

Berufssakademie Sachsen
Staatliche Studienakademie Dresden
Studiengang Wirtschaftsinformatik
Hans-Grundig-Straße 5
01307 Dresden

Projektbeleg zum Modul SE-W-30

Helpdeskssystem

eingereicht von:

Solveig Thomas, 09.05.1989

Jens Hetze, 28.05.1986

Martin Krockert, 20.08.1985

Johannes Schaffrath, 17.03.1989

Tag der Themenübergabe: 2. Dezember 2013

Tag der Einreichung: 10. März 2014

Inhalt

Verzeichnis der verwendeten Abkürzungen und Symbole	5
1 Einleitung	6
2 Lastenheft	7
2.1 Beschreibung des Unternehmens	7
2.2 Ziele des IT-Systems	7
2.2.1 Gründe für die Einführung des IT-Systems	7
2.2.2 Kurzbeschreibung der erwarteten Leistung und Ziele.....	7
2.3 Anforderungen an das IT-System und das Projekt	8
2.3.1 Funktionale Anforderungen.....	8
2.3.2 Anforderungen an das Projekt.....	8
3 Vorstellung des Projektteams.....	9
4 Pflichtenheft	11
4.1 Unternehmenscharakteristik.....	11
4.2 Ist-Zustand der Arbeitsgebiete	11
4.2.1 Überblick und Zusammenhänge	11
4.2.2 Bisherige Verfahren für die Arbeitsgebiete	12
4.3 Zielsetzung	12
4.4 Anforderungen an die geplante Software	12
4.4.1 Fachliche Anforderungen	12
4.4.2 Technische Anforderungen	14
4.5 Mengengerüst.....	15
4.6 Anforderungen an die Hard- und Systemsoftware	16
4.7 Zahl der Mitarbeiter für die Umstellung.....	16
4.8 Definition des zeitlichen Rahmens	17
4.9 Struktur der Webseite	14
5 Systemarchitektur	18
5.1 Auswahl verwendeter Technologien.....	18

5.1.1	Entwicklungsumgebung	18
5.1.2	Auswahl der Entwicklungsumgebung.....	20
5.1.3	Datenbanksystem	20
5.1.4	Auswahl der verwendeten Java-Technologie	21
5.1.5	Framework zur Darstellung	21
5.1.6	Auswahl des Frameworks	22
5.2	Aufbau des Systems	23
6	Implementierung	24
6.1	Datenbankstruktur.....	24
6.1.1	Tabellen und Relationen	24
6.1.2	Trigger	24
6.2	Applikationsstruktur.....	25
6.2.1	Modularer Aufbau	25
6.2.2	Objektvererbung	26
6.2.3	Umsetzung des MVC-Modells.....	26
6.3	Module	27
6.3.1	Rights – Rechte	27
6.3.2	Users – Benutzer	27
6.3.3	Announcements – Benachrichtigungen.....	27
6.3.4	Tickets	27
6.3.5	Solutions – Lösungen.....	28
6.3.6	Search – Suche	28
6.3.7	Reporting – Auswertung.....	28
6.4	Umsetzung der Webseitenstruktur	29
7	Überprüfung und Wartung.....	30
7.1	Testablauf	30
7.2	Modultests	30
7.2.1	Users – Benutzer	30
7.2.2	Rights – Rechte	30
7.2.3	Announcements – Benachrichtigungen.....	30
7.2.4	Tickets	30
7.2.5	Solutions – Lösungen.....	31
7.2.6	Search – Suche	31

7.2.7	Reporting – Auswertung.....	31
7.3	Testergebnisse	31
8	Fazit und Ausblick.....	32
8.1	Projektfazit	32
8.2	Erweiterungsmöglichkeiten	32
	Literaturverzeichnis	33
	Quellenverzeichnis	34
	Verzeichnis der Abbildungen.....	35
	Verzeichnis der Tabellen	36
	Verzeichnis der Anlagen.....	37
	Anlagen.....	38

Verzeichnis der verwendeten Abkürzungen und Symbole

3NF = Dritte Normalform

AWT = Abstract Window Toolkit

CRUD = Create, Read, Update und Delete

DAO = Data Access Object

ERM = Entity-Relationship-Modell

FAQ = Frequently Asked Questions

GB = Gigabyte

GUI = Graphical User Interface

IDE = Integrated Development Environment

JSF = Java Server Faces

KW = Kalenderwoche

LTS = Long Term Services

MB = Megabyte

MVC = Model-View-Controller

SQL = Structured Query Language

UML = Unified Modeling Language

XHTML = Extensible Hypertext Markup Language

1 Einleitung

Software ist bereits heute ein wichtiger Bestandteil unseres Lebens. Viele denken bei Software meist an Computerspiele oder an Apps für das Smartphone, doch auch in der Waschmaschine, im Kaffeeautomat oder im Auto befindet sich heute Software und täglich kommen weitere Einsatzbereiche hinzu. Doch wie schreibt man eigentlich gute Software? Sie muss natürlich die Bedürfnisse des Anwenders bedienen. Diese hören nicht schon mit dem richtigen Ergebnis auf, sondern gehen über ein ansprechendes Design hin zu einer einfachen Handhabung. Um das Prädikat einer guten Software zu erreichen muss das Rundumpaket stimmen. Allerdings ist es mit der Anwendersicht noch nicht getan. Die Software muss auch aus Entwicklersicht gute Software sein. Das heißt sie muss gut lesbar, anpassbar und erweiterbar sein. Denn oft ist es notwendig Software an neue Gegebenheiten, beziehungsweise neue Bedürfnisse der Anwender anzupassen. Wie Grady Booch einmal sagte:

"Sauberer Code ist einfach und direkt. Sauberer Code liest sich wie wohlgeschriebene Prosa. Sauberer Code verdunkelt niemals die Absicht des Designers, sondern ist voller griffiger Abstraktionen und geradliniger Kontrollstrukturen". (Martin, 2009)

Diese Aussage ist die Grundlage und das Ziel für unser Projekt. Wir wollen aus Sicht eines beauftragten Softwareunternehmens ein Ticketsystem, das benutzerfreundlich ist und aus sauberem Code besteht, für die Muster GmbH programmieren. Aufbauend auf dem Wasserfallmodell wurde zunächst das fiktive Lastenheft (siehe 2 Lastenheft) der Muster GmbH erstellt. Dieses ist die Grundlage für das spätere Produkt. Daraufhin haben wir das entsprechende Pflichtenheft (siehe 4 Pflichtenheft) erstellt, um klar abzugrenzen, was das Ticketsystem leisten und was es nicht leisten soll. Im Kapitel 5 Systemarchitektur befindet sich die Abwägung und Auswahl der einzelnen Technologien, die zum Einsatz kommen sollen. Daran schließt sich das Kapitel 6 Implementierung, in welchem es um die eigentliche Implementierung und somit um die Grundlage für sauberen Code geht, an. Zu einem guten Softwareprodukt gehören natürlich auch entsprechende Softwaretests. Diese wurden in Kapitel 7 Überprüfung und Wartung ausgeführt und dokumentiert.

2 Lastenheft

2.1 Beschreibung des Unternehmens

Name des Unternehmens: Muster GmbH

Anschrift: Musterstraße 24
12345 Musterhausen

Unternehmensgröße: 340 Mitarbeiter(innen)

Branche: Dienstleistung

Beschreibung des Portfolios: Gebäude- und Liegenschaftsbetreuung

Kontaktperson: Marcel Mustermann
Tel.: 01234 9876-54

2.2 Ziele des IT-Systems

2.2.1 Gründe für die Einführung des IT-Systems

Die Erbringung von internen Helpdesk-Dienstleistungen erfolgt derzeit unkoordiniert und teilweise in unterschiedlicher Qualität. Des Weiteren ist die Auslastung der einzelnen Helpdesk-Mitarbeiter sehr unterschiedlich und Probleme, die mehrere Benutzer betreffen, lassen sich nur mit Verzögerung identifizieren.

2.2.2 Kurzbeschreibung der erwarteten Leistung und Ziele

Durch die Einführung eines browserbasierten Helpdesksystems soll die Leistung der Helpdesk-Mitarbeiter messbar gemacht werden. Dies ermöglicht eine gleichmäßige Auslastung. Die Qualität des erbrachten Supports soll gesteigert werden. Außerdem ist es das Ziel IT-Probleme, die mehrere Mitarbeiter betreffen, besser zu identifizieren und zu adressieren. Die Einführung des IT-Systems soll bis März 2014 erfolgen.

2.3 Anforderungen an das IT-System und das Projekt

2.3.1 Funktionale Anforderungen

Die funktionalen Anforderungen werden in den folgenden Punkten beschrieben:

- Das Ticketsystem muss plattformunabhängig sein.
- Es muss eine verbreitete Programmiersprache verwendet werden, um die Wartung und Anpassung des Systems zu ermöglichen.
- Die Tickets müssen sich eindeutig einem Nutzer und einem Bearbeiter zuordnen lassen.
- Aus Gründen der Nachvollziehbarkeit muss die Lösung eines Tickets dokumentiert werden. Das Erfassen der Lösung darf sich deshalb nicht umgehen lassen.
- Eine Wissensdatenbank für häufige Probleme sollte für die Helpdesk-Mitarbeiter zur Verfügung gestellt werden.
- Um die Messbarkeit der erbrachten Helpdesk-Leistung zu ermöglichen, müssen verschiedene Auswertungsmöglichkeiten zur Verfügung gestellt werden.
- Das Ticketsystem soll aus Gründen der Akzeptanz intuitiv zu bedienen sein.
- Des Weiteren soll im Havariefall die Möglichkeit bestehen, die Nutzer über bereits bekannte Probleme zu informieren. Damit werden zusätzliche Tickets vermieden.

2.3.2 Anforderungen an das Projekt

Die Anforderungen an das Projekt werden in den folgenden Punkten zusammengefasst:

- Das IT-System soll bis zum 1. März 2014 eingeführt sein.
- Das Projektteam verwendet ein, für die Softwareentwicklung übliches, Phasenmodell.
- Die Mitarbeiter erhalten eine den Anforderungen an das IT-System entsprechende Schulung.
- Die Funktionsfähigkeit des Helpdesksystems ist anhand von entsprechenden Softwaretests zu belegen.

3 Vorstellung des Projektteams

Tabelle 3-1: Vorstellung der Person Solveig Thomas

Mitglied:	Solveig Thomas
Aufgabenbereich:	Frau Thomas implementiert im Projekt den Reportingbereich. Dabei beschäftigt sie sich zuerst mit den möglichen Kennzahlen, welche sich aus der Datenlage bilden lassen. Im anschließenden Implementierungsprozess überlegt sie sich eine geeignete grafische Darstellung der Kennzahlen, was eine entsprechende Datenbankabfrage im Hintergrund mit einschließt. Des Weiteren arbeitet sie einige Testfälle ab.
Qualifikation:	Erste Erfahrungen im Bereich der Softwareentwicklung konnte Frau Thomas im ersten Semester ihres Studiums mit den Programmiersprachen C und C++ sammeln. Im zweiten Semester vertiefte sie ihre Kenntnisse im Bereich Datenbanken und im dritten Semester mit dem Modul Software Engineering im Bereich Webentwicklung.

Tabelle 3-2: Vorstellung der Person Martin Krockert

Mitglied:	Martin Krockert
Aufgabenbereich:	Auf Grund seiner Erfahrungen hat Herr Krockert den wohl größten Anteil an der Entwicklung und der Umsetzung des Projektes. Er legt den Grundstein für die gemeinsame Arbeit, indem er die Datenbank erstellt und einbindet, die Applikations-Struktur entwirft, ein Template-Layout der Webseite anlegt und die Benutzerverwaltung sowie das Rechtesystem programmiert. Während des Projektes steht er als Ansprechpartner bei Problemen zur Verfügung. Im finalen Abschnitt der Entwicklung liegt die Zusammenführung der Anwendung in seiner Verantwortung.
Qualifikation:	Herr Krockert ist seit 2003 im Bereich IT-Administration und Softwareentwicklung tätig. Während dieser Zeit hat er zahlreiche Software- und IT-Projekte begleitet. Die während dieser Zeit gesammelten Erfahrungen stellt er dem Team bei der technischen Planung und Entwicklung zur Verfügung.

Tabelle 3-3: Vorstellung der Person Jens Hetze

Mitglied:	Jens Hetze
Aufgabenbereich:	<p>Die Aufgaben von Herrn Hetze liegen in diesem Projekt in der Umsetzung der Schnittstelle zwischen den Tickets und den Lösungen sowie der Bereitstellung der Suchfunktion. Dazu ist es nötig sich über SQL-Abfragen, welche zur Suche genutzt werden können, Gedanken zu machen. Die Programmierung der darauf aufbauenden Methoden ist die Kernaufgabe von Herrn Hetze.</p> <p>Hinzukommend bearbeitet er die Form, Rechtschreibung und Ausgestaltung der Dokumentation.</p>
Qualifikation:	<p>Herr Hetze ist seit 2002 im Bereich der IT-Administration tätig. Er hat bereits einige Datenbankanwendungen mit Lotus Notes entwickelt. Lotus Notes von IBM basiert auf Java. Die Datenbankanwendungen wurden jedoch nicht mit Java, sondern mit der Lotus Notes eigenen Programmiersprache Lotus Script programmiert. Herrn Hetze war die objektorientierte Programmierung bis zu Beginn des Studiums unbekannt. In diesem erweiterte er sein Wissen der prozeduralen Programmierung um die Objektorientierung.</p>

Tabelle 3-4: Vorstellung der Person Johannes Schaffrath

Mitglied:	Johannes Schaffrath
Aufgabenbereich:	<p>Herr Schaffrath kümmert sich um die Programmierung des Ticket-handlings. Außerdem überprüft er den Quellcode der anderen Projektmitglieder auf Sauberkeit und Einheitlichkeit.</p>
Qualifikation:	<p>Mit dem Beginn seines Studiums sammelte Herr Schaffrath erste Erfahrung im Bereich der Softwareentwicklung. Angefangen mit dem Modul Programmierung für die Programmiersprachen C und C++, über eine Praxisphase im Bereich ABAP Entwicklung hin zur Programmiersprache Java im Modul Software Engineering erweiterte er stetig seine Kompetenzen.</p>

4 Pflichtenheft

4.1 Unternehmenscharakteristik

Name und Anschrift:	Muster GmbH Musterstraße 24 12345 Musterhausen
Branche, Dienstleistung:	Dienstleistung; Gebäude- und Liegenschaftsbetreuung
Unternehmensstruktur:	Funktional
Zahl der Beschäftigten:	340 Mitarbeiter(innen)
Zahl der Niederlassungen:	5 Niederlassungen
Unternehmensgröße:	Mittelständisches Unternehmen
Wachstumsrate:	4%

4.2 Ist-Zustand der Arbeitsgebiete

4.2.1 Überblick und Zusammenhänge

Das Anfordern von internen Helpdesk-Dienstleistungen erfolgt derzeit telefonisch oder per E-Mail an die einzelnen Helpdesk-Mitarbeiter. Die Lösung des Problems wird derzeit nicht dokumentiert und es erfolgt kein Monitoring. Aufgrund dessen sind die Mitarbeiter unterschiedlich ausgelastet und Probleme, die mehrere Nutzer betreffen, werden meist mit Verzögerung identifiziert. Vereinzelt kommt es vor, dass Probleme zwar aufgenommen werden, aber im Prozessablauf untergehen und dadurch nicht abgearbeitet werden.

4.2.2 Bisherige Verfahren für die Arbeitsgebiete

Derzeit werden die Probleme telefonisch aufgenommen oder durch die Hilfesuchenden per E-Mail an einzelne Helpdesk-Mitarbeiter gesandt. Die Abarbeitung der Probleme erfolgt individuell und ohne Kontrolle durch den zuständigen IT-Leiter.

Tabelle 4-1: Bewertung des Ist-Zustandes

Bewertung des Ist-Zustandes:	
<p>Stärken:</p> <ul style="list-style-type: none">• Hohe Akzeptanz aufgrund des persönlichen Kontaktes• Geringer Dokumentationsaufwand	<p>Schwächen:</p> <ul style="list-style-type: none">• Unterschiedliche Auslastung der Helpdesk-Mitarbeiter• Keine Festlegung von Prioritäten bei der Bearbeitung von Problemen• Späte Erkennung von globalen Problemen• Organisationsform ist nicht für zukünftiges Wachstum geeignet• Es ist keine Wissensdatenbank vorhanden• Adäquate Auswertungsmöglichkeiten sind nicht gegeben

4.3 Zielsetzung

Die Ziele und der erwartete quantifizierbare Nutzen des zu entwickelnden Systems bestehen darin, dass die Mitarbeiterauslastung ausgeglichen sein soll, Problemlösungen nachvollzogen werden können und eine Auswertung zur Verfügung steht.

4.4 Anforderungen an die geplante Software

4.4.1 Fachliche Anforderungen

4.4.1.1 Überblick und Zusammenhänge

Das Ticketsystem soll die Arbeit der Helpdesk-Mitarbeiter vereinfachen und für Dritte nachvollziehbar machen. Des Weiteren soll jeder Mitarbeiter die Möglichkeit erhalten seine erstellten Tickets zu verfolgen und Informationen beziehungsweise Lösungen zu bekannten Problemen zu recherchieren.

4.4.1.2 Detaillierte Anforderungen

User – Benutzer

- Jeder Benutzer verfügt über einen eindeutigen Benutzernamen und ein Passwort.

Rights – Rechte

- Es stehen drei Rollen (Administrator, Techniker und Anwender) zur Verfügung, um den Benutzern entsprechende Rechte zuzuweisen.

Announcements – Benachrichtigungen

- Das IT-System verfügt über einen Bekanntmachungsbereich für aktuelle Meldungen, welcher von Administratoren und Technikern gepflegt wird.

Tickets

- Jeder Benutzer kann Tickets erstellen. Diese sind dem jeweiligen Ersteller eineindeutig zugeordnet.
- Ein Ticket enthält die im Datenbankschema (siehe Anlage 1: Datenbankschema) ersichtlichen Felder.
- Bei jedem Ticket ist die Lösungsbeschreibung ein Pflichtfeld.
- Des Weiteren darf es nicht möglich sein Tickets zu löschen.

Solutions – Lösungen

- Zum IT-System gehört eine Wissensdatenbank mit Lösungen, die von den Administratoren und Technikern gepflegt wird und jedem Nutzer zur Verfügung gestellt werden kann.

Search – Suche

- Die Suche stellt den Benutzern das Wiederauffinden von Datensätzen zur Verfügung.
- Sie wird nur ausgeführt, wenn ein Suchstring eingegeben wurde.
- Der Benutzer kann entweder eine Volltextsuche oder eine tabellenbezogene Suche (Erweiterte Suche) ausführen.
- Bei Ausführung einer Volltextsuche suchen die Rechte "Administrator" und "Techniker" in allen Tabellen. "Anwender" suchen hingegen nur in "Tickets" und "Lösungen".
- Die Benutzer mit den Rechten "Administrator" und "Techniker" durchsuchen alle Tickets. "Anwender" durchsuchen nur eigene Tickets.
- Bei Ausführung einer tabellenbezogenen Suche (Erweiterte Suche) werden die Spaltenauswahl und die Sortierauswahl dynamisch auf die gewählte Tabelle angepasst.
- Der Benutzer kann entweder in allen Spalten oder in einer ausgewählten Spalte suchen.
- Die Sortierung kann aufsteigend oder absteigend erfolgen.
- Standardmäßig, wenn keine einschränkenden Kriterien gewählt wurden, wird eine Volltextsuche mit aufsteigender Sortierung nach ID ausgeführt.

Die Funktionsweise der Suche unter den oben genannten Anforderungen wird in Anlage 6: Entscheidungstabelle zum Modul "Search" dargestellt.

Reporting – Auswertung

- Die Administratoren haben die Möglichkeit über eine Oberfläche Reports abzubilden.

Die oben genannten Anforderungen für das geplante Helpdeskssystem werden mit den Systemgrenzen laut Anlage 3: Systemgrenzen des Helpdeskssystems festgelegt. Die zur Verfügung stehenden Aktionen der verschiedenen Benutzerrechte ist in den Anlagen Anlage 4: Use-Case-Diagramm des Helpdeskssystems und Anlage 5: Rechtematrix des Helpdeskssystems dargestellt.

Struktur der Webseite

Die Webseite wird in folgende Abschnitte unterteilt (siehe Abbildung 4-1: Struktur der Webseite), Navigation (North), Anmeldung (East), Datenbereich (Center) und Seitenfuß (South). Dieser Aufbau wird über eine Vorlage realisiert und ist maßgebend für ein einheitlich übergreifendes Layout.

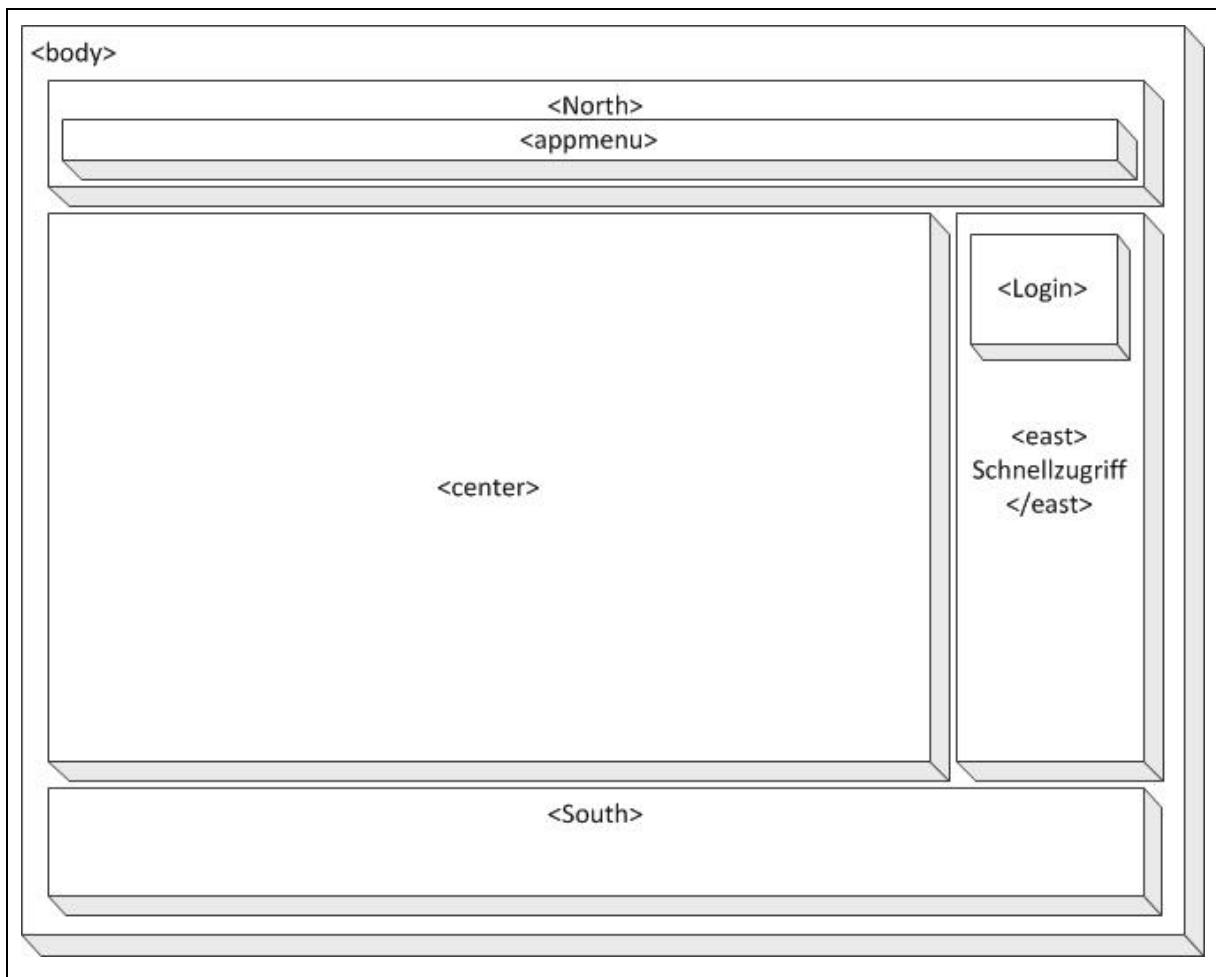


Abbildung 4-1: Struktur der Webseite

4.4.2 Technische Anforderungen

Qualitätsanforderungen

- Die anfallenden Daten werden in einer Datenbank gespeichert. Die für die Registrierung notwendigen Kennwörter werden mit den SHA-256 Hash-Algorithmen verschlüsselt.

- Die Anwendung wird webbasiert umgesetzt. Als Programmiersprache wird Java verwendet.

Dokumentation und Schulung

- Für die Muster GmbH wird eine Dokumentation aller vorhandenen Komponenten erstellt.
- Die Administratoren der Muster GmbH erhalten durch das Projektteam eine Einweisung in das Helpdesk-System, mit welcher sie das IT-System einrichten und bedienen können.
- Die Mitarbeiter der Muster GmbH werden anschließend von den Administratoren geschult.

4.5 Mengengerüst

Für das System werden die folgenden Mengen und Häufigkeitsangaben verwendet:

- 340 potentielle Ticketschreiber
- Zirka 170 Tickets pro Monat bei 0,5 Tickets pro Tag und Person
- Etwa ein Eintrag in der Lösungsdatenbank pro 20 erstellter Tickets

4.6 Anforderungen an die Hard- und Systemsoftware

Tabelle 4-2: Anforderungen an das Serversystem

Anforderungen an das Serversystem:		
Software:	Betriebssystem:	<ul style="list-style-type: none">• Linux LTS ab Kernel Version 3.0
	Hostsystem:	<ul style="list-style-type: none">• Apache Tomcat 7.0 oder höher• Serverbasiertes SQL Management System
Hardware:	Mindestens:	<ul style="list-style-type: none">• 512 MB Arbeitsspeicher• 1 GB freier Festplattenspeicher• Single Core Prozessor mit mehr als einem GHz Taktfrequenz
	Empfohlen:	<ul style="list-style-type: none">• 2 GB Arbeitsspeicher• 10 GB freien Festplattenspeicher• 2,5 GHz XEON CPU

Tabelle 4-3: Anforderungen an die Clients

Anforderungen an die Clients:	
Software:	<p>Die Anwendung ist auf jedem Clientsystem benutzbar, auf dem ein webfähiger Browser installiert ist. Die Anwendung wird getestet und soll folgende Browser unterstützen:</p> <ul style="list-style-type: none">• Microsoft Internet Explorer 9 oder höher• Mozilla Firefox 21 oder höher• Google Chrome Version 25 oder höher
Hardware:	Der Client sollte, um eine optimale Darstellung der Weboberfläche zu gewährleisten, eine Auflösung von mindestens 1280x1024 Pixeln unterstützen.

4.7 Zahl der Mitarbeiter für die Umstellung

Für die erfolgreiche Umsetzung des Projektes wird mit vier Mitarbeitern geplant. Um die Einrichtung bei der Muster GmbH zu realisieren, wird ein Administrator dieser benötigt.

4.8 Definition des zeitlichen Rahmens

Das Projekt beginnt am 3. Januar 2014. Eine entsprechende Vorabversion der Software wird dem Kunden bis zum 31. Januar 2014 zur Verfügung gestellt. In der ersten Märzwoche – Kalenderwoche (KW) 10 des Jahres 2014 – wird das System beim Kunden installiert. In dieser Woche finden auch die Benutzerschulungen statt. Am 10. März 2014 soll das System in den Produktivbetrieb eingebunden sein.

5 Systemarchitektur

5.1 Auswahl verwendeter Technologien

5.1.1 Entwicklungsumgebung

Theoretisch kann die Wahl der Entwicklungsumgebung jedem Entwickler selbst überlassen werden. Dennoch haben wir uns entschieden unsere Entwicklung auf eine einheitliche Basis zu stellen, um eventuellen Komplikationen vorzubeugen und schnelle gegenseitige Hilfe sicherzustellen. Daraufhin wurden die drei wichtigsten Entwicklungsumgebungen kurz ausgetestet und in ihrer Handhabung bewertet.

Tabelle 5-1: Entwicklungsumgebung „IntelliJ IDEA“

Funktion	Beschreibung	Punkte 1-3
Installation:	+ Einfache Installation durch ein einzelnes Installationspaket	3
GUI:	+ Viele Möglichkeiten die Oberfläche anzupassen + Viele Fenster mit sehr vielen Optionen, die beliebig positioniert werden können – Kann schnell unübersichtlich werden	2
Features:	+ Sehr schneller kontextsensitiver Editor + Sehr viele Codeautomatismen	2
Erweiterbarkeit:	– Möglich, aber im Vergleich stellte sich heraus, dass es nur eine geringe Auswahl an sinnvollen Erweiterungen gibt	1

Tabelle 5-2: Entwicklungsumgebung „Netbeans“

Funktion	Beschreibung	Punkte 1-3
Installation:	<ul style="list-style-type: none"> – Es sind zwei Installationspakete notwendig – IDE und Weberweiterung – im Grunde aber eine einfache Installation 	1
GUI:	<ul style="list-style-type: none"> + Sehr übersichtlich strukturiert + Schnelle und einfach verständliche Benutzeroberfläche 	3
Features:	<ul style="list-style-type: none"> + Mit der Weberweiterung wird ein eigener Apache- und Glassfish-Server mitgeliefert + Codedebugging-Plug-Ins für Google Chrome und Mozilla Firefox 	3
Erweiterbarkeit:	<ul style="list-style-type: none"> + Sehr viele Erweiterungen, als auch eine stetig entwickelnde Community 	2

Tabelle 5-3: Entwicklungsumgebung „Eclipse“

Funktion	Beschreibung	Punkte 1-3
Installation:	<ul style="list-style-type: none"> + Einfache Installation – Großes 800 MB Installationspaket – Nachinstallation für Webentwicklung notwendig 	1
GUI:	<ul style="list-style-type: none"> + Viele voreingestellte Fenster – Fensterverhalten beim Schließen – Unübersichtliche Anordnung, lange Optionslisten 	1
Features:	<ul style="list-style-type: none"> + Einfache Fehlerverwaltung und gute Lösungsvorschläge + Einfaches Nachladen von Plug-Ins, wenn diese benötigt werden 	3
Erweiterbarkeit:	<ul style="list-style-type: none"> + Größte Erweiterungsbibliothek im Vergleich 	3

5.1.2 Auswahl der Entwicklungsumgebung

Das Team hat sich auf Basis der Entscheidungsmatrix (siehe Tabelle 5-4: Entscheidungsmatrix zur Auswahl der Entwicklungsumgebung) für Netbeans entschieden. Die einfache Oberfläche und das Codedebugging sind herausragende Merkmale von Netbeans. Es bietet deswegen für unsere Anforderungen an das Projekt die beste Rundumversorgung.

Tabelle 5-4: Entscheidungsmatrix zur Auswahl der Entwicklungsumgebung

	IntelliJ IDEA	Netbeans	Eclipse
Installation:	3	2	1
GUI:	2	3	1
Features:	2	3	3
Erweiterbarkeit:	1	2	3
Summe:	8	10	8

5.1.3 Auswahl des Datenbanksystem

Aufgrund der Projektanforderungen, Portabilität und Systemunabhängigkeit standen in diesem Bereich nur drei valide Optionen zur Verfügung. Unter den Drei blieb nach genauerer Analyse nur „SQLite“ als geeignetes Datenbanksystem übrig.

HSQldb

„HSQLDB“ ist ein portables in Java geschriebenes Datenbankmanagementsystem. Es ist in seinem Funktionsumfang für das Projekt jedoch nicht geeignet, da Binary Large Objekts und Transaktionen nicht integriert sind.

MySQL

„MySQL“ ist ein serverbasiertes Datenbankmanagementsystem. Durch Programme dritter Parteien ist es möglich die MySQL-Datenbank portierbar zu machen. Die Datenbank ist dadurch jedoch nicht mehr systemunabhängig und damit keine Option für das Projekt.

SQLite

„SQLite“ ist ein in C geschriebenes Datenbankmanagementsystem, das alle Projektanforderungen erfüllt. Die Datenbank selbst benötigt keinerlei Konfiguration und lässt sich mittels einer *.war-Datei auf jedes beliebige System verteilen. (Haldar, 2009)

5.1.4 Auswahl der verwendeten Java-Technologie

Bei der Technologiewahl wurde vom Auftraggeber Java als Codebasis vorgegeben. Innerhalb von Java existieren jedoch weitere technologische Unterscheidungen. In den Ausgabemechanismen. Für die Webentwicklung wird in den Kategorien in „**Servlets**“, „**Java Server Pages**“ (JSP) oder „**Java Server Faces**“ (JSF) unterschieden.

JSF ist in dieser Auswahl das technologisch am weitesten fortgeschrittene Werkzeug zur Darstellung von grafischen Elementen und ist spezialisiert auf Webanwendungen. Des Weiteren werden für die JSF sehr gute Layout-Frameworks angeboten. Diese erleichtern die optisch einheitliche Gestaltung der Webseite.

5.1.5 Framework zur Darstellung

Bei der Darstellung der HTML-Elemente innerhalb von JSF kann auf drei teilweise sich stark unterscheidende Frameworks zurückgegriffen werden. Das Wort Framework sollte in diesem Zusammenhang nicht missverstanden werden, da diese im eigentlichen sinne Bibliothekserweiterungen für die Darstellung des HTML-Kontextes sind.

Tabelle 5-5: Framework „RichFaces“ (redhat)

Funktion	Beschreibung	Punkte 1-3
Komponenten:	– 39 Komponenten und Komponenten Entwickler Kit	1
Implementierung:	– Zusätzliche Abhängigkeiten notwendig	0
Dokumentation:	+ Tutorial nur von Drittanbietern	1
Features:	+ Auslösung der Clientaktualisierung durch serverseitige Ereignisse – „RichFaces Push“	2

Tabelle 5-6: Framework „IceFaces“ (ICEsoft Technologies Inc.)

Funktion	Beschreibung	Punkte 1-3
Komponenten	+ 70 Komponenten und 40 Erweiterungen aus der „PrimeFaces“-Bibliothek	1
Implementierung	– Zusätzliche Abhängigkeiten notwendig	2
Dokumentation	+ Schaukasten nur für wichtige Bibliotheksteile + Viele Frequently Asked Questions (FAQ) im Forum des Anbieters	3
Features	+ Aktualisierung von Teilbereichen einer Webseite mit „Direkt 2 DOM“ Technologie	3

Tabelle 5-7: Framework „PrimeFaces“ (primetec)

Funktion	Beschreibung	Punkte 1-3
Komponenten	+ 120 Komponenten	3
Implementierung	+ Einfaches Hinzufügen über eine einzelne Datei – PrimeFaces-{Version}.jar	3
Dokumentation	+ Vollständige Dokumentation als E-Book vorhanden + Schaukasten allen zur Verfügung stehenden Elementen mit Codebeispielen	3
Features	+ jQuery als Codebasis für alle Erweiterungen	1

5.1.6 Auswahl des Frameworks

„RichFaces“ ist für das Projekt als Framework nicht geeignet, da es keine Unterstützung für grafische Auswertungen bietet. Die Entscheidungsmatrix zeigt das „IceFaces“ und „PrimeFaces“ ähnlich hohe Punktzahlen erreicht haben. Am Ende hat sich im Vorabtest „PrimeFaces“ durchgesetzt. Ausschlaggebend waren die einfache Implementierung und die hervorragende Dokumentation.

Tabelle 5-8: Entscheidungsmatrix zur Auswahl des Frameworks

	RichFaces	IceFaces	PrimeFaces
Komponenten:	1	1	3
Implementierung:	0	2	3
Dokumentation:	1	3	3
Features:	2	3	1
Summe:	4	9	10

5.2 Aufbau des Systems

Die Projektstruktur orientiert sich an der Model-View-Controller-Architektur (MVC). Bei dieser Einteilung werden die Codeteile in drei Teile aufgespalten. Ziel dabei ist ein leicht zu lesendes Programmuster, welches spätere Erweiterungen und Änderungen vereinfachen soll.

Die Bezeichnung der Codeteile im Projekt ist wie folgt benannt:

Tabelle 5-9: Umsetzung der MVC-Architektur

View:	{Tabellenname}{Aktion}.xhtml
Model:	{Tabellenname}Bean.java
Controller:	{Tabellenname}DAO.java

Das Model ist der Datenhalter für die von der View angeforderten Daten. Die Anforderung aus der View wird vom Controller verarbeitet und der View wieder zur Verfügung gestellt. Das MVC-System ist mit den gegebenen Mitteln nicht korrekt abzubilden, fungiert aber als richtungsweisendes Konzept für das Projekt. Als funktionsbeschränkende Größe ist hier der Apache Server zu nennen, da dieser keine Unterstützung für „Enterprise Beans“ besitzt. Weitere Informationen zum MVC-Modell und dessen Umsetzung im Projekt sind unter 6.2 Applikationsstruktur zu finden.

6 Implementierung

6.1 Datenbankstruktur

6.1.1 Tabellen und Relationen

Die erstellte Tabellenstruktur besteht aus acht Tabellen und mehr als acht Beziehungen (siehe Anlage 1: Datenbankschema und Anlage 2: ERM). Die Tabellen unterteilen sich dabei in Stammdaten und Bewegungsdaten. Die Unterscheidung in diese beiden Kategorien wird anhand der Änderungshäufigkeit vorgenommen. Stammdaten enthalten alle notwendigen Informationen die für die Verarbeitung der Anfragen notwendig sind. Zu den Stammdatentabellen zählen users, rights, categories, status und priorities. Zu den Bewegungsdaten zählen die Tabellen announcements, tickets, solutions und attachment.

Die Relationen und Tabellenaufteilung innerhalb der Datenbank sind entsprechend der dritten Normalform (3NF). Eine Besonderheit ist die „attachment“ Tabelle, da sie Anhänge aus Tickets, als auch aus Solutions aufnehmen kann und dafür einen zusammengesetzten Schlüssel aus „id_table“ und „tablename“ besitzt.

6.1.2 Trigger

Zur Vereinfachung der Abfragestruktur innerhalb der Anwendung werden für Standardaktionen Trigger auf Tabellenebene verwendet. Die Trigger kommen zum Einsatz um das Setzen eines Datums beim Einfügen eines Datensatzes und das Setzen eines Änderungsdatums in der Tickettabelle zu automatisieren.

Durch das Einfügen eines Datensatzes in die Tabelle „tickets“ wird ein zusätzlicher Insert-Befehl über den Trigger „tickets_insert“ (siehe Abbildung 6-1: Codebeispiel des Triggers „tickets_insert“) ausgelöst, der das Tupel mit dem aktuellen Datum im Attribut „date_created“ aktualisiert. Durch diesen nachgestellten SQL-Befehl wird zusätzlich der Trigger „tickets_changed“ (siehe Abbildung 6-2: Codebeispiel des Triggers „tickets_changed“) aktiv und das Attribut „date_changed“ geändert. Mithilfe dieser Kombination werden von Beginn an beide Felder mit Datumswerten beschrieben. Jedoch wird nur das Attribut „tickets_changed“ bei weiteren Änderungen angesprochen und das aktuelle Datum im Attribut „date_changed“ gesetzt.

```
CREATE TRIGGER tickets_insert AFTER INSERT ON tickets
BEGIN
    UPDATE tickets set date_created = date('now')
    where tickets.ID = NEW.ID;
END;
```

Abbildung 6-1: Codebeispiel des Triggers „tickets_insert“

```

CREATE TRIGGER tickets_changed AFTER UPDATE ON tickets
BEGIN
    UPDATE tickets set date_changed = date('now')
    where tickets.ID = OLD.ID;
END;

```

Abbildung 6-2: Codebeispiel des Triggers „tickets_changed“

6.2 Applikationsstruktur

6.2.1 Modularer Aufbau

Die Struktur innerhalb des Projekts hält sich an den Standardaufbau von Netbeans. Zudem ist der modulare Aufbau des MVC-Konzeptes mit den vorhandenen Mitteln so nahe wie möglich realisiert worden.

Im rot gekennzeichneten Bereich der Abbildung 6-3 finden sich alle Module namentlich wieder. Webseitenteile die außerhalb der Module liegen sind allgemeingültig und werden über das Template angesteuert. Zum Beispiel wird bei fehlenden Rechten automatisch die „ErrorAccessDenied.xhtml“ in das Template eingebunden.

Die Steuerung der Webseite erfolgt im blau gekennzeichneten Bereich. Auch hier finden sich die Modulbezeichnungen als Pakete wieder. Innerhalb eines Paketes befindet sich der zum Modul passende Controller und das Modell (siehe dazu 5.2 Aufbau des Systems und 6.2.3 Umsetzung des MVC-Modells).

Zu den Modulen gibt es noch zwei weitere Pakete. Das Paket „database“ beinhaltet die Klassen zur Steuerung des Datenbankzugriffs und das Paket „tools“ stellt hilfreiche Methoden zur Datenkonvertierung zwischen den SQL- und Java-Datentypen sowie einige Anzeigefunktionen für die Benutzeroberfläche bereit. Zudem ist die generische Modellstruktur für die Controller-Klassen und das Interface für die Modelklassen hier untergebracht (siehe 6.2.2 Objektvererbung).

Der grüne Bereich enthält extern hinzugefügte Bibliotheken. Dazu zählen die SQL-Treiber, JSF und die Apache Tomcat-Erweiterung für das Projekt.

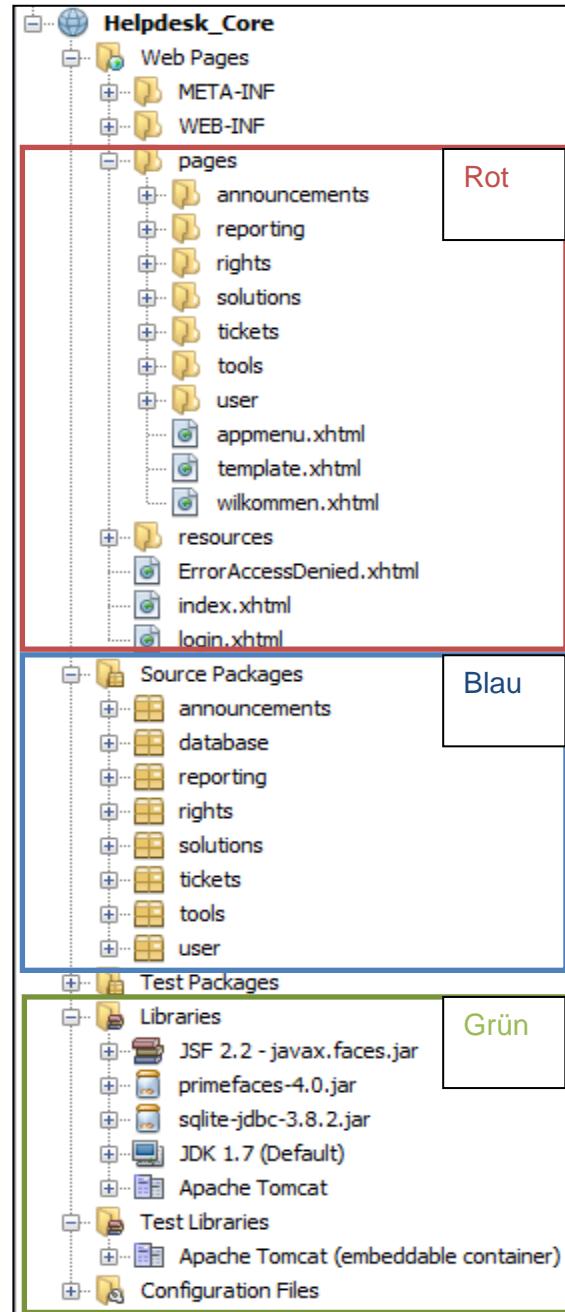


Abbildung 6-3: Projektstruktur des Helpdesks

6.2.2 Objektvererbung

Alle Module, die eine Datenbanktabelle als Basis haben, sind nach dem gleichen Muster aufgebaut. In der Abbildung 6-4: Vererbungsmodell ist das verwendete Modell, anhand des Benutzers beispielhaft, dargestellt. Kern dieses Modells ist die Schnittstelle „InterfaceBean“ und die abstrakte Klasse „AbstractDAO“. Die Schnittstelle stellt zwei Standardmethoden für das Datenmodell bereit.

Anhand dieser Schnittstelle kann das „DAO“ das Datenmodell in einer generischen Liste sammeln und durchsuchen. Die dafür benötigten Methoden werden von der Klasse „AbstractDAO“ bereitgestellt. Die programmtechnische Umsetzung in den entsprechenden Klassen befindet sich als Anlage 9: Codebeispiel der Data Access Object-Klasse angefügt.

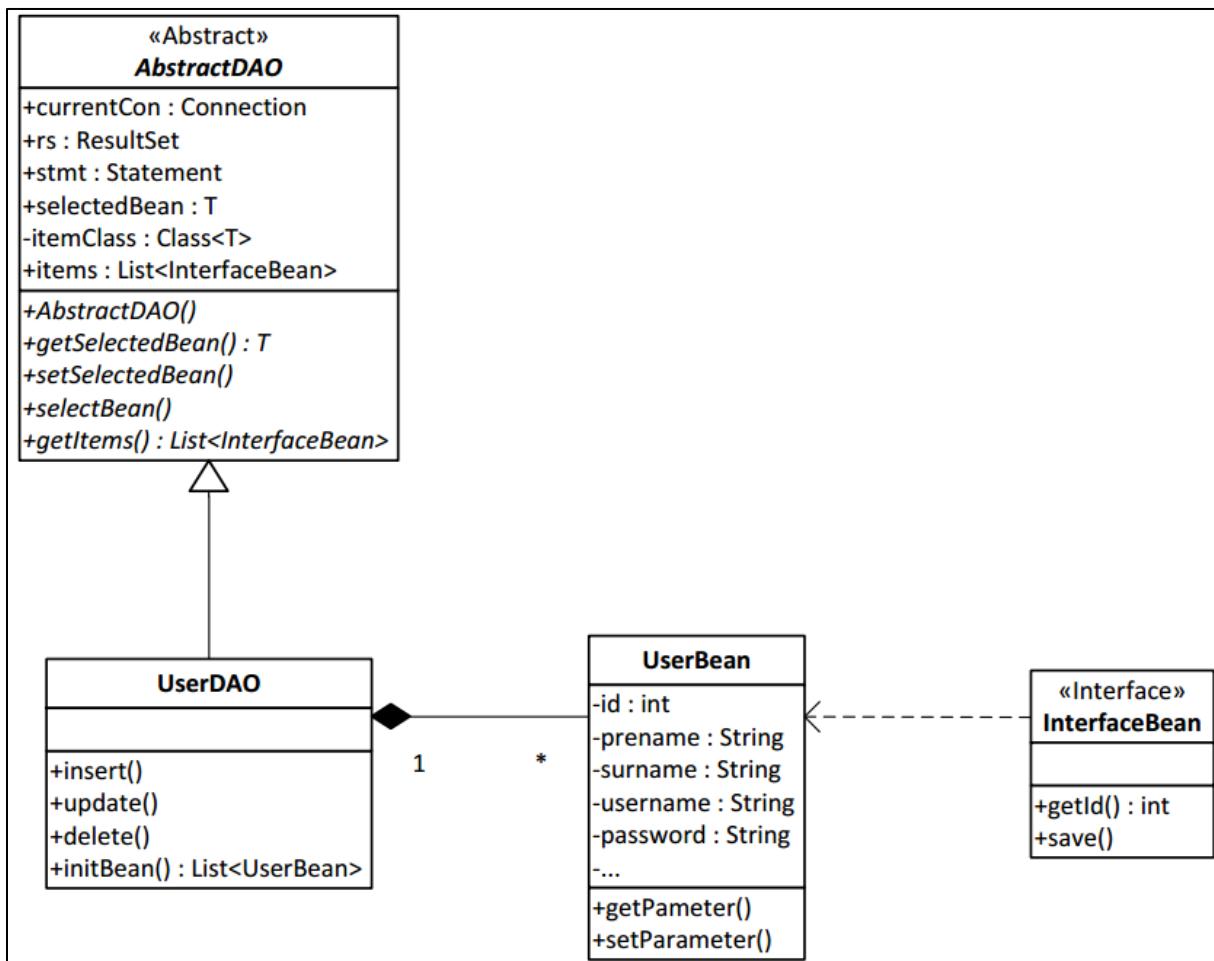


Abbildung 6-4: Vererbungsmodell

6.2.3 Umsetzung des MVC-Modells

6.2.3.1 Modell: Bean

Im Projekt entspricht die „{Tabelle}Bean“ dem Datenschema einer Tabelle. Die Instanz einer Bean ist äquivalent zu einem Tupel aus der jeweiligen Abfrage der Datenbank. Als Methoden besitzt diese Klasse ausschließlich Getter und Setter für die jeweiligen Parameter und eine Methode „save()“, die sich selbst als Objekt übergibt.

6.2.3.2 Controller: DAO – Data Access Object

Das Objekt „{Tabelle}DAO“ reguliert alle Interaktionen zwischen der Datenbank, der Darstellungsebene und dem Datenmodel-Bean. Das sind die sogenannten „Create, Read, Update und Delete“-Operationen (CRUD). Durch die Vererbung der abstrakten Klasse „Abstract-DAO“ werden dem Objekt die benötigten Parameter für den Datenbankzugriff übergeben.

6.2.3.3 View: Darstellungstemplate, *.xhtml und jQuery

Die Anzeige der Ergebnisse einer Abfrage erfolgt über JSF-Seiten im XHTML-Format. Für die einheitliche Illustration der Seiten ist „PrimeFaces“ als Framework verantwortlich. Im Hintergrund von „PrimeFaces“ arbeitet die jQuery API. Mit Hilfe derer kann das Layout der Seite ohne Schwierigkeiten an neue Designs angepasst werden. Eine Einbindung eigener Oberflächenelemente auf Basis von jQuery ist ohne Probleme möglich.

6.3 Module

6.3.1 Rights – Rechte

Das Modul „Rights“ verwaltet die auf der Webseite zur Verfügung stehenden Rechte. Die standardmäßig angelegten Rechte sind Administrator, Techniker und Benutzer. Dabei ist ein Administrator automatisch auch Techniker und ein Techniker analog auch ein Benutzer. Das Hinzufügen weiterer Rechte ist im aktuellen Stand der Entwicklung nicht möglich, da dies keine Auswirkung auf die Funktionsweise des Systems hätte. Jedoch ist das Umbenennen der Rechte möglich, da intern mit den zugehörigen IDs gearbeitet wird.

6.3.2 Users – Benutzer

Über das Modul „Users“ können die Benutzer des Helpdesksystems verwaltet werden. Dem Benutzer können in diesem Bereich entsprechende Rechte gegeben, sein Passwort zurückgesetzt und Informationen, wie zum Beispiel Vorname, Nachname, Telefonnummer und E-Mail-Adresse, über ihn hinterlegt werden.

6.3.3 Announcements – Benachrichtigungen

Nach dem Anmelden ist auf der Startseite ein Bereich mit Nachrichten für alle Benutzer eingeblendet. Dieser kann von Administratoren sowie Technikern verwendet werden um auf wichtige Gegebenheiten hinzuweisen. Jede Meldung ist mit einem Gültigkeitsdatum versehen. Ist dieses abgelaufen wird die Nachricht automatisch aus dem Anzeigebereich entfernt. Sie kann jedoch für Revisionszwecke wieder eingeblendet werden.

6.3.4 Tickets

Dieses Modul umfasst das Erstellen und Bearbeiten von Tickets. Die Tickets werden dem entsprechenden Bearbeiter zugeordnet und in der Datenbank abgelegt. Jeder Benutzer verfügt über die Möglichkeit sich seine Tickets anzuzeigen. Der Techniker verfügt zusätzlich über die Möglichkeit sich Tickets anzuzeigen, die er bearbeiten soll. Der Administrator kann sich die Tickets von allen Benutzern anzeigen lassen. Die Tickets lassen sich nach verschie-

denen Kriterien sortieren. In den Tickets können Dateien als Anhänge abgelegt werden. Die Anhänge werden beim Hochladen temporär gespeichert und erst beim Speichern des Tickets in die Datenbank geschrieben (siehe Anlage 10: Codebeispiel des Dateiuploads).

6.3.5 Solutions – Lösungen

Das Modul „Solutions“ oder Lösungen bildet die Wissensdatenbank für IT-Probleme ab. Darin sind alle wiederverwendbaren Lösungen zu bereits aufgetretenen oder voraussichtlichen Problemen enthalten. Es gibt öffentliche Lösungen und Lösungen nur für Administratoren und Techniker. Der Grund dafür ist der Datenschutz. Es könnten Kennwörter oder abteilungssensible Daten in den Lösungen enthalten sein. Den Lösungen können wie den Tickets Dateianhänge angehören. Diese sind mit Hilfe desselben Moduls gelöst.

6.3.6 Search – Suche

Das Modul „Search“ oder auch Suche wird im Helpdeskssystem zum Wiederauffinden von Datensätzen genutzt. Es steht eine schnelle Suche über alle Tabellen der Datenbank oder eine erweiterte Suche mit Auswahl der zu durchsuchenden Tabelle oder des zu durchsuchenden Feldes und die Sortierung der Suchergebnisse zur Verfügung. Der Vorgang der Suche wird wie in Anlage 7: Aktivitätsdiagramm zum Modul "Search" beschrieben abgearbeitet. Der Suchalgorithmus ist in ausprogrammierter Form als Anlage 11: Codebeispiel der Suche – Suchalgorithmus angefügt.

Für die Umsetzung der Suche waren neue Klassen mit den Namen „TableCollection“, „Table“ und „Field“ notwendig. Ein dazugehöriges Klassendiagramm ist unter Anlage 8: Klassendiagramm zu TableCollection, Table und Field hinterlegt. Die Objekte dieser Klassen werden mit den zu durchsuchenden Tabellen und Feldern gefüllt. Derzeit ist darin jedoch nur eine Auswahl dieser enthalten, da in einer weiteren Version der automatische Import aus den Metadaten der Datenbank geplant ist.

6.3.7 Reporting – Auswertung

Ein weiterer Menüpunkt ist die Auswertungsseite. Diese stellt eine Unterstützung des IT-Leiters bzw. der Mitarbeiter dar, die die Tickets bearbeiten. Hier haben nur Techniker und Administratoren Zugriff. Dort zu finden sind drei Tabs mit unterschiedlichen Auswertungen.

Tab 1: Hier ist die aktuelle Ist-Situation der Tickets für eine bestimmte Woche abgebildet. Der Zeitraum ist einstellbar auf Jahr und Monat bis zu den einzelnen Kalenderwochen.

Tab 2: Im zweiten Tab findet man eine Darstellung der durchschnittlichen Reaktions- und Bearbeitungszeit der Tickets in Tagen.

Tab 3: Dieses Diagramm zeigt, welcher Mitarbeiter Probleme welcher Art hatte.

Ziel der Auswertungen ist eine unkomplizierte Aufklärung über die aktuelle Situation im Unternehmen. Somit können mögliche Notstände, z. B. anhand häufig auftretender Ticket-Beschreibungen oder Kategorie-Einteilungen, schnell erkannt, als auch entsprechende Gegenmaßnahmen eingeleitet werden. Ein Szenario könnte immer wiederkehrende Hardware-

probleme sein. Außerdem helfen die ausgewerteten Informationen, z. B. die Reaktionszeit auf die Tickets, dabei, einen kontinuierlichen Optimierungsprozess anzustreben. Ein Ausschnitt des Programmiercodes ist im Anhang zu finden.

6.4 Umsetzung der Webseitenstruktur

Die Umsetzung der im Pflichtenheft unter 0 **Struktur der Webseite** vorgegebenen Architektur ist wie in Abbildung 6-5: Umsetzung der Webseitenstruktur dargestellt realisiert.

Ersteller	Kategorie	Priorität	Status	Bearbeiter	Betreff	Problembeschreibung	Anhänge	Erstelldatum	Letzte Änderung
Martin Krockert	Peripherie	Niedrig	Offen	Solveig Thomas	Mein PC geht nicht an!	Habe schon alles versucht	Keine	18.02.2014	20.02.2014
Ellie Phant	Software	Mittel	In Arbeit	Martin Krockert	Kann mich nicht Einloggen.	Anmelddaten sind vermutlich inkorrekt.	Keine	10.01.2014	10.01.2014
Luzi Groggy	Hardware	Hoch	In Arbeit	Jens Hetze	Drucker kaputt.	Error 3485 erscheint auf dem Display nach dem Anschalten.	Keine	10.01.2014	13.01.2014
Cheyenne Newingham	Hardware	Hoch	Geschlossen	Jens Hetze	Rechnergeschwindigkeit	Mein Rechner benötigt morgens fast 15 Minuten bis er arbeitsbereit ist, Probleme bei der Start des Outlook's u. Compass. Welche Möglichkeiten der Geschwindigkeitserhöhung bestehen?	Keine	18.02.2014	19.02.2014
Sau Beeks	Software	Niedrig	Geschlossen	Jens Hetze	Email Signatur	Problem doppelzeilige Anzeige der Signatur	Keine	25.02.2014	28.02.2014
Tracie Sacks	Peripherie	Niedrig	Geschlossen	Jens Hetze	Zählerstandsmeldung an Saxocom	Eimal im Quartal müssen die Zählerstände der Drucker und Kopiergeräte abgelesen werden. Die Werte sind in den entsprechenden Excel Tabellen einzutragen, abzuspeichern und an Saxocom zu schicken.	Keine	19.01.2014	21.01.2014
Shaun Ellen	Hardware	Niedrig	In Arbeit	Solveig Thomas	Anrufbeantworter	Bitte um Einweisung in die Verwendung des Anrufbeantworters.	Keine	15.12.2013	20.12.2013

Abbildung 6-5: Umsetzung der Webseitenstruktur

7 Überprüfung und Wartung

7.1 Testablauf

Im Testablauf wurden auf drei verschiedene Kategorien eingegangen, das Design, die Funktionsfähigkeit und die Einhaltung der Rechte. Die Tests werden zudem unter den drei am meisten benutzen Browsern – „Mozilla Firefox“, „Google Chrome“ und „Microsoft Internet Explorer“ – durchgeführt. Dazu wurde für den Test der Software ein Testplan erstellt (siehe Anlage 13: Testplanvorlage für das Helpdeskssystem).

7.2 Modultests

Der unter 7.1 Testablauf beschriebene Ablauf wurde für die einzelnen Module individuell angepasst. Alle Module werden auf ein einheitliches Layout, homogenes Design und gleiche Handhabbarkeit geprüft.

7.2.1 Users – Benutzer

Die Benutzer müssen auf Anzeig-, Erstell-, Bearbeit- und Archivierbarkeit getestet werden. Benutzer dürfen nicht gelöscht werden, da dadurch die referentielle Integrität verletzt wird. Des Weiteren muss beim Test des Moduls „Benutzer“ die Auswahlmöglichkeit der Rechte geprüft werden. Nur Administratoren dürfen das Modul „Benutzer“ verwenden.

7.2.2 Rights – Rechte

Die Anwendung der Rechte wird in den einzelnen Modulen geprüft. Das Modul „Rechte“ muss nicht kontrolliert werden, da die darin enthaltenen Rechte fest im System verankert sind.

7.2.3 Announcements – Benachrichtigungen

Die Benachrichtigungen müssen auf Anzeig-, Erstell-, Bearbeit- und Löschbarkeit getestet werden. Annoucements werden nach der Gültigkeitsprüfung des Datums nicht mehr angezeigt, sobald diese nicht bestanden wurde. Administratoren und Techniker dürfen im Modul „Benachrichtigungen“ lesen, bearbeiten und löschen. Anwender dürfen die Benachrichtigungen nur lesen.

7.2.4 Tickets

Die Tickets müssen auf Anzeig-, Erstell-, Bearbeitbarkeit getestet werden. Ein Löschen von Tickets ist im System nicht vorgesehen. Zudem ist die Unterscheidung nach Benutzerrechten zu prüfen. Administratoren sehen alle Tickets, Techniker und Anwender sehen die selbst erstellten und die als Bearbeiter zugewiesenen Tickets.

7.2.5 Solutions – Lösungen

Die Lösungen müssen auf Anzeig-, Erstell-, Bearbeit- und Löschbarkeit getestet werden. Es wird geprüft, ob nur die öffentlichen Lösungen den Anwendern sichtbar sind. Als Besonderheit sind in diesem Modul das Anhängen von Anlagen und deren korrekte Ausgabe zu prüfen.

7.2.6 Search – Suche

Der Test der Suche kann auf Grundlage der unter Anlage 6: Entscheidungstabelle zum Modul "Search" eingefügten Tabelle stattfinden. Werden die unter Aktionen angegebenen Rückgaben wie in der Entscheidungstabelle ausgeführt, ist der Test der Suche erfolgreich.

7.2.7 Reporting – Auswertung

Im Modul „Auswertung“ werden die Korrektheit der angezeigten Werte und der Export der Datentabelle in verschiedene Ausgabeformate geprüft. Dabei ist ebenfalls zu testen, dass nur Administratoren und Techniker die Auswertung lesen können.

7.3 Testergebnisse

Die Testergebnisse beruhen auf den ausgefüllten Testplänen. Diese sind in der Anlage 14: Ausgefüllte Testpläne zu den einzelnen Modulen ersichtlich.

Bei den Tests ist aufgefallen, dass die Diagrammaktualisierung in der Auswertung bei vorheriger leerer Wochenauswertung nicht wieder aktualisiert. Außerdem werden beim Seitenwechsel, d. h. Verlassen und Wiederkehr der Seite Tickets, die Ticketfilter intern nicht zurückgesetzt. Es wird weiterhin die gefilterte Liste angezeigt. Des Weiteren werden die Datumswerte in der Ticketliste bei der Sortierung nicht als Datumswert sondern als Stringwert interpretiert.

Das Programm ist lauffähig und erfüllt die gegebenen Anforderungen. Die festgestellten Mängel werden in der nächsten Version mit einem Patch behoben. Außerdem können die gewünschten Funktionserweiterungen (siehe 8.2 Erweiterungsmöglichkeiten) darin enthalten sein.

8 Fazit und Ausblick

8.1 Projektfazit

Das Projekt verlief vollständig im vorgegebenen Zeit- und Kostenrahmen. Durch das schreiben der Applikation als Webanwendung ist eine freie Skalierbarkeit des Systems gegeben. Die Wartung der Software ist durch die vollständige Verwaltung aller Einstellungen über die Webseite und durch Benutzer mit entsprechenden Rechten möglich.

Zur regelmäßigen Wartung ist täglich ein Backup der Datenbank zu erstellen. Ein Backup des Webservers ist nur bei Änderungen an der Webseite selbst notwendig.

Die Schulung der Mitarbeiter wurde wie geplant in der ersten März-Woche in kleinen abteilungsinternen Gruppen durchgeführt.

8.2 Erweiterungsmöglichkeiten

Die modulare Struktur des Programms ermöglicht eine schnelle und einfache Implementierung von neuen Modulen. Auch der Einfluss von sogenannten „Side-Effects“ wird durch diese Struktur minimiert.

Nach der erfolgreichen Einführung sollten Überlegung zu zusätzlichen Modulen getroffen werden. Sinnvolle Erweiterungen wären zum Beispiel eine Geräteliste in Kombination mit Verträgen. So können Geräte an Benutzer geknüpft werden und einen entsprechenden Servicevertrag zugewiesen bekommen. Das ermöglicht eine Rückverfolgung von häufig defekten Geräten, als auch eine schnelle Abarbeitung von Störungen durch das Vorhandensein von allen benötigten Informationen.

Zudem ist eine Erweiterung des Rechtesystems auf ein dynamisch anpassbares System notwendig, d. h. wenn weitere Funktionen und Module hinzukommen kann man diesen Benutzergruppen zuordnen.

Mit dieser Erweiterung sollte auch die Suche so angepasst werden, dass die Tabellensammlung aus der Mastertabelle der Datenbank erstellt wird. Außerdem sind die anzuseigenden und zu durchsuchenden Felder an Hand von Signalfeldern zu bestimmen.

Literaturverzeichnis

Haldar, S. (2009). *Inside SQLite*. O'Reilly Media.

Martin, R. C. (2009). *Clean Code*. mitp.

Quellenverzeichnis

ICEsoft Technologies Inc. (kein Datum). Abgerufen am 1. Februar 2014 von
<http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>

primetec. (kein Datum). Abgerufen am 1. Februar 2014 von <http://www.primefaces.org>

redhat. (kein Datum). Abgerufen am 1. Februar 2014 von <http://www.jboss.org/richfaces>

Verzeichnis der Abbildungen

Abbildung 4-1: Struktur der Webseite	14
Abbildung 6-1: Codebeispiel des Triggers „tickets_insert“	24
Abbildung 6-2: Codebeispiel des Triggers „tickets_changed“	25
Abbildung 6-3: Projektstruktur des Helpdesks	25
Abbildung 6-4: Vererbungsmodell	26
Abbildung 6-5: Umsetzung der Webseitenstruktur	29

Verzeichnis der Tabellen

Tabelle 3-1: Vorstellung der Person Solveig Thomas	9
Tabelle 3-2: Vorstellung der Person Martin Krockert	9
Tabelle 3-3: Vorstellung der Person Jens Hetze.....	10
Tabelle 3-4: Vorstellung der Person Johannes Schaffrath	10
Tabelle 4-1: Bewertung des Ist-Zustandes	12
Tabelle 4-2: Anforderungen an das Serversystem.....	16
Tabelle 4-3: Anforderungen an die Clients.....	16
Tabelle 5-1: Entwicklungsumgebung „IntelliJ IDEA“	18
Tabelle 5-2: Entwicklungsumgebung „Netbeans“	19
Tabelle 5-3: Entwicklungsumgebung „Eclipse“	19
Tabelle 5-4: Entscheidungsmatrix zur Auswahl der Entwicklungsumgebung	20
Tabelle 5-5: Framework „RichFaces“ (redhat)	21
Tabelle 5-6: Framework „IceFaces“ (ICEsoft Technologies Inc.)	21
Tabelle 5-7: Framework „PrimeFaces“ (primetec).....	22
Tabelle 5-8: Entscheidungsmatrix zur Auswahl des Frameworks	22
Tabelle 5-9: Umsetzung der MVC-Architektur	23

Verzeichnis der Anlagen

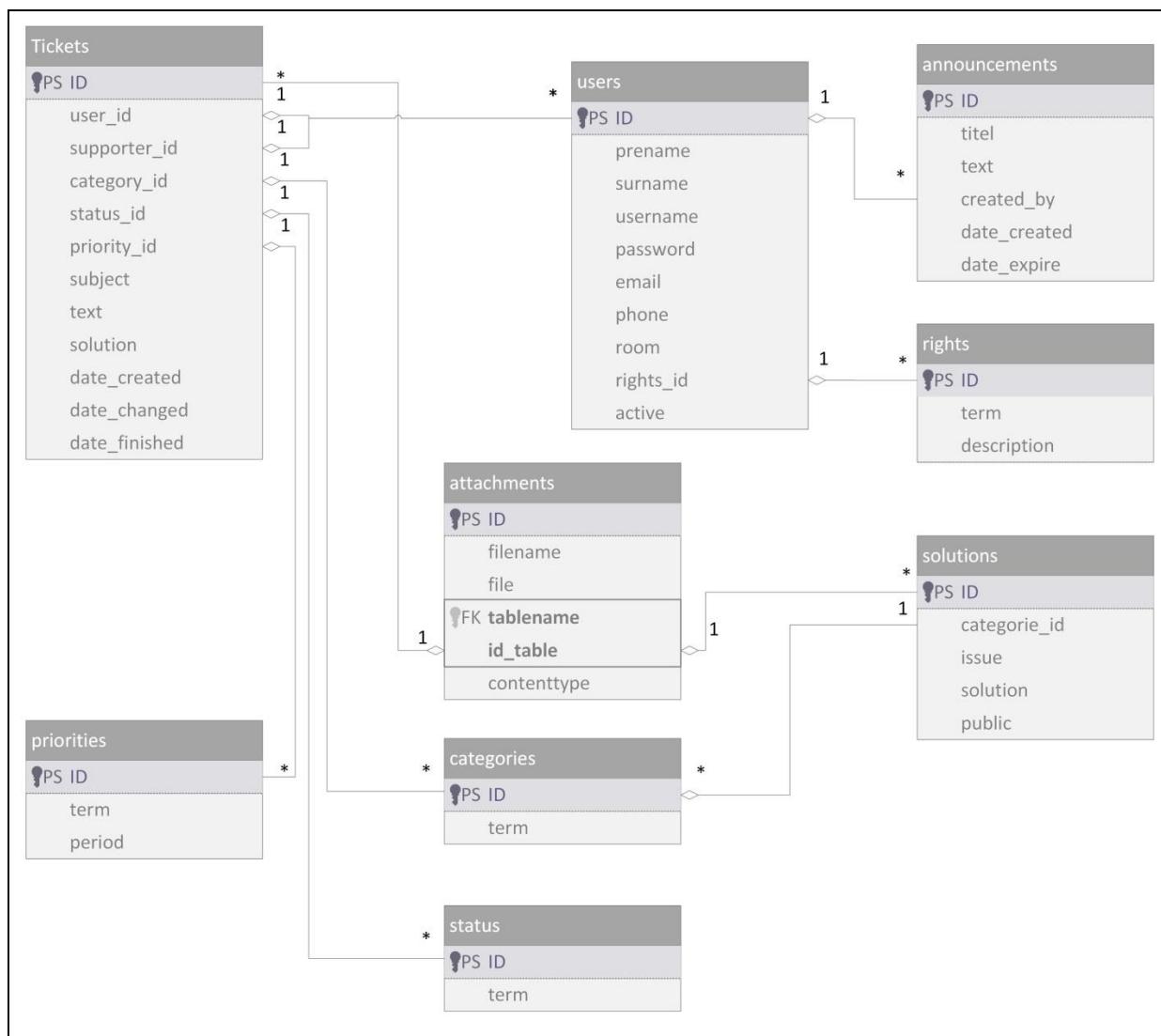
Anlage 1: Datenbankschema.....	39
Anlage 2: ERM	40
Anlage 3: Systemgrenzen des Helpdesksystems	41
Anlage 4: Use-Case-Diagramm des Helpdesksystems.....	42
Anlage 5: Rechtematrix des Helpdesksystems	43
Anlage 6: Entscheidungstabelle zum Modul "Search".....	47
Anlage 7: Aktivitätsdiagramm zum Modul "Search"	48
Anlage 8: Klassendiagramm zu TableCollection, Table und Field	49
Anlage 9: Codebeispiel der Data Access Object-Klasse.....	50
Anlage 10: Codebeispiel des Dateiuploads	51
Anlage 11: Codebeispiel der Suche – Suchalgorithmus	52
Anlage 12: Codebeispiel des Reportings.....	53
Anlage 13: Testplanvorlage für das Helpdesksystem	54
Anlage 14: Ausgefüllte Testpläne zu den einzelnen Modulen.....	60

Anlagen

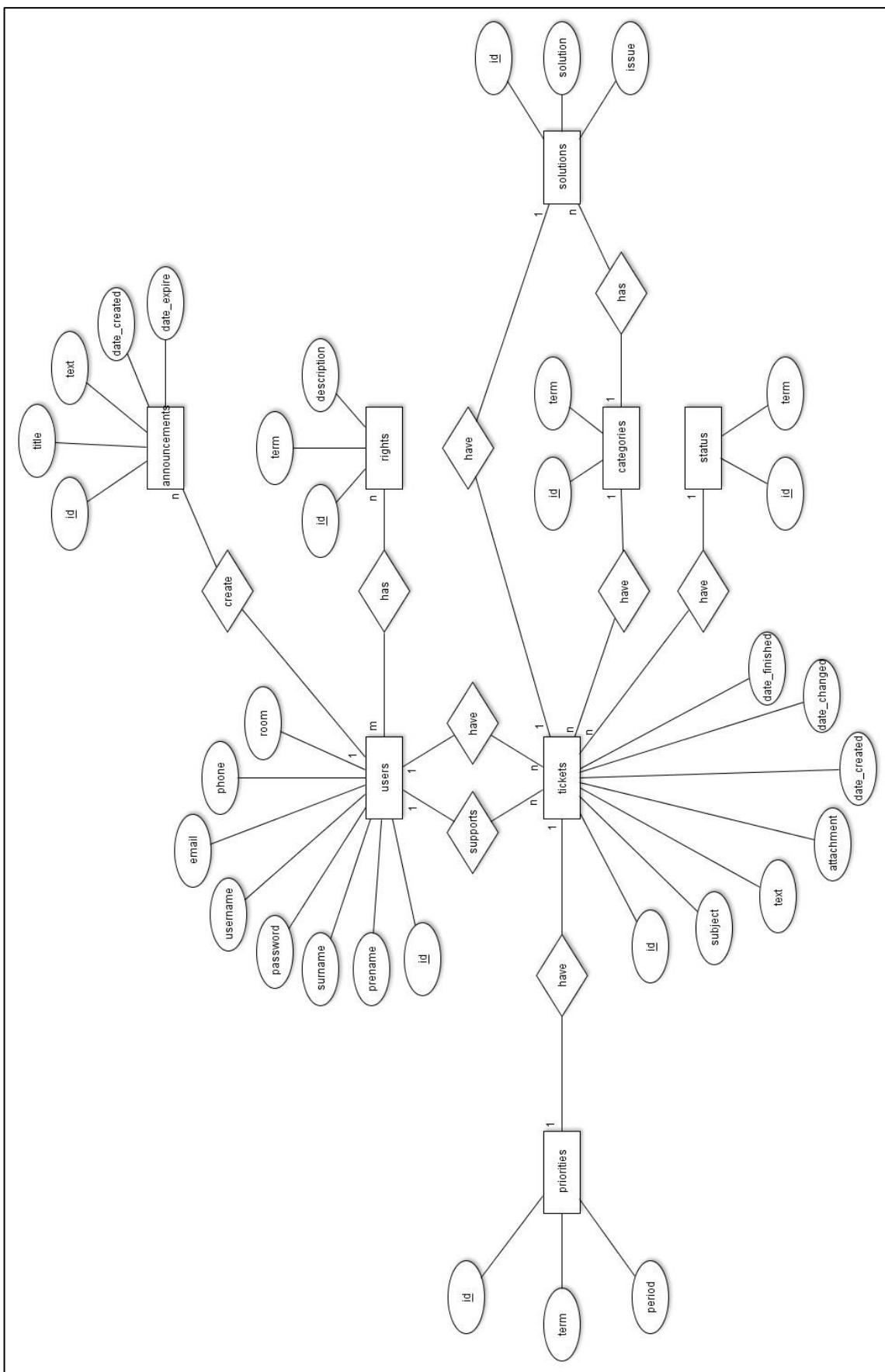
Auf den folgenden Seiten finden Sie die Anlagen zum Dokument.

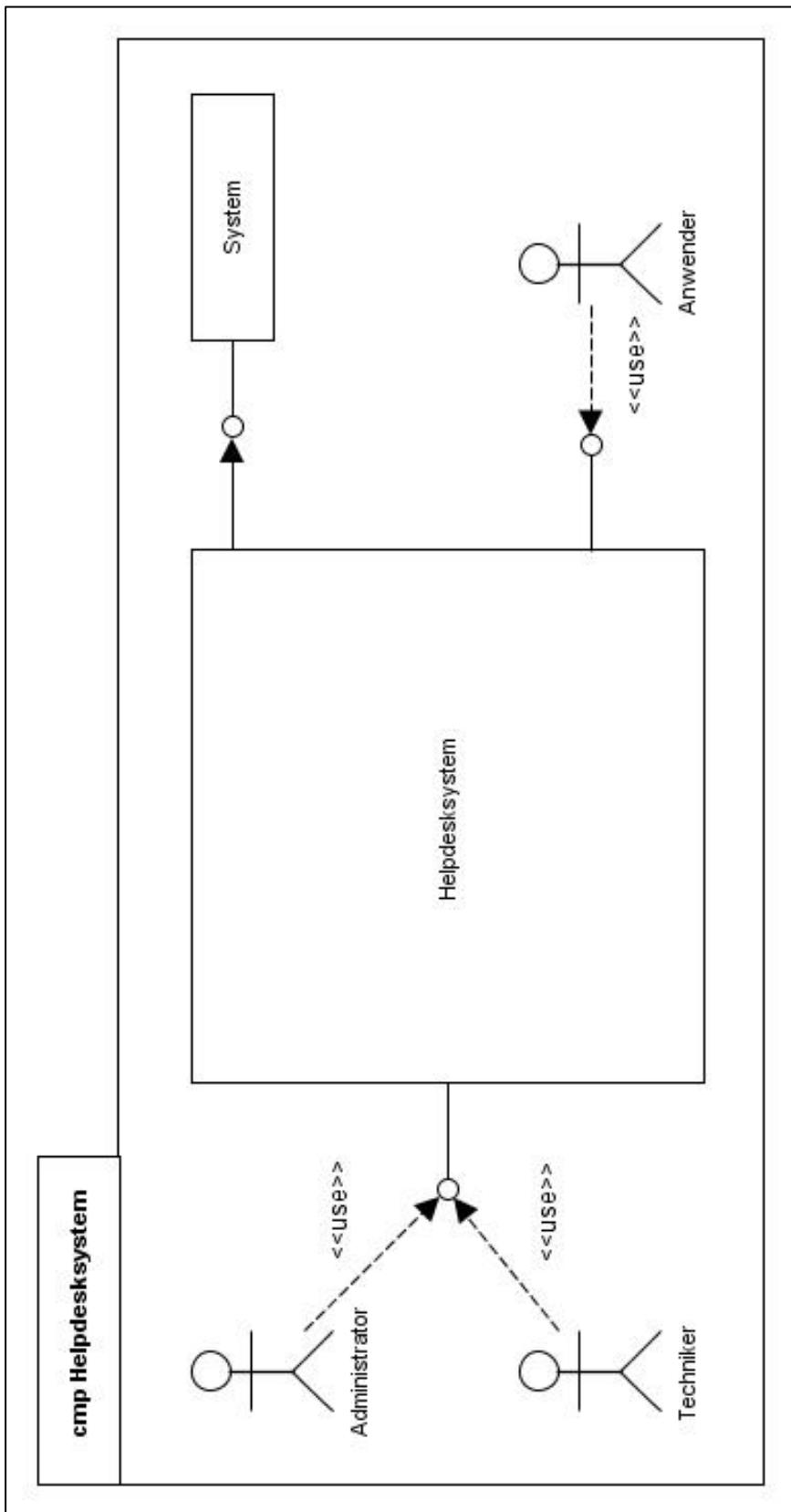
Datenbankschema

Anlage 1

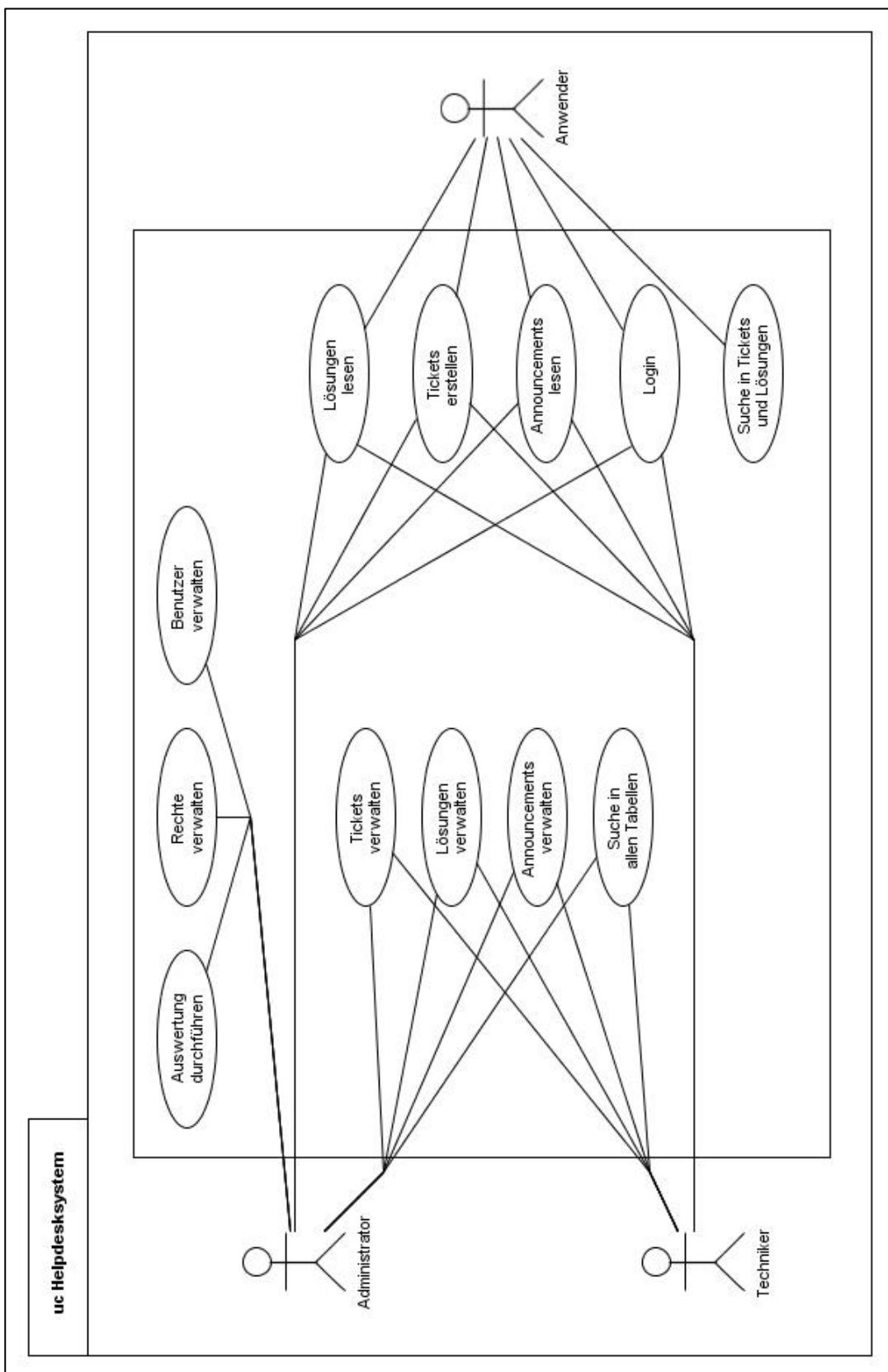


Anlage 1: Datenbankschema





Anlage 3: Systemgrenzen des Helpdesksystems



Anlage 4: Use-Case-Diagramm des Helpdesksystems

Vorgang	Administrator	Techniker	Anwender
Login	X	X	X
Rechte verwalten	X		
Benutzer verwalten	X		
Announcements lesen	X	X	X
Announcements verwalten	X	X	
Tickets erstellen	X	X	X
Tickets verwalten	X	X	
Lösungen verwalten	X	X	
Lösungen lesen	X	X	X
Suche in allen Tabellen	X	X	
Suche in Tickets und Lösungen			X
Auswertung durchführen	X		

Anlage 5: Rechtematrix des Helpdesksystems

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J
1 Hat Benutzerrecht "Admin"?																				
2 Hat Benutzerrecht "Techniker"?																				
3 Hat Benutzerrecht "Anwender"?																				
4 Suchstring ist eingegeben?																				
5 Führt eine Volltextsuche aus? (N=Erweiterte Suche)																				
6 Es wird die Tabelle "Tickets" durchsucht?																				
7 Es wurde eine bestimmte Suchspalte gewählt?																				
8 Es wurde eine bestimmte Sortierspalte gewählt?																				
9 Es wird aufsteigend sortiert?																				
A Ausführung der Suche																				
A Ergebnisse aus allen Tabellen																				
A Ergebnisse aus Tabellen "Lösungen" und "Tickets"																				
A Ergebnisse aus der gewählten Tabelle																				
H Spalten- und Sortierauswahl mit Spalten der Ausgabe																				
H Spalten- und Sortierauswahl mit Spalten der gewählten Tabelle																				
A Suchergebnis enthält nur eigene Tickets																				
A Suchergebnis enthält Tickets von verschiedenen Benutzern																				
A Suche in der allen Spalten																				
A Suche in der gewählten Spalte																				
A Sortierung nach Spalte "ID"																				
A Sortierung nach der gewählten Spalte																				
A Suchergebnis aufsteigend sortiert																				
A Suchergebnis absteigend sortiert																				

Legende:

A = Ausführung, d. h. diese Aktion wird beim Ausführen der Suchanfrage gebraucht.
H = Handling, d. h. diese Aktion wird bei der Eingabe der Suche gebraucht.

Legende:

A = Ausführung, d. h. diese Aktion wird beim Ausführen der Suchanfrage gebraucht.
H = Handling, d. h. diese Aktion wird bei der Eingabe der Suche gebraucht.

		41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
	Bedingungen	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1	Hat Benutzerrecht "Admin"?	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J
2	Hat Benutzerrecht "Techniker"?																				
3	Hat Benutzerrecht "Anwender"?																				
4	Suchstring ist eingegeben?																				
5	Führt eine Volltextsuche aus? (N=Erweiterte Suche)																				
6	Es wird die Tabelle "Tickets" durchsucht?																				
7	Es wurde eine bestimmte Suchspalte gewählt?																				
8	Es wurde eine bestimmte Sortierspalte gewählt?																				
9	Es wird aufsteigend sortiert?																				
	Ausführung der Suche	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	A Ergebnisse aus allen Tabellen																				
	A Ergebnisse aus Tabellen "Lösungen" und "Tickets"																				
	A Ergebnisse aus der gewählten Tabelle	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	H Spalten- und Sortierauswahl mit Spalten der Ausgabe																				
	H Spalten- und Sortierauswahl mit Spalten der gewählten Tabelle	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	A Suchergebnis enthält nur eigene Tickets																				
	A Suchergebnis enthält Tickets von verschiedenen Benutzern	X																			
	A Suche in der allen Spalten	X																			
	A Suche in der gewählten Spalte	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	A Sortierung nach Spalte "ID"	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	A Sortierung nach der gewählten Spalte																				
	A Suchergebnis aufsteigend sortiert																				
	A Suchergebnis absteigend sortiert	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Legende:

A = Ausführung, d. h. diese Aktion wird beim Ausführen der Suchanfrage gebraucht.
H = Handling, d. h. diese Aktion wird bei der Eingabe der Suche gebraucht.

Entscheidungstabelle zum Modul "Search"

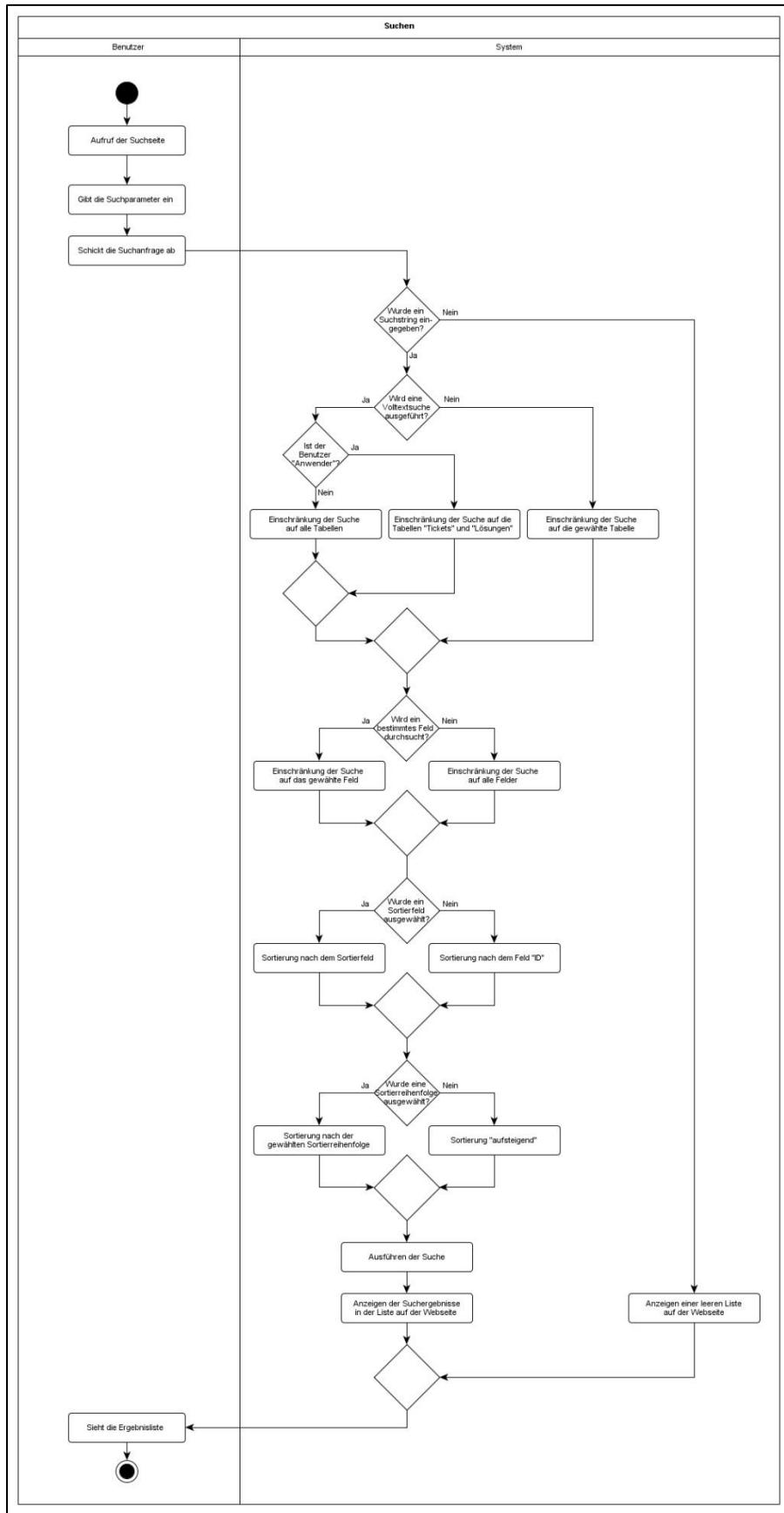
		Bedingungen	61	62	63	64	65	66	67	68	69	70	71	72	73
1	Hat Benutzerreicht "Admin"?		N	N	N	N	N	N	N	N	N	N	N	N	N
2	Hat Benutzerreicht "Techniker"?		N	N	N	N	N	N	N	N	N	N	N	N	N
3	Hat Benutzerreicht "Anwender"?		J	J	J	J	J	J	J	J	J	J	J	J	J
4	Suchstring ist eingegeben?		J	J	J	J	J	J	J	J	J	J	J	J	J
5	Führt eine Volltextsuche aus? (N=Erweiterte Suche)		N	N	N	N	N	N	N	N	N	N	N	N	N
6	Es wird die Tabelle "Tickets" durchsucht?		J	J	J	J	J	J	J	J	J	J	J	J	J
7	Es wurde eine bestimmte Suchspalte gewählt?		J	N	N	N	J	J	J	J	J	J	J	J	J
8	Es wurde eine bestimmte Sortierspalte gewählt?		N	J	N	N	J	J	J	J	J	J	J	J	J
9	Es wird aufsteigend sortiert?		N	J	N	J	N	J	N	J	N	J	N	J	N
A	Ausführung der Suche		X	X	X	X	X	X	X	X	X	X	X	X	X
A	Ergebnisse aus allen Tabellen														
A	Ergebnisse aus Tabellen "Lösungen" und "Tickets"														
A	Ergebnisse aus der gewählten Tabelle		X	X	X	X	X	X	X	X	X	X	X	X	X
H	Spalten- und Sortierauswahl mit Spalten der Ausgabe														
H	Spalten- und Sortierauswahl mit Spalten der gewählten Tabelle		X	X	X	X	X	X	X	X	X	X	X	X	X
A	Suchergebnis enthält nur eigene Tickets		X	X	X	X									
A	Suchergebnis enthält Tickets von verschiedenen Benutzern														
A	Suche in der allen Spalten		X	X	X	X									
A	Suche in der gewählten Spalte		X				X	X	X	X	X	X	X	X	X
A	Sortierung nach Spalte "ID"		X				X	X	X	X	X	X	X	X	X
A	Sortierung nach der gewählten Spalte			X	X			X	X	X	X	X	X	X	X
A	Suchergebnis aufsteigend sortiert			X	X			X	X	X	X	X	X	X	X
A	Suchergebnis absteigend sortiert		X	X	X										

Legende:

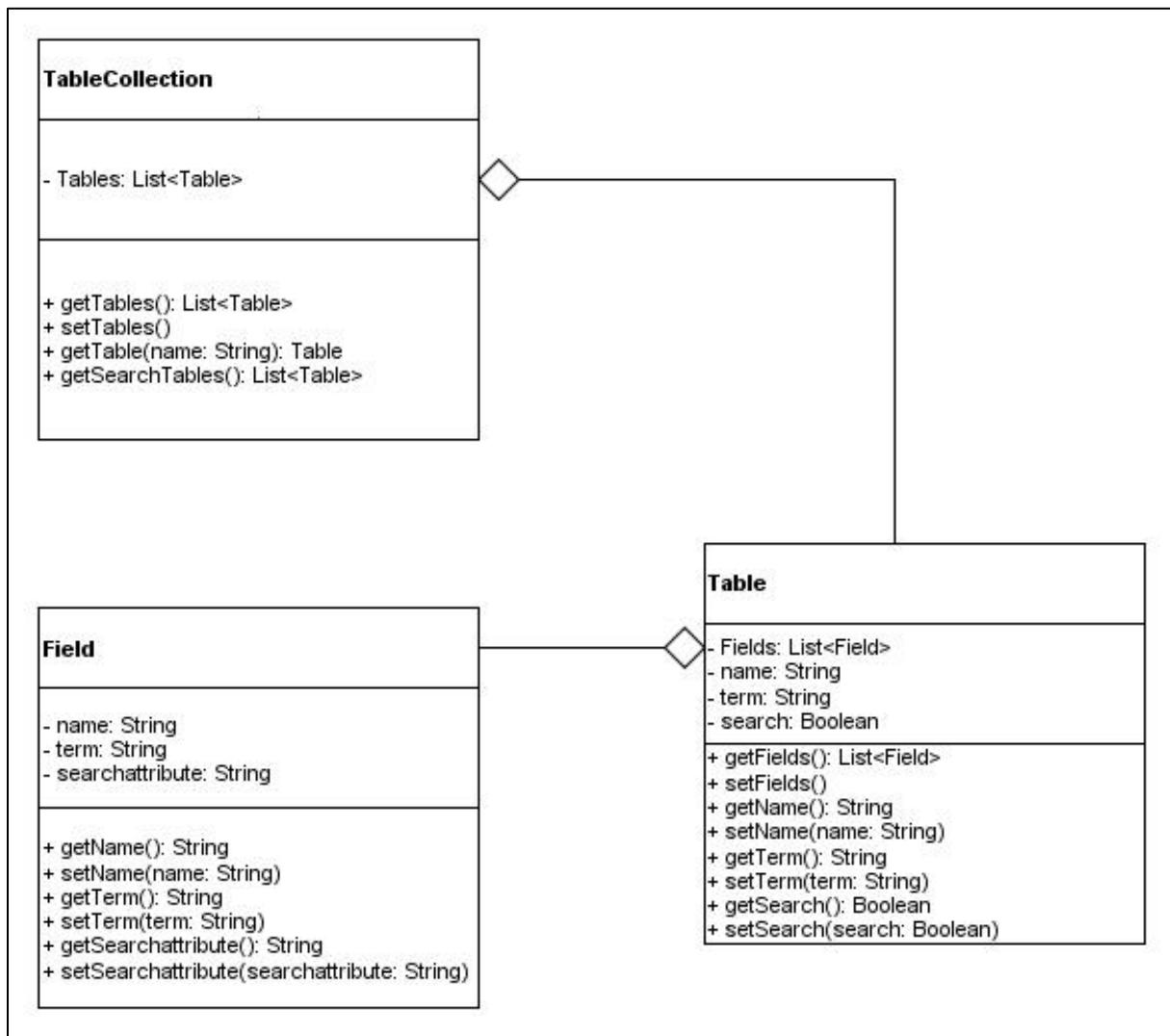
A = Ausführung, d. h. diese Aktion wird beim Ausführen der Suchanfrage gebraucht.
H = Handling, d. h. diese Aktion wird bei der Eingabe der Suche gebraucht.

Aktivitätsdiagramm zum Modul "Search"

Anlage 7



Anlage 7: Aktivitätsdiagramm zum Modul "Search"



Anlage 8: Klassendiagramm zu TableCollection, Table und Field

```
/**  
 *  
 * @author Martin Krockert  
 * Abstrakte Klasse zum vererben der Grundstuktur für Data Access Objects  
 * @param <T> Generische Klasse für Alle DAO's  
 */  
public abstract class AbstractDAO<T> implements Serializable {  
    // Variablen Deklaration  
    public Connection currentCon = null;  
    public ResultSet rs = null;  
    public Statement stmt = null;  
    public T selectedBean;  
    private Class<T> itemClass;  
    private List<InterfaceBean> items = new ArrayList<InterfaceBean>();  
  
    // BEGIN: Getter & Setter  
    public AbstractDAO(Class<T> itemClass) {  
        this.itemClass = itemClass;  
    }  
  
    public T getSelectedBean() {  
        return selectedBean;  
    }  
  
    public void setSelectedBean(T selected) {  
        this.selectedBean = selected;  
    }  
  
    public List<InterfaceBean> getItems() {  
        return items;  
    }  
    // END: Getter & Setter  
  
    // Setzen des Ausgewählten Beans anhand der ID  
    public void selectBean(int id) {  
        for (InterfaceBean bean : items) {  
            if(bean.getId() == id) {  
                selectedBean = (T)bean;  
            }  
        }  
    }  
  
    /**  
     * @author Martin Krockert  
     * Interface für Beans zum sammeln in der  
     * Generischen list des Data Access Objekts  
     */  
  
    public interface InterfaceBean {  
        public int getId();  
        public void save();  
    }  
}
```

Anlage 9: Codebeispiel der Data Access Object-Klasse

```
public void handleFileUpload(FileUploadEvent event) {  
    try {  
        AttachmentBean fileBean = new AttachmentBean();  
        // speichern der Details zur Datei im Bean  
        fileBean.setFilename(event.getFile().getFileName());  
        fileBean.setContentType(event.getFile().getContentType());  
  
        // speichern der Datei Als Inputstream - broken BUG PF 4RC1  
        InputStream inputstream = event.getFile().getInputStream();  
  
        // Workaround  
        // Erstellen einer Temporären Datei.  
        File tmp = File.createTempFile(fileBean.getFilename(), "tmp");  
  
        // BEGIN: Schreiben der Datei in einen ByteArray  
        FileOutputStream fos = new FileOutputStream(tmp);  
        int read;  
        byte[] file = new byte[1048576];  
        while ((read = inputstream.read(file)) != -1) {  
            fos.write(file, 0, read);  
        }  
        fos.close();  
  
        fileBean.setFile(file);  
        fileBean.setFromDB(false);  
        items.add(fileBean);  
        // ENDE  
    } catch (IOException ex) {  
        Logger.getLogger(AttachmentBean.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

Anlage 10: Codebeispiel des Dateiuploads

Codebeispiel der Suche – Suchalgorithmus

Anlage 11

```

/*
 * Diese Methode erstellt die SQL-Abfrage aus den eingegebenen Werten der erweiterten Suche.
 * Falls nichts eingegeben wurde, wird eine leer Ergebnisliste angezeigt.
 * Es wird die einfache Suche ausgeführt, wenn lediglich das Attribut searchstring gesetzt ist.
 * Die einfache Suche durchsucht als Rolle "Anwender" nur die Tickets und Lösungen.
 * Als Rolle "Admin" oder "Techniker" werden alle Tabellen durchsucht.
 * @author Jens
 * @return SQL-Abfrage als String
 */
public String createSql() throws SQLException
{
    String sql="";
    if (this.table == null) (this.table="");
    if (this.field == null) (this.field="");
    if (this.searchstring == null) (this.searchstring="");
    if (this.sort_field == null || this.sort_field.equals("") == true) (this.sort_field="id");
    if (this.sort == null || this.sort.equals("") == true) (this.sort="ASC");

    // Falls Searchstring leer ist, wird eine leere SQL-Abfrage zurückgegeben
    if (this.searchstring.equals("") == true)
    { return sql; }

    TableCollection col = new TableCollection();
    int x = col.getTables().size(); // Anzahl der Einträge in TableCollection
    int z = 0; //Zur Zählung der durchsuchten Tabellen

    for(int i=0;i<x;i++)
    {
        Table tab = col.getTable(i);
        String title = "";
        String description = "";

        // Prüfung auf Volltext oder Tabellensuche in diesem Index der TableCollection oder Teilsuche in Lösungen und Tickets,
        // wenn nur Anwenderrechte vorhanden sind.
        if ((this.table.equals("") == false && tab.getName().equals(this.table) == true) //gewählte Tabelle ist die aktuelle Tabellen
        || (this.table.equals("") == true && tab.getSearch() == true && (Helper.isAdmin() == true || Helper.isTechnician() == true))
        || (this.table.equals("") == true && tab.getSearch() == true && Helper.isUser() == true
        && (tab.getName().equals("solutions")==true || tab.getName().equals("tickets"))))

        {
            // Mehrere Tabellen mit UNION zusammenfügen bei der Volltextsuche
            if (this.table.equals("") == true && z > 0) { sql = sql + " UNION "; }

            // Zusammenbauen der WHERE-Variable für die SQL-Abfrage
            int y = tab.getFields().size();
            String where="";

            if (Helper.isUser() == true && tab.getName().equals("tickets") == true)
            { where = where + "("; }

            for (int j = 0 ; j < y ; j++)
            {
                Field feld = tab.getFields().get(j);

                if (this.field.equals("") == true || (this.field.equals("") == false && feld.getName().equals(this.field) == true))
                {
                    if (this.field.equals("") == true && j>0)
                    { where = where + " OR "; }

                    where = where + feld.getName() + " LIKE '%" + searchstring + "%' ";
                }
            }

            // Wenn der Benutzer die Rolle Anwender hat, dann bekommt er nur seine Tickets ausgegeben.
            if (Helper.isUser() == true && tab.getName().equals("tickets") == true)
            { where = where + " AND users_id=" + Helper.getId() + " "; }

            // Besetzung, welche Attribute dieser Tabelle der title und welche die description werden.
            for (int k=0; k<tab.getFields().size(); k++)
            {
                Field feld = tab.getFields().get(k);
                if (feld.getSearchattribute().equals("title") == true)
                { title = feld.getName(); }

                if (feld.getSearchattribute().equals("description") == true)
                { description = feld.getName(); }
            }

            // Erstellen der SQL für die aktuelle Tabelle
            sql = sql + "SELECT ID as id, " + title + " AS title, " + description + " AS description, '" + tab.getTerm() + "' AS table";
            sql = sql + " FROM " + tab.getName();
            // Where nur bei Einzelsuche und Volltextsuche ohne ausgewähltem Feld, wenn kein UNION stattfindet
            if (this.table.equals("") == false || this.field.equals("") == true) { sql = sql + " WHERE " + where; }

            // Sortierung nur bei Einzelsuche, wenn kein UNION stattfindet
            if (this.table.equals("") == false) { sql = sql + " ORDER BY " + this.sort_field + " " + this.sort; }

            //Hochzählen der durchsuchten Tabellen
            if (tab.getSearch() == true) { z++; }
        }
    }

    // Where der Volltextsuche, nachdem das UNION zusammengebaut ist
    if (this.table.equals("") == true && this.field.equals("") == false)
    { sql = "SELECT * FROM (" + sql + ") WHERE " + this.field + " LIKE '%" + this.searchstring + "%'; " }

    // Sortierung der Volltextsuche, nachdem das UNION zusammengebaut ist
    if (this.table.equals("") == true)
    { sql = sql + " ORDER BY " + this.sort_field + " " + this.sort; }

    return sql;
}

```

Anlage 11: Codebeispiel der Suche – Suchalgorithmus

Beispielhafter Ausschnitt des Programmcodes des Moduls „Reporting“

Ein besonderes Gimmick im Modul Reporting ist, dass der Administrator die Möglichkeit hat sich die abgebildete Tabelle als XML oder CSV Dateien ausgeben zu lassen. Damit könnte er erweiterte Analyse anfertigen bzw. die Daten an anderer Stelle weitergeben und verarbeiten.

Genutzt wurde hierzu der von JSF bereitgestellte DataExporter, welcher in der View Datei der MVC-Architekt implementiert ist. Sie nennt sich reports.xhtml.

```

<p:column>
    <f:facet name="header">
        <h:outputText value="Ticketanzahl" />
    </f:facet>
    <h:outputText value="#{cat.tickets}" />
</p:column>
</p:dataTable>

<h:panelGrid columns="2" >
    <p:panel header="Export Data">
        <h:commandLink>
            <p:graphicImage value="/resources/images/csv.png" />
            <p:dataExporter type="csv" target="tbl" fileName="cat" />
        </h:commandLink>

        <h:commandLink>
            <p:graphicImage value="/resources/images/xml.png" />
            <p:dataExporter type="xml" target="tbl" fileName="cat" />
        </h:commandLink>
    </p:panel>
</h:panelGrid>
```

Um die Tabelle zu füllen, wurde im Hintergrund in der ReportDAO-Klasse, welche eine Tochterklasse der AbstractDAO-Klasse ist, eine Methode implementiert, die die gewünschten Daten aus der Datenbank abfragt und diese an die ReportBean-Klasse, zur Datenzwischenlagerung, übergibt.

```

public List<ReportBean> initBeanCategories() throws SQLException {
    items.clear();
    currentCon = ConnectionManager.getConnection();
    try {
        stmt=currentCon.createStatement();
        rs = stmt.executeQuery("select u.surname, u.prename, c.term, count(*) as tickets "
                + "from tickets t "
                + "left join users u on u.ID = t.users_id "
                + "left join categories c on c.ID = t.categories_id "
                + "group by u.surname, u.prename, c.term");
        while (rs.next()) {
            ReportBean bean = new ReportBean();
            bean.setSurname(rs.getString("surname"));
            bean.setPrename(rs.getString("prename"));
            bean.setTerm(rs.getString("term"));
            bean.setTickets(rs.getInt("tickets"));
            items.add(bean);
        }
        rs.close();
        stmt.close();
    } catch (SQLException ex) {
        System.out.println("An Exception has occurred! " + ex);
    }
    currentCon.close();
    return items;
}
```

# Testfall	Implementation	Test bestanden?	Bemerkungen
Anzeige			
1 Einheitliches Design?			
2 Stimmen die Ausgabeformate?			
3 Rechtschreibung in Ordnung?			
Funktionen			
3 Funktionieren die Filter?			
4 Funktioniert die Sortierung?			
5 Funktioniert das Einfügen?			
6 Funktioniert das Bearbeiten?			
7 Funktioniert das Löschen?			
8 Haben die Auswahllisten die korrekten Werte?			
Unterschiede durch Rechte			
9 Hat der Administrator Rechte entsprechend der Rechtematrix?			
10 Hat der Techniker Rechte entsprechend der Rechtematrix?			
11 Hat der Anwender Rechte entsprechend der Rechtematrix?			

Testplan – Reporting		Geprüft von: Martin Krockert	Geprüft am: 02.03.2014	
# Testfall	Anzeige	Implementierung	Test bestanden?	Bemerkungen
	Anzeige			
	Einheitliches Design?	Ja	Ja	
	Stimmen die Ausgabeformate?	Ja	Ja	
	Rechtschreibung in Ordnung?	Ja	Ja	
	Funktionen			
	Funktionieren die Filter?	Nein		
	Funktioniert das Blättern?	Ja	Ja	Nur im 3. Tab vorhanden und da getestet.
	Funktioniert die Sortierung?	Nein		
	Funktioniert das Einfügen?	Nein		
	Funktioniert das Bearbeiten?	Nein		
	Funktioniert das Löschen?	Nein		
	Haben die Auswahllisten die korrekten Werte?	Ja	Ja	
	Modulspezifische Eigenschaften			
	Stichprobenkontrolle der angezeigten Werte und Diagramme in der Datenbank kontrollieren	Ja	Ja	
	Diagrammdarstellung korrekt?	Ja	Nein	Wochenauswertung: Beim Leeren der ResultSets funktioniert die Neuauswahl anderer Wochen nicht. (Diagramm wird trotz Daten nicht dargestellt.)
	Unterschiede durch Rechte			
	Hat der Administrator Rechte entsprechend der RechteMatrix?	Ja	Ja	
	Hat der Techniker Rechte entsprechend der RechteMatrix?	Ja	Ja	
	Hat der Anwender Rechte entsprechend der RechteMatrix?	Ja	Ja	

Testplan – Announcements		Geprüft von: Jens Heize	Geprüft am: 02.03.2014	
# Testfall	Anzeige	Implementierung	Test bestanden?	Bemerkungen
	Einheitliches Design?	Ja	Ja	
	Stimmen die Ausgabeformate?	Ja	Ja	
	Rechtschreibung in Ordnung?	Ja	Ja	
Funktionen				
	Funktionieren die Filter?	Nein		
	Funktioniert das Blättern?	Nein		
	Funktioniert die Sortierung?	Nein		
	Funktioniert das Einfügen?	Ja	Ja	
	Funktioniert das Bearbeiten?	Ja	Ja	
	Funktioniert das Löschen?	Ja	Ja	
	Haben die Auswahllisten die korrekten Werte?	Ja	Ja	
Modulspezifische Eigenschaften				
	Werden alle Benachrichtigungen bis zum aktuellen Datum angezeigt?	Ja	Ja	
	Eingabeformular auf Pflichtfelder prüfen	Ja	Ja	
	Korrekte Eingaben/Eingabevervalidierung	Ja	Ja	
	Funktioniert das Ein-/Ausblassen nicht relevanter Elemente?	Ja	Ja	
Unterschiede durch Rechte				
	Hat der Administrator Rechte entsprechend der Rechtematrix? Kann lesen/schreiben/löschen	Ja	Ja	
	Hat der Techniker Rechte entsprechend der Rechtematrix? Kann lesen	Ja	Ja	
	Hat der Anwender Rechte entsprechend der Rechtematrix?	Ja	Ja	

Testplan – Benutzer		Geprüft von: Solveig Thomas 02.03.2014			
#	Testfall	Anzeige	Implementierung	Test bestanden?	Bemerkungen
	Einheitliches Design?		Ja	Ja	
	Stimmen die Ausgabeformate?		Ja	Ja	
	Rechtschreibung in Ordnung?		Ja	Ja	
Funktionen					
	Funktionieren die Filter?		Ja	Ja	
	Funktioniert das Blättern?		Ja	Ja	
	Funktioniert die Sortierung?		Ja	Ja	
	Funktioniert das Einfügen?		Ja	Ja	
	Funktioniert das Bearbeiten?		Ja	Ja	
	Funktioniert das Löschen?		Ja	Ja	
	Haben die Auswahllisten die korrekten Werte?		Ja	Ja	
Modulspezifische Eigenschaften					
	Anzeige aller Benutzer		Ja	Ja	
	Korrekte Rechte Drop-Down		Ja	Ja	
	Korrekte Pflichtfelder		Ja	Ja	
	Eingabevervalidierung: Required		Ja	Ja	
	Eingabevervalidierung: Maskierung der Telefonnummer		Ja	Ja	
	Eingabevervalidierung: Regex-Match für die E-Mail-Adresse		Ja	Ja	
Unterschiede durch Rechte					
	Hat der Administrator Rechte entsprechend der Rechtematrix?		Ja	Ja	
	Zugriff nur für Admins				
	Hat der Techniker Rechte entsprechend der Rechtematrix?		Ja	Ja	
	Hat der Anwender Rechte entsprechend der Rechtematrix?		Ja	Ja	

Testplan – Lösungen		Geprüft von: Martin Krockert	Geprüft am: 02.03.2014	
#	Testfall	Implementierung	Test bestanden?	Bemerkungen
Anzeige				
	Einheitliches Design?	Ja	Ja	
	Stimmen die Ausgabeformate?	Ja	Ja	
	Rechtschreibung in Ordnung?	Ja	Ja	
Funktionen				
	Funktionieren die Filter?	Nein		
	Funktioniert das Blättern?	Ja	Ja	
	Funktioniert die Sortierung?	Nein		
	Funktioniert das Einfügen?	Ja	Ja	
	Funktioniert das Bearbeiten?	Ja	Ja	
	Funktioniert das Löschen?	Ja	Ja	
	Haben die Auswahllisten die korrekten Werte?	Ja	Ja	
Modulspezifische Eigenschaften				
	Eingabeformular auf Pflichtfelder prüfen	Ja	Ja	
	Korrekte Eingaben/Eingabeverifikation	Ja	Ja	
	Funktioniert der Dateiupload?	Ja	Ja	
	Funktioniert der Dateidownload?	Ja	Ja	
	Funktioniert das Anzeigen von freigegebenen Lösungen für Anwender?	Ja	Ja	
Unterschiede durch Rechte				
	Hat der Administrator Rechte entsprechend der Rechtematrix? Kann lesen/schreiben/löschen	Ja	Ja	
	Hat der Techniker Rechte entsprechend der Rechtematrix? Kann lesen	Ja	Ja	
	Hat der Anwender Rechte entsprechend der Rechtematrix?	Ja	Ja	

Testplan – Suche		Geprüft von: Johannes Schaffrath	Geprüft am: 02.03.2014		
#	Testfall	Anzeige	Implementierung	Test bestanden?	Bemerkungen
	Einheitliches Design?	Ja	Ja	Ja	
	Stimmen die Ausgabeformate?	Ja	Ja	Ja	
	Rechtschreibung in Ordnung?	Ja	Ja	Ja	
	Funktionen				
	Funktionieren die Filter?	Nein			
	Funktioniert das Blättern?	Ja	Ja	Ja	
	Funktioniert die Sortierung?	Nein			
	Funktioniert das Einfügen?	Nein			
	Funktioniert das Bearbeiten?	Nein			
	Funktioniert das Löschen?	Nein			
	Haben die Auswahllisten die korrekten Werte?	Ja	Ja	Ja	
	Modulspezifische Eigenschaften				
	Stichprobenkontrolle der angezeigten Suchergebnisse.	Ja	Ja	Ja	
	Kontrolle der Suchergebnisse für Administratoren.	Ja	Ja	Ja	
	Kontrolle der Suchergebnisse für Techniker.	Ja	Ja	Ja	
	Kontrolle der Suchergebnisse für Anwender.	Ja	Ja	Ja	
	Unterschiede durch Rechte				
	Hat der Administrator Rechte entsprechend der Rechtematrix?	Ja	Ja	Ja	
	Hat der Techniker Rechte entsprechend der Rechtematrix?	Ja	Ja	Ja	
	Hat der Anwender Rechte entsprechend der Rechtematrix?	Ja	Ja	Ja	

Testplan – Tickets		Geprüft von: Jens Hetze 02.03.2014	Geprüft am:	
# Testfall	Anzeige	Implementierung	Test bestanden?	Bemerkungen
Anzeige	Einheitliches Design?	Ja	Ja	
	Stimmen die Ausgabeformate?	Ja	Ja	
	Rechtschreibung in Ordnung?	Ja	Ja	
Funktionen	Funktionieren die Filter?	Ja	Nein	Bei Seitenwechsel werden die Filter nicht zurückgesetzt.
	Funktioniert das Blättern?	Ja	Ja	
	Funktioniert die Sortierung?	Ja	Nein	Sortieren der Datumsfelder funktioniert nach dem String-Wert und nicht nach dem Datumswert.
	Funktioniert das Einfügen?	Ja	Ja	
	Funktioniert das Bearbeiten?	Ja	Ja	
	Funktioniert das Löschen?	Ja	Ja	
	Haben die Auswahllisten die korrekten Werte?	Ja	Ja	
Modulspezifische Eigenschaften				
	Eingabeformular auf Pflichtfelder prüfen	Ja	Ja	
	Korrekte Eingaben/Eingabevervalidierung	Ja	Ja	
	Funktioniert der Dateiupload?	Ja	Ja	
	Funktioniert der Dateidownload?	Ja	Ja	
	Funktioniert das Suchfeld über die gesamte Tickettabelle?	Ja	Ja	
	Funktioniert das Speichern und zur Lösung kopieren von offenen Tickets? - Formular Ticket bearbeiten	Ja	Ja	
	Funktioniert das Kopieren der Lösung von geschlossenen Tickets zur Lösungsdatenbank? – Ticketliste	Ja	Ja	
Unterschiede durch Rechte				
	Hat der Administrator Rechte entsprechend der Rechtematrix?	Ja	Ja	
	Kann lesen/schreiben/loschen			
	Hat der Techniker Rechte entsprechend der Rechtematrix? Kann lesen	Ja	Ja	
	Hat der Anwender Rechte entsprechend der Rechtematrix?	Ja	Ja	

Selbstständigkeitserklärung

Eidesstattliche Erklärung

Wir erklären an Eides statt, dass wir die vorliegende Arbeit (entsprechend der genannten Verantwortlichkeit) selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt haben.

Ort, Abgabetermin

Unterschriften der Verfasser