

Architekturdokumentation

<Miraculix>



erstellt von

Gruppe B

HydraSoft GmbH

Template Revision: 5.0 DE (Rev. 2)

April 2011

We acknowledge that this document uses material from the arc 42 architecture template, <http://www.arc42.de>. Created by Dr. Peter Hruschka & Dr. Gernot Starke. For additional contributors see arc42.de/about/contributors.html



Änderungsübersicht

Version	Datum	Bearbeiter	Beschreibung
1.B	24.05.2013	P. Hegnauer	Abschluss Beta Version
1.A	26.05.2013	A. Di Lascio	Abschluss Alpha Version

Tabelle 1, Änderungsübersicht

Basisdokumente

Dokument	Beschreibung

Tabelle 2, Basisdokumente



Inhaltsverzeichnis

1. EINFÜHRUNG UND ZIELE	5
AUFGABENSTELLUNG	6
1.1 QUALITÄTSZIELE	7
1.2 STAKEHOLDER	8
2. RANDBEDINGUNGEN	9
2.1 TECHNISCHE RANDBEDINGUNGEN	9
FUNKTIONALE ANFORDERUNGEN	9
2.2 ORGANISATORISCHE RANDBEDINGUNGEN	10
2.3 KONVENTIONEN	11
2.4 ENTWICKLUNGSWERKZEUGE	11
3. KONTEXTABGRENZUNG	12
3.1 FACHLICHER KONTEXT	12
4. LÖSUNGSSTRATEGIE	14
5. BAUSTEINSICHT	15
5.1 EBENE 0	16
5.2 EBENE 1	17
5.3 EBENE 2	20
6. LAUFZEITSICHT	30
6.1 LAUFZEITSZENARIO 1	30
6.2 LAUFZEITSZENARIO 2	31
7. VERTEILUNGSSICHT	32
8. TYPISCHE MUSTER, STRUKTUREN UND ABLÄUFE	33
8.1 TYPISCHE MUSTER UND STRUKTUREN	33
8.2 TYPISCHE ABLÄUFE	34
9. TECHNISCHE KONZEPTE	35
9.1 PERSISTENZ	35
9.2 BENUTZUNGSOBERFLÄCHE	35
9.3 ERGONOMIE	36
9.4 ABLAUFSTEUERUNG	36
9.5 TRANSAKTIONSBEHANDLUNG	36
9.6 SESSIONBEHANDLUNG	37
9.7 SICHERHEIT	37
9.8 KOMMUNIKATION UND INTEGRATION MIT ANDEREN IT-SYSTEMEN	37
9.9 VERTEILUNG	37
9.10 PLAUSIBILISIERUNG UND VALIDIERUNG	37
9.11 AUSNAHME-/FEHLERBEHANDLUNG	38
9.12 MANAGEMENT DES SYSTEMS & ADMINISTRIERBARKEIT	39
9.13 LOGGING, PROTOKOLLIERUNG, TRACING	39
9.14 GESCHÄFTSREGELN	39
9.15 PARALLELISIERUNG UND THREADING	39
INTERNATIONALISIERUNG	40
9.16 MIGRATION	40
9.17 TESTBARKEIT	40
9.18 SKALIERUNG, CLUSTERING	55
9.19 HOCHVERFÜGBARKEIT	55
10. ENTWURFSENTSCHEIDUNGEN	56
10.1 ENTWURFSENTSCHEIDUNG 1	56

10.2	ENTWURFSENTSCHEIDUNG 2	57
10.3	ENTWURFSENTSCHEIDUNG 3	58
11.	QUALITÄTSSZENARIEN.....	59
11.1	QUALITÄTSBAUM.....	59
11.2	BEWERTUNGSSZENARIEN	61
12.	RISIKEN.....	62
13.	GLOSSAR.....	63
14.	ANHANG.....	64
14.1	BEDIENUNGSANLEITUNG.....	64
14.2	INSTALLATIONS- UND DEPLOYMENTANWEISUNGEN	74
14.3	ZEITPLAN	75
14.4	ARBEITSZEITEN	75
14.5	SITZUNGSPROTOKOLLE.....	75
14.6	PFLICHTENHEFT	75
14.7	NAMENSKONVENTIONEN	75
14.8	KLASSENDIAGRAMM.....	75
14.9	ERD DIAGRAMM.....	75
14.10	SOURCE CODE	75



1. Einführung und Ziele

Die TotalService GmbH ist ein Unternehmen, welches sich auf Reparaturen von elektronischen Haushaltsgeräten spezialisiert hat. Die Reparaturen werden von einem Servicetechniker vor Ort ausgeführt. Der finanzielle sowie administrative Teil werden durch das BackOffice abgewickelt und geleitet.

Da das Unternehmen rapide gewachsen ist und mittlerweile über 30 Servicetechniker beschäftigt, benötigt die TotalService GmbH eine unterstützende Applikation, welche sich auf die eigenen Bedürfnisse einstellt.

Durch den Auftrag der TotalService GmbH sollen wir im Rahmen einer Projektarbeit eine Applikation erstellen, welche die Anforderungen der TotalService GmbH vollumfänglich erfüllt. Es soll ein strukturierter und klarer Prozess verfolgt werden, welcher durch die Applikation unterstützt wird.

Neu soll es möglich sein Kunden-, Auftrags- oder auch Mitarbeiterdaten auf einer Datenbank zu speichern. Die Daten sind redundant und jederzeit und überall abrufbar. So kann der Prozess für das BackOffice optimiert werden. Sie sind nun schneller und müssen nicht mühsam die Daten aus einer Excel-Tabelle auslesen. Da die Daten persistent auf der Datenbank liegen, können auch keine Kundenaufträge oder Kundendaten verloren gehen.

Nicht nur für das BackOffice soll die Applikation einen Vorteil bringen sondern auch der Servicetechniker profitiert davon. Er kann einfach von zu Hause aus am Vorabend seine Aufträge aufrufen und diese ausdrucken. So kann er am nächsten Tag direkt zum ersten Kunden gehen, und muss nicht vorher noch ins Geschäft. Vor Ort kann er direkt den Rapport ausfüllen und quittieren. Falls nun noch ein Problem vorhanden ist, kann er mit Hilfe unserer Applikation direkt mit dem Kunden einen neuen Termin vereinbaren. Die Kommunikation über das BackOffice entfällt somit.

Da die TotalService GmbH kundenorientiert arbeiten möchte, hat die neue Software auch für den Kunden Vorteile. Er kann sich von zu Hause aus einloggen und hat eine Übersicht über seine Tickets und Benutzerdaten. Ausserdem hat er die Möglichkeit, falls ein Gerät defekt ist, direkt ein Ticket aufzugeben und muss nicht zwingend Kontakt mit der Firma aufnehmen. Nachdem ein Ticket abgeschlossen wurde, erhält der Kunde automatisch eine E-Mail, welche eine Umfrage beinhaltet. In dieser Umfrage kann der Kunde unser System und unsere Arbeit bewerten. So generiert die TotalService GmbH einen Mehrwert. Sie kann direkt auf das Feedback der Kunden eingehen.

Die Applikation ist benutzerfreundlich und hat ein einheitliches Erscheinungsbild. Die tägliche Arbeit des BackOffice wird dadurch deutlich vereinfacht. Die TotalService GmbH gewinnt durch „Miraculix“ einen deutlichen Mehrwert durch steigende Produktivität und kundenorientierte Prozesse.



Aufgabenstellung

Seit drei Jahren ist TotalService GmbH für den Service von zahlreichen Haushaltsgeräten eines namhaften Haushaltsgeräteherstellers in der Zentralschweiz zuständig. Dabei werden Waschmaschinen, Wäschetrockner, Kühlschränke, Kochherds, Backöfen, Geschirrspüler, etc. gewartet. Die Zusammenarbeit wurde schrittweise ausgebaut und zurzeit beschäftigt die TotalService GmbH mehr als 30 Servicetechniker, die täglich bei ihren Kundinnen und Kunden unterwegs sind. Damit die Organisation einfacher bewältigt werden kann, soll eine verteilte Applikation (≥ 3 Tier) entwickelt werden, mit der das tägliche Geschäft einfacher bewältigt werden kann.

Geschäftsprozess:

Die Applikation beinhaltet folgende wesentliche Anforderungen:

- ✓ die Verwaltung aller notwendigen Servicetechniker-Daten, die für die Reparaturarbeiten nötig sind
- ✓ die Verwaltung aller notwendigen Kundendaten
- ✓ die Verwaltung von allen Servicevertragsdaten
- ✓ die Erfassung von telefonischen Meldungen (Fehlerbeschreibung, Serie- und Service-Nummer, Gerätetyp, Hersteller etc.) durch das BackOffice
- ✓ die Suche nach dem nächsten freien Termin pro Servicetechniker und die Erstellung des Reparaturauftrags für den Servicetechniker durch das BackOffice
- ✓ das elektronische Bereitstellen der Service-Aufträge für den Servicetechniker. Die elektronischen Service-Aufträge müssen vom Servicetechniker auch in einer gut lesbaren Form ausgedruckt werden können.
- ✓ die Erstellung des Reparaturreports durch den Servicetechniker in elektronischer Form

Da alle Servicetechniker mit Notebooks ausgerüstet und immer online sind, muss die Applikation den Zugriff auf relevanten Daten zu jedem beliebigen Zeitpunkt ermöglichen. Der Servicetechniker soll in der Lage sein, seine Aufgaben abzufragen, damit er selber auch einen Termin mit dem Kunden vor Ort vereinbaren kann. Des Weiteren muss die Applikation dem Servicetechniker die Möglichkeit bieten, den Reparatur-Rapport unmittelbar nach der ausgeführten Reparatur (vor Ort) zu erstellen und in der Anwendung zu speichern.

Der Zugriff auf die Applikation und Daten muss in allen Fällen durch ein entsprechendes Anmeldeverfahren geschützt werden. Die Applikation wird einerseits vom BackOffice und andererseits von Servicetechniker benutzt. Auch die Kunden haben über einen Webservice Zugriff auf die Applikation. Das BackOffice befindet sich im Intranet, während die Servicetechniker und die Kunden einen entfernten Zugriff vornehmen. Die Verwaltung der anfallenden Daten muss mit einem zuverlässigen DBMS realisiert werden.



1.1 Qualitätsziele

Nr./Priorität	Qualitätsziel	Kurzbeschreibung/Verweis
1.	Performanz	Die Datenbank sollte Daten enthalten, die möglichst aktuell sind. Neue Daten müssen korrekt und sofort abrufbar sein.
2.	Sicherheit	Die relationale Datenbank ist ein Schlüsselement für den Erfolg der Firma. Aus diesem Grund müssen die Daten vor fremden Zugriffen geschützt werden. Durch ein Anmeldeverfahren soll die Datensicherheit, der Datenschutz und die Daten-Integrität gewährleistet sein.
3.	Flexibilität	Unsere Applikation soll von überall her erreichbar sein. Der Servicetechniker und der Kunde erlangen somit eine grössere Flexibilität und sind nicht mehr stark an das BackOffice gebunden.
4	Erweiterbarkeit	Aufgrund der ständigen Modernisierung muss die Applikation leicht anpassbar sein. Die Erweiterbarkeit ist somit äußerst essentiell für das Unternehmen, um auch in Zukunft konkurrenzfähig zu bleiben. Die Veränderungen des täglichen Business sollen sich leicht in die Applikation implementieren lassen.
5	Benutzerfreundlichkeit	Ein einfaches und einheitliches Erscheinungsbild vereinfacht die Benutzung der Applikation und steigert die Produktivität. Das GUI muss für Personen bedienbar sein, die über kein allzu grosses Flair für IT verfügen.

Tabelle 3, Qualitätsziele



1.2 Stakeholder

Name/Rolle	Ziel/Berührungspunkt	Notwendige Beteiligung
Servicetechniker	Die Applikation ist für den Servicetechniker ein wichtiges Tool bei der Arbeit. Er muss Rapporte ausfüllen und ausdrucken sowie neue Termine vereinbaren können.	Schulung notwendig
BackOffice	Das BackOffice benötigt die Applikation am meisten. Mit der Applikation lässt es sich übersichtlich und strukturiert arbeiten. Aufträge können genau zugewiesen werden und die Verrechnungen verlaufen ohne Redundanz. Des Weiteren müssen die Informationen leicht abrufbar sein, um die Kunden zu bedienen.	Schulung notwendig
Konkurrenz	Die TotalService GmbH hebt sich durch die neue Applikation und Prozessen von der Konkurrenz ab.	keine
Lieferanten	Der namhafte Gerätehersteller stellt eine Liste ihrer Ersatzteile und –geräte zur Verfügung. Diese wird in der Datenbank eingelesen.	keine
Kunden	Der Kunde kann sich mit Hilfe unserer Anwendung bei uns registrieren und ein Ticket erstellen, welches dann vom Back-Office entgegengenommen wird.	Schulung durch Bedienungs- und Installationsanleitung

Tabelle 4, Stakeholder



2. Randbedingungen

2.1 Technische Randbedingungen

ID	Randbedingung	Beschreibung
TECH01	Systembetrieb	Das System ist auf Onlinebetrieb entwickelt.
TECH02	Software-Infrastruktur	Webserver & Datenbanksysteme
TECH03	Verfügbarkeit der Laufzeitumgebung	7x24-StundenBetrieb ► siehe 9.20 Hochverfügbarkeit
TECH04	Grafische Oberfläche	Diese soll durch den Servicetechniker, den Sachbearbeiter des BackOffice sowie den Kunden zugänglich sein.
TECH05	Bibliotheken, Frameworks und Komponenten	Einsatz von Frameworks und Open-Source-Bibliotheken
TECH06	Programmiersprachen	Java und PostgreSQL
TECH07	Analyse- und Entwurfsmethoden	MVC, Modell- View- Controller ► siehe 8.1 Typische Abläufe und Strukturen
TECH08	Datenstrukturen	3-Tier-Architektur mit Webservice, RMI und DB
TECH09	Programmierschnittstellen	Datenbankschnittstellen und RMI / Webservice
TECH10	Programmiervorgaben	JUnit & TDD
TECH11	Technische Kommunikation	Synchron oder asynchron

Tabelle 5, Technische Randbedingungen

Funktionale Anforderungen

► siehe Kapitel 1 „Aufgabenstellung“



2.2 Organisatorische Randbedingungen

ID	Randbedingung	Beschreibung
Organisation und Struktur		
ORG01	Organisationsstruktur beim Auftraggeber	Sind fix definiert von Hr. Jordan Suçur & Hr. Erwin Mathis. Allfällige Änderungen werden durch sie kommuniziert.
ORG02	Eigenentwicklung oder externe Vergabe	Hierbei handelt es sich um eine schulische Projektarbeit.
ORG03	Entwicklung als Produkt oder zur eigenen Nutzung?	Die Entwicklung dient als Produktlösung für die fiktive TotalService GmbH.
Ressourcen (Budget, Zeit, Personal)		
ORG04	Zeitplan	Der Zeitplan für das Projekt wurde von der Gruppe erstellt. Darauf wird geachtet, die gesetzte Deadline einzuhalten. ► siehe Anhang „Zeitplan“
ORG05	Zeitplan und Funktionsumfang	Da der Abgabetermin auf den 24.05.2013, 12:00 Uhr festgelegt und nicht änderbar ist, ist der Termin als hoch priorisiert zu betrachten.
ORG06	Team	Das Team besteht aus folgenden 7 Studenten der Hochschule Luzern – Wirtschaft, Studiengang Wirtschaftsinformatik Vollzeit, 2. Semester: <div style="display: flex; justify-content: space-between;"> <div> Silvana Maissen Patrick Hegnauer Alessio Di Lascio Cedric Amstalden Christoph Blättler Michael Ziehlmann Stefan Habbacher </div> <div> Projektleiterin Projektleiter Stv. </div> </div>
ORG07	Release-Plan	Der volle Funktionsumfang wird am 27.05.2013 vorgestellt. Die anderen Meilensteine sind dem Zeitplan zu entnehmen. ► siehe Anhang „Zeitplan“
Organisatorische Standards		
ORG08	Vorgehensmodell	Keine internen Standards vorhanden.
ORG09	Entwicklungswerkzeuge	► siehe Kapitel 2.4 „Entwicklungswerkzeuge“
ORG10	Konfigurations- und Versionsverwaltung	► siehe Kapitel 2.4 „Entwicklungswerkzeuge“
ORG11	Testwerkzeuge und Prozesse	Testing mit JUnit
ORG12	Abnahme- und Freigabeprozesse	Alle Entscheidungen werden in wöchentlichen Meetings gefällt. Diese wurden stets vom Projektleiter-Stellvertreter geführt.

Tabelle 6, Organisatorische Randbedingungen



2.3 Konventionen

ID	Randbedingung	Beschreibung
KONV01	Programmierkonventionen	Nicht vorhanden
KONV02	Dokumentationskonventionen	Nicht vorhanden
KONV03	Namenskonventionen	► siehe Anhang „Namenskonventionen“

Tabelle 7, Konventionen

2.4 Entwicklungswerkzeuge

Eclipse JUNO
4.4.2 X32



Java FX Scene Builder
Version 1.0



Apache Tomcat
Version 7.0



Jasper Reports
PlugIn Eclipse
Version 5.1.0



pgAdmin III
Version 1.16.1



ObjectAid
Version 1.1.4.2



Enterprise Architect
Version 9.3



SVN Subversion
Version 1.7.9



3. Kontextabgrenzung

Schnittstelle/ Nachbarsystem	Ausgetauschte Daten (Datenformate, Medien)	Technologie/Protokoll
JAVA Application	Quellcode	Via Webservice
PostgreSQL	Objekte mit Attribute	RMI

Tabelle 8, Kontextabgrenzung

3.1 Fachlicher Kontext

3.1.1 Komponentendiagramm

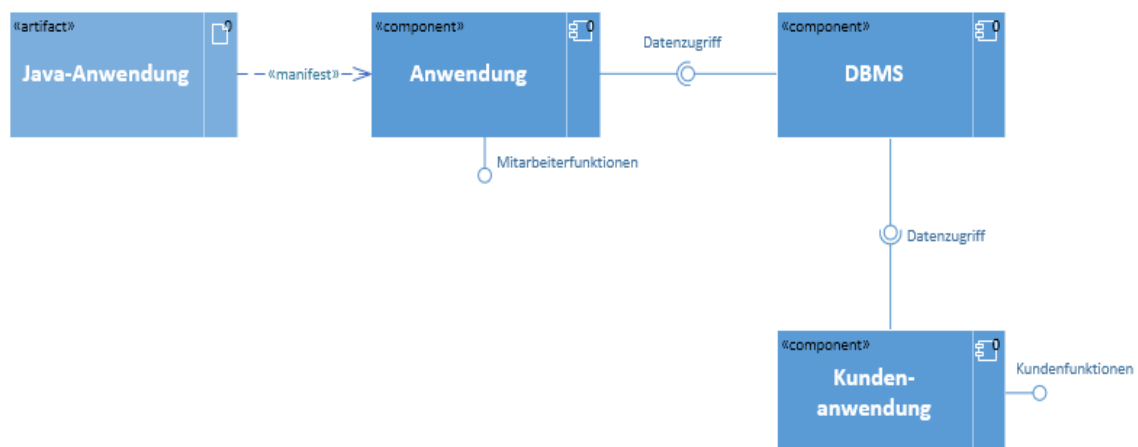


Abbildung 1, Komponentendiagramm

3.1.2 Paketdiagramm Miraculix Client Sachbearbeiter

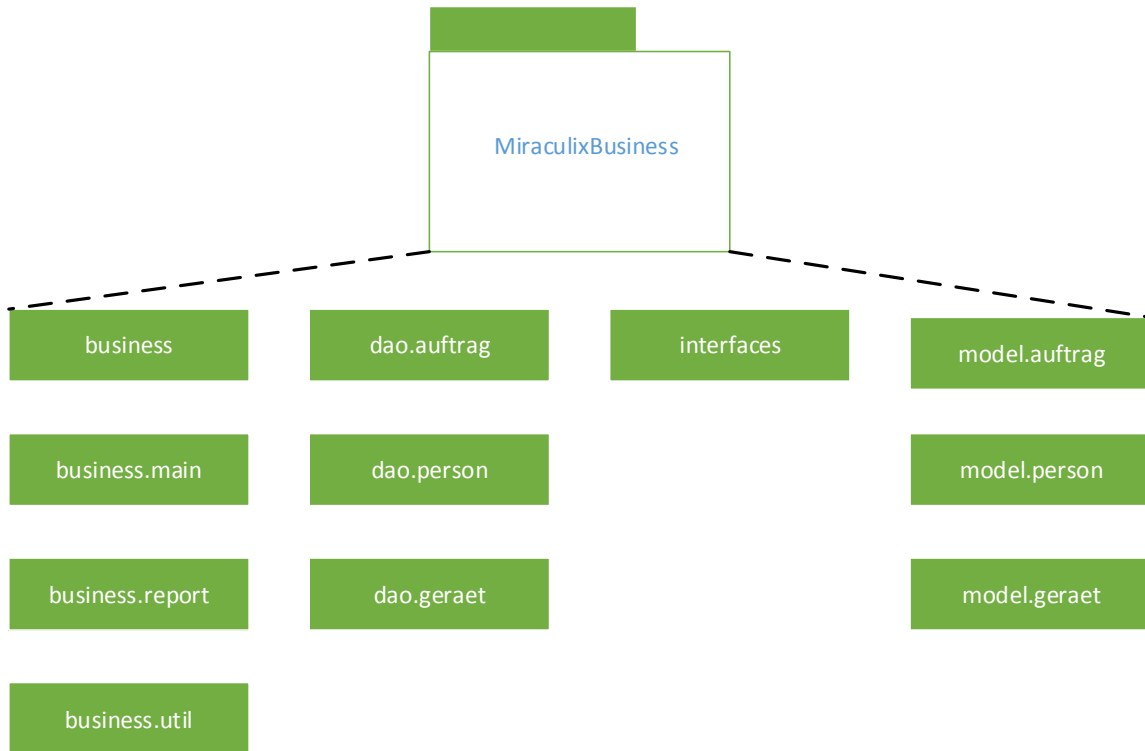


Abbildung 2, Paketdiagramm Sachbearbeiter

3.1.3 Paketdiagramm Miraculix Business

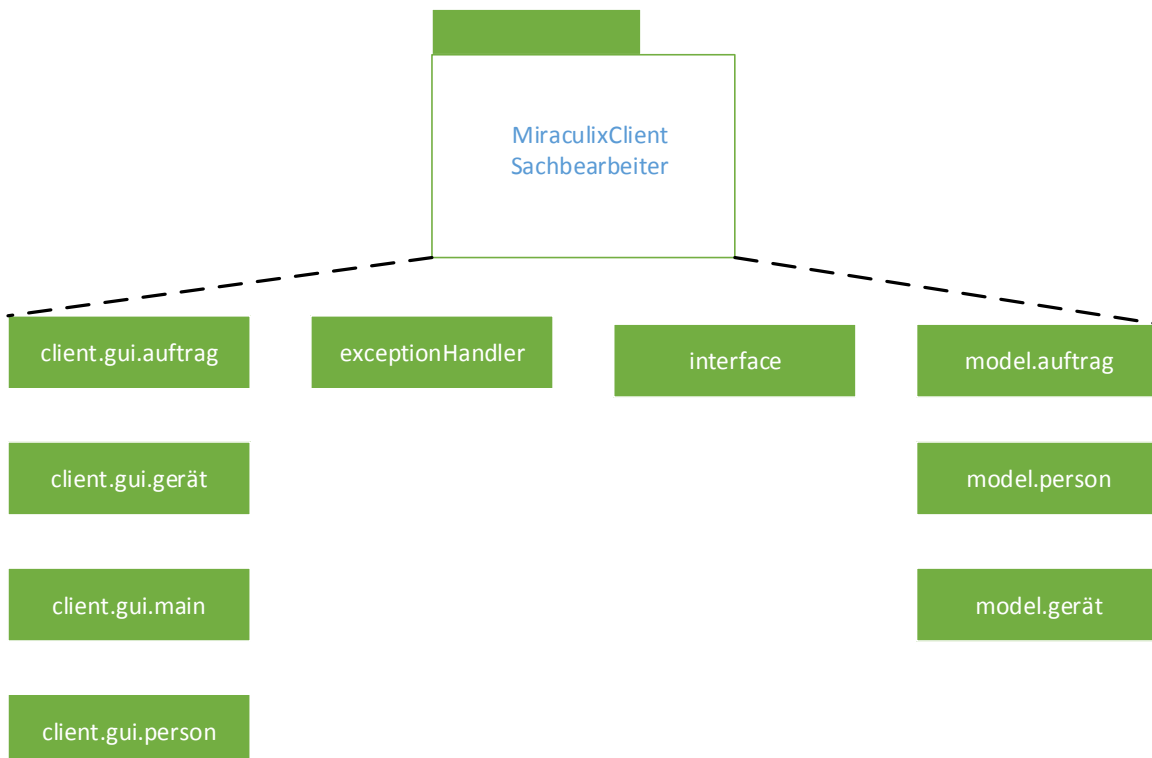


Abbildung 3, Paketdiagramm Business



4. Lösungsstrategie

Das Projektteam hat entschieden, die Applikation in der objektorientierten Programmiersprache Java zu schreiben. Gründe wie Performance, Sicherheit, Image und Know-How haben für diesen Entscheid massgebend gesprochen.

Da Swing immer mehr aus der Mode kommt und veraltet aussieht, haben wir uns für das mächtige Tool JavaFX entschieden, um unsere GUI zu realisieren. Die neumodische Oberfläche von JavaFX macht die Arbeit sehr angenehm. Zudem hat JavaFX viel mehr Möglichkeiten als Swing: Bilder können einfach eingebunden, die Oberfläche samt Buttons können mittels CSS gestaltet werden und vieles mehr. Für die Gestaltung der GUI wurde die OpenSource-Software SceneBuilder benutzt. Dies ist im Moment die einzige Software welche mit JavaFX kompatibel ist.

Die Datensätze werden auf der relationalen Datenbank PostgreSQL hinterlegt, die uns zur Verfügung gestellt wurde.

Es werden drei verschiedene Ansichten programmiert:

- Die BackOffice-Ansicht wird mittels RMI implementiert
- Die Servicetechniker greift mittels Webservice auf unsere Datenbank und Business Logik
- Der Kunde wird ebenfalls mit Webservice realisiert

Wir verzichten auf das MockTesting, da die Datenbank schon sehr früh zur Verfügung stand und wir direkt auf der Datenbank die Tests durchführen konnten.

Um die Rapporte als pdf-Dateien herzustellen stützen wir uns auf die OpenSource-Software Jaspersoft Report.



5. Bausteinsicht

Die folgende Abbildung zeigt alle Bausteine unserer Applikation:

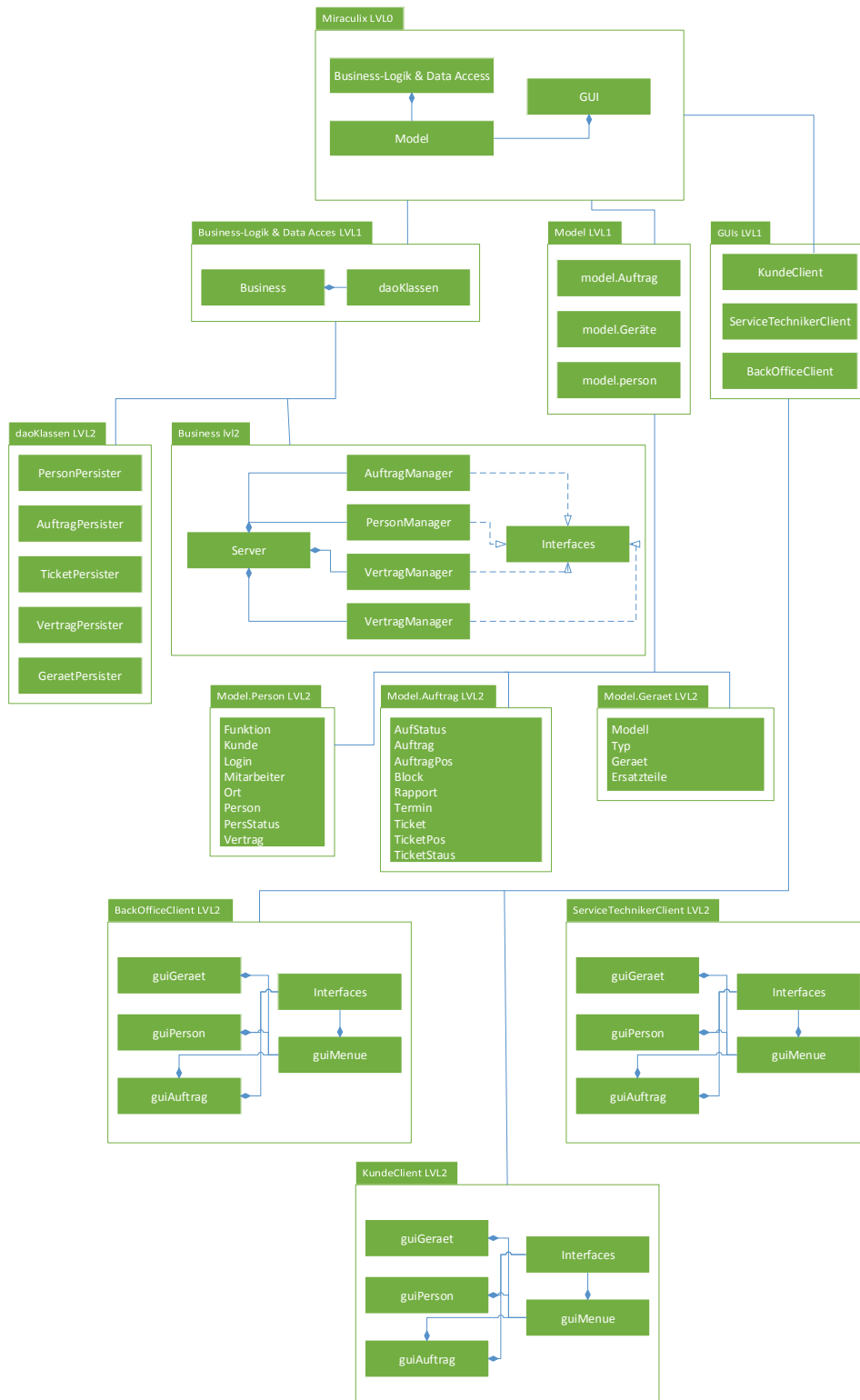


Abbildung 4, Bausteinschicht

5.1 Ebene 0

Lokale Bausteine von WHITEBOX << Miraculix LVL0 >>	
<<BLACKBOXNAME>>	Kurzer Beschrieb
Business-Logik & Data Access	Beinhaltet Logik & Datenbankmanagement
GUI	Beinhaltet grafische Oberflächen und Kontrollermethoden
Model	Beinhaltet Modelklassen

Blackbox-Layout für <<Business-Logik & Data Access LVL0>	
Zweck / Verantwortlichkeit	Beinhaltet die ganze Logik des Programmes und regelt Zugriff zur Datenbank
Schnittstellen	Folgende Business-Interfaces werden benötigt, um auf Businesslogik zugreifen zu können: IVertrag, IPerson, IAuftrag, IGeraet
Variabilität	Da der Baustein vollständig abgekapselt ist, können sich alle Algorithmen ändern, ohne dass andere Komponenten etwas davon mitkriegen.
Leistungsmerkmale	Keine Einschränkungen
Ablageort / Datei	Business-Komponente (business-package und dao-packages)

Blackbox-Layout für <<Model LVL0>	
Zweck / Verantwortlichkeit	Beinhaltet Alle Modeklassen
Schnittstellen	
Variabilität	
Leistungsmerkmale	Keine Einschränkungen
Ablageort / Datei	Business-Komponente (business-package und dao-packages)



Blackbox-Layout für <<GUIs LVL0>	
Zweck / Verantwortlichkeit	Beinhaltet alle GUI und Benutzerschnittstellen.
Schnittstellen	
Variabilität	
Leistungsmerkmale	Keine Einschränkungen
Ablageort / Datei	Business-Komponente (business-package und dao-packages)

5.2 Ebene 1

Lokale Bausteine von WHITEBOX << Model LVL1 >>	
<<BLACKBOXNAME>>	Kurzer Beschrieb
Model.Auftrag	Modelklasse für Auftrag
Model.Person	Modelklasse für Person
Model.Geraete	Modelklasse für Geraete
Lokale Bausteine von WHITEBOX << Business-Logik & Data Access LVL1 >>	
<<BLACKBOXNAME>>	Kurzer Beschrieb
Business	Enthält Methoden für logische Operationen
daoKlassen	Enthält Methoden zur Datenpersistierung
Lokale Bausteine von WHITEBOX << GUIs LVL1 >>	
<<BLACKBOXNAME>>	Kurzer Beschrieb
KundeClient	Stellt die Applikation für den Kunden bereit.
ServiceTechnikerClien	Stellt die Applikation für den Servicetechniker bereit.
BackOfficeClien	Stellt die Applikation für den Backoffice Mitarbeiter bereit.



Blackbox-Layout für <<daoKlassen>> LVL1	
Zweck / Verantwortlichkeit	Die daoKlassen sind für jegliche Kommunikation mit der Datenbank verantwortlich. Über sie werden Objekte gespeichert, zusammengeführt und abgefragt.
Schnittstellen	Kommuniziert mit dem Business Baustein und stellt ihm datenbankorientierte Methoden zur Verfügung. Kommuniziert via EntityManager mit der pgDatenbank
Variabilität	Sollen zusätzliche Operationen zur Speicherung, Bearbeitung oder Löschung von Objekten benötigt werden, müssen die daoKlassen erweitert werden.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business dao.auftrag, dao.geraet, da.person

Blackbox-Layout für <<Business LVL1>>	
Zweck / Verantwortlichkeit	Dieser Baustein übernimmt die Verteilung aller benötigten Methoden. Er ist dafür verantwortlich, dass sämtliche logischen Operationen in den Benutzerinterfaces durchgeführt werden können und zur Verfügung stehen.
Schnittstellen	Kommuniziert direkt mit den dao Klassen und ruft Methoden zur Bearbeitung von Datenbankobjekten auf.
Variabilität	Soll die Applikation an Funktionalität erweitert werden, kommen alle logischen Methoden in eine der Business Klassen.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.



Blackbox-Layout für <<model.Person>> LVL1	
Zweck / Verantwortlichkeit	Model.Person stellt alle Attribute und deren Setter- und Gettermethoden zur Verfügung. Ebenfalls sind alle OR-Mapping relevanten Felder und Beziehungen definiert.
Variabilität	Zusätzliche Attribute und OR-Mapping Erweiterungen.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.model.person

Blackbox-Layout für <<model.Geraete>> LVL1	
Zweck / Verantwortlichkeit	Model.Geraete stellt alle Attribute und deren Setter- und Gettermethoden zur Verfügung. Ebenfalls sind alle OR-Mapping relevanten Felder und Beziehungen definiert.
Variabilität	Zusätzliche Attribute und OR-Mapping Erweiterungen.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.model.geraete

Blackbox-Layout für <<model.Auftrag>> LVL1	
Zweck / Verantwortlichkeit	Model.Auftrag stellt alle Attribute und deren Setter- und Gettermethoden zur Verfügung. Ebenfalls sind alle OR-Mapping relevanten Felder und Beziehungen definiert.
Schnittstellen	
Variabilität	Zusätzliche Attribute und OR-Mapping Erweiterungen.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.model.auftrag



Blackbox-Layout für <<KundenClient >> LVL1	
Zweck / Verantwortlichkeit	Stellt dem Kunden sämtliche grafischen Benutzeroberflächen zur Verfügung.
Schnittstellen	WS-Kommunikation mit Businesslogik
Variabilität	Anpassungen der Benutzeroberfläche.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSKundenClient Client.gui.*

Blackbox-Layout für <<ServiceTechnikerClient >> LVL1	
Zweck / Verantwortlichkeit	Stellt dem Servicetechniker sämtliche grafischen Benutzeroberflächen zur Verfügung.
Schnittstellen	WS-Kommunikation mit Businesslogik
Variabilität	Anpassungen der Benutzeroberfläche.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.*

Blackbox-Layout für <<BackofficeClient >> LVL1	
Zweck / Verantwortlichkeit	Stellt dem Sachbearbeiter sämtliche grafischen Benutzeroberflächen zur Verfügung.
Schnittstellen	RMI-Kommunikation mit Businesslogik
Variabilität	Anpassungen der Benutzeroberfläche.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.*
Sonstige Verwaltungsinformationen	



5.3 Ebene 2

Lokale Bausteine von WHITEBOX << daoKlassen LVL2 >>	
<<BLACKBOXNAME>>	Kurzer Beschrieb
PersonPersister	Stellt Methoden zur Kommunikation mit der Datenbank bereit.
AuftragPersister	Stellt Methoden zur Kommunikation mit der Datenbank bereit.
VertragPersister	Stellt Methoden zur Kommunikation mit der Datenbank bereit.
TicketPersister	Stellt Methoden zur Kommunikation mit der Datenbank bereit.
GeraetPersister	Stellt Methoden zur Kommunikation mit der Datenbank bereit.

Lokale Bausteine von WHITEBOX << BackOfficeClient, ServiceTechnikerCliet, KundenClient >>	
<<BLACKBOXNAME>>	Kurzer Beschrieb
guiGeraet	Liefert grafische Benutzeroberflächen
guiPerson	Liefert grafische Benutzeroberflächen
guiAuftrag	Liefert grafische Benutzeroberflächen
Interfaces	Stellt Methoden zur Verfügung
guiMenue	Kommuniziert mit Business-Logik



Blackbox-Layout für <<PersonPersister>> LVL2	
Zweck / Verantwortlichkeit	Der PersonPersister enthält alle nötigen Methoden um IPerson Objekte (Kunden oder Mitarbeiter) zu speichern oder zusammenzuführen.
Schnittstellen	Stellt dem PersonManager seine Methoden zur Verfügung. Schreibt direkt in die Datenbank.
Variabilität	Erweiterungen im Bereich Speichern und Editieren von Datensätzen müssen hier implementiert werden.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business dao.person.PersonPersister

Blackbox-Layout für <<VertragPersister>> LVL2	
Zweck / Verantwortlichkeit	Der VertragPersister enthält alle nötigen Methoden um IVertrag Objekte zu speichern oder zusammenzuführen.
Schnittstellen	Stellt dem VertragManager seine Methoden zur Verfügung. Schreibt via EntityManager in die Datenbank.
Variabilität	Erweiterungen im Bereich Speichern und Editieren von Datensätzen müssen hier implementiert werden.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business dao.person.VertragPersister



Blackbox-Layout für <<GeraetPersister>> LVL2	
Zweck / Verantwortlichkeit	Der GeraetPersister enthält alle nötigen Methoden um IGeraet Objekte zu speichern oder zusammenzuführen.
Schnittstellen	Stellt dem GeraetManager seine Methoden zur Verfügung. Schreibt via EntityManager in die Datenbank.
Variabilität	Erweiterungen im Bereich Speichern und Editieren von Datensätzen müssen hier implementiert werden.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business dao.geraet.GeraetPersister

Blackbox-Layout für <<AuftragPersister>> LVL2	
Zweck / Verantwortlichkeit	Der AuftragPersister enthält alle nötigen Methoden um IAuftrag Objekte zu speichern oder zusammenzuführen.
Schnittstellen	Stellt dem Manager seine Methoden zur Verfügung. Schreibt via EntityManager in die Datenbank.
Variabilität	Erweiterungen im Bereich Speichern und Editieren von Datensätzen müssen hier implementiert werden.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business dao.auftrag.AuftragPersister
Offene Punkte (nach Bedarf)	



Blackbox-Layout für <<TicketPersister>> LVL2	
Zweck / Verantwortlichkeit	Der TicketPersister enthält alle nötigen Methoden um Ticket Objekte zu speichern oder zusammenzuführen.
Schnittstellen	Stellt dem AuftragManager seine Methoden zur Verfügung. Schreibt via EntityManager in die Datenbank.
Variabilität	Erweiterungen im Bereich Speichern und Editieren von Datensätzen müssen hier implementiert werden.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business dao.auftrag.TicketPersister

Blackbox-Layout für <<AuftragManager>> LVL2	
Zweck / Verantwortlichkeit	Der AuftragManager enthält alle Methoden um Auftragsobjekte zu erstellen und zu bearbeiten. Die Methodennamen- und Signaturen werden vom Interface IAuftrag zur Verfügung gestellt.
Schnittstellen	Implementiert das Interface IAuftrag. Importiert AuftragPersister. Importiert Methoden TicketPersister.
Variabilität	Soll die Funktionalität von Miraculix erweitert werden, wird das Interface IAuftrag mit einer Methode erweitert und diese dann mit Codebody im AuftragManager realisiert.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.AuftragManager



Blackbox-Layout für <<VertragManager>> LVL2	
Zweck / Verantwortlichkeit	Der VertragManager enthält alle Methoden um Vertragsobjekte zu erstellen und zu bearbeiten. Die Methodennamen- und Signaturen werden vom Interface IVertrag zur Verfügung gestellt.
Schnittstellen	Implementiert das Interface IVertrag. Importiert Methoden VertragPersister.
Erfüllt Anforderungen (nach Bedarf)	
Variabilität	Soll die Funktionalität von Miraculix erweitert werden, wird das Interface IVertrag mit einer Methode erweitert und diese dann mit Codebody im AuftragManager realisiert.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.VertragManager
Sonstige Verwaltungsinformationen	

Blackbox-Layout für <<PersonManager>> LVL2	
Zweck / Verantwortlichkeit	Der PersonManager enthält alle Methoden um Vertragsobjekte zu erstellen und zu bearbeiten. Die Methodennamen- und Signaturen werden vom Interface IPerson zur Verfügung gestellt.
Schnittstellen	Implementiert das Interface IPerson. Importiert PersonPersister.
Variabilität	Soll die Funktionalität von Miraculix erweitert werden, wird das Interface IPerson mit einer Methode erweitert und diese dann mit Codebody im AuftragManager realisiert.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.PersonManager



Blackbox-Layout für <<GeraeteManager>> LVL2	
Zweck / Verantwortlichkeit	Der GeraeteManager enthält alle Methoden um Geraeteobjekte zu erstellen und zu bearbeiten. Die Methodennamen- und Signaturen werden vom Interface IVertrag zur Verfügung gestellt.
Schnittstellen	Implementiert das Interface IGeraete. Importiert GeraetePersister.
Variabilität	Soll die Funktionalität von Miraculix erweitert werden, wird das Interface IGeraete mit einer Methode erweitert und diese dann mit Codebody im AuftragManager realisiert.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business. GeraeteManager

Blackbox-Layout für <<Interfaces>> LVL2	
Zweck / Verantwortlichkeit	Stellt den ManagerKlassen ihre Methoden zur Verfügung. Werden im Interface mit Name- und Methodensignatur definiert.
Schnittstellen	Interface IVertrag spezifiziert VertragManager Interface IPerson spezifiziert PersonManager Interface IVertrag spezifiziert AuftragManager Interface IGeraet spezifiziert GeraeteManager
Variabilität	Methoden zur Funktionalitätserweiterung werden hier spezifiziert. Der konkrete Codebody in der Managerklasse implementiert.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business Interfaces.IVertrag Interfaces.IPerson Interfaces.Geraet Interfaces.Auftrag

Blackbox-Layout für <<Server>> LVL2	
Zweck / Verantwortlichkeit	Stellt die Registry und die entfernten Objekte zur Verfügung.
Schnittstellen	Importiert PersonManager Importiert VertragManager Importiert GeraetManager Importiert AuftragManager
Variabilität	Methoden zur Funktionalitätserweiterung werden hier spezifiziert. Der konkrete Cod-body in der Managerklasse implementiert.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	Miraculix Business business.main.Server

Blackbox-Layout für <<guiGeraet >> LVL2	
Zweck / Verantwortlichkeit	Stellt sämtliche Benutzeroberflächen im Zusammenhang mit Geräten zur Verfügung.
Schnittstellen	Importiert guiMenue. Kommuniziert via gui-Menue mit der Businesslogik. Importiert IGeraet.
Variabilität	Anpassungen der Benutzeroberfläche im Bereich Geraete.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.geraete MiraculixWSServiceKunde Client.gui.geraete MiraculixRMIBackOffice Client.gui.geraete



Blackbox-Layout für <<guiPerson >> LVL2	
Zweck / Verantwortlichkeit	Stellt sämtliche Benutzeroberflächen im Zusammenhang mit Kunde zur Verfügung.
Schnittstellen	Importiert guiMenue. Kommuniziert via guiMenue mit der Businesslogik. Importiert IPerson.
Variabilität	Anpassungen der Benutzeroberfläche im Bereich Kunde.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.kunde MiraculixWSServiceKunde Client.gui.kunde MiraculixRMIBackOffice Client.gui.kunde

Blackbox-Layout für <<Interfaces >> LVL2	
Zweck / Verantwortlichkeit	Stellt den Benutzeroberflächen alle nötigen Methoden zur Verfügung.
Schnittstellen	Importiert guiMenue. Kommuniziert via guiMenue mit der Businesslogik. Kommuniziert via Webservice oder RMI mit der Businesslogik.
Variabilität	
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.main MiraculixWSServiceKunde Client.gui.main MiraculixRMIBackOffice Client.gui.main



Blackbox-Layout für <<guiAuftrag >> LVL2	
Zweck / Verantwortlichkeit	Stellt sämtliche Benutzeroberflächen im Zusammenhang mit Auftrag zur Verfügung.
Schnittstellen	Importiert guiMenue. Kommuniziert via guiMenue mit der Businesslogik. Importiert IAuftrag
Variabilität	Anpassungen der Benutzeroberfläche im Bereich Auftrag.
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.auftrag MiraculixWSServiceKunde Client.gui.auftrag MiraculixRMIBackOffice Client.gui.auftrag

Blackbox-Layout für <<guiMenue >> LVL2	
Zweck / Verantwortlichkeit	Stellt alle nötigen Methoden zur Verfügung um mit der Businesslogik zu kommunizieren.
Schnittstellen	Importiert guiMenue. Kommuniziert via guiMenue mit der Businesslogik. Kommuniziert via Webservice oder RMI mit der Businesslogik.
Variabilität	
Leistungsmerkmale	Sollte bei Betrieb immer zur Verfügung stehen. Ansonsten ist die Software nicht lauffähig.
Ablageort / Datei	MiraculixWSServiceTechniker Client.gui.main MiraculixWSServiceKunde Client.gui.main MiraculixRMIBackOffice Client.gui.main
Sonstige Verwaltungsinformationen	



6. Laufzeitsicht

6.1 Laufzeitszenario 1

Das folgende Szenario zeigt an, wie ein Ticket erstellt wird:

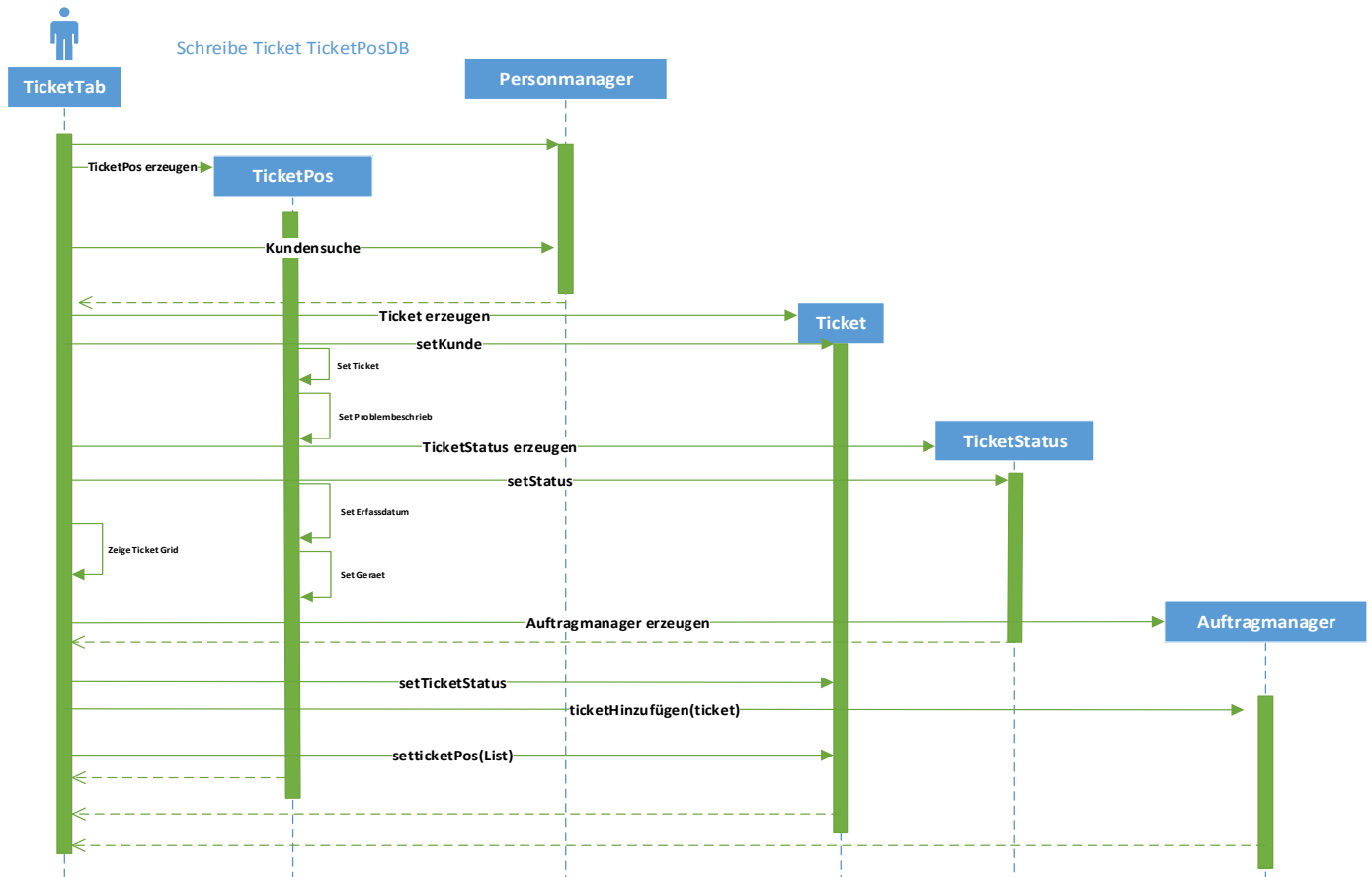


Abbildung 5: Sequenzdiagramm Schreibe TicketPosDB

6.2 Laufzeitzszenario 2

In diesem Laufzeitzszenario wird dargestellt, was passiert wenn ein Kunde uns telefonisch kontaktiert, oder wenn er sich über seine Applikation einloggt und ein Ticket erstellt.

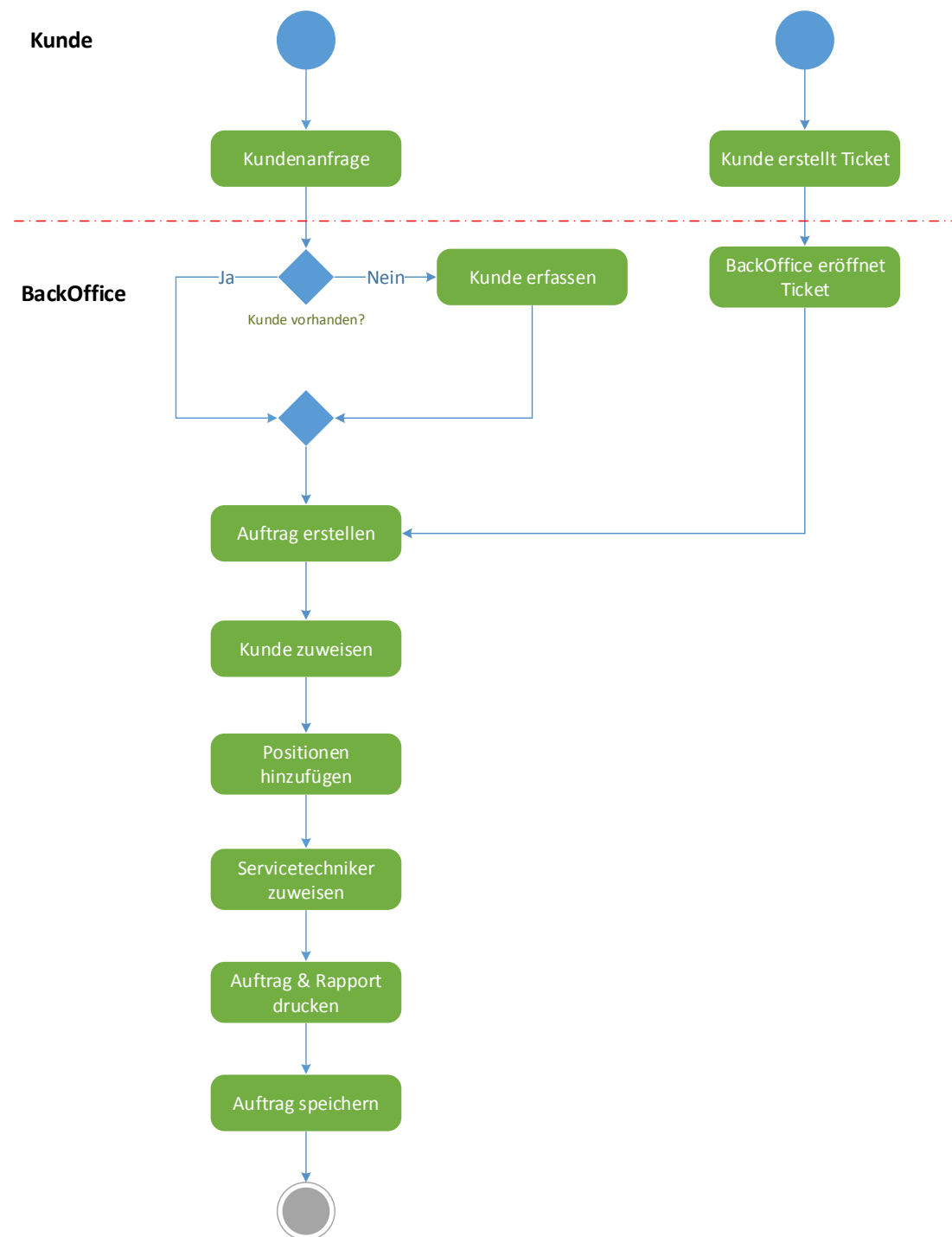


Abbildung 6, Aktivitätsdiagramm

7. Verteilungssicht

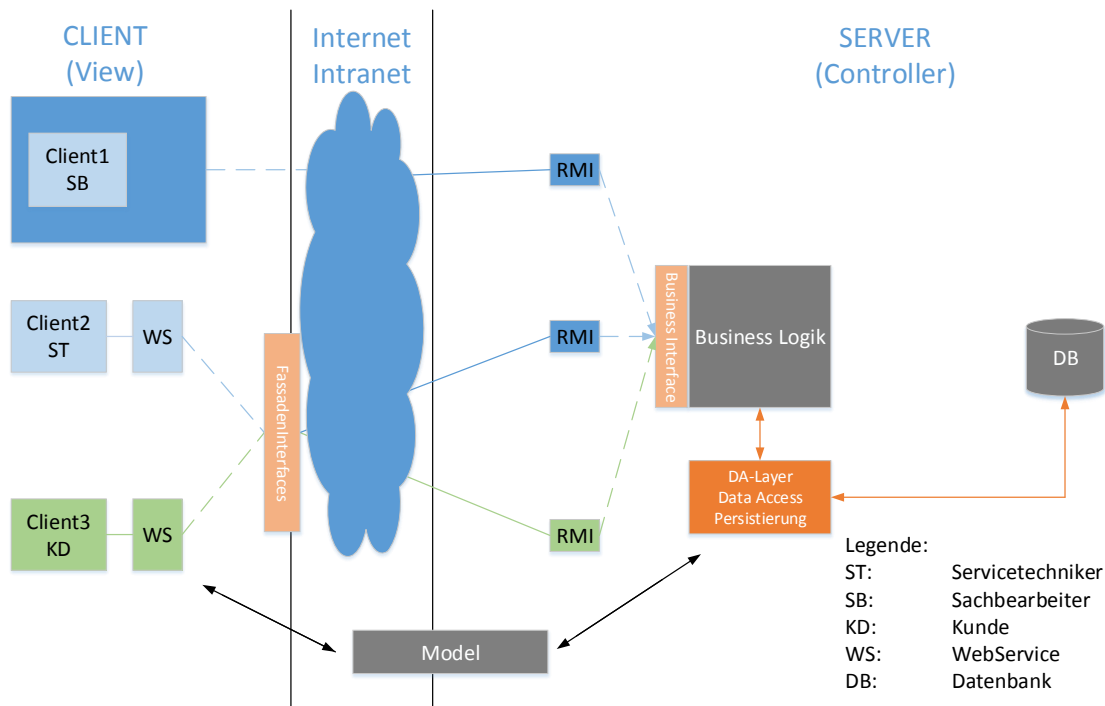


Abbildung 7, Verteilungskontext

Miraculix ist eine verteilte 3-Tier-Softwarearchitektur. Der Zugriff geschieht über drei mögliche Clients. Im Unternehmen geschieht dies über den Sachbearbeiter-Client, welcher sich im Intranet befindet und so über RMI und die Business-Interfaces mit der Businesslogik kommuniziert. Das heisst, die SachbearbeiterClient.jar wird auf einer Windowsmaschine im Unternehmen ausgeführt.

Die beiden anderen Clients (Kunde und Servicetechniker) befinden sich ausserhalb unseres Netzes. Auch diese Dateien (KundeClient.jar und ServicetechnikerClient.jar) werden jeweils auf einer Windowsmaschine im Unternehmen ausgeführt. Deshalb benutzen diese jeweils einen Webservice, um über Fassaden-Interfaces auf die Business-Logik zuzugreifen. Die Webservices laufen auf dem TomCat-Server Version 7.0. Der Datenbankserver kann nur von der Business-Logik und nur über die Persistierungsschicht (DAO-Klassen) angesprochen werden. Dies ermöglicht eine klare Trennung der Schichten.

8. Typische Muster, Strukturen und Abläufe

8.1 Typische Muster und Strukturen

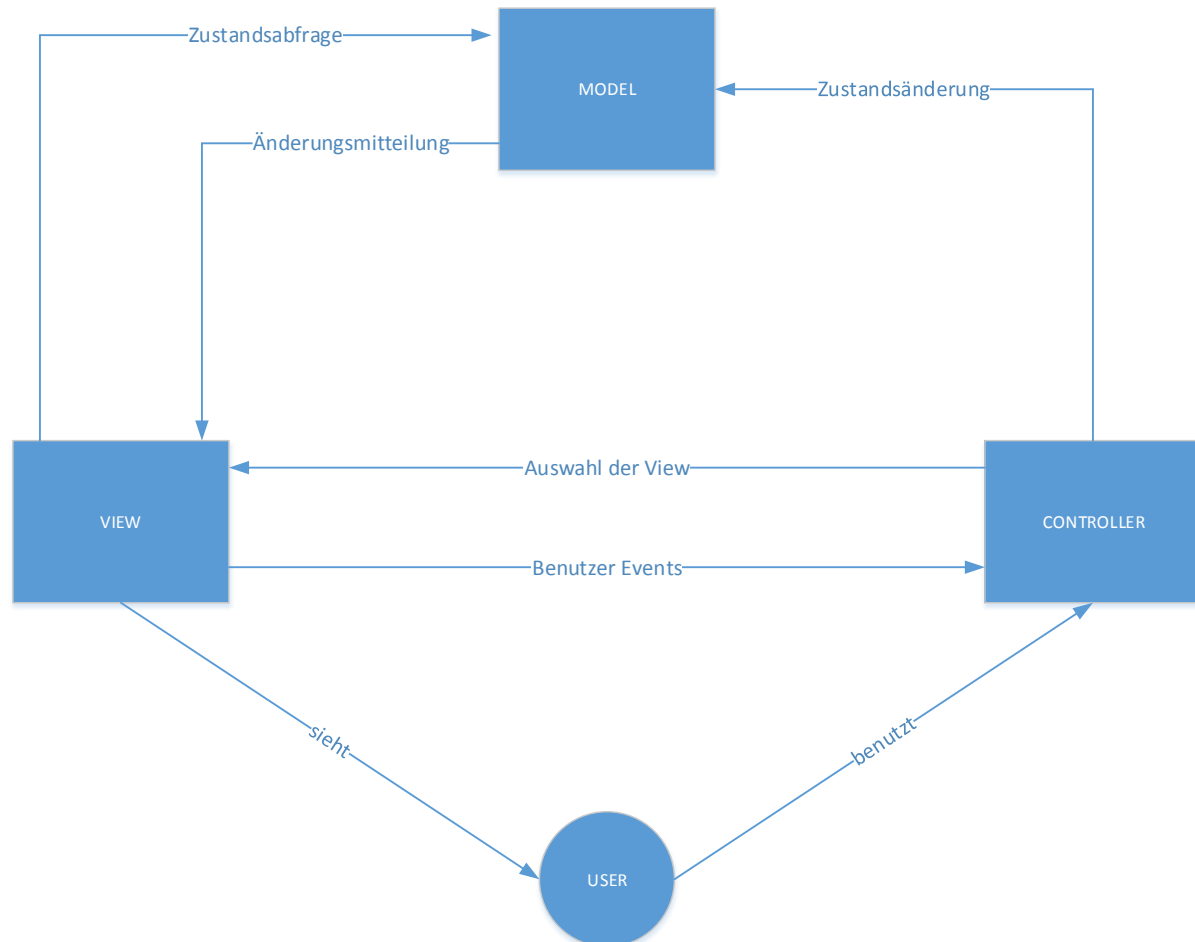


Abbildung 8, MVC Modell

Ablauf des MVC Modells

Der Ablauf beginnt beim User. Er benutzt den Controller (Steuerung). Gibt es irgendwelche Zustandsänderungen, so wird dies an das Model (Modell) weitergeleitet. Das Model teilt die Änderungen der View mit. Zudem macht die View stetig Zustandsabfragen. Der User kann auf dem Controller die View aussuchen. Die Präsentationsschicht ist zuständig für die Darstellung der Daten. Sie gibt aber auch die Benutzer-Events an den Controller weiter.

Praktisches Beispiel:

Der Sachbearbeiter will einen neuen Kunden erstellen. Er gibt die nötigen Daten in den entsprechenden Feldern ein (View). Der Controller holt sich die eingegebenen Daten von der View und erstellt daraus ein Objekt. Diese wird durch EclipseLink an das Model übergeben. Das Model schreibt daraufhin das umgewandelte Objekt in die Datenbank. Der Anwender erhält nun auf der View die Bestätigung, dass die Daten erfolgreich gespeichert wurden.

8.2 Typische Abläufe

Hier wird ein typischer Ablauf aufgezeigt: Erstellen eines neuen Auftrages:

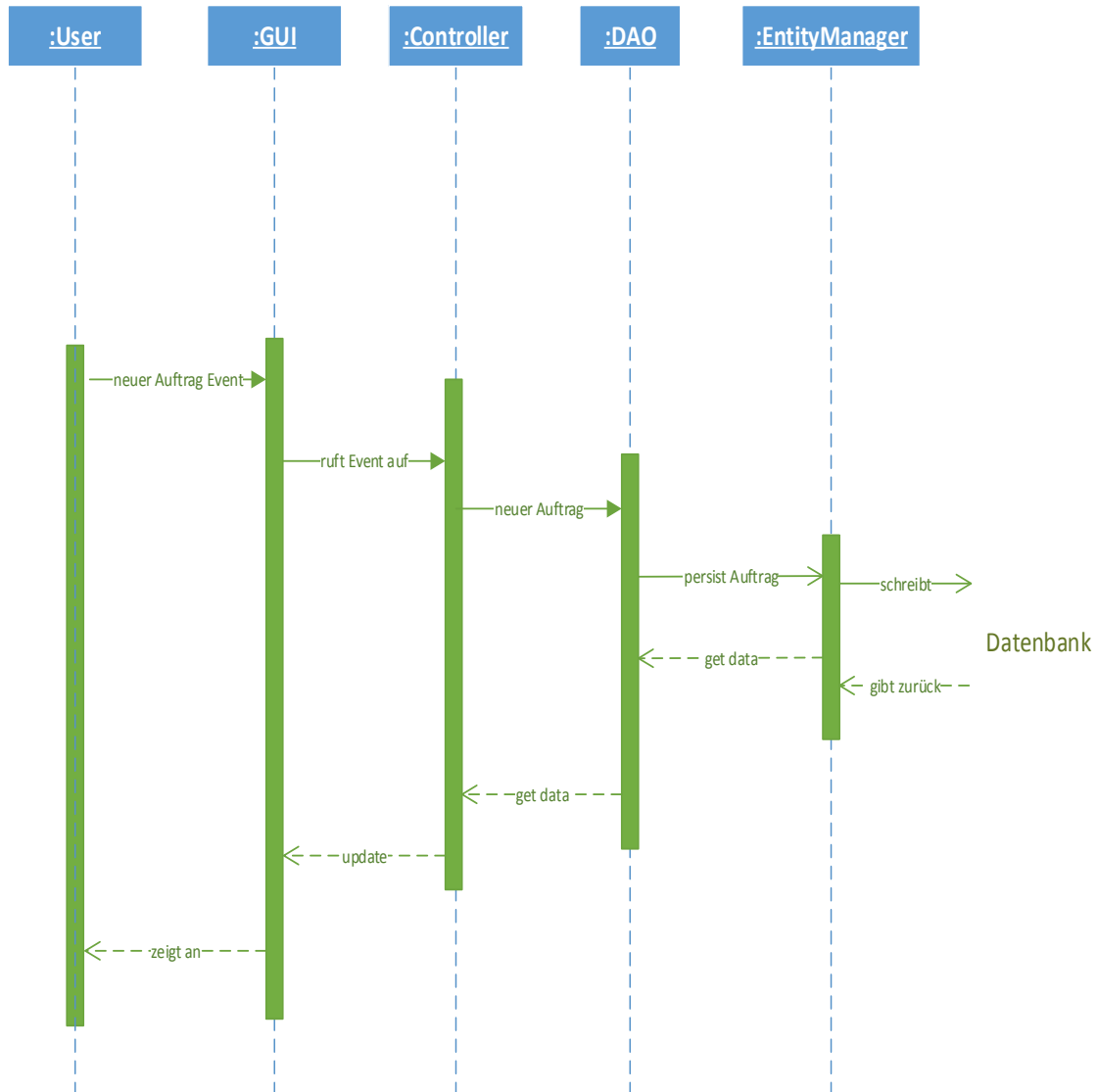


Abbildung 9, Sequenzdiagramm MVC

9. Technische Konzepte

9.1 Persistenz

Die Daten werden auf dem GUI eingegeben. Solange die Daten nicht gespeichert sind, sind sie flüchtig, d.h. bei einem Systemabsturz würden diese nicht mehr existieren. Sobald die entsprechenden Buttons gedrückt werden, sind die eingegebenen Daten auf unsere Datenbank vorhanden.

Diese Datenbank ist von überall zugänglich. So ist es möglich von überall auf die Daten zuzugreifen. Die Flexibilität ist so sehr gross.

9.2 Benutzungsoberfläche

Unsere Benutzeroberfläche wurde mit JavaFX realisiert. Das Graphical User Interface (GUI) stellt dem Benutzer eine freundliche Oberfläche dar.

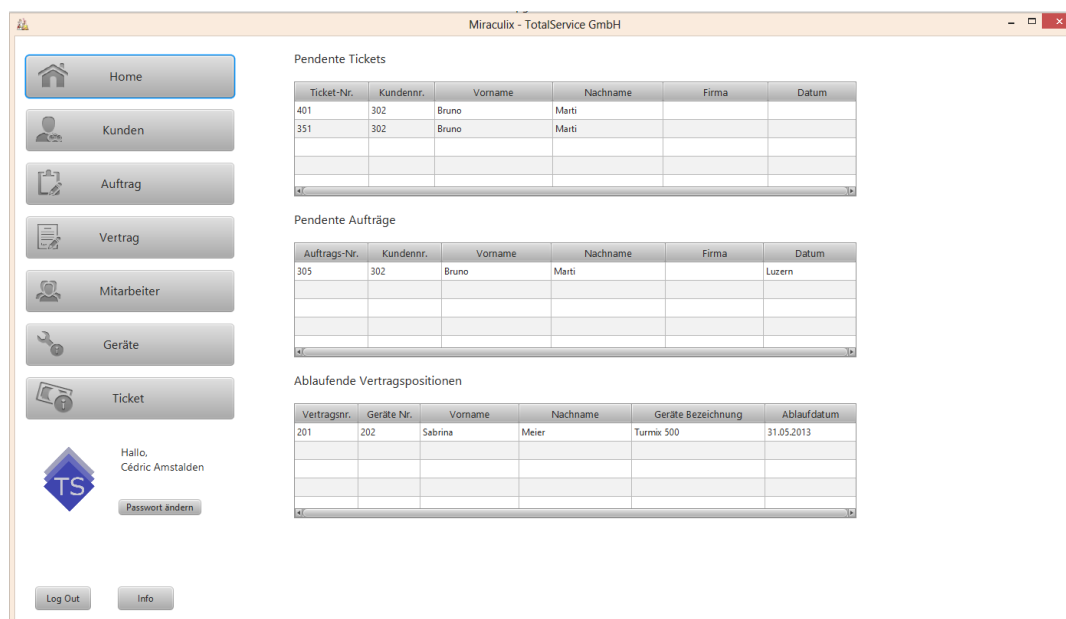


Abbildung 10, Sachbearbeiter Home

Das GUI wurde mit dem von JavaFX zur Verfügung gestellter SceneBuilder erstellt. Dort werden die verschiedenen Ansichten gestaltet. Die Daten werden im Dateiformat .fxml gespeichert. Zu jeder .fxml Datei gibt es eine Controller-Klasse, welche Actions auf den Buttons oder Labels aufruft. Die Controller-Klasse setzt die .fxml Datei also „aktiv“. Die Controller-Klasse und die .fxml Datei sind im gleichen Package. So können Actions in der .fxml Datei aufgerufen werden.

Die verschiedene Clients haben auch unterschiedliche Ansichten; jede Ansicht ist auf die Funktion des jeweiligen Clients zugeschnitten. Somit ist die Autorisierung gewährleistet (der Kunde kann zum Beispiel keine neuen Mitarbeiter erstellen).

9.3 Ergonomie

Unser System sollte eine Verfügbarkeit von 100% aufweisen. Dies ist natürlich nicht möglich, da viele Faktoren zu dieser Verfügbarkeit stören. Fällt einmal der Strom aus, so ist die Verfügbarkeit von 100% schon nicht mehr garantiert.

► *Siehe 9.20 „Hochverfügbarkeit“*

Die Benutzeroberfläche ist freundlich und modern gestaltet. Sie wurde kundenspezifisch erstellt. So wurde auf die Wünsche des Kunden eingegangen und die Software kann optimal genutzt werden. Keine unnötigen Komponenten sind vorhanden, welche Speicherplatz auf dem Rechner brauchen. Die verschiedenen Fenster sind optisch gleich aufgebaut, um den Wiedererkennungswert zu steigern, und die Benutzerfreundlichkeit zu erhöhen.

► *Siehe 9.2 „Benutzeroberfläche“*

9.4 Ablaufsteuerung

► *Siehe 8.1 „Typische Muster und Strukturen“*

9.5 Transaktionsbehandlung

Die Transaktionsbehandlung funktioniert über den Entity-Manager. Bevor die Daten in die Datenbank geschrieben werden, werden sämtliche Aufgaben vom Entity-Manager ausgeführt. Werden Fehler gefunden, werden die Daten weder in die Datenbank geschrieben noch gelesen.

Transaktionen gemäss dem ACID Prinzip:

Atomicity: Die Daten werden ganz oder gar nicht übertragen, sind nicht alle Daten resp. Felder ausgefüllt, werden die Daten nicht gespeichert.

Consistency: Nach Ausführung der Transaktion sind die Daten fest vorhanden. Das heisst, sie sind auf der Datenbank persistent abgespeichert.

Isolation: Bei gleichzeitigen Ausführungen von Transaktionen beeinflussen diese sich gegenseitig nicht.

Durability: Die Daten sind dauerhaft auf der Datenbank vorhanden, können also nicht verloren gehen.



9.6 Sessionbehandlung

Nicht relevant.

9.7 Sicherheit

Jeder Benutzer hat ein eigenes Login. Nur mit korrektem Benutzername und Passwort kann auf die Software zugegriffen werden. Durch diese Authentisierung wird geprüft, ob der Benutzer im System erfasst ist, oder nicht. Hat der Kunde ein Login, welches auf unserem Server gespeichert ist, hat er Zugriff auf seine Daten.

Die Daten sind einmalig auf dem Server vorhanden. Dank der relationalen Datenbank gibt es keine Doubletten.

Aus sicherheitstechnischen Gründen wird im Betrieb jeden Abend ein Backup der gesamten Datenbank erstellt.

Es gibt zwei Usergruppen, Mitarbeiter und Kunde, welche verschiedene Rechte besitzen. Die Gruppe Mitarbeiter ist wiederum in zwei Rollen unterteilt:

- Sachbearbeiter, mit Lese- und Schreibrechte (Vollzugriff)
- Servicetechniker, mit Lese- und Schreibrechte beim Rapport

Die Kundengruppe hat nur Lese- und Schreibrechte bei ihren eigenen Benutzerdaten und bei den Tickets.

9.8 Kommunikation und Integration mit anderen IT-Systemen

Die Kommunikation mit anderen IT-Systemen ist mit Web Service gewährleistet.

9.9 Verteilung

► Siehe 7 „Verteilungssicht“

9.10 Plausibilisierung und Validierung

Typ und Modelle

Wenn ein bestimmter Typ ausgewählt wird, wird durch eine Plausibilitätsprüfung das zuständige Modell angezeigt.

Integer Felder

In Felder, welche als Datentyp „Integer“ definiert sind, können keine anderen Zeichen eingetragen werden, sonst gibt es einen Fehler.



9.11 Ausnahme-/Fehlerbehandlung

Alle unsere Programmfehler und Ausnahmen werden gleich behandelt. Je nach Programmfehler wird ein Fenster aufgerufen, welches den entsprechenden Fehlercode ausgibt. Die Dialog-Fenster besteht aus einer Nachricht und dem Fehlercode.

Wir haben verschiedene Meldungen definiert: es gibt Informations-, Bestätigungs- und Errormeldungen.

Informationsmeldungen stellen kein Fehler dar, sie informieren lediglich den User über ein fehlendes Feld oder wenn keine Ergebnisse vorhanden sind.

Die Bestätigungsmeldung erwartet vom Benutzer eine Entscheidung, ob der gewählte Vorgang tatsächlich durchgeführt werden soll. Wird dem zugesagt („YES“), dann wird der Vorgang fortgesetzt. Klickt man jedoch auf einem der beiden anderen Buttons, wird der Vorgang abgebrochen und das Dialog-Fenster geschlossen.

Eine Errormeldung erscheint bei nicht erfolgreichen Transaktionen. Eine mögliche Ursache ist, dass die Datenbank nicht angesprochen werden kann. Bei diesen Fällen sollte man sich bei der HydraSoft GmbH-Hotline melden.



Abbildung 11, Informationsmeldung

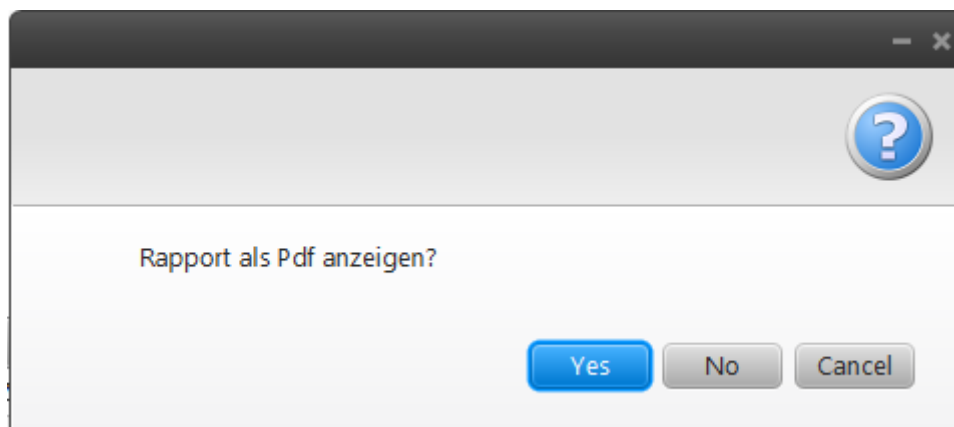


Abbildung 12, Bestätigungsmeldung



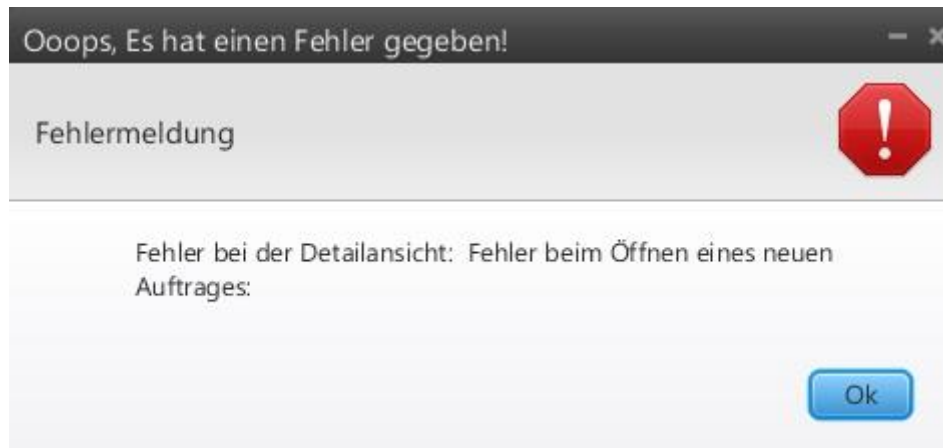


Abbildung 13, Errormeldung

9.12 Management des Systems & Administrierbarkeit

Unser IT-System greift auf dem Server zu, welcher in Horw in der Hochschule T&A steht. Der Server ist von überall zu erreichen. So kann der Administrator der Datenbank auch von fern Daten mutieren.

Der Administratorzugriff bleibt bei der HydraSoft GmbH, da die TotalService GmbH momentan über keine IT-Fachleute verfügt.

9.13 Logging, Protokollierung, Tracing

Der Logger protokolliert uns alle Fehler in eine Log-Datei. Aufgrund des Know-Hows benutzen wir hierzu den ApacheLog4j. Herzlichen Dank hierfür an Her Jordan Suçur für den Input.

Damit verschiedene Personen am Projekt arbeiten können wurde SVN installiert. So können mehrere Personen am gleichen Projekt zeitgleich arbeiten. Damit man nachvollziehen kann, wer welche Änderung durchgeführt hat, musste man dies beim Upload protokollieren. Diese Kommentare wurden in einer History gespeichert.

9.14 Geschäftsregeln

Die TotalService GmbH ist nur eine virtuelle gegründete Firma, dadurch bestehen auch keine Geschäftsregeln.

9.15 Parallelisierung und Threading

Wird automatisch durch RMI gehandhabt.



Internationalisierung

Die TotalService GmbH spezialisiert sich vor allem auf die Zentralschweiz. Eine nationale Expansion ist in Planung.

9.16 Migration

Die Firma TotalService GmbH hat kein Altsystem. Daher müssen keine bestehenden Daten vom Altsystem in unser neues System übertragen werden. Zudem ist kein Parallelbetrieb nötig, da kein Altsystem abgelöst wird.

Die Kundendaten, welche bei der TotalService GmbH bereits als .csv Datei vorliegen, werden von uns in einem SQL Skript in die Datenbank eingelesen, sodass keine Kundendaten verloren gehen. Zukünftig kann das BackOffice die Daten mit Hilfe unserer Software selber einlesen.

9.17 Testbarkeit

Tests Allgemein

Testbarkeit spielt eine wichtige Rolle. Hier werden die Methoden der Klassen getestet. Tests sind die Grundlage für jedes Projekt.

Unit Test

Wir haben unsere Tests mit Hilfe des Tools JUnit-Test durchgeführt. So schrieben wir für die wichtigsten Klassen einige Test Klassen. Wir wollten testen, ob die verschiedenen Methoden wie speichern, löschen oder bearbeiten funktionieren.

Funktionstest (BlackBoxTesting):

Nun folgen einige Funktionsteste. Diese haben wir beim Abschluss der Software durchgeführt, um die Software auf Fehler zu überprüfen.

► *Siehe Anhang „Source Code“*



Test Nr	Site
1	AuftragSuchen
	Beschreibung
	Aufträge können gesucht werden
	Resultat
	Resultate werden in Tabelle dargestellt. Falls keine Ergebnisse vorhanden sind, wird man mittels einer Informationsnachricht benachrichtigt.
	Massnahmen
	Keine

Test Nr.	Site
2	Auftrag Kunde suchen
	Beschreibung
	Suchen eines Kunde
	Resultat
	Die Kunden oder der Kunde erscheint in der untenstehende Resultattabelle
	Massnahmen
	Keine

Test Nr	Site
3	AuftragErfassen
	Beschreibung
	Mitarbeiter können erfasst werden
	Resultat
	Auftrag kann erstellt werden. Im Tab können Rapport und verschiedene Positionen hinzugefügt werden.
	Massnahmen
	Keine



Test Nr	Site
4	MitarbeiterDetail
	Beschreibung
	Bearbeiten von Mitarbeiter
	Resultat
	Mitarbeiter kann nur durch BackOffice bearbeitet werden. Änderungen können nur nach drücken des entsprechenden Buttons vorgenommen werden.
	Massnahmen
	Keine

Test Nr	Site
5	KundeUebersicht
	Beschreibung
	Übernehmen der Werte
	Resultat
	Falls ein Kunde auf der Suche ausgesucht wird, kann er durch anklicken ausgesucht werden. Die Werte werden ins neue Fenster übernommen
	Massnahmen
	Keine

Test Nr	Site
6	KundeHome
	Beschreibung
	Zeigt nur offene Tickets an
	Resultat
	Es werden nur die Tickets angezeigt, welche "in bearbeitung" oder "eröffnet" sind
	Massnahmen
	Keine



Test Nr	Site
7	VertragUebersicht
	Beschreibung
	Gesamtvertrag wird berechnet
	Resultat
	Angaben werden korrekt zusammen gerechnet
	Massnahmen
	Keine

Test Nr.	Site
8	Vertrag, Kunden zuweisen, Kundensuche
	Beschreibung
	Kundensuche
	Resultat
	Die Resultate der Kundensuche werden unten angezeigt
	Massnahmen
	Keine

Test Nr.	Site
9	Vertrag, Gerät hinzufügen
	Beschreibung
	Geräte erfassen
	Resultat
	Das Gerät wurde erfasst, in der untenstehender Tabelle wird das erstellte Gerät angezeigt
	Massnahmen
	Keine



Test Nr.	Site
10	Vertrag, abspeichern
	Beschreibung
	Den Vertrag abspeichern
	Resultat
	Der erstellte Vertrag wurde gespeichert. Das Panel Kunde zuweisen wird inaktiv gesetzt.
	Massnahmen
	Keine

Test Nr.	Site
11	Vertrag, zurück gehen
	Beschreibung
	Mit dem Zurück Button wieder ins Startmenü des Vertrages gelangen
	Resultat
	Das Anfangsmenü des Vertrages erscheint wieder.
	Massnahmen
	Keine

Test Nr	Site
12	VertragSuchen
	Beschreibung
	Zurücksetzen der Werte
	Resultat
	Werte werden korrekt zurück gesetzt
	Massnahmen
	Keine



Test Nr	Site
13	Login
	Beschreibung
	Überprüfen von Daten
	Resultat
	Username und Passwort werden überprüft, falls es einen Fehler gibt, wird entsprechend eine Fehlermeldung angezeigt
	Massnahmen
	Keine

Test Nr	Site
14	Mitarbeitererfassen
	Beschreibung
	Speichern nur mit gültigen Daten
	Resultat
	Mitarbeiter kann nur gespeichert werden, wenn alle Felder ausgefüllt sind.
	Massnahmen
	Keine

Test Nr	Site
15	Generell
	Beschreibung
	Exceptions werden dokumentiert
	Resultat
	Exceptions werden mit Hilfe des Logger4J in eine History eingetragen.
	Massnahmen
	Keine



Test Nr	Site
16	SachbearbeiterHome
	Beschreibung
	Anzeige von auslaufenden Verträgen
	Resultat
	Es werden nur Verträge angezeigt, welche kein Verlängerungsdatum haben und in den nächsten 30 Tagen ablaufen
	Massnahmen
	Keine

Test Nr	Site
17	TerminSuche
	Beschreibung
	Termin für Kunden auswählen
	Resultat
	Es können nur Servicetechniker ausgewählt werden, welche an diesem Termin noch frei sind
	Massnahmen
	Keine

Test Nr	Site
18	Ersatzteile
	Beschreibung
	Ersatzteile hinzufügen
	Resultat
	Ersatzteile können nur hinzugefügt werden, wenn korrekte Daten eingegeben werden.
	Massnahmen
	Keine



Test Nr.	Site
19	Ersatzteil
	Beschreibung
	Ersatzteil kann geändert werden
	Resultat
	Ersatzteile können mit Hilfe des Aktualisieren-Buttons geändert werden. Wenn im Feld Preis Buchstaben eingegeben werden, erscheint eine Fehlermeldung und die Änderung wird nicht vorgenommen.
	Massnahmen
	Keine

Test Nr	Site
20	KundeSuchen
	Beschreibung
	Suche von Kunden
	Resultat
	Kunden können gesucht werden. Daten werden in das Fenster KundeDetail übernommen und können nicht bearbeitet werden
	Massnahmen
	Keine

Test Nr.	Site
21	Kunden erstellen
	Beschreibung
	Kunden konnte erfasst werden
	Resultat
	Kunde konnte erstellt werden. Es erscheint eine Meldung dass der Kunde erfasst wurde.
	Massnahmen
	Keine



Test Nr.	Site
22	Kunden auswählen
	Beschreibung
	Auswählen eines Kunde
	Resultat
	Der ausgewählte Kunde erscheint in einem neuen Fenster mit ausgefüllten Daten.
	Massnahmen
	Keine

Test Nr.	Site
23	MitarbeiterTab
	Beschreibung
	Mitarbeiter Austrittsdatum zurücksetzen
	Resultat
	Mit einem Klick auf den Button Austritt zurücksetzen wird das Austrittsdatum gelöscht. Die Checkbox neben dem Feld wird deaktiviert
	Massnahmen
	Keine

Test Nr.	Site
24	MitarbeiterTab
	Beschreibung
	Mitarbeiter Austrittsdatum erfassen
	Resultat
	Das Austrittsdatum kann mit der Kalenderkomponente erfasst werden. Die Checkbox neben dem Datum wird aktiviert.
	Massnahmen
	Keine

Test Nr.	Site
25	MitarbeiterTab
	Beschreibung
	Mitarbeiterdaten ändern
	Resultat
	Durch einen Klick auf den Bearbeiten-Button können die Felder mit den Mitarbeiterdaten geändert werden. Mit einem Klick auf den Speichern-Button wird der bestehende Datensatz nur überschrieben, wenn korrekte Daten erfasst wurden. Die Mitarbeiter-Nr. kann nicht geändert werden.
	Massnahmen
	Keine

Test Nr.	Site
26	MitarbeiterTab
	Beschreibung
	Mitarbeiterübersicht verlassen
	Resultat
	Mit einem Klick auf den Zurück-Button gelangt man zurück in die Mitarbeiter Suche
	Massnahmen
	Keine

Test Nr.	Site
27	MitarbeiterTab
	Beschreibung
	In Mitarbeiterübersicht gelangen
	Resultat
	Nach der Suche nach einem Mitarbeiter wird mit einem Klick auf den gewünschten Mitarbeiter die Übersicht geöffnet
	Massnahmen
	Keine

Test Nr.	Site
28	MitarbeiterTab
	Beschreibung
	Nach Funktion der Mitarbeiter suchen
	Resultat
	Mittels Dropdownfenster kann Wahlweise nach Sachbearbeitern oder Servicetechniker gesucht werden.
	Massnahmen
	Keine

Test Nr.	Site
29	MitarbeiterTab
	Beschreibung
	Nach Name des Mitarbeiter suchen
	Resultat
	Mittels Eingabe eines Mitarbeiternamens kann nach dem entsprechenden Mitarbeiter gesucht werden. Besteht kein Mitarbeiter mit diesem Namen, wird dies mit einer Meldung angezeigt
	Massnahmen
	Keine

Test Nr.	Site
30	MitarbeiterTab
	Beschreibung
	Nach aktiven/inaktiven Mitarbeitern suchen
	Resultat
	Mit dem Radiobutton kann auf Wunsch nach aktiven oder inaktiven Mitarbeitern gesucht werden. Wenn keine Ergebnisse vorhanden sind, wird dies mit einer entsprechenden Meldung angezeigt
	Massnahmen
	Keine



Test Nr	Site
31	Rapport
	Beschreibung
	Eintragen der Geräte
	Resultat
	Bei bestehendem Kunde werden die vorhandenen Geräte in den RapportTab eingetragen
	Massnahmen
	Keine

Test Nr	Site
32	AuftragPos
	Beschreibung
	Suchen von Servicetechniker
	Resultat
	Bei Suche von Servicetechniker wird ein PopUp Fenster geöffnet und der Servicetechniker kann gesucht werden
	Massnahmen
	Keine

Test Nr	Site
33	TerminSuchen
	Beschreibung
	Suchen eines Termin mit Datum
	Resultat
	Es kann nur ein Servicetechniker gesucht werden, wenn das Datum angegeben wird. Andernfalls wird eine Fehlermeldung angezeigt.
	Massnahmen
	Keine



Test Nr	Site
34	AuftragUebersicht
	Beschreibung
	Erstellen des PDF
	Resultat
	Rapport wird als PDF exportiert und kann ausgedruckt werden
	Massnahmen
	Keine

Test Nr.	Site
35	Auftrag, Auftrag erstellen
	Beschreibung
	Auftrag erfassen(Position
	Resultat
	Die Position des Auftrages konnte nicht hinzugefügt werden.
	Massnahmen
	Keine

Test Nr	Site
36	VertragUebersicht
	Beschreibung
	Anzeigen der im Vertrag enthaltenen Geräte
	Resultat
	Wird ein bestehender Vertrag ausgewählt, so werden in der Übersicht die dazu gehörigen Geräte angezeigt
	Massnahmen
	Keine



Test Nr	Site
37	GeraeteTab
	Beschreibung
	GeraeteTyp kann aktualisiert werden
	Resultat
	Wenn ein GeraeteTyp erstellt wird, kann er mit Hilfe des Aktualisieren Button aktualisiert werden.
	Massnahmen
	Keine

Test Nr	Site
38	GeraeteTab
	Beschreibung
	GeraeteModell kann hinzugefügt werden
	Resultat
	GeräteModell können hinzugefügt werden und werden sofort in der Tabelle angezeigt
	Massnahmen
	Keine

Test Nr.	Site
39	GeraetModell
	Beschreibung
	ModellTyp zu GeraetModell hinzufügen
	Resultat
	Bereits erstellte GeraetTyp können mit einem Dropdownfeld an das entsprechende GeraetModell angeheftet werden
	Massnahmen
	Keine



Test Nr.	Site
40	GeraeteModell
	Beschreibung
	GeraeteModell kann aktualisiert werden
	Resultat
	Wenn bestehendes GeraeteModell ausgewählt wird, erscheint ein neues Fenster, in welchem die Daten geändert werden können. Mit Hilfe des Speichern-Buttons wird das GeraeteModell aktualisiert.
	Massnahmen
	Keine

Test Nr.	Site
41	GeraetTyp
	Beschreibung
	GeraetTyp kann hinzugefügt werden
	Resultat
	GeraetTyp kann erfolgreich hinzugefügt werden und wird prompt in der oberen Tabelle angezeigt.
	Massnahmen
	Keine

Test Nr	Site
42	VertragTab
	Beschreibung
	Kundezuweisen
	Resultat
	Beim Erstellen eines neuen Vertrages kann man einen Kunden zuweisen.
	Massnahmen
	Keine



Test Nr	Site
43	PasswortÄndern
	Beschreibung
	Passwort ändern
	Resultat
	Das Passwort kann erfolgreich geändert werden.
	Massnahmen
	Keine

Test Nr	Site
44	TicketSuche
	Beschreibung
	Nach Ticket suchen
	Resultat
	Tickets werden in der Tabelle angezeigt
	Massnahmen
	Keine

9.18 Skalierung, Clustering

Die Datenbank, welche wir für unser System brauchen ist ohne Probleme erweiterbar. Bei steigender Benutzerzahl oder beim Einsatz zusätzlicher Anwendungen ist die installierte Rechnerleistung leicht anpassbar. Unser System ist stufenlos erweiterbar und in kleinen Architekturen (z.B. 20 Arbeitsplätze) ebenso zu nutzen wie in grossen (z.B. 1000 Arbeitsplätzen). Das System wächst jederzeit mit den Anforderungen der Unternehmung. Die Schichten der 3-Tier Architektur sind separat erneuerbar.

9.19 Hochverfügbarkeit

Unser Ziel ist es eine Verfügbarkeit von 99.99 % zu erreichen. Dies entspricht der Verfügbarkeitsklasse 4. Konkret können wir eine maximale Ausfallzeit von 52 Minuten pro Jahr mit einberechnen. Da die Datenbank hinter einer USV ist, kann auch bei einem Stromunterbruch im weiter gearbeitet werden. Im Katastrophenfall wären immer noch Backups vorhanden, damit die Kundendaten nicht verloren gehen.



10. Entwurfsentscheidungen

10.1 Entwurfsentscheidung 1

10.1.1 Fragestellung

Welches Programm nehmen wir, um das GUI zu gestalten und realisieren?

10.1.2 Rahmenbedingungen

Das GUI sollte möglichst einfach zu bedienen sein und sollte der neusten Technik entsprechen. Es muss kompatibel mit Java sein, da Java unsere Programmiersprache ist.

10.1.3 Annahmen

Keine

10.1.4 Betrachtete Alternativen

Swing:

Das Programm und das Aussehen sind nicht mehr zeitgemäss.

10.1.5 Entscheidung

Wir haben uns für JavaFX entschieden, da es moderner- und anpassungsfähiger ist wie Swing. Das Design ist freundlicher und attraktiver.



10.2 Entwurfsentscheidung 2

10.2.1 Fragestellung

Wie funktioniert der Prozessablauf auf dem GUI?

10.2.2 Rahmenbedingungen

Der Ablauf sollte wie das Design einheitlich sein. Durch den Wiedererkennungswert ist eine Einarbeitung viel leichter zu vollziehen. Die verschiedenen Abläufe sind darum ähnlich aufgebaut. Konkret besteht unser Hauptfenster aus einem fixen Bereich (links mit Buttons) und rechts mit einem dynamischen Bereich.

10.2.3 Annahmen

Wir nahmen an, dass es für die Anwender einfacher ist, schnell sich in die Anwendung einzulesen, wenn überall der gleiche Prozess stattfindet. Es sollte auch übersichtlich sein, welches Fenster im Moment offen ist. Darum wird fast alles im gleichen Fenster geöffnet, somit hat man eine klare Übersicht.

10.2.4 Betrachtete Alternativen

Keine, der Prozess wurde einmalig besprochen, einstimmig angenommen und dann auch so ausgeführt.

10.2.5 Entscheidung

► Siehe 6.3 „Laufzeitszenario 3“



10.3 Entwurfsentscheidung 3

10.3.1 Fragestellung

Wie sieht die Paketstruktur aus?

10.3.2 Rahmenbedingungen

Die Paketstruktur sollte logisch und übersichtlich aufgebaut werden.

10.3.3 Annahmen

Keine

10.3.4 Betrachtete Alternativen

Keine, die Paketstruktur wurde einmalig besprochen und realisiert.

Durch die dazugewonnenen Inputs musste die Paketstruktur jedoch im späterem Verlauf angepasst werden.

10.3.5 Entscheidung

► Siehe 3.1.2 „Paketdiagramm Miraculix Client Sachbearbeiter“

► Siehe 3.1.3 „Paketdiagramm Miraculix Business“



11. Qualitätsszenarien

11.1 Qualitätsbaum

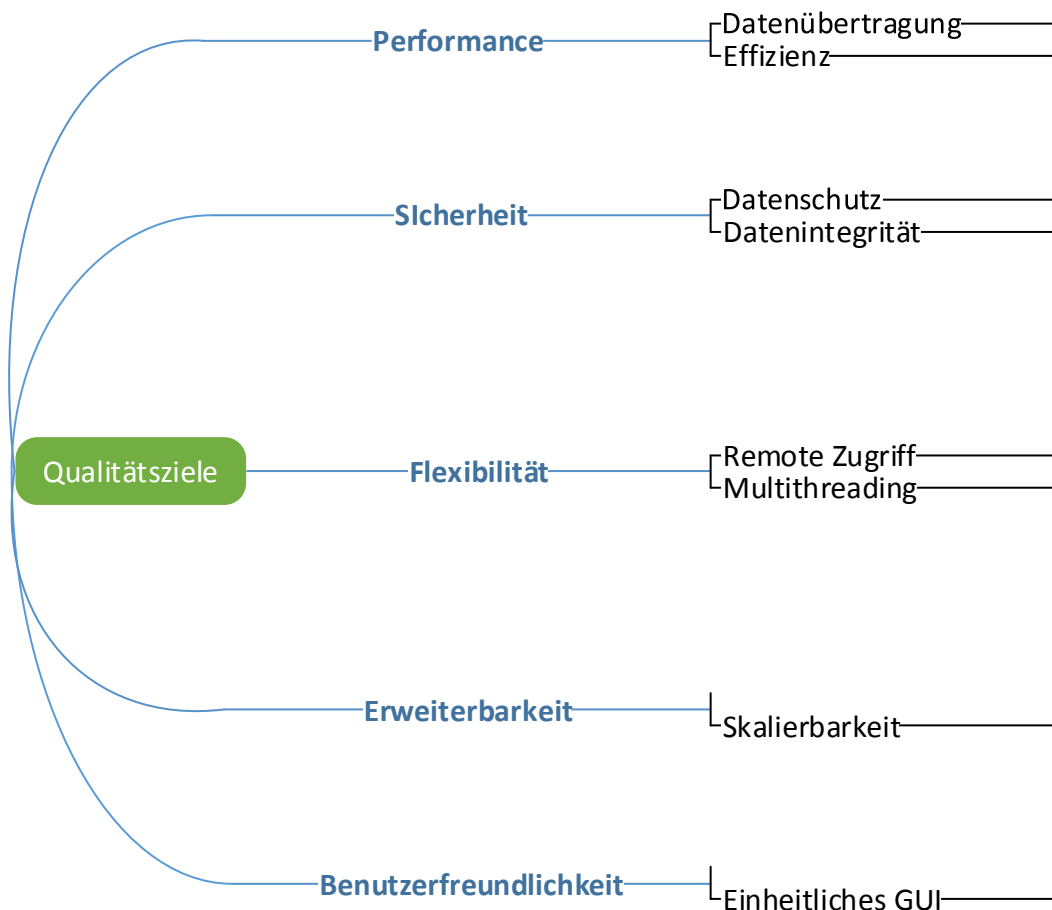


Abbildung 14, Qualitätsbaum

Datenübertragung

Nachdem neue Daten in der Datenbank gespeichert wurden, sind sie für alle autorisierten Benutzer sofort abrufbar. Bemerkbar ist dies zum Beispiel, wenn ein neuer Mitarbeiter oder Kunde eingetragen wird. Nach dem Speichern ist es möglich, die Suche durchzuführen und die neu erstellte Person zu finden.

Effizienz

Um das Qualitätsziel „Performance“ zu erreichen setzen wir RMI, Webservice und PostgreSQL ein. Jede Komponente ist Zuständig für diverse Schlüsselaufgaben im gesamten Systemprozess.

► Siehe 7.0 „Verteilungssicht“



Datenschutz

Um auf das Programm und die Daten zugreifen zu können, bedarf es zuerst einer Authentisierung durch ein Login. Wird das Login angenommen, erscheinen die autorisierten Module des jeweiligen Benutzers. Zugriff auf fremde Module ist nicht möglich, da jeder Benutzer durch ein Persontyp (Mitarbeiter oder Kunde) unterschieden wird. Der Persontyp „Mitarbeiter“ wird weiter in den Funktionen „Sachbearbeiter“ und „Servicetechniker“ aufgeteilt. Somit wird gewährleistet, dass für jeden Benutzer nur die Daten und Funktionen von seiner Zuständigkeit sichtbar und abrufbar sind.

Datenintegrität

Die relationale Datenbank ist in logische Tabellen aufgebaut. Beim Design der Datenbank wurde darauf geachtet, dass keine Doubletten erfasst werden können. Des Weiteren wurden die nötigen Felder mit „not-null“ gesichert, um keine null-Werte zuzulassen. Um die Datenintegrität zu erhöhen wurden den Attributen die zutreffenden Datentypen zugeteilt. Es ist zum Beispiel nicht möglich, eine Postleitzahl mit Buchstaben in der Datenbank zu speichern.

Remote-Zugriff

Mit Webservice ist es möglich, von einem beliebigen Standort auf unsere Business-Logik zuzugreifen. Es wird lediglich eine Internetverbindung vorausgesetzt.

Multithreading

Diese Aufgabe wird von RMI durchgeführt. Sie ist dafür verantwortlich, dass verschiedene Benutzer gleichzeitig auf unsere Applikation und unsere Daten zugreifen können.

Skalierbarkeit

► Siehe 9.18 „Skalierung, Clustering“

Einheitliches GUI

► Siehe 9.2 „Benutzeroberfläche“ ff.



11.2 Bewertungsszenarien

Nutzungsszenarien (Qualitätsziel Flexibilität):

Starten des Programm aus der Sicht des Servicetechnikers.

Der Servicetechniker möchte vor Ort beim Kunden den aktuellen Auftrag rapportieren. Dazu startet er unsere Software. Mit dem persönlichen Login meldet sich der Servicetechniker bei unserer Datenbank an, dazu muss er nur mit dem Internet verbunden sein. Da jeder Servicetechniker mit einem UMTS-Stick der Swisscom ausgestattet ist, ist er immer mit dem Internet verbunden. Nun kann er auf den aktuellen Auftrag zugreifen und diesen rapportieren. Sobald der Servicetechniker den Rapport zum zugehörigen Auftrag quittiert, wird der Status auf „abgeschlossen“ gesetzt. Das BackOffice sieht dies und kann nun den Auftrag verrechnen. Der Servicetechniker loggt sich aus und kann zum nächsten Kunden gehen.

Nutzungsszenarien (Qualitätsziel Benutzerfreundlichkeit):

Starten des Programm aus der Sicht des BackOffice.

Das BackOffice loggt sich mit dem persönlichen Login ein. Es erscheint die Menüansicht. Diese ist sehr benutzerfreundlich aufgebaut. Links hat das BackOffice alle wichtigen Buttons zur Verfügung wie Auftrag, Vertrag oder Mitarbeiter. Das BackOffice kann nun einfach per Mausklick die gewünschte Seite öffnen. Es wird kein neues Fenster geöffnet sondern es wird in der gleichen Ansicht im Parent geöffnet. Somit bleibt die ganze Software übersichtlich.



12. Risiken

Das Risikomanagement umfasst sämtliche Maßnahmen zur systematischen Erkennung, Analyse, Bewertung, Überwachung und Kontrolle von Risiken. Hier werden die uns wichtigsten Risiken aufgezählt.

Risiko 1:

Datenverlust von Kundendaten oder internen Daten.

Eintrittswahrscheinlichkeit:

Die Eintrittswahrscheinlichkeit ist klein.

Schadenshöhe:

Je nach Verlust grösser oder kleiner

Massnahmen:

Jeden Tag wird ein Backup der Kundendaten durchgeführt. Datenbank ist hinter einer USV, so ist auch bei Stromausfall die Datenbank weiterhin online.



13. Glossar

Abbildungsverzeichnis

Abbildung 1, Komponentendiagramm.....	12
Abbildung 2, Paketdiagramm Sachbearbeiter.....	13
Abbildung 3, Paketdiagramm Business	13
Abbildung 4, Bausteinschicht.....	15
Abbildung 5:Sequenzdiagramm Schreibe TicketPosDB	30
Abbildung 6, Aktivitätsdiagramm	31
Abbildung 7, Verteilungskontext	32
Abbildung 8, MVC Modell	33
Abbildung 9, Sequenzdiagramm MVC.....	34
Abbildung 10, Sachbearbeiter Home	35
Abbildung 11, Informationsmeldung	38
Abbildung 12, Bestätigungsmeldung	38
Abbildung 13, Errormeldung.....	39
Abbildung 14, Qualitätsbaum.....	59

Tabellenverzeichnis

Tabelle 1, Änderungsübersicht	2
Tabelle 2, Basisdokumente	2
Tabelle 3, Qualitätsziele	7
Tabelle 4, Stakeholder.....	8
Tabelle 5, Technische Randbedingungen.....	9
Tabelle 6, Organisatorische Randbedingungen	10
Tabelle 7, Konventionen.....	11
Tabelle 8, Kontextabgrenzung.....	12



14. Anhang

14.1 Bedienungsanleitung

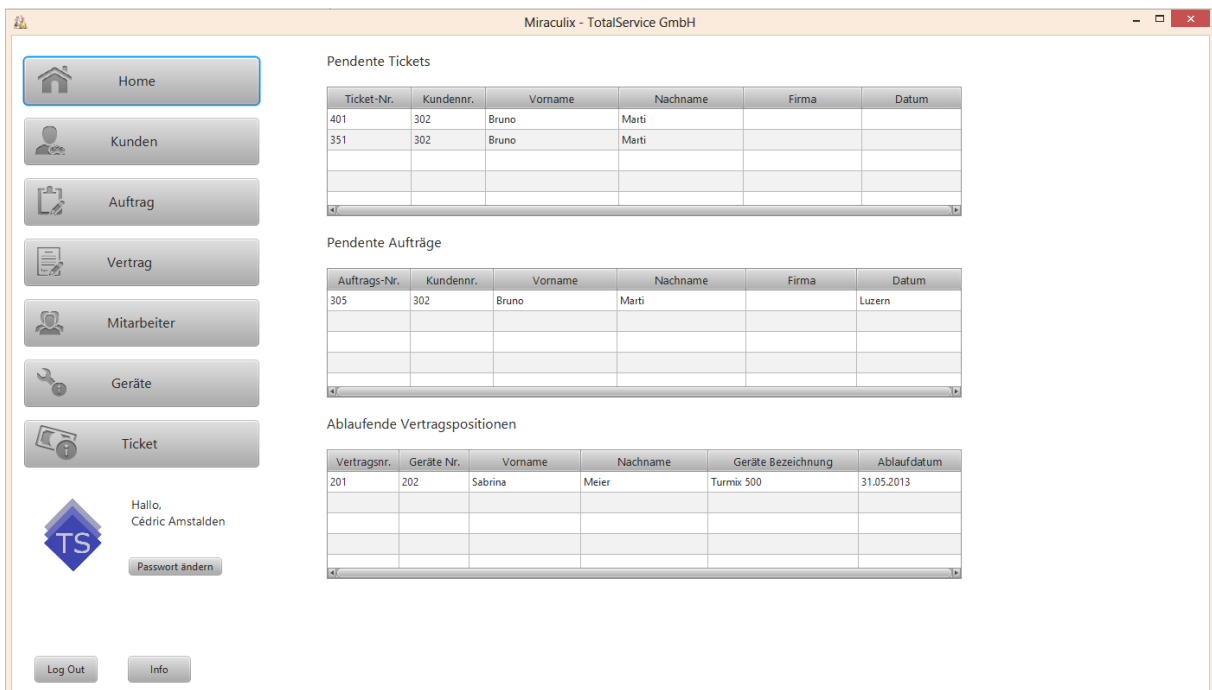
Anleitung für den Sachbearbeiter

Benutzer: cedric

Passwort: test

Einleitung

Nach dem Start der Applikation erscheint das LoginFenster. Geben Sie den Benutzernamen und das Passwort ein und klicken Sie auf bestätigen. Danach erscheint das Homefenster des Sachbearbeiters:



The screenshot shows the 'Miraculix - TotalService GmbH' application window. On the left is a sidebar with navigation buttons: Home (selected), Kunden, Auftrag, Vertrag, Mitarbeiter, Geräte, and Ticket. Below these buttons is a user profile section with a 'TS' logo, the name 'Hallo, Cédric Amstalden', and a 'Passwort ändern' button. At the bottom of the sidebar are 'Log Out' and 'Info' buttons. The main content area displays three tables:

Pendente Tickets

Ticket-Nr.	Kundennr.	Vorname	Nachname	Firma	Datum
401	302	Bruno	Marti		
351	302	Bruno	Marti		

Pendente Aufträge

Auftrags-Nr.	Kundennr.	Vorname	Nachname	Firma	Datum
305	302	Bruno	Marti		Luzern

Ablaufende Vertragspositionen

Vertragsnr.	Geräte Nr.	Vorname	Nachname	Geräte Bezeichnung	Ablaufdatum
201	202	Sabrina	Meier	Turmix 500	31.05.2013

1. Home: Dieser Button führt zum oben gezeigten Homescreen der auch dazu der Startbildschirm ist. Dort erhält der Sachbearbeiter eine Übersicht über die pendenten Tickets, Aufträge und ablaufende Vertragspositionen. Wird auf ein Element in der Tabelle geklickt, wechselt das Programm auf die entsprechende Seite und die Einträge können bearbeitet werden.

2. Kunden: Bei Klick wird man auf die Kundensuche weitergeführt. Dor kann anhand verschiedener Kriterien nach Kunden gesucht werden und es können neue Kunden erfasst werden. Wurde eine erfolgreiche Suche durchgeführt, wird durch ein Klick auf ein Ergebnis die Detailansicht geöffnet und die Kundenstammdaten können bearbeitet werden.

Suche

Kundennummer	<input type="text"/>
Firmenname	<input type="text"/>
Vorname	<input type="text"/>
Nachname	<input type="text"/>
Strasse	<input type="text"/>
PLZ / Ort	<input type="text"/> / <input type="text"/>
Kundenstatus	<input type="radio"/> In Bearbeitung <input type="radio"/> Aktiv <input type="radio"/> Inaktiv
	<input type="button" value="Suchen"/> <input type="button" value="Zurücksetzen"/>

Resultate

Kundennr.	Firmenname	Nachname	Vorname	Strasse	Ort	Status	Login	
302		Marti	Bruno	Musterweg 2	Luzern	aktiv	Ja	
304		Meier	Sabrina	Musegggasse 34	Meggen	aktiv	Ja	

Durch Klick auf eines der Suchergebnisse öffnet sich die Detailansicht:

Kundenerfassung

Kundennummer	<input type="text" value="302"/>
Firmenname	<input type="text"/>
Anrede	<input type="text" value="Herr"/>
Vorname	<input type="text" value="Bruno"/>
Nachname	<input type="text" value="Marti"/>
Strasse	<input type="text" value="Musterweg 2"/>
PLZ / Ort	<input type="text" value="6001"/> / <input type="text" value="Luzern"/>
Telefon	<input type="text" value="0793456232"/>
Mobile	<input type="text"/>
Geschäftlich	<input type="text"/>
Email	<input type="text" value="cedric.amstalden@stud.hslu.ch"/>
Kundenstatus	<input type="radio"/> In Bearbeitung <input checked="" type="radio"/> Aktiv <input type="radio"/> Inaktiv
	<input type="button" value="Bearbeiten"/>

Nach dem Klick auf Bearbeiten können alle Felder editiert und danach gespeichert werden. Ebenfalls besteht die Möglichkeit ein Kundelogin generieren zu lassen oder das Passwort zurückzusetzen falls schon ein Login besteht.



Wählt man Neukunde erfassen erscheint das unterstehende Fenster. Dort müssen alle mit * gekennzeichneten Felder ausgefüllt werden. Auch ein Status muss gesetzt werden. Wird Login versenden auf Ja gesetzt, generiert die Software automatisch ein Login und sendet es an die eingetragene Emailadresse.

Kundenerfassung

Firmenname	<input type="text"/>		
Anrede	<div><div></div><div></div></div>	*	
Vorname	<input type="text"/>	*	
Nachname	<input type="text"/>	*	
Strasse	<input type="text"/>		
PLZ / Ort	<div><div><input type="text"/></div><div>*</div><div>/</div><div><input type="text"/></div></div>	*	
Telefon / Handy	<input type="text"/>	*	
Telefon 2 / Handy 2	<input type="text"/>		
Geschäftlich	<input type="text"/>		
Email	<input type="text"/>	*	
Kundenstatus	<div><div></div> In Bearbeitung</div>	<div><div></div> Aktiv</div>	*
Logindaten versenden?	<div><div></div> Ja</div>	<div><div></div> Nein</div>	*

Speichern

3. Auftrag: Im Auftragstab könne neue Aufträge erfasst werden und es kann nach bestehenden Aufträgen gesucht werden.

Suche

Auftragsnummer:

Kundennummer:

Kundenname:

Datum Auftrag: bis

Servicetechniker:

Status: ☐ In Bearbeitung ☐ Bearbeitet ☐ Abgeschlossen

[illegible]

Klickt man auf neuen Auftrag erstellen muss zuerst der Kunde hinzugefügt werden. Dieser kann via Suche gefunden werden und durch Klick hinzugefügt werden:

Kunde
Positionen
Rapport
Übersicht

Suche

Kundennummer

Firmenname

Vorname

Nachname

Strasse

PLZ / Ort /

Kundenstatus ☐ In Bearbeitung ☐ Aktiv ☐ Inaktiv

Resultate

Kundennr.	Firmenname	Nachname	Vorname	Strasse	Ort	Status	Login	
302		Marti	Bruno	Musterweg 2	Luzern	aktiv	Ja	

Unter Position können Geräte mit Problembeschrieb hinzugefügt werden und es kann ein Servicetechniker hinzugefügt werden.

Kunde
Positionen
Rapport
Übersicht

Position

Gerät:

Preis Total:

Erfassdatum:

Servicetechniker:

Termin:

Problem Beschrieb:

Positionen:

Position Nr.	Geräte Nr.	Termin	Problembeschrieb
Kein Content in Tabelle			

Der Servicetechniker wird via Suche hinzugefügt. Zuerst wird nach einem verfügbaren Servicetechniker gesucht und danach den genauen Termin festgelegt werden.

Termin suchen

Suche

Vorname:

Nachname:

Datum:

28.05.2013

Block:

☐ A (08:00 - 10:00)
 ☐ B (10:00 - 12:00)
 ☐ C (13:00 - 15:00)
 ☐ D (15:00 - 17:00)

Suchen

Zurücksetzen

Resultate

MA-Nr.	Vorname	Name	
10001	Max	Muster	

Nach dem Klick auf Hinzufügen wird die Position gespeichert:

Kunde

Positionen

Rapport

Übersicht

Position

Gerät:

Preis Total:

Erfassdatum:

Servicetechniker:

Suchen

Termin:

Problem Beschrieb:

Hinzufügen

Zurücksetzen

Positionen:

Position Nr.	Geräte Nr.	Termin	Problembeschrieb
1	154 Waschmaschine V-Zug W320	28.05.2013	Die Waschmaschine ist nicht mehr dicht.

Weiter

Zurück

Im Tab Rapport kann nach erfolgreichem Abschluss der Rapport generiert werden.

Kunde	Positionen	Rapport	Übersicht
Service Techniker MA-Nr: Name, Vorname:		Kunde KundenNr: 302 Name, Vorname: Marti , Bruno	
<div>▼ Geräte</div> <div>154 Waschmaschine V-Zug W320</div>			

Die Übersicht bietet vor dem Speichern nochmals die Gelegenheit zur Kontrolle:

Kunde	Positionen	Rapport	Übersicht
Zurück			
Kundenangaben: Auftrags-Nr: 0 Kunden-Nr: 302 Firmenname: Vorname: Bruno Nachname: Marti Strasse: Musterweg 2 PLZ / Ort: 6001 , Luzern			
Gerät	Problem Beschrieb		
154 Waschmaschine V-Zug W320	Die Waschmaschine ist nicht mehr dicht.		

Drucken Abschliessen



4. Vertrag: In der Vertragsübersicht kann nach Verträgen gesucht werden und es können neue Verträge erfasst werden.

Suche

Vertragsnummer:

Kundennummer:

Nachname:

Ort:

Servicenummer:

Gerätetyp:

Gerätemodell:

Herstellungsjahr:

Seriennummer:

Gültig von: bis

Resultate

Vertragsnr.	Kundennr.	Name	Vorname	Adresse	PLZ	Ort	
153	302	Marti	Bruno	Musterweg 2	6001	Luzern	

Klickt man auf neuen Vertrag muss man zuerst den Kunden hinzufügen. Dieser wird über die Suche gefunden und mit einem Klick markiert:

Suche

Kundennummer:

Firmenname:

Vorname:

Nachname:

Strasse:

PLZ / Ort: /

Kundenstatus: ☐ In Bearbeitung ☐ Aktiv ☐ Inaktiv

Resultate

Kundennr.	Firmenname	Nachname	Vorname	Strasse	Ort	Status	Login	
456		Stark	Tony	Küstenhaus 1	Zürich	aktiv	Ja	



Danach kann man in den Tab Gerät hinzufügen wechseln und eine beliebige Anzahl Geräte hinzufügen:

Kunden zuweisen Geräte hinzufügen Übersicht

Servicenummer	<input type="text" value="478125468"/>
Gerätetyp	<input type="button" value="Waschmaschine"/> ▼
Gerätemodell	<input type="button" value="V-Zug W320"/> ▼
Herstellungsjahr	<input type="text" value="2010"/>
Verkaufsdatum	<input type="text" value="24.05.2013"/>
Seriennummer	<input type="text" value="987654321"/>
Gültig bis	<input type="text" value="24.05.2015"/>
Betrag	<input style="border: 2px solid blue;" type="text" value="8000"/>

5. Mitarbeiter: Hier kann nach Mitarbeitern gesucht werden und es können neue Mitarbeiter erfasst werden.

[illegible]

Auch hier führt ein Klick auf das Suchergebnis in die Detailsicht wo der Mitarbeiter bearbeitet werden kann:

Mitarbeiter-Nr:	<input type="text" value="10001"/>	Mitarbeiter-Nr:	<input type="text" value="10001"/>
Anrede:	<input type="text" value="Herr"/>	Anrede:	<input type="text" value="Herr"/>
Vorname:	<input type="text" value="Max"/>	Vorname:	<input type="text" value="Max"/>
Nachname:	<input type="text" value="Muster"/>	Nachname:	<input type="text" value="Muster"/>
Funktion:	<input type="text" value="Servicetechniker"/>	Funktion:	<input type="text" value="Servicetechniker"/>
Adresse:	<input type="text" value="Glaubenweg 2"/>	Adresse:	<input type="text" value="Glaubenweg 78"/>
PLZ / Ort:	<input type="text" value="7000"/> <input type="text" value="Chur"/>	PLZ / Ort:	<input type="text" value="7000"/> <input type="text" value="Chur"/>
Telefonnr:	<input type="text" value="04134567233"/>	Telefonnr:	<input type="text" value="04134567233"/>
Mobilennr:	<input type="text" value="07423476223"/>	Mobilennr:	<input type="text" value="07423476223"/>
Email:	<input type="text" value="max.muster@totalservice.com"/>	Email:	<input type="text" value="max.muster@totalservice.com"/>
Eintritt:	<input type="text" value="01.11.2011"/>	Eintritt:	<input type="text" value="01.11.2011"/>
Austritt:	<input type="text" value="dd.MM.yyyy"/>	Austritt:	<input type="text" value="dd.MM.yyyy"/>
<input type="button" value="Austritt zurücksetzen"/>		<input type="button" value="Austritt zurücksetzen"/>	
Stundenansatz:	<input type="text" value="150.0"/>	Stundenansatz:	<input type="text" value="150.0"/>

Bei Neuerfassung von Mitarbeitern muss darauf geachtet werden, dass alle Felder korrekt sind und die korrekte Emailadresse eingetragen ist. Auch hier wird der Benutzername und das Passwort beim Erstellen automatisch an die eingetragene Emailadresse gesendet.

6. Geräte: Unter Geräte werden die verschiedenen Gerätetypen, Gerätemodelle und dazugehörige Ersatzteile verwaltet. Im Startfenster können mit einem Klick auf Hinzufügen neue Gerätetypen erfasst werden.

Gerätetyp	Gerätemodell	Ersatzteile
Geräte		
Bezeichnung		
Waschmaschine		
Mixer		
Geschirrspüler		
Luftbefeuchter		

Im Tab Gerätemodell können Geräte mit detaillierten Informationen aufgenommen werden:

Gerätetyp
Gerätemodell
Ersatzteile

Gerätemodell

Modell	Typ	Servicebeitrag pro Jahr	Max Verlängerung in Jahren	Garantie in Jahren	Hersteller	
V-Zug W320	Waschmaschine	100.0	5	3	V-Zug	
Turmix 500	Mixer	15.0	5	1	Turmix	
V-Zug SL-230	Geschirrspüler	80.0	6	2	V-Zug	

FreshAir
Luftbefeuchter
22
2
V-Zug
Hinzufügen

7. Tickets: Hier können nach Tickets gesucht diese bearbeitet werden.

Ticketsuche

Ticket-Nr:

Kunden-Nr:

Vorname:

Nachname:
















































Position Erfassungsdatum: bis

Status: ☒ eröffnet ☐ abgeschlossen

[illegible]

14.2 Installations- und Deploymentanweisungen

1. Starten Sie die MiraculixBusiness.bat Datei und warte Sie bis auf der Konsole „Server erfolgreich gestartet.“ steht.
2. Kopieren Sie nun die Web Archive Files
(,MiraculixWebServiceKunde.war‘ und ,MiraculixWebServiceServicetechniker.war‘)
in das webapps-Verzeichnis Ihrer TomCat-Installation.
3. Beachten Sie, dass folgende Bibliotheken im TomCat lib-Verzeichnis zwingend vorhanden sein müssen:

 annotations-api.jar	 catalina.jar	 catalina-ant.jar	 catalina-ha.jar
 catalina-tribes.jar	 ecj-4.2.2.jar	 eclipselink.jar	 el-api.jar
 FastInfoset.jar	 gmbal-api-only.jar	 ha-api.jar	 jasper.jar
 jasper-el.jar	 javax.annotation.jar	 javax.persistence_2.0.4.jar	 jaxb-api.jar
 jaxb-impl.jar	 jaxb-xjc.jar	 jaxp-api.jar	 jaxws-api.jar
 jaxws-rt.jar	 jaxws-rt-javadoc.jar	 jaxws-tools.jar	 jaxws-tools-javadoc.jar
 jsp-api.jar	 jsr181-api.jar	 mail.jar	 management-api.jar
 mimepull.jar	 policy.jar	 saaj-api.jar	 saaj-impl.jar
 servlet-api.jar	 stax2-api.jar	 stax2-api-source.jar	 stax-ex.jar
 streambuffer.jar	 tomcat-api.jar	 tomcat-coyote.jar	 tomcat-dbcp.jar
 tomcat-i18n-es.jar	 tomcat-i18n-fr.jar	 tomcat-i18n-ja.jar	 tomcat-jdbc.jar
 tomcat-util.jar	 woodstox-core-asl.jar	 woodstox-core-asl-source.jar	

4. Führen Sie die startup.bat Datei im Tomcat bin Verzeichnis aus.
5. Starten Sie die gewünschten Client-Batch-Dateien (MiraculixSachbearbeiterClient.bat, MiraculixWSKundeClient.bat oder MiraculixWSServicetechnikerClient.bat) und melden Sie sich mit Ihrem entsprechenden Login an.



14.3 Zeitplan

- ▶ *siehe separates Dokument „Zeitplan“*

14.4 Arbeitszeiten

- ▶ *siehe separates Dokument „Arbeitszeiten“*

14.5 Sitzungsprotokolle

- ▶ *siehe separates Dokument „Sitzungsprotokolle“*

14.6 Pflichtenheft

- ▶ *siehe separates Dokument „Pflichtenheft“*

14.7 Namenskonventionen

- ▶ *siehe separates Dokument „Namenskonventionen“*

14.8 Klassendiagramm

- ▶ *siehe separates Dokument „Klassendiagramm“*

14.9 ERD Diagramm

- ▶ *siehe separates Dokument „ERD Diagramm“*

14.10 Source Code

- ▶ *siehe separaten Ordner „Source Code Miraculix – Gruppe B“*

