10: Qualitätsanforderungen*

Dieser Abschnitt nennt die Ziele und Maßnahmen des Qualitätsmanagements und ist daher insbesondere für den Auftraggeber und unseren QM-Ansprechpartner interessant.

## 2 Zielbestimmung

Für das Verwaltungssystem FH-Info der Fachhochschule Kaiserslautern (siehe dazu *Kapitel 4.1.1: FH-Info*) „soll ein Abrechnungsmodul für Reisekosten der Mitarbeiter der FH erstellt werden. Dazu muss die bisherige Verwaltung, die komplett auf Papier basiert, umgestellt werden. Alle Mitarbeiter sind bereits mit ihren Stammdaten im FH-Info System erfasst. Darauf aufbauend soll die Möglichkeit geboten werden, dass jeder Mitarbeiter seine Dienstreisedaten über Formulare eingeben kann und die Verwaltungsmitarbeiter diese dann in ihrer Sicht dieses Moduls weiter bearbeiten und fertig stellen können. Dabei sollen die Verwaltungsmitarbeiter durch das Modul so weit wie möglich durch z. B. automatische Berechnungen, Wertetabellen etc. unterstützt werden.

Das Modul soll alle Mitarbeiter der FH, die einen Dienstreiseantrag stellen und abrechnen wollen, und die Verwaltungsmitarbeiter, die diese Anträge bearbeiten müssen, weitestgehend unterstützen. Die Verwaltungsmitarbeiter sollen dabei durch automatisierte Berechnungen, pflegbare Stammdaten, Wertetabellen etc. entlastet werden.

Um mögliche Fehler zu minimieren soll das Modul den Mitarbeiter da, wo es geht, durch Vorgabe der Prozessreihenfolge leiten. Da keine intensive Mitarbeiterschulung durchgeführt werden soll, muss die Bedienung intuitiv möglich sein (d. h. ohne lesen umfangreicher Benutzerhandbücher). Online-Hilfen sollten wo erforderlich vorhanden sein.“ Quelle: [Lastenheft]

Im Folgenden werden die Muss- und Kann-Kriterien definiert. Die detaillierten Anwendungsfälle sind in *Kapitel 6: Produktfunktionen* zu finden.

### 2.1 Muss-Kriterien

- Objektorientierte Analyse für das gesamte System
- System- und Datenbank-Entwurf für das gesamte System
- Implementierung
  - Sicht des Dienstreisenden: Alle Use Cases außer „Filter anwenden“
  - Sicht des Dekanats: Alle Use Cases außer „Filter anwenden“

### 2.2 Kann-Kriterien

- Implementierung
  - Sicht des Dienstreisenden: Use Case „Filter anwenden“
  - Sicht des Dekanats: Use Case „Filter anwenden“
  - Sicht der Verwaltung: Alle Use Cases

## **2.3 Abgrenzungskriterien**

Folgende Kriterien sollen bewusst nicht erreicht werden.

(1) Sicht des Dekanats: Aggregierte Darstellung

Im Lastenheft wird für die Sicht des Dekanats eine aggregierte Darstellung der bisherigen Kosten für alle Angehörigen des Fachbereichs gefordert. Nach Rücksprache mit Herrn Brackly und Herrn Walter wurde diese Anforderung gestrichen.

(2) Genehmigung bei Drittmittelprojekten

Ein aus Drittmittelprojekten finanzierter Antrag auf Reisekostenvergütung muss durch den Geldgeber genehmigt werden. Dies geschieht auch in Zukunft außerhalb des FH-Info-Systems und muss daher beim Entwurf der Software nicht berücksichtigt werden.

(3) Private Reisen

In der Praxis können Dienstreisen mit privaten Reisen verknüpft werden. Dieser Sonderfall soll beim Entwurf der Software nicht berücksichtigt werden.

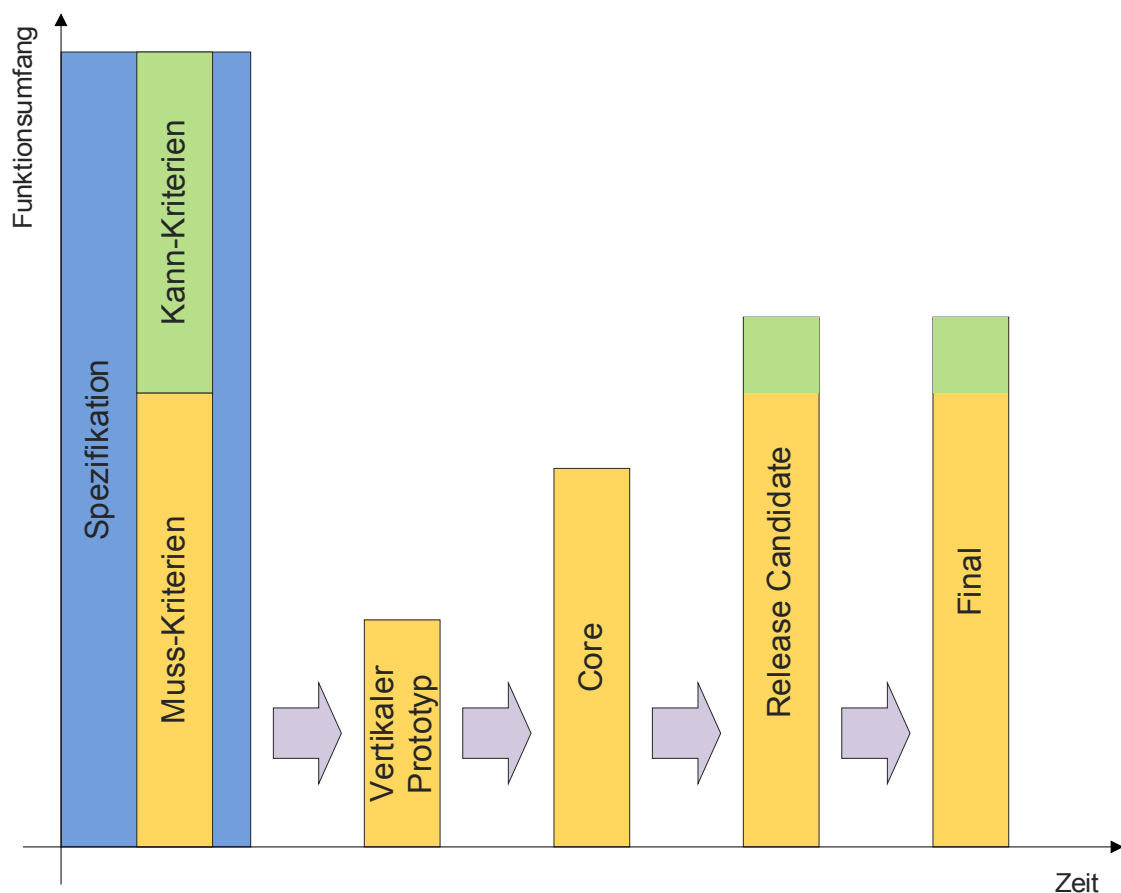
(4) Benutzerhandbuch

Das Modul soll intuitiv bedienbar sein, es soll neben der Online-Hilfe kein Benutzerhandbuch erstellt werden.

## 3 Vorgehen

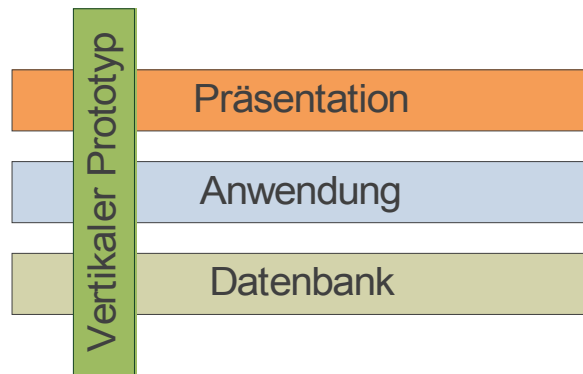
### 3.1 Vorgehensmodell

Es wird zunächst eine vollständige Spezifikation erstellt, die sowohl die Muss- als auch die Kann-Funktionalitäten abdeckt. Anschließend wird diese Spezifikation inkrementell umgesetzt. Die Entwicklungsschritte sind mit den Meilensteinen (siehe *Kapitel 3.2: Meilensteine*) verknüpft und richten sich nach den im Dokument [Kickoff Präsentation] festgelegten Anforderungen.



#### 3.1.1 Vertikaler Prototyp

Der vertikale Prototyp realisiert nur einen kleinen Ausschnitt der Funktionalität und zeigt an diesem Beispiel die Verwendung der Drei-Schichten-Architektur mit Datenbank-, Anwendungs- und Präsentationsebene.



Der vertikale Prototyp wird folgende Funktionalität bieten.

- (1) Ein Dienstreisender erstellt einen Antrag auf Genehmigung einer Dienstreise und speichert ihn.
- (2) Ein Dienstreisender lässt sich die gespeicherten Anträge auf Genehmigung einer Dienstreise anzeigen.

### 3.1.2 Core

Der Core beinhaltet die Kernfunktionalitäten des Systems und ist ausreichend stabil, damit Anwender das System testen können.

Welche konkreten Funktionen implementiert werden sollen hängt vom Projektverlauf ab und wird bis zum Meilenstein „Vertikaler Prototyp“ mit dem Auftraggeber geklärt.

### 3.1.3 Release Candidate

Der Release Candidate ist die letzte Testversion der Software. Nach der Implementierung des Release Candidates erfolgt ein Code Freeze, d. h. ab dieser Version werden keine weiteren Produktfunktionen implementiert. Von nun an werden lediglich vom Auftraggeber geforderte Änderungen und Erweiterungen umgesetzt, sowie Fehler behoben und die Online-Hilfe vervollständigt.

Welche konkreten Funktionen implementiert werden sollen hängt vom Projektverlauf ab und wird bis zum Meilenstein „Core“ mit dem Auftraggeber geklärt.

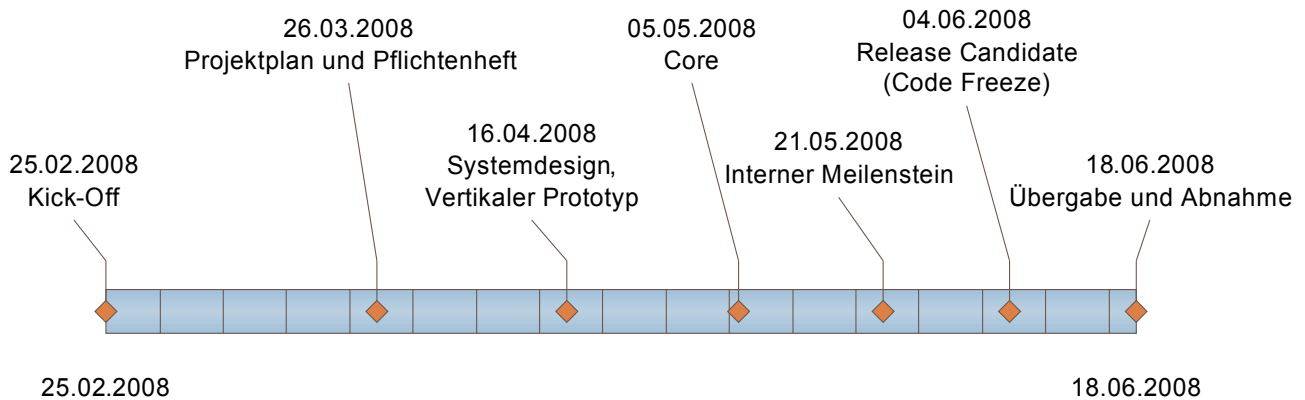
### 3.1.4 Final

Der Final ist die Endversion, die dem Kunden übergeben wird. Hinsichtlich des Funktionsumfangs entspricht der Final dem Release Candidate, allerdings ist der Final stabil, d. h. alle Tests wurden durchgeführt und alle bekannten Bugs sind behoben.



## 3.2 Meilensteine

Die Präsentationen beim Kunden definieren die Meilensteine des Projektes. Zusätzlich hat das Team Confidentia einen internen Meilenstein, 9 Tage vor dem Code Freeze, festgelegt.



### 3.2.1 Meilenstein „Projektplan und Pflichtenheft“ (26.03.2008)

- Es berichten zumindest: PM, QM, DBA, SYSA
- Abzugeben sind:
  - Projektplan
  - Statusbericht
  - Installationsnachweis für ein Issue-Tracking System (Bugzilla, ...)
  - QM-Dokumentation:
    - Lenkung von Dokumenten
    - Testplan (verbindlich und vorher im Review (G. Schmidt))
  - Pflichtenheft incl.
    - Use Case-Diagramm und –Beschreibungen
    - Fachliches Klassendiagramm
- Bis zu diesem Termin müssen Sie außerdem sämtliche Voraussetzungen für die Entwicklung geschaffen haben
  - CVS-Nutzung

Quelle: [Kickoff Präsentation]

### 3.2.2 Meilenstein „Systemdesign / Vertikaler Prototyp“ (16.04.2008)

- Es berichten alle!
- Abzugeben sind:

- *Statusbericht*
- *Nachweis des aktiven Einsatzes des Issue-Tracking-Systems*
- *Testspezifikationen*
- *Systementwurf*
  - *Übersichtsdiagramm Systemarchitektur*
  - *Datenbankentwurf*
  - *Spezifikation der Systemkomponenten und Schnittstellen*
  - *Entwurf der Benutzungsoberfläche*
- *Quelltexte*
- *Ggf. aktualisierte Versionen der Dokumente aus der vorangehenden Phase*

Quelle: [Kickoff Präsentation]

### **3.2.3 Meilenstein „Core“ (05.05.2008)**

- *Es berichten mindestens: PM, Tester, SYSA, DBA*
- *Abzugeben sind:*
  - *Statusbericht*
  - *Testdokumentation incl. JUnit-Tests*
  - *Quelltexte*
  - *Ggf. aktualisierte Versionen der Dokumente aus den vorangehenden Phasen*

Quelle: [Kickoff Präsentation]

### **3.2.4 Interner Meilenstein (21.05.2008)**

Der interne Meilenstein definiert den Termin, an dem das Team entscheidet, welche Produktfunktionen der Release Candidate (und somit auch der Final) enthalten wird.

### **3.2.5 Meilenstein „Release Candidate“ (04.06.2008)**

- *Es berichten alle !*
- *Abzugeben sind:*
  - *Statusbericht*
  - *Testdokumentation incl. JUnit-Tests*
  - *Quelltexte*
  - *Ggf. aktualisierte Versionen der Dokumente aus den vorangehenden Phasen*

Quelle: [Kickoff Präsentation]

### **3.2.6 Meilenstein „Übergabe und Abnahme“ (18.06.2008)**

- *Abzugeben sind:*
  - *Statusbericht (Abschlussbericht)*
  - *Alle erstellten Dokumente und Produkte*
    - *Aus den Quelltexten muss hervorgehen, von wem welche Klasse erstellt wurde*
  - *Endgültige Versionen der Dokumente aus den vorangehenden Phasen, die dem tatsächlich realisierten System entsprechen*
  - *CD-ROM mit allen Quellen und Dokumenten sowie einem installationsfähigen System incl. Setup mit allen benötigten Programmen und Bibliotheken*

Quelle: [Kickoff Präsentation]

### **3.3 Work Breakdown Structure**

#### **(1) Planung**

1. Projektplan
2. Qualitätsplan
3. Testplan

#### **(2) Analyse**

1. Benutzerrollen
2. Workflow als Aktivitätsdiagramm
3. Use Case Diagramme und Beschreibungen
4. Fachliches Klassendiagramm
5. GUI-Prototypen

#### **(3) Entwurf**

1. Systemarchitektur
  1. Übersichtsdiagramm Systemarchitektur
  2. Spezifikation der Systemkomponenten und Schnittstellen
  3. Vollständiges Klassendiagramm
  4. GUI-Entwurf
2. DB-Entwurf
  1. ERM
  2. RDM
  3. Normalisierung
  4. Constraints, Trigger, Stored Procedures

#### **(4) Implementierung**

1. Vertikaler Prototyp
2. Core
3. Release Candidate
4. Final

#### **(5) Tests**

1. JUnit-Tests
2. Integrationstests
3. Abnahmetest

- (6) Dokumentation
  - 1. Online-Hilfe
  - 2. Installationsanleitung
- (7) Installation und Abnahme
  - 1. CD zusammenstellen

### 3.4 Fortschrittskontrolle: Indikatoren

Um den Fortschritt des Projektes zu kontrollieren, werden die folgenden Indikatoren verwendet, eingetragen sind die SOLL-Werte für die einzelnen Meilensteine.

		<b>Planung 26.03.08</b>	<b>Prototyp 16.04.08</b>	<b>Core 14.05.08</b>	<b>RC 04.06.08</b>	<b>Final 18.06.08</b>
<b>LOC gesamt</b>	SOLL	0	1200	5100	6500	6700
	IST	0	1973	4973	10947	12213
<b>Java Statements</b>	SOLL	0	600	1600	2000	2100
	IST	0	607	1540	4159	4681
<b>Lines of Documentation</b>	SOLL	0	150	400	500	525
	IST	0	158	480	1188	1268
<b>Verhältnis Statements / Kommentarzeilen</b>	SOLL	-	4 : 1	4 : 1	4 : 1	4 : 1
	IST	-	3,8 : 1	3,2 : 1	3,5 : 1	3,7 : 1
<b>Anzahl implementierte Klassen und Interfaces</b>	SOLL	0	6	16	27	27
	IST	0	11	16	28	28
<b>Implementierte Testmethoden</b>	SOLL	0	0	15	20	20
	IST	0	0	11	24	24
<b>Gestartete Testmethoden</b>	SOLL	0	0	15	20	20
	IST	0	0	11	24	24
<b>Offene Testmethoden</b>	SOLL	0	0	5	0	0
	IST	0	0	9	0	0
<b>Anzahl offener Fehler</b>	SOLL	0	0	10	20	0
	IST	0	0	2	4	0

Die Auswahl der Indikatoren orientiert sich am Dokument [Kickoff Präsentation].

## 4 Produkteinsatz

### 4.1 Anwendungsbereich und Produktumgebung

#### 4.1.1 FH-Info

„Das FH-Info-System ist ein Portalsystem der FH Kaiserslautern mit zahlreichen Anwendungen für Studierende und Mitarbeiter. Die aktuell eingesetzten Module decken unter anderem die Bereiche Personal- und Raumverwaltung, Stunden- und Prüfungsplanerstellung, Verwaltung von Studienarbeiten und Forschungsprojekten, ein TrafficMonitor und die Notenabfrage für die Studierenden ab.“ Quelle: [Lastenheft]

### 4.2 Zielgruppen

#### 4.2.1 Zukünftige Benutzer

Im Rahmen der objektorientierten Analyse wurden die folgenden Rollen ermittelt.

1. **Dienstreisender:** Der Dienstreisende stellt „Anträge auf Genehmigung einer Dienstreise“, und „Anträge auf Reisekostenvergütung“ nachdem er die Reise durchgeführt hat. Diese Rolle kann von jeder anderen hier aufgeführten Rolle bzw. von jedem FH- Mitarbeiter eingenommen werden.
2. **Dekanat:** Das Dekanat genehmigt „Anträge auf Genehmigung einer Dienstreise“, oder lehnt letztere ab.
3. **Verwaltung:** Die Verwaltung erhält vom Dienstreisenden „Anträge auf Genehmigung einer Dienstreise“ und „Anträge auf Reisekostenvergütung“ zur Bestätigung. Des weiteren ist sie zuständig für die Reisekostenberechnung.
  - (a) **Präsident/ Beauftragter:** ist bevollmächtigt, „Anträge auf Genehmigung einer Dienstreise“ und „Anträge auf Reisekostenvergütung“ zu genehmigen bzw. abzulehnen. Des weiteren führt er die Reisekostenberechnung durch. Der Präsident/ Beauftragter hat zwangsläufig auch alle Berechtigungen des Sachbearbeiters. Diese Rolle wird am Standort Zweibrücken von Dipl.-Verw. (FH) Johannes Kuhn ausgefüllt.
  - (b) **Sachbearbeiter:** kann „Anträge auf Genehmigung einer Dienstreise“ editieren. Des weiteren kann er Reisekostenberechnungen als sog. „Feststeller“ auf Korrektheit überprüfen. Diese Rolle wird am Standort Zweibrücken von Frau Theis ausgefüllt.
4. **Admin:** Der Admin pflegt die zum Modul zugehörigen Wertetabellen (Tagegeld, Übernachtungspauschalen, Wegstreckenentschädigung, Mitnahmeentschädigung, Kürzungsbeträge).

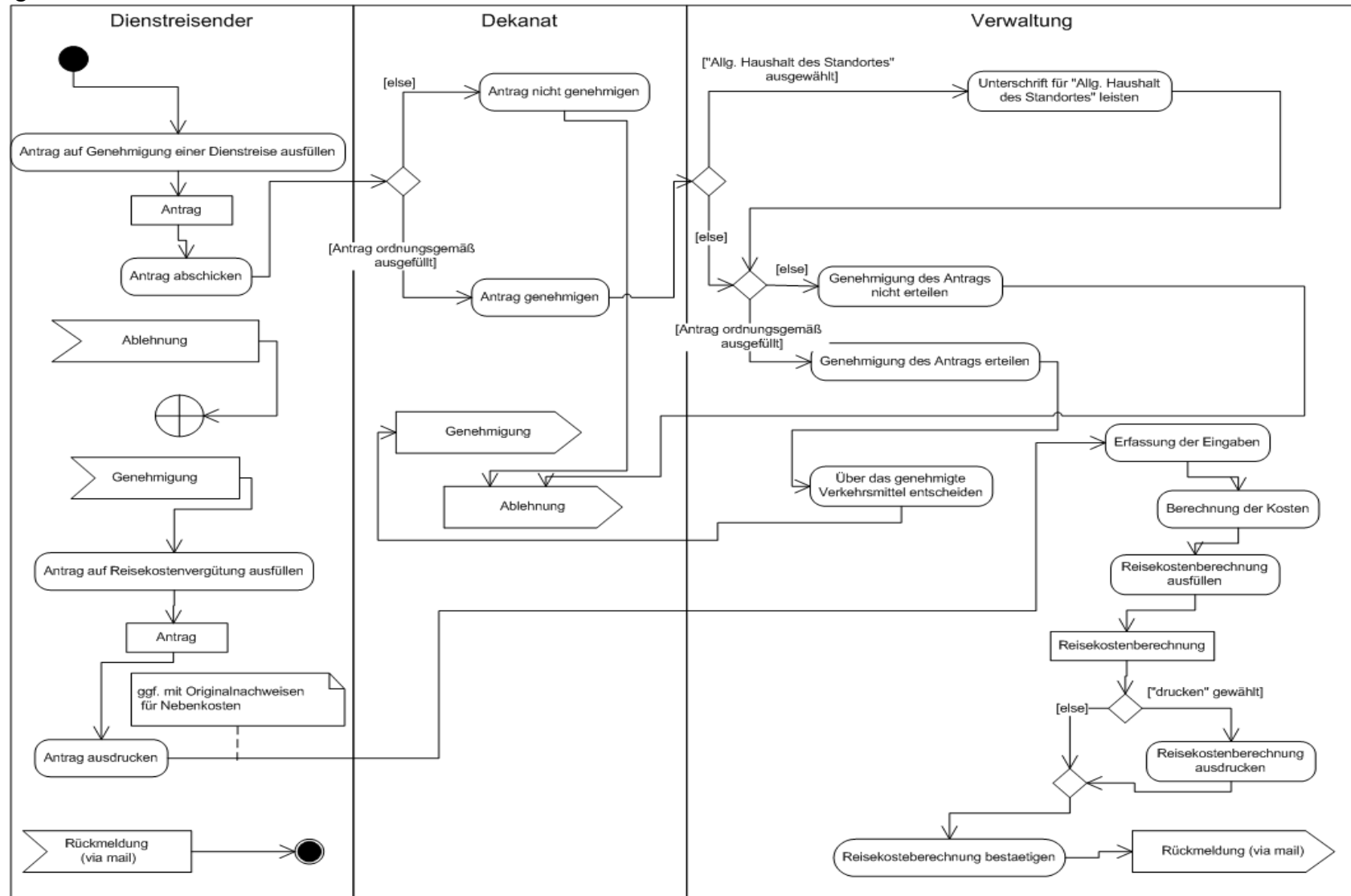
Quelle: [Objektorientierte Analyse]

### **4.2.2 Zukünftige Entwickler**

Da das Team Confidentia nach Projektabschluss keine weiteren Arbeiten am Reisekosten-Abrechnungsmodul durchführt, muss für die zukünftigen Entwickler eine Entwickler-Dokumentation (siehe *Kapitel 9.1: Entwickler-Dokumentation*) erstellt werden, damit sie die Software warten und weiterentwickeln können.

### 4.3 Workflow

Es folgt ein Aktivitätsdiagramm, das den Ablauf von Anträgen auf Genehmigung einer Dienstreise, Anträgen auf Reisekostenvergütung und die Berechnung der Reisekosten erläutert.



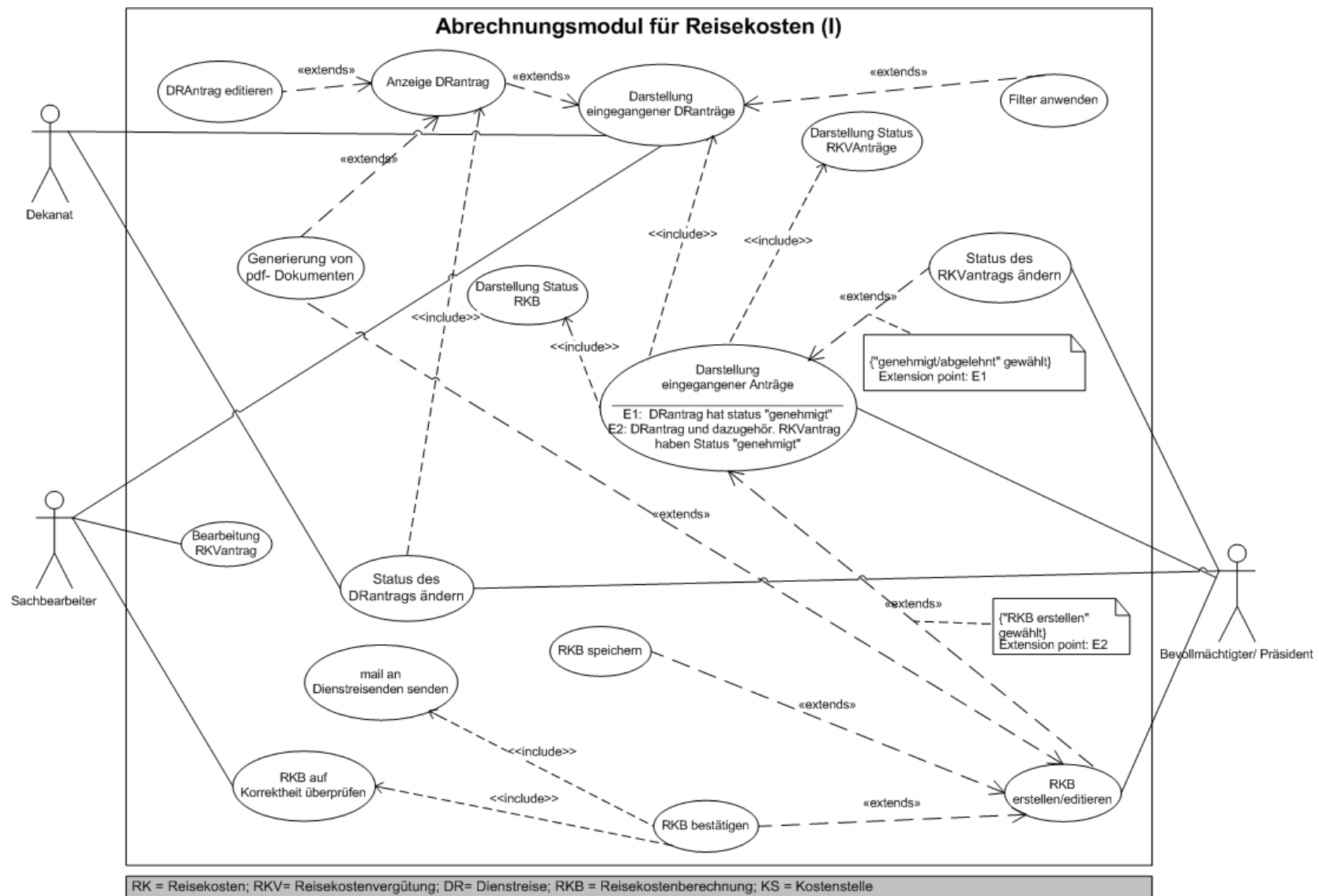


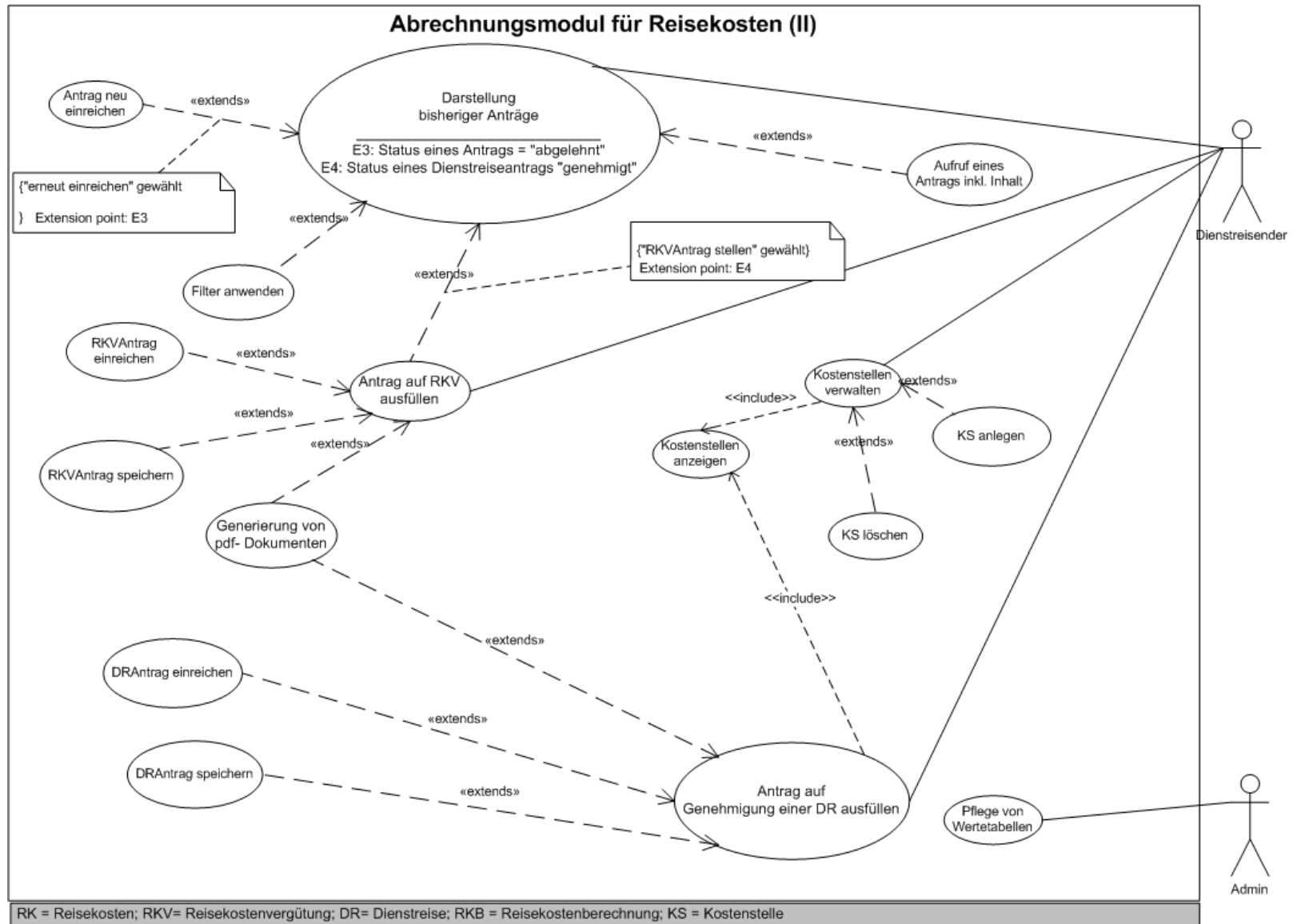
## 5 Technologie und Architektur

*„Es ist eine Client-Server-Lösung als Web-Applikation unter Java 1.5 mit Java-Servlet- und JSP-Technik und Tomcat 5.5 als Servlet Engine zu entwickeln. Zur Entwicklung ist das Struts-Konzept anzuwenden. Als Entwicklungsumgebung sollte Eclipse 3.2 oder höher oder JDeveloper 10.0.3.2 verwendet werden.*

*Die Datenbank ist MaxDB, die Kommunikation soll mittels Standard-SQL stattfinden. Die Anbindung soll über JDBC geschehen. Zugriffe auf die Datenbank aus der Anwendung (XXXDTO-Klassen) müssen über eigene Schnittstellenklassen (XXXDB-Klassen) mit statischen Methoden realisiert werden. Durch die Verwendung von Standard SQL-99 (Release 2003) SQL inklusive JDBC ist die Portierbarkeit auf andere relationale DB-Systeme sicherzustellen. Für Tests werden Spieldaten zur Verfügung gestellt.“ Quelle: [Lastenheft]*

## 6 Produktfunktionen





Die Beschreibungen der einzelnen Use Cases sind im Dokument [Objektorientierte Analyse] zu finden.

## **7 Benutzeroberfläche**

Basierend auf den ersten Gesprächen mit den zukünftigen Benutzern wurden Prototypen für die Benutzeroberflächen erstellt.

### **7.1 Menüstruktur**

Das gesamte Modul wird über das Tab „Dienstreise“ im FH-Info-Portal angesteuert werden können unabhängig von der Rolle.

Für verschiedene Rollen wird es allerdings verschiedene Unter- Tabs geben. Bisher ist folgende Struktur geplant:

#### **Dienstreisender:**

- „Dienstreise“
  - „Übersicht“ (Status von Dienstreiseanträgen betrachten, Reisekostenvergütungsanträge stellen, deren Status betrachten, den Status von Reisekostenberechnung betrachten)
  - „Dienstreiseantrag“ (hier kann ein Dienstreiseantrag gestellt werden)
  - „Kostenstellen verwalten“ (Kostenstellen anlegen, löschen)
  - „Hilfe“

#### **Zusätzlich für Dekanatsmitarbeiter:**

- „Dienstreise“
  - „Eingegangene Anträge“ (Anträge aufrufen, ablehnen, genehmigen)

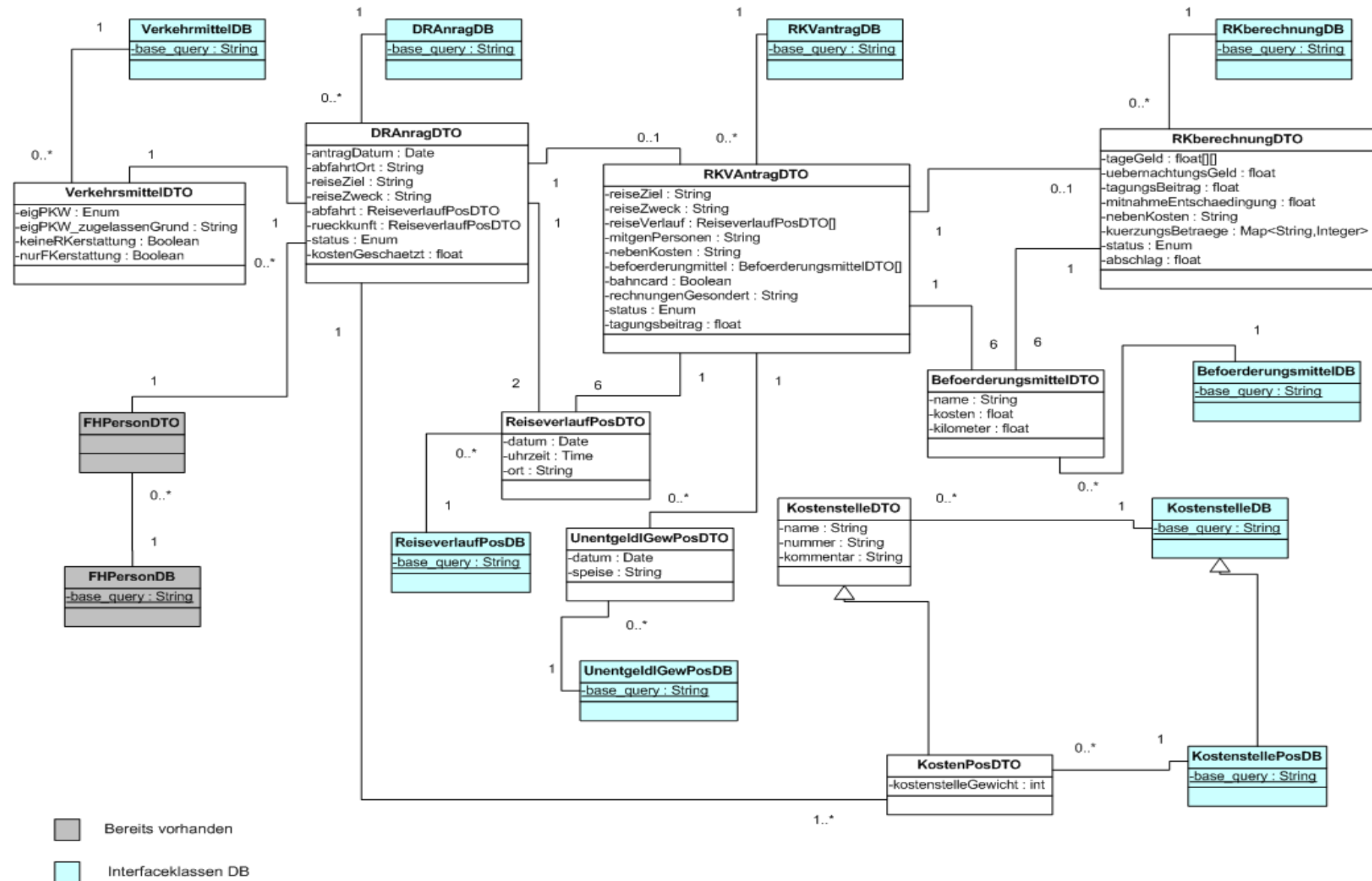
#### **Zusätzlich für Verwaltungsmitarbeiter:**

- „Dienstreise“
  - „Eingegangene Anträge“ (Anträge aufrufen, ablehnen, bestätigen, Reisekostenvergütungsanträge ablehnen, genehmigen, Reisekostenberechnungen durchführen)

### **7.2 Layout der einzelnen Seiten**

Die Vorschläge für die Layouts einzelner Seiten sind im Dokument [Objektorientierte Analyse] zu finden.

## 8 Fachliches Klassendiagramm



## **9 Dokumentation**

### **9.1 Entwickler-Dokumentation**

Mit dem Ende des Projektes gibt das Team Confidentia das Reisekosten-Abrechnungsmodul vollständig an den Auftraggeber ab. Damit zukünftige Entwickler die Software warten und weiterentwickeln können, ist eine ausführliche und selbsterklärende Dokumentation erforderlich. Diese wird im Rahmen des Systementwurfs erstellt und umfasst:

- Eine Übersicht über die gesamte Systemarchitektur
- Den Entwurf der darunter liegenden Datenbank
- Die Spezifikation der Systemkomponenten und Schnittstellen
- Den Entwurf der Benutzeroberfläche

### **9.2 Quellcode-Dokumentation**

Um die Funktionalität der einzelnen Klassen und Methoden zu beschreiben, wird Javadoc eingesetzt. Die Beschreibungen werden als Javadoc-Kommentare im Quellcode verfasst, daraus wird anschließend eine HTML-Dokumentation generiert.

### **9.3 Installationsanleitung**

Im Rahmen des Projektes wird eine Installationsanleitung für das Reisekosten-Abrechnungsmodul erstellt, durch die der Auftraggeber imstande ist, die Software selbstständig auf seinem System zu installieren.

### **9.4 Online-Hilfe**

Um die zukünftigen Benutzer zu unterstützen, wird eine Online-Hilfe erstellt.

Die Online-Hilfe wird dabei immer als Unter-Tab integriert (unterhalb des Haupt-Tabs „Dienstreise“). Hierbei sollen wir zum Einen die Erwartungswerte der einzelnen Eingabefelder erläutern, aber auch bei Bezeichnungen wie z. B. „PKW genehmigt“ vs. „PKW zugelassen“ eine Hilfestellung geben. Die konkreten Hilfetexte werden bis zum Meilenstein „Core“ ermittelt.

## 10 Qualitätsplan

### (1) Einleitung

In diesem Abschnitt werden die Qualitätsmaßnahmen erläutert, die für das Reisekostenabrechnungsmodul getroffen wurden.

### (2) Qualitätsplanung

Durch die Qualitätsplanung wird die Qualität der Software sichergestellt. In folgenden Dokumenten werden Angaben zur Qualitätsplanung gemacht:

- [Dokumentenlenkung]
- [CVS-Regeln]
- [Testplan]

### (3) Zu erstellende Dokumente

Während des Projekts werden Dokumente entstehen, die am Ende des Projektes abzugeben sind. Diese Dokumente halten sich an die Vorgaben des Kick-Off-Meetings. Welche Dokumente abzugeben sind, kann man in dem Dokument [Liste der zu erstellenden Dokumente] nachlesen.

### (4) Überprüfung der abzugebenden Dokumenten

Die Dokumente sind spätestens 2 Tage vor Abgabe zu überprüfen.

### (5) Werkzeuge und Plattformen

Entwicklungssprache	Java 1.5, JSP , Java Servlets
Datenbank	MaxDB
SDK	Eclipse Europa
Application Server	Tomcat 5.5
Entwicklungskonzept	Struts
Dokumentation	JavaDoc
Projektplanung	MS Project 2003
Modellierungswerkzeug	MS Visio 2003
Tests	JUnit
Versionsverwaltung	CVS

### (6) Teststrategie

Die Teststrategie wird im Dokument Testplan festgelegt. Dieser Testplan ist verbindlich für das Projekt und zeigt die Testmethoden auf, die in dem Projekt benutzt werden.

(7) Qualitätsanforderungen,-ziele

Ziel unseres Qualitätsmanagement ist es durch qualitätssichernde Maßnahmen alle Qualitätsmerkmale so zu erfüllen, dass der Kunde zufrieden ist. Dabei wird ein besonderes Augenmaß auf die im Lastenheft angegebenen Qualitätsmerkmale gelegt. Im Lastenheft sind folgende Qualitätsmerkmale genannt:

- Benutzbarkeit
- Abdeckung der Funktionalität
- Zuverlässigkeit
- Effizienz
- Änderbarkeit

Mit folgenden Maßnahmen soll die Qualität der Merkmalen gesichert werden:

- Benutzbarkeit: Der Kunde wird in die Entstehung der Benutzeroberfläche mit einbezogen. Schon in der Analyse-Phase werden Prototypen für die GUI erstellt, anschließend wird Feedback vom Kunden eingeholt und umgesetzt. Bei der Erstellung der Oberflächen wird darauf geachtet, dass die Eingabe-Masken nicht unübersichtlich werden, sondern dass sie leicht verstanden werden können, ohne in ein größeres Handbuch schauen zu müssen. Es sollte höchstens eine Online-Hilfe zu Rate gezogen werden. Um zu testen, ob die Bedienung des Systems intuitiv ist, werden zu gegebenen Zeitpunkten Usability Tests mit den zukünftigen Benutzern durchgeführt.
- Abdeckung der Funktionalität: Um eine Abdeckung der Funktionalität zu erreichen wird von Anfang an nach dem FH-Info-Styleguide programmiert. Damit wird verhindert, dass es später Integrationsprobleme mit dem FH-Info-Portal geben wird. Um während der Implementierung auf Probleme aufmerksam zu werden, werden regelmäßig JUnit-Tests durchgeführt. Um Fehler in der Software zu dokumentieren benutzen wir ein Issue-Tracking-System namens Mantis, so dass alle Fehler gemeldet und bearbeitet werden.
- Zuverlässigkeit: Beim Zugriff auf die Datenbank wird ein Transaktionskonzept benutzt, um die Datenbank immer in einen konsistenten Zustand zu halten. Während des Projekts wird mit JUnit-Tests sichergestellt, dass das System sich in einen zuverlässigen Zustand befindet.
- Effizienz: Um das Modul effizient nutzen zu können muss beachtet werden, dass das Datenbankmodell so minimal wie möglich gehalten wird. Auch soll im Systementwurf darauf geachtet werden, dass es keine Klassen und Use Cases gibt die man nicht benötigt.
- Änderbarkeit: Um das System änderbar zu machen werden wir von Anfang an unser komplettes Vorgehen dokumentieren. Besonders zu beachten ist dabei die Dokumentation des Datenbank- und Systementwurfs, sowie die Quellcode-Dokumentation. Ein weiterer Schritt zur Änderbarkeit besteht darin, dass Benutzeroberfläche, Programmlogik und Datenbank durch Struts und das MVC-Pattern getrennt werden.



## 11 Referenzierte Dokumente

Dokument	Version	CVS-Pfad
Lastenheft	1.1	\Lastenheft\Lastenheft Reiseabrechnung.pdf
Kickoff Präsentation	n/a	\Organisation\SWT-Projekt-SS08-Kickoff.pdf
Objektorientierte Analyse	1.1	\Spezifikation\Objektorientierte Analyse.odt
Testplan	1.0	\Organisation\Testplan.odt
Dokumentenlenkung	1.0	\Organisation\Dokumentenlenkung.odt
CVS-Regeln	1.1	\Organisation\CVS_Regeln.odt
Liste der zu erstellenden Dokumente	1.1	\Organisation\Liste der zu erstellenden Dokumenten.odt