

Projekt Feukora 2016

Detaillierte Projektdokumentation angelehnt an Arc42

Projekt Feukora 2016

Detaillierte Projektdokumentation angelehnt an Arc42

HSLU – Hochschule Luzern Wirtschaft

Bachelor Wirtschaftsinformatik

27. Mai 2016

Autoren:

Lerngruppe 171 - Projektgruppe B

Alexandra Lengen, alexandra.lengen@stud.hslu.ch

Dominik Stirnimann, dominik.stirimann@stud.hslu.ch

Luca Raneri, luca.raneri@stud.hslu.ch

Matthias Perrollaz, matthias.perrollaz@stud.hslu.ch

Olivia Wassmer, olivia.wassmer@stud.hslu.ch

Pascal Steiner, pascal.steiner@stud.hslu.ch

Betreuende Dozenten:

Erwin Mathis, erwin.mathis@hslu.ch

Jordan Sucur, jordan.sucur@hslu.ch

Inhaltsverzeichnis

1	Einführung und Ziele	8
1.1	Aufgabenstellung.....	8
1.2	Qualitätsziele	8
1.3	Stakeholder.....	9
2	Randbedingungen	12
2.1	Technische Randbedingungen.....	12
2.2	Organisatorische Randbedingungen	13
2.3	Konventionen	14
3	Kontextabgrenzung	14
3.1	Fachlicher Kontext	14
3.2	Technischer oder Verteilungskontext.....	15
3.3	Schnittstellen	15
4	Lösungsstrategie.....	15
5	Bausteinsicht	16
5.1	Ebene 0.....	17
5.1.1	System Feukora (Blackbox)	17
5.2	Ebene 1	17
5.2.1	Applikationsserver (Blackbox).....	17
5.2.2	Datenbankserver (Blackbox)	18
5.2.3	Webservice (Blackbox)	19
5.2.4	Extranet Client (Blackbox)	20
5.2.5	Intranet Client (Blackbox).....	20
5.3	Ebene 2	21
5.3.1	Businesslogik-Komponente (Blackbox)	21
5.3.2	RMI-Komponente (Blackbox)	22
5.3.3	Modell-Komponente (Blackbox)	23
5.3.4	Persistence-Komponente (Blackbox)	23
6	Laufzeitsicht.....	24
6.1	Laufzeitszenario 1: Messwerte eingeben	24
6.2	Laufzeitszenario 2: Auftrag erstellen	25
6.3	Laufzeitszenario 3: Terminübersicht drucken	26
7	Verteilungssicht.....	27
7.1	Infrastruktur Ebene 1.....	27
7.1.1	Verteilungsdiagramm.....	27
7.1.2	Verteilungsartefakte	27
7.1.3	Spezifikation der Knoten	28

7.2	Fachliche Strukturen und Modelle	29
7.3	Typische Muster und Strukturen	30
8	Technische Konzepte.....	30
8.1	Persistenz.....	30
8.2	Benutzungsoberfläche.....	30
8.2.1	Feuerungskontrolleur (Client Extern).....	31
8.2.2	Backoffice (Client Intern).....	31
8.3	Ergonomie	31
8.4	Ablauf- Dialog und Workflowsteuerung.....	31
8.4.1	Flussdiagramm Client Intern	32
8.4.2	Flussdiagramm Client Extern.....	33
8.5	Geschäftsregeln	34
8.6	Plausibilisierung und Validierung	34
8.7	Transaktionsbehandlung	34
8.8	Sessionsbehandlung und Zustandsbehandlung.....	34
8.9	Caching	34
8.10	Sicherheit.....	35
8.11	Kommunikation und Integration mit anderen IT Systemen	35
8.12	Verteilung	35
8.13	Ausnahme- und Fehlerbehandlung	35
8.14	Systemmanagement und Administrierbarkeit	35
8.15	Logging, Protokollierung und Tracing	36
8.16	Konfigurierbarkeit.....	36
8.17	Parallelisierung und Threading.....	36
8.18	Internationalisierung	36
8.19	Skalierung	36
8.20	Hochverfügbarkeit & Desaster-Recovery	36
8.21	Migration	36
8.22	Testbarkeit.....	36
8.23	Codegenerierung	36
8.24	Build-Management	37
9	Entwurfsentscheidungen.....	37
10	Qualitätsszenarien.....	39
10.1	Qualitätsbaum	39
10.2	Bewertungsszenarien	39
10.2.1	Erweiterungsmöglichkeiten:.....	39
10.2.2	Wartbarkeit:.....	39
10.2.3	Testbarkeit:	39
10.2.4	Analysierbarkeit:	39

10.2.5	Wartezeit/Effizienz:	40
10.2.6	Verfügbarkeit:	40
10.2.7	Aktualität:	40
10.2.8	Datenschutz (Sicherheit):.....	40
10.2.9	Übersichtlichkeit/Bedienbarkeit:.....	40
11	Risiken	40
12	Glossar.....	41
13	Arc42 Dokumentationen und darüber hinaus.....	41
13.1	Vorschläge und Anregungen für die Applikation Feukora.....	41
13.2	Arc42 Dokumentation	42
14	Requirements Dokumentation.....	43
15	UseCase Dokumentation.....	44
15.1	Login	44
15.1.1	Benutzernamen eingeben.....	44
15.1.2	Login.....	45
15.1.3	Passwort eingeben.....	45
15.2	Liegenschafts- und Eigentümerverwaltung.....	46
15.2.1	Liegenschaften und Eigentümer anzeigen	46
	Liegenschaften und Eigentümer verwalten.....	46
15.2.2	Neue Liegenschaft/Eigentümer erfassen.....	47
15.2.3	Bestehende Liegenschaften und Eigentümer bearbeiten	47
15.3	Terminverwaltung	49
15.3.1	Termin hinzufügen.....	49
15.3.2	Termin löschen.....	49
15.3.3	Termine bearbeiten	50
15.3.4	Terminübersicht starten	50
15.3.5	Terminübersicht drucken.....	51
15.3.6	Freies Terminfenster suchen	51
15.4	Messdatenverwaltung	52
15.4.1	Messdaten anzeigen	52
15.4.2	Neue Messdaten erfassen	53
15.4.3	Bestehende Messdaten bearbeiten.....	53
15.4.4	Bestehende Messdaten verwalten	54
15.5	Wärmeerzeugerverwaltung.....	54
15.5.1	Neuen Wärmeerzeuger erfassen	55
15.5.2	Wärmeerzeuger anzeigen.....	55
15.5.3	Wärmeerzeuger verwalten	56
15.5.4	Bestehende Wärmeerzeuger bearbeiten	56
15.6	Brennerverwaltung.....	58

15.6.1	Brenner anzeigen	58
15.6.2	Brenner verwalten	58
15.6.3	Neuen Brenner erfassen	59
15.6.4	Bestehende Brenner bearbeiten.....	59
7.	Rapportdatenverwaltung.....	60
15.6.5	Neuer Feuerungsrapport erstellen und öffnen.....	60
15.6.6	Feuerungsrapport bearbeiten.....	61
15.6.7	Feuerungsrapport speichern.....	61
16	Klassendiagramme	63
17	Deployment Informationen.....	64
17.1	Deployment Dateien.....	64
17.2	Installationsanleitung	64
17.2.1	Installation Datenbank «Feukora».....	64
17.2.2	Installation und Start Serverapplikation	66
17.2.3	Installation und Start Webservice.....	66
17.2.4	Client Intern starten.....	66
17.2.5	Client Extern starten	66
18	TDD und JUnit.....	67
18.1	Erfahrung mit TDD	67
18.2	JUnit for the Win.....	67
18.3	Testklassen	68
18.3.1	Persister	68
18.3.2	Business-Logik Layer	68
18.3.3	Web-Service	68
18.4	Durchführung der JUnit Tests	69
19	Funktionale Tests	69
19.1	Testumgebung	69
19.2	Testzeitpunkt	70
19.3	Testergebnis	70
19.4	Testbeschreibungen	70
20	Datenbank Dokumentation.....	72
20.1	ERD/Attributsliste	72
20.2	Datenbank-Server Informationen.....	73
20.3	Test-Datenbank	73
21	Projektmitglieder Beiträge	74
21.1	Alexandra Lengen	74
21.1.1	Individuelles Management Summary zum Projekt Feukora	74
21.1.2	Individuelles Portfolio zum Projekt Feukora	75
21.2	Dominik Stirnimann	78

21.2.1	Individuelles Management Summary zum Projekt Feukora Dominik Stirnimann	78
21.2.2	Individuelles Portfolio zum Projekt Feukora Dominik Stirnimann	78
21.3	Luca Raneri	83
21.3.1	Individuelles Management Summary zum Projekt Feukora Alexandra Lengen	83
21.3.2	Individuelles Portfolio zum Projekt Feukora Luca Raneri	84
21.4	Matthias Perrollaz	88
21.4.1	Individuelles Management Summary zum Projekt Feukora	88
21.4.2	Individuelles Portfolio zum Projekt Feukora	89
21.5	Olivia Wassmer	93
21.5.1	Individuelles Management Summary zum Projekt Feukora	93
21.5.2	Individuelles Portfolio zum Projekt Feukora	94
21.6	Pascal Steiner	100
21.6.1	Individuelles Management Summary zum Projekt Feukora	100
21.6.2	Individuelles Portfolio zum Projekt Feukora	100
22	Source-Code von selber beschriebenem Code	103
23	Anhang	104
23.1	Dokumentation Änderungen und Autoren	104
23.2	Anwesenheitsliste	104
23.3	Genaue Aufgaben-Zeitverteilung im Projekt	105
23.4	Anhang Projektleitung	107
23.5	Anhang Entwicklung	120
23.6	Anhang GUI	124
23.7	Projekt Feukora Source Code	128
1.0	Modelklassen	128
2.0	Persister	128
3.0	Businesslogik	128
4.0	RMI	128
5.0	Client intern	128
6.0	Webservice	128
7.0	Client Extern	128

Abbildungsverzeichnis

Abbildung 1 - Fachlicher Kontext Feukora	15
Abbildung 2 - Bausteinsicht Feukora Gesamtbild	16
Abbildung 3 - Bausteinsicht Feukora Vertiefungsebene 0	17
Abbildung 4 - Bausteinsicht Feukora Vertiefungsebene 1	17
Abbildung 5 - Bausteinsicht Feukora Vertiefungsebene 2	21
Abbildung 6 - Laufzeitszenario 1 Feukora Messwerte eingeben	24
Abbildung 7 - Laufzeitszenario 2 Feukora Auftrag erstellen	25
Abbildung 8 - Laufzeitszenario 3 Feukora Terminübersicht drucken	26

Abbildung 9 - Verteilungssicht Feukora	27
Abbildung 10 - Fachliche Strukturen Feukora	29
Abbildung 11 - Typische Muster und Strukturen Feukora	30
Abbildung 12 - Persistenz Prozess Aufbau	30
Abbildung 13 - Workflow Client Intern	32
Abbildung 14 - Workflow Client Extern.....	33
Abbildung 15 - Grundlagen Entity Manager.....	34
Abbildung 16 – Exception-Handling	35
Abbildung 17 - Qualitätsbaum Feukora	39
Abbildung 18 - UseCase Login	44
Abbildung 19 - UseCase Liegenschafts- und Eigentümerverwaltung.....	46
Abbildung 20 - UseCase Terminverwaltung.....	49
Abbildung 21 - UseCase Messdatenverwaltung.....	52
Abbildung 22 - UseCase Wärmeerzeugerverwaltung	54
Abbildung 23 - UseCase Brennerverwaltung	58
Abbildung 24 - Rapportdatenverwaltung.....	60
Abbildung 25 - Klassendiagramm Feukora.....	63
Abbildung 26 - Neue Login Rolle 1	65
Abbildung 27 - Neue Login Rolle 2	65
Abbildung 28 - Backup erstellen	65
Abbildung 29 - Ausführen eines JUnit Test	69
Abbildung 30 - ERD Feukora.....	72

1 Einführung und Ziele

Diese Dokumentation liefert den strategischen, wie auch technischen Überblick über die Architektur des Projektes Feukora. Mithilfe der hier niedergeschriebenen Ausführungen sollen die Entwurfsentscheide des Projektteams nachvollzogen werden können und der Einstieg in die Anwendung der Software erleichtert werden.

1.1 Aufgabenstellung

Hauptziel ist es die momentan manuelle Übermittlung der Feuerungskontrolle-Daten an die Geschäftsstelle Feuerungskontrolle durch eine Software zu ersetzen, welche ebenfalls mit einer Terminfunktion ergänzt wird.

Feuerungskontrolleure welche im Auftrag ihrer Heizfirmen, Heizanlagen bei Kunden überprüfen müssen, sollen künftig die abgelesenen Daten für ihren Feuerungskontrolle-Rapport direkt mit dem Laptop an die Geschäftsstelle GFK übermitteln können. Dabei sollen sämtliche Daten ebenfalls in einer Datenbank gesichert werden.

Zusätzlich zu dieser digitalisierten Übermittlung soll ein Terminkalender Tool entstehen, auf welchem das Backoffice die Termine für die einzelnen Feuerungskontrolleure eingeben kann. Außerdem sollen auch die einzelnen Kontrolleure für sich selber Termine eingeben können.

Die ausführliche Aufgabenstellung kann aus der Übersicht im [Anhang](#) Dokument „INM21_Projektaufgabe_FS16_V_1.7“ entnommen werden. Sowie die Anforderungen in [Kapitel 15 \(Requirements Dokumentation\)](#).

Das System soll die Daten aus dem Feuerungskontrolle-Rapport speichern und an die GFK übermitteln. Dadurch können Fehler durch unleserlich ausgefüllte Rapporte vermieden werden. Ebenfalls fällt durch die elektronische Übermittlung die Wartezeit der Postübermittlung weg.

Durch das integrierte Terminkalender-Tool kann das Backoffice die Termine einheitlich verwalten. Sie werden direkt aktualisiert, dadurch können Fehlerquellen wie das falsche Eintragen der Termine vermieden werden. Ebenfalls werden keine falschen Termine eingetragen, da der Kalender immer up-to-date ist. So kann der Kontrolleur sich eigene Termine selbstständig blockieren.

Die Feuerungskontrolleure erhalten somit eine erleichterte Eingabe. Ihre Termine können Sie zusammen mit dem Backoffice effizient und zeitsparend verwalten.

1.2 Qualitätsziele

Die folgende Auswahl der wichtigsten Qualitätsziele zeigt den Fokus der Architektur auf. Selbstverständlich bauen diese Ziele auf dem Ziel der vollständigen Erfüllung der Anforderungen auf. Diese ist für das Projekt wie auch das Projektteam essentiell.

Priorität	Qualitätsziel	Erläuterung
1	VERFÜGBARKEIT & AKTUALITÄT	24/7 Zugriff und immer „up to date“ Die Software sollte stets für Heizfirmen sowie die GFK verfügbar sein, um keinen Unterbruch in der Übertragung der erfassten Kontrolldaten zu erhalten.
2	BENUTZERFREUNDLICHKEIT	Eingabefreundlichkeit für Heizungskontrollen & übersichtliche Terminverwaltung Kontrolleure sollen Daten einfach eingeben, anpassen und übertragen können.
3	PERFORMANCE	Effizientes Arbeiten ermöglichen Die Software sollte stets ohne Wartezeit bedienbar sein.
4	SICHERHEIT	Langfristiger Datenschutz in Zeiten von BigData Daten (Kundendaten sowie auch Geschäftsdaten) sollen sicher gegen Außen sein

Die Qualitätsziele werden während des gesamten Projekts stets berücksichtigt, des Weiteren werden sie zur Kontrolle der Ergebnisse weiter in den Qualitätsszenarien bewertet. Diese finden sich im Abschnitt [10. Qualitätsmerkmale](#).

1.3 Stakeholder

Am Projekt Feukora sind viele Stakeholder beteiligt. Folgende Stakeholder Gruppen kommen in Kontakt mit der Software:

- Seitens Auftragsgeber, die Geschäftsstelle für Feuerungskontrollen
- Als Endnutzer – Angestellte der Heizfirmen, welche die Software bei der GFK beziehen
- Sowie Kundes-Kunden, also die Besitzer der Heizungen welche kontrolliert werden müssen
- Zu guter Letzt das Projektteam, und somit jedes Projektmitglied welches seinen Beitrag an die Entwicklung der Software leistet

Jeder Stakeholder hat andere Intentionen die Software zu benutzen, folgende Tabelle verschafft einen detaillierteren Überblick:

ROLLE / BEREICH	STAKEHOLDER-GRUPPE	BESCHREIBUNG	ZIEL / INTENTION	KONTAKT
GESÄFTSSTELLE FÜR FEUERUNGS-KONTROLLEN (GFK)	Auftraggeber	Verfügen über technische und fachliche Kenntnisse um das Projekt vollumfänglich bewerten zu können.	Möchten eine möglichst gute Lösung für die Automatisierung der Datenübermittlung an die Feuerungskontrolle-Dienststelle	• E. Mathis • J. Sucur
FEUERUNGS-KONTROLLEURE	Endnutzer	Sind vor Ort beim Kunden um Daten für die Kontrolle abzulesen und diese zu übermitteln. Machen meist direkt auch einen Service neben der Feuerungskontrolle. Ihre Aufträge werden grundsätzlich über das Backoffice verwaltet. Sie tragen sich jedoch auch selber Termine ein.	Möchten eine erleichterte Übermittlung der Daten anstatt via Post und Papier, direkt mit dem Laptop. Möchten ebenfalls eigene Termine in Kalender eintragen können.	• Keiner
BACK-OFFICE SACHBEARBEITER	Endnutzer	Verwaltet alle Aufträge/Termine der Feuerungskontrolleure und ist Ansprechperson für Kunden.	Will einfacher Termine verwalten können und diese direkt aktualisieren können. Will zudem direkt Zugriff auf alle Feuerungsrapport-Daten haben.	• Keiner
HEIZUNGS-BESITZER	Kundes-Kunden	Werden von Heizungsfirmen kontrolliert. Dabei werden Feuerungskontrollen sowie Service gemacht. Haben nicht zwingend Fachwissen über ihre Infrastruktur.	Möchten möglichst kurz gestört werden und nicht viel Arbeit mit Service oder Kontrolle haben	• Keiner
PROJEKTLINTER	Projektteam	Überwacht die einzelnen Bereiche des Projektes, koordiniert Besprechungen und den Verlauf des Projektes. Ist für Entscheidungen verantwortlich wenn keine Einigung seitens des Teams gefunden werden konnte.	Hat grosses Interesse an guter Kommunikation innerhalb des Teams und effizienter Arbeitsweise. Möchte die Zeitplanung einhalten und alle Anforderungen des Auftraggebers mithilfe des Teams erfüllen.	• Dominik Stirnimann
DEPLOYMENT	Projektteam	Führt die Applikation beim Kunden ein und stellt sämtliche Hilfestellungen dafür zur Verfügung	Ist auf eine zeitgerechte Fertigstellung der Applikation angewiesen, damit alles getestet und für den Release vorbereitet werden kann. Möchte die Einführung beim Kunden ohne Schwierigkeiten vollziehen können.	• Olivia Wassmer • Pascal Steiner
SOFTWARE-ARCHITEKTEN	Projektteam	Verfügen über strategische sowie technische Kenntnisse um die Architektur der Applikation zu entwerfen und überwachen	Hauptziel ist es eine einwandfreie Planung des Projektes zu erbringen welche den Verlauf der Entwicklung vereinfacht und übersichtlich macht.	• Olivia Wassmer • Pascal Steiner

DATENMANAGEMENT	Projektteam	Entwerfen Struktur der Datenbank sowie die Verbindung zu den benutzten Schnittstellen. Sind ebenfalls für die Sicherheit der Daten verantwortlich.	Beschäftigen sich ausführlich mit Anforderungen und der Planung bevor die Entwicklung gestartet wird. Möchten Gefäss für die umfängliche Erfassung der Kontrolldaten bieten und diese absichern. Sind auf gute Kommunikation mit Schnittstellen angewiesen.	• Matthias Perrollaz • Luca Raneri • Pascal Steiner
USER INTERFACES	Projektteam	Entwerfen das Grafical User Interfaces unter dem Ziel der Benutzerfreundlichkeit	Entwerfen die Oberfläche welche der Kunde schlussendlich benutzt, haben das Ziel dies so benutzerfreundlich und ansprechend wie möglich zu gestalten.	• Alexandra Lengen • Pascal Steiner
REQUIREMENTS ENGINEERING	Projektteam	Für die vollumfängliche und klare Einholung der Anforderungen verantwortlich.	Haben das Ziel alle Belange der Anforderungen des Kunden aufzunehmen um den Softwarearchitekten eine klare und vollständige Grundlage zur Architektur zu liefern.	• Alexandra Lengen • Dominik Stirnimann
DOKUMENTATION	Projektteam	Erstellt Dokumentation der Projektarbeit um Verständliche Übersicht über die Architektur und alle weiteren Belange des Projekts abzudecken.	Möchte eine vollständige Dokumentierung aller „Abteilungen“ um eine vollständige Dokumentation erstellen zu können.	• Olivia Wassmer

2 Randbedingungen

Das Projektteam hat den Auftrag - hinsichtlich der technischen Randbedingungen - eine verteilte Applikation zu gestalten. Dies bringt den Vorteil, dass die verschiedenen Prozesse so stets parallel laufen und die gewünschte Aktualität der Daten erreicht werden kann. Weitere Randbedingungen werden im folgenden Abschnitt behandelt.

2.1 Technische Randbedingungen

Die folgenden Randbedingungen sind von den Requirements abgeleitet. Sie legen die Basis auf welcher das Projekt aufgebaut wird.

RANDBEDINGUNG	BESCHREIBUNG
HARDWARE	PC / Notebooks Es werden Computer (Back Office) und Laptops (flexibel für Feuerungskontrolleure) verwendet.
SOFTWARE	Windows 7 Die Computer sollten das Betriebssystem Windows mind. Version Windows7 verwenden. Eclipse Für die Entwicklung der Applikation wird Eclipse benutzt.
ENTWICKLUNGSUMGEBUNG	
TDD	JUnit 4 Für das „Test Driven Developement“ werden JUnit Test der Version 4 benutzt.
PROGRAMMIERSPRACHE	Java, SQL Die Applikation wird nach Vorgabe in der Programmiersprache Java geschrieben. Die Schnittstelle zur Datenbank wird mit SQL ergänzt.
VERSIONSMANAGEMENT SYSTEM	Github Zur Verwaltung der Versionen der Applikationsbestandteile verwendet das Team Github
APPLIKATIONSSERVER	Apache Tomcat 8 Die Applikation soll auf Apache Tomcat 8 lauffähig sein. Diese Software wird vom Auftraggeber verwendet.
DATENBANK MANAGEMENT SYSTEM	PostgreSQL Zur Verwaltung wird PostgreSQL benutzt. Die Datenbank ist somit JPA-konform und unterstützt SQL. Bedingung ist, dass die DB unabhängig vom restlichen System und als Teil einer verteilten Applikation ist. JPA Zur Realisierung der Schnittstelle zwischen Datenbank und Java wird Java Persistence API benutzt. RMI Als Middleware wird die „Remote Method Invocation“ benutzt zur Kommunikation zwischen Intranet Client und Server.
SCHNITTSTELLEN	SOAP Webservice Der Webservice SOAP wird zur Kommunikation zwischen Extranet Client und Server benutzt.

LOGSYSTEM	Log4j Benutzt wird das Logsystem Log4j welches mit seinem Preis-Leistungs-Verhältnis überzeugt
GUI	JavaFX Scene Builder Die Grafische Benutzeroberfläche wird mit JavaFX erstellt.
SYSTEMBETRIEB	Zu Arbeitszeiten Die Verfügbarkeit der Applikation wird während der Arbeitszeiten der Unternehmen, sprich zwischen 7 – 18 Uhr, gewährleistet.
VERFÜGBARKEIT LAUFZEITUMGEBUNG	Durchgehend Der Server der Applikation Feukora ist durchgehend erreichbar. Jedoch können ausserhalb des Systembetriebs Updates vorgenommen werden, welche für kurze Unterbrechungen sorgen können.

2.2 Organisatorische Randbedingungen

Für die Organisation des Projekts liegen folgende Randbedingungen vor:

RANDBEDINGUNG	BESCHREIBUNG
TEAMKONSTELLATION	Das Projektteam besteht aus 6 Mitgliedern: Alexandra Lengen, Dominik Stirnimann, Luca Raneri, Matthias Perrollaz, Olivia Wassmer und Pascal Steiner
KNOW-HOW	Alle Mitglieder sind auf dem gleichen Wissensstand in der Wirtschaftsinformatik und besuchen das 2. Semester im Bachelor für Wirtschaftsinformatik an der Hochschule Luzern. Neben dem Studium sind keine Vorkenntnisse in der Wirtschaftsinformatik vorhanden.
ARBEITSAUFTEILUNG	Die Aufteilung der einzelnen Bereiche wurde nach Wissen und Interesse gemacht, vor allem jedoch anhand letzterem. Das Team hat sich die Bedingung von möglichst gleichmässiger Arbeitseinteilung gesetzt.
VORGEHEN	Es wird nach der pragmatischen Vorlage arc42 vorgegangen. Ergänzt durch die Hinweise aus Modul INM22 an der HSLU.
ABGABETERMIN	Für das Projekt steht die Unterrichtszeit des 2. Semesters zur Verfügung -genauer die Zeit zwischen 22. Februar und 26. Mai 2016. Der Abgabetermin ist der 27. Mai 2016 um 12:00 Uhr. Zu diesem Zeitpunkt muss die Applikation auf der HSLU Infrastruktur (Übernimmt Rolle des Auftraggebers) lauffähig installiert sein.
PROJEKTPRÄSENTATION	Das Projekt wird am 30. Mai 2016 dem Auftraggeber präsentiert (20-25min) und allfällige Fragen werden beantwortet.

2.3 Konventionen

Alle Dokumente werden im Cloudservice „Dropbox“ gespeichert. Zusammengefügt werden schlussendlich alle Dokumente wie auch Codestücke in einer einheitlichen Dokumentation. Diese wird anhand von arc42 erstellt. Für die einzelnen Bausteine sollen folgende Konventionen beachtet werden:

KONVENTION	BESCHREIBUNG
DATEIEN	Sämtliche Dateien werden mit dem Dokumentnamen, Projektnamen abgespeichert.
CODE	Code wird über GitHub gespeichert und stets aktuell gehalten. https://github.com/HSLU-WI-Feukora2016-GruppeB/Feukora_HSLU_2016_GruppeB
JAVADOC	Der gesamte Code wird jeweils direkt bei der Erstellung vom Autor verständlich und in deutscher Sprache dokumentiert.
DIAGRAMME	Alle Diagramme welche die Architekturlogik grafisch unterstützen, werden mit der Software „Enterprise Architect“ gezeichnet.

3 Kontextabgrenzung

3.1 Fachlicher Kontext

Die Applikation des Projekt Feukora steht im Mittelpunkt von drei Haupt-Akteuren; den Heizungskontrolleuren, den Sachbearbeitern aus dem Back-Office sowie der Geschäftsstelle für Feuerungskontrolle.

Die Applikation kann im Rahmen der Kontextabgrenzung auf folgende Kernprozesse heruntergebrochen werden:

- Die Eingabe von Kontrolldaten, sowie die Eingabe und das Auslesen von Terminen durch den Heizungskontrolleur.
- Die Eingabe von Stammdaten, sowie die Eingabe und das Auslesen von Terminen durch den Sachbearbeiter oder die Sachbearbeiterin im Back-Office.
- Die Weiterleitung des elektronischen Feuerungskontroll-Rapports an die Geschäftsstelle für Feuerungskontrolle (GFK).

Im folgenden Diagramm kann der fachliche Kontext auf einen Blick entnommen werden. Weitere Ausführungen zum Business Kontext sind unter anderem in [Kapitel 8.1](#) sowie in den detaillierten Use Cases in [Kapitel 15](#) verfügbar.

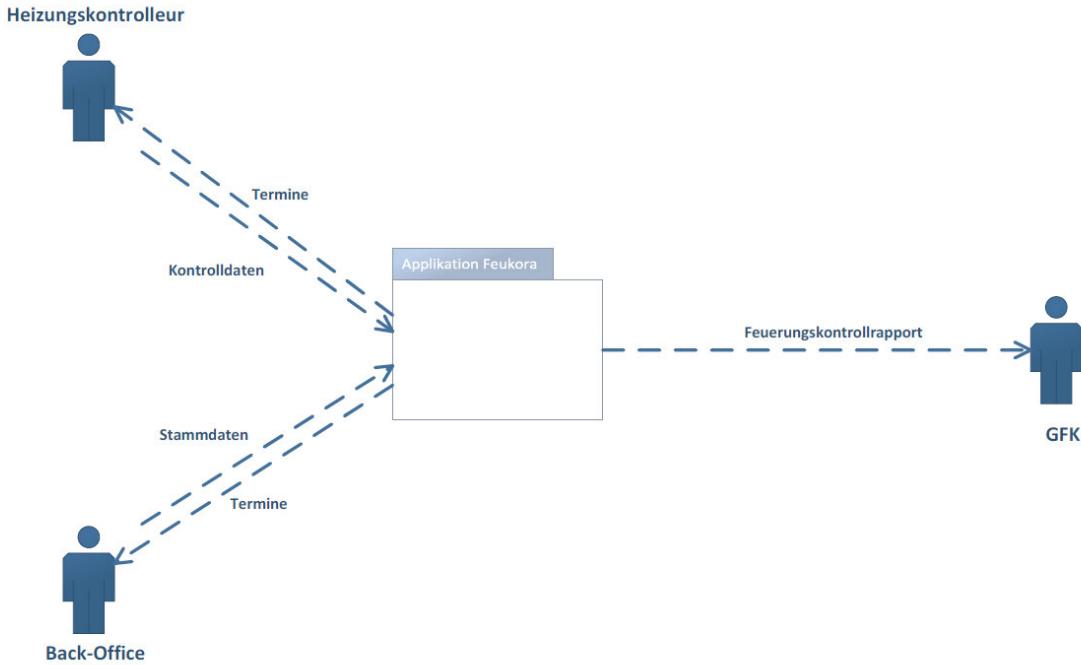


Abbildung 1 - Fachlicher Kontext Feukora

3.2 Technischer oder Verteilungskontext

Die Applikation besteht aus einem einzelnen System an Komponenten, welches auf der Schicht der technischen Kontextabgrenzung keine weiteren Details liefert.

Details zum Aufbau der Applikation sind in der [Bausteinsicht \(Kapitel 5\)](#) und [Laufzeitsicht \(Kapitel 6\)](#) definiert, Informationen zum Verteilungskontext in [Verteilungssicht \(Kapitel 7\)](#).

3.3 Schnittstellen

Das System hat in sich keine externen Schnittstellen. Es benötigt einzig eine Verbindung zum Internet wenn die Applikation wie geplant, auf verschiedene Geräte, verteilt wird.

Der Ausdruck der gewünschten Terminübersicht wird auf dem Computer direkt lokal gespeichert und hat somit auch keine weiteren Schnittstellen.

Mit Ausblick in die Zukunft möchten wir hier die externe Schnittstelle zur Geschäftsstelle für Feuerungskontrollen erwähnen. In einem nächsten Schritt könnte die Applikation sehr gut ergänzt werden durch eine Anbindung an den zentralen Server der GFK. Die Feuerungsrapporte müssten so nicht weiter gedruckt und übermittelt werden sondern könnten direkt auf den Server gepusht werden via einer weiteren Webservice Schnittstelle.

4 Lösungsstrategie

Die Applikation Feukora wird wie gewünscht als verteilte Applikation entwickelt. Durch die klare Aufgabenstellung ist der Spielraum für die Software Architektur relativ gering. Darum wird das System so simpel und benutzerfreundlich. Sowie nach den Requirements der Geschäftsstelle für Feuerungskontrolle realisiert.

Dabei greifen die Heizungskontrolleure im Aussendienst von Ihrem Laptop via Webservice auf die Applikation zu. Die Sachbearbeiter im Back-Office nutzen die Applikation, vom Computer im Intranet, welcher mit RMI

verbunden ist. Beide Parteien speisen Teile an Stammdaten sowie Termindaten ins System ein. Diese Daten werden via JPA in die Datenbank (PostgreSQL) gespeichert.

Durch eine Authentisierung der User bei der Anmeldung ins System, werden die verschiedenen Zugriffsrechte realisiert und kontrolliert. Des Weiteren gibt es einen Superuser (der Systemadministrator), welcher die Applikation unterhält.

5 Bausteinsicht

Die folgende Bausteinsicht liefert einen genaueren Blick auf die einzelnen Komponenten der Applikation Feukora, und wo sie sich befinden. Es werden unter anderem ihr Zweck, ihre Schnittstellen, Abhängigkeiten und weitere Informationen erläutert. Die erfüllten Anforderungen, welche sich auf die einzelnen Bausteine beziehen sind hier erwähnt und detailliert in der [Requirements Dokumentation \(Kapitel 14\)](#) nachzulesen.

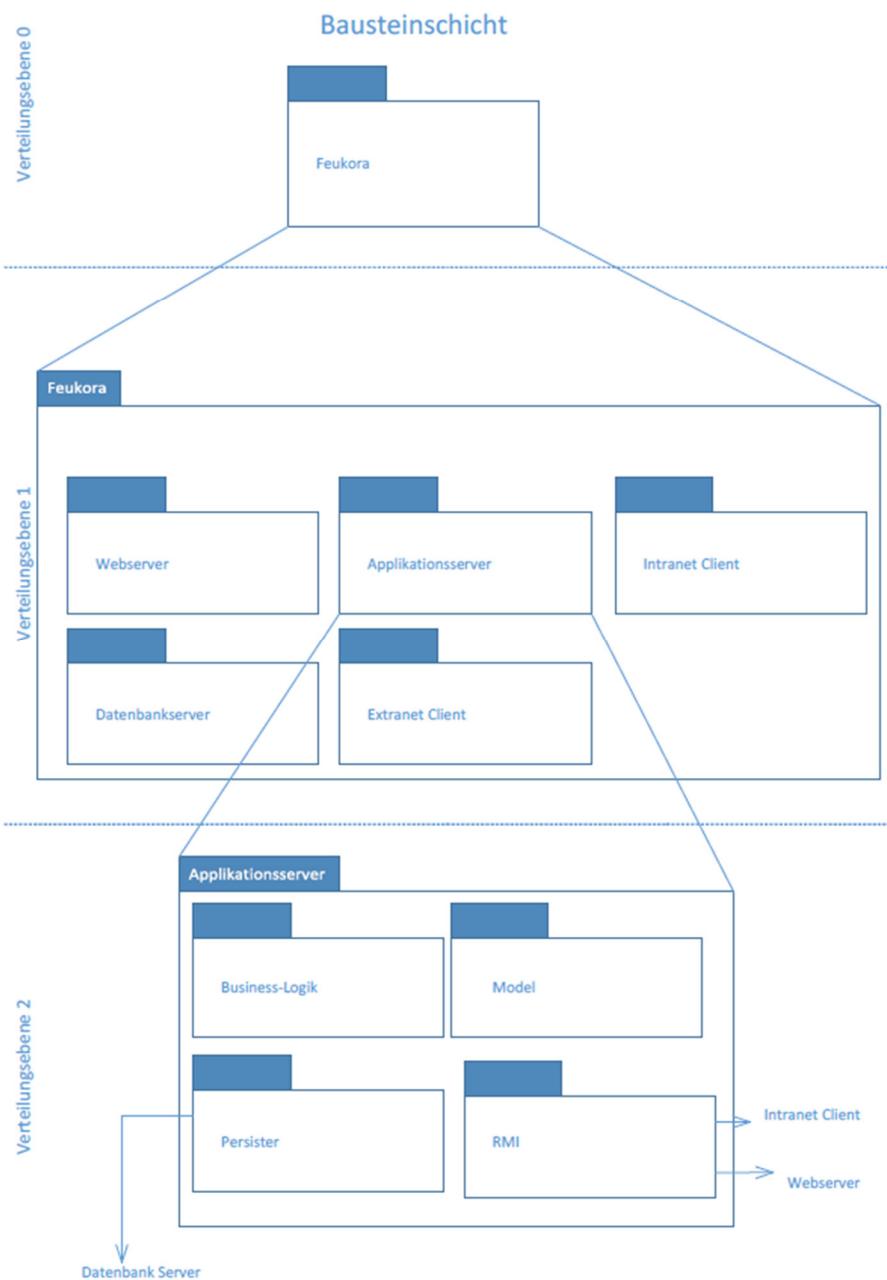


Abbildung 2 - Bausteinsicht Feukora Gesamtbild

5.1 Ebene 0

5.1.1 System Feukora (Blackbox)

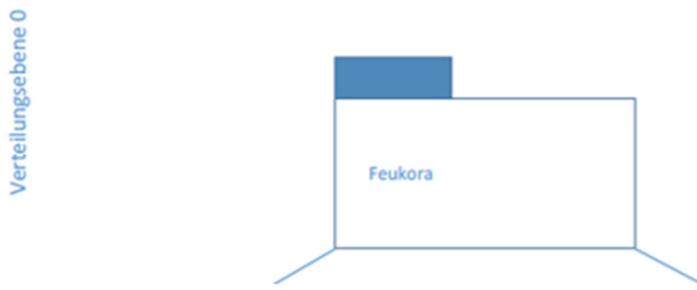


Abbildung 3 - Bausteinsicht Feukora Vertiefungsebene 0

NAME	FEUKORA
ZWECK / VERANTWORTLICHKEIT	Dieser Baustein stellt die gesamte Applikation dar. In der Kontextabgrenzung (Kapitel 3) ist die Blackbox näher erläutert.

5.2 Ebene 1

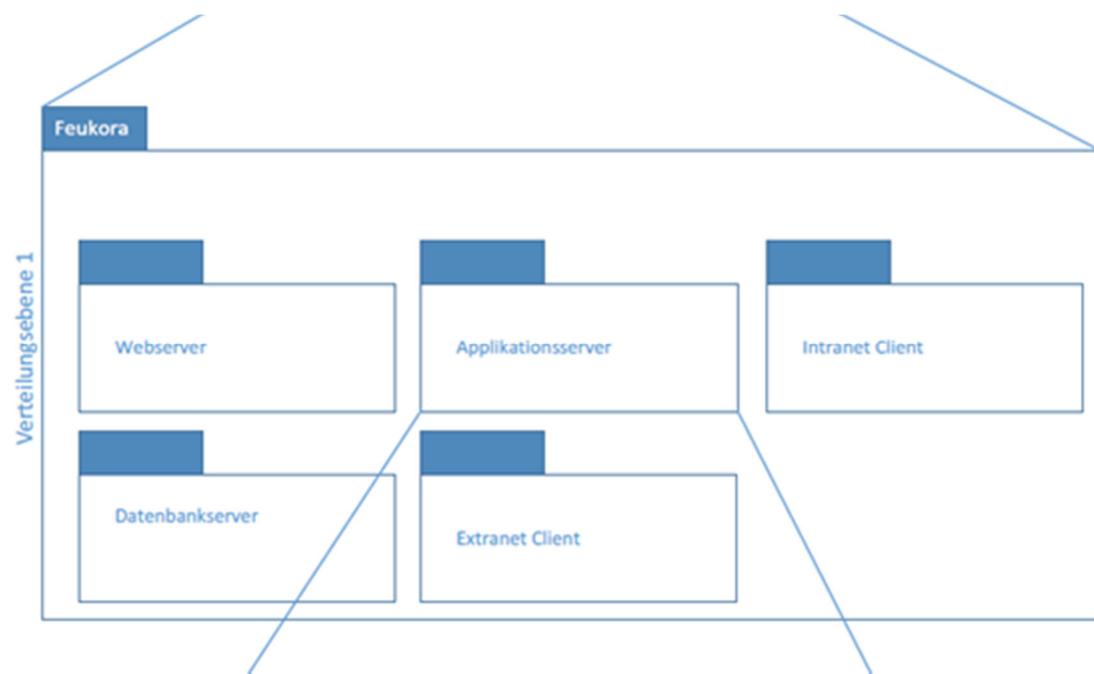


Abbildung 4 - Bausteinsicht Feukora Vertiefungsebene 1

5.2.1 Applikationsserver (Blackbox)

NAME	APPLIKATIONSSERVER
ZWECK / VERANTWORTLICHKEIT	Dieser Baustein ist die Hauptkomponente der Applikation. Sie beinhaltet die zentrale Businesslogik und die Modell-Komponenten. Zudem verantwortet sie die Kommunikation zu Schnittstellen des Systems.

SCHNITTSTELLE 1	<p>Intranet Client (via RMI)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Neue Stammdaten (neue Kunden, neue Wärmeerzeuger/Brenner) - Termine (sämtliche Termine für Feuerungskotrolleure) <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersicht (Wochenübersicht aller Termine)
SCHNITTSTELLE 2	<p>Webservice (via SOAP)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Kontrolldaten - Stammdaten (Wärmeerzeuger, Brenner) - Termine <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersicht (Persönliche Terminübersicht)
SCHNITTSTELLE 3	<p>Datenbankserver (via EclipseLink, JPA)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Status Speicherung <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (zur Speicherung) <p>Dieser Baustein ist abhängig von der Laufzeitumgebung der Serverinfrastruktur der HSLU.</p>
ABHÄNGIGKEITEN	Run-feukora-gruppeB-applikationsserver-bat (rmi) Run-feukora-gruppeB-init.bat Lib Folder
CODE-ARTEFAKTE	Req014
ERFÜLLTE ANFORDERUNGEN	Dieser Baustein verändert sich vorerst nicht. Ausser dies wird infolge einer Funktionserweiterung gewünscht.
VARIABILITÄT	
WEITERE INFORMATIONEN	Verschiedene Autoren, koordiniert durch Olivia Wassmer. Die Komponente wurde bei der Erarbeitung des Projektes erstellt und befindet sich noch auf Versionsstufe 1. Weitere Versionen werden sich durch Erweiterungen oder Optimierungen ergeben.

5.2.2 Datenbankserver (Blackbox)

NAME	DATENBANKSERVER
ZWECK / VERANTWORTLICHKEIT	Dieser Baustein stellt die Speicherung der eingegebenen Stammdaten und Kontrolldaten sicher. Diese werden vom Applikationsserver via einer Schnittstelle an die Datenbank weitergeleitet.
SCHNITTSTELLE 1	<p>Applikationsserver (via EclipseLink, JPA)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (zur Speicherung) <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Status Speicherung

ABHÄNGIGKEITEN	Dieser Baustein ist abhängig vom Datenbankserver PostgreSQL sowie der Laufzeitumgebung der Serverinfrastruktur der HSLU.
CODE-ARTEFAKTE	keine
ERFÜLLTE ANFORDERUNGEN	Req01, Req02, Req03, Req04, Req05, Req06, Req07, Req10
VARIABILITÄT	Dieser Baustein verändert sich vorerst nicht.
WEITERE INFORMATIONEN	Autor des DB Scripts ist Matthias Perrollaz. Die Datenbank wurde eingerichtet von Matthias Perrollaz, Luca Raneri und Pascal Steiner.

5.2.3 Webservice (Blackbox)

NAME	WEBSERVICE
ZWECK / VERANTWORTLICHKEIT	Dieser Baustein verantwortet die Kommunikation zwischen dem Extranet Client und dem Applikationsserver.
SCHNITTSTELLE 1	<p>Applikationsserver (via RMI)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersicht (Persönliche Terminübersicht) <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Kontrolldaten - Stammdaten (Wärmeerzeuger, Brenner) - Termine
SCHNITTSTELLE 2	<p>Extranet Client (via SOAP)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Kontrolldaten - Stammdaten (Wärmeerzeuger, Brenner) - Termine <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersicht (Persönliche Terminübersicht)
ABHÄNGIGKEITEN	Der Baustein Webservice ist abhängig vom Server Tomcat sowie der Laufzeitumgebung der Serverinfrastruktur der HSLU.
CODE-ARTEFAKTE	Run-feukora-gruppeB-webservice.bat Feukora-gruppeB-webservice.jar Lib Folder
ERFÜLLTE ANFORDERUNGEN	Req08, Req09
VARIABILITÄT	Dieser Baustein verändert sich vorerst nicht. Ausser es wir eine Erweiterung geplant.
WEITERE INFORMATIONEN	Dominik Stirnimann implementierte den Webservice. Unterstützt beim Testing durch Olivia Wassmer

5.2.4 Extranet Client (Blackbox)

NAME	EXTRANET CLIENT
ZWECK / VERANTWORTLICHKEIT	Stellt das Grafical User Interface (GUI) für den Heizungskontrolleur zur Verfügung. Dort können Stamm- / Kontrolldaten eingegeben werden und über den Webservice an den Applikationsserver und schliesslich an die Datenbank weitergeleitet werden.
SCHNITTSTELLE 1	<p>Webservice</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersicht (Persönliche Terminübersicht) <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Kontrolldaten - Stammdaten (Wärmeerzeuger, Brenner) - Termine
ABHÄNGIGKEITEN	Der Baustein «Extranet Client» ist abhängig von der Laufzeitumgebung des Laptops des jeweiligen Heizungskontrolleurs sowie von der Schnittstelle zum Webservice wodurch er Daten via Applikationsserver und Datenbank abspeichern resp. anzeigen lassen kann.
CODE-ARTEFAKTE	Feukora-grupeB-appCE.jar Run-feukora-gruppeB-appCE.bat Lib Folder Req08, Req09
ERFÜLLTE ANFORDERUNGEN	
VARIABILITÄT	Dieser Baustein verändert sich vorerst nicht. Ausser es werden mehr Funktionen für die Feuerungskontrolleure benötigt.
WEITERE INFORMATIONEN	Autoren sind Pascal Steiner und Alexandra Lengen. Unterstützt beim Testing durch Olivia Wassmer

5.2.5 Intranet Client (Blackbox)

NAME	INTRANET CLIENT
ZWECK / VERANTWORTLICHKEIT	Stellt das Grafical User Interface (GUI) für den Sachbearbeiter im Backoffice zur Verfügung. Dort können Stamm- / Kontrolldaten eingegeben werden und über RMI an den Applikationsserver und schliesslich an die Datenbank weitergeleitet werden.
SCHNITTSTELLE 1	<p>Applikationsserver (via RMI)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersicht (Persönliche Terminübersicht) <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Kontrolldaten - Stammdaten (Wärmeerzeuger, Brenner) - Termine
ABHÄNGIGKEITEN	Der Intranet Client ist als Baustein abhängig von der Laufzeitumgebung des Computers des jeweiligen

CODE-ARTEFAKTE	Back-Office Sachbearbeiters sowie von der RMI Verbindung zum Applikationsserver. Feukora-gruppeB-appCI.jar Run-feukora-gruppeB-appCI.bat Lib Folder Req07, Req12
ERFÜLLTE ANFORDERUNGEN VARIABILITÄT	Dieser Baustein verändert sich vorerst nicht. Ausser es werden mehr Funktionen für die Sachbearbeiter benötigt.
WEITERE INFORMATIONEN	Autoren sind Pascal Steiner und Alexandra Lengen. Unterstützt beim Testing durch Olivia Wassmer

5.3 Ebene 2

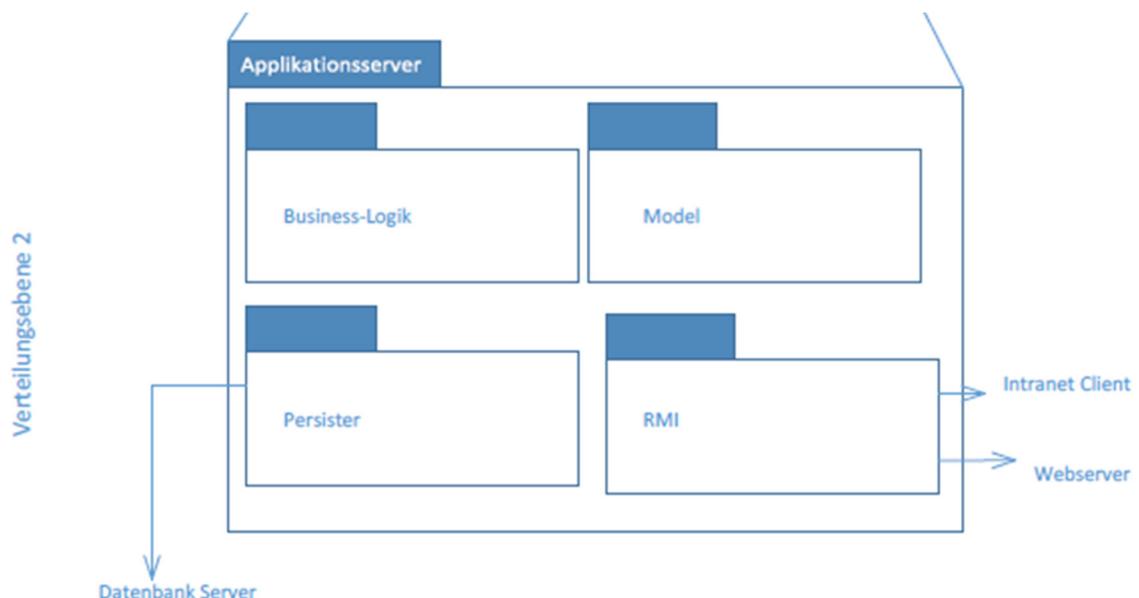


Abbildung 5 - Bausteinsicht Feukora Vertiefungsebene 2

5.3.1 Businesslogik-Komponente (Blackbox)

NAME	BUSINESSLOGIK-KOMPONENTE
ZWECK / VERANTWORTLICHKEIT	Beinhaltet die Businesslogik der Applikation, speziell erwähnenswert sind die Hauptprozesse «Kontrolldaten speichern» sowie «Terminverwaltung»
SCHNITTSTELLE 1	<p>Model-Komponente Eingehende Informationen: - Instanzen/Instanzvariablen der Modelklassen Ausgehende Informationen: - Werte für die Instanzvariablen der Modelklassen</p>
SCHNITTSTELLE 2	<p>RMI-Komponente Eingehende Informationen: - Kontrolldaten - Stammdaten - Termine / Aufträge Ausgehende Informationen:</p>

	<ul style="list-style-type: none"> - Stammdaten (Anzeige aller Stammdaten) - Terminübersichten
SCHNITTSTELLE 3	<p>Persister-Komponente (via EclipseLink, JPA)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Abgefragte Daten und Objekte - Status Speicherung <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten und Objekte zur Speicherung <p>Dieser Baustein ist abhängig von der Laufzeitumgebung der Serverinfrastruktur der HSLU.</p> <p>Zusammengefasst im höheren Baustein</p> <p>Req14</p> <p>Dieser Baustein verändert sich vorerst nicht, ausser es sollen mehr Funktionen hinzugefügt werden.</p> <p>Autor des Interfaces, der Implementierung sowie der Tests: Olivia Wassmer</p>
ABHÄNGIGKEITEN	
CODE-ARTEFAKTE	
ERFÜLLTE ANFORDERUNGEN	
VARIABILITÄT	
WEITERE INFORMATIONEN	

5.3.2 RMI-Komponente (Blackbox)

NAME	RMI-KOMPONENTE
ZWECK / VERANTWORTLICHKEIT	Beinhaltet die Funktionalität um eine Verbindung zwischen Client Intern und Applikationsserver herzustellen, sowie für die Verbindung zwischen Webservice und Businesslogik.
SCHNITTSTELLE 1	<p>Businesslogik-Komponente</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Abgerufene Objekte oder Daten aus der Datenbank <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Abfragen nach Objekte und Daten
SCHNITTSTELLE 2	<p>Webservice (via SOAP)</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Abfragen nach Objekte und Daten <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Abgerufene Objekte oder Daten aus der Datenbank
SCHNITTSTELLE 3	<p>Intranet Client</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Abfragen nach Objekte und Daten <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Abgerufene Objekte oder Daten aus der Datenbank <p>Dieser Baustein ist abhängig von der Laufzeitumgebung der Serverinfrastruktur der HSLU.</p> <p>Zusammengefasst im höheren Baustein</p> <p>Req08, Req09</p> <p>Dieser Baustein verändert sich vorerst nicht, ausser es sollen mehr Funktionen hinzugefügt werden oder neue Verbindungen zu neuen Komponenten erstellt werden.</p> <p>Autor dieses Bausteins ist Matthias Perrollaz.</p>
ABHÄNGIGKEITEN	
CODE-ARTEFAKTE	
ERFÜLLTE ANFORDERUNGEN	
VARIABILITÄT	
WEITERE INFORMATIONEN	

5.3.3 Modell-Komponente (Blackbox)

NAME	MODELL-KOMPONENTE
ZWECK / VERANTWORTLICHKEIT	Beinhaltet die Struktur der Entitäten sowie deren Inhalt (Attribute).
SCHNITTSTELLE 1	<p>Businesslogik-Komponente</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Werte für die Instanzvariablen der Modelklassen sowie Objektreferenzen <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Instanzen/Instanzvariablen der Modelklassen
SCHNITTSTELLE 2	<p>Persistence-Komponente</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - keine <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Annotationen und Struktur der Entitäten
ABHÄNGIGKEITEN	Dieser Baustein ist abhängig von der Laufzeitumgebung der Serverinfrastruktur der HSLU.
CODE-ARTEFAKTE	Zusammengefasst im höheren Baustein
ERFÜLLTE ANFORDERUNGEN	-
VARIABILITÄT	Dieser Baustein verändert sich vorerst nicht, ausser die Struktur der Entitäten verändert sich.
WEITERE INFORMATIONEN	Autor ist Olivia Wassmer, Anpassungen durch Matthias Perrollaz und Olivia Wassmer

5.3.4 Persistence-Komponente (Blackbox)

NAME	PERSISTENCE-KOMPONENTE
ZWECK / VERANTWORTLICHKEIT	Dieser Baustein ist verantwortlich für die Verbindung zur Datenbank und das abspeichern, updaten, löschen etc.
SCHNITTSTELLE 1	<p>Datenbankserver</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Objekte und Daten welche von höheren Komponenten angefordert werden <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Objekte und Daten welche gespeichert werden sollen.
SCHNITTSTELLE 2	<p>Businesslogik-Komponente</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Stammdaten und Objekte zur Speicherung <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - Abgefragte Daten und Objekte - Status Speicherung
SCHNITTSTELLE 3	<p>Model-Komponente</p> <p>Eingehende Informationen:</p> <ul style="list-style-type: none"> - Annotationen und Struktur der Entitäten <p>Ausgehende Informationen:</p> <ul style="list-style-type: none"> - keine
ABHÄNGIGKEITEN	Dieser Baustein ist abhängig von der Laufzeitumgebung der Serverinfrastruktur der HSLU.
CODE-ARTEFAKTE	Zusammengefasst in höherem Baustein
ERFÜLLTE ANFORDERUNGEN	Req01, Req02, Req03, Req04, Req05, Req06, Req07
VARIABILITÄT	Dieser Baustein verändert sich vorerst nicht, ausser die Prozesse für die Speicherung werden verändert.

6 Laufzeitsicht

6.1 Laufzeitszenario 1: Messwerte eingeben

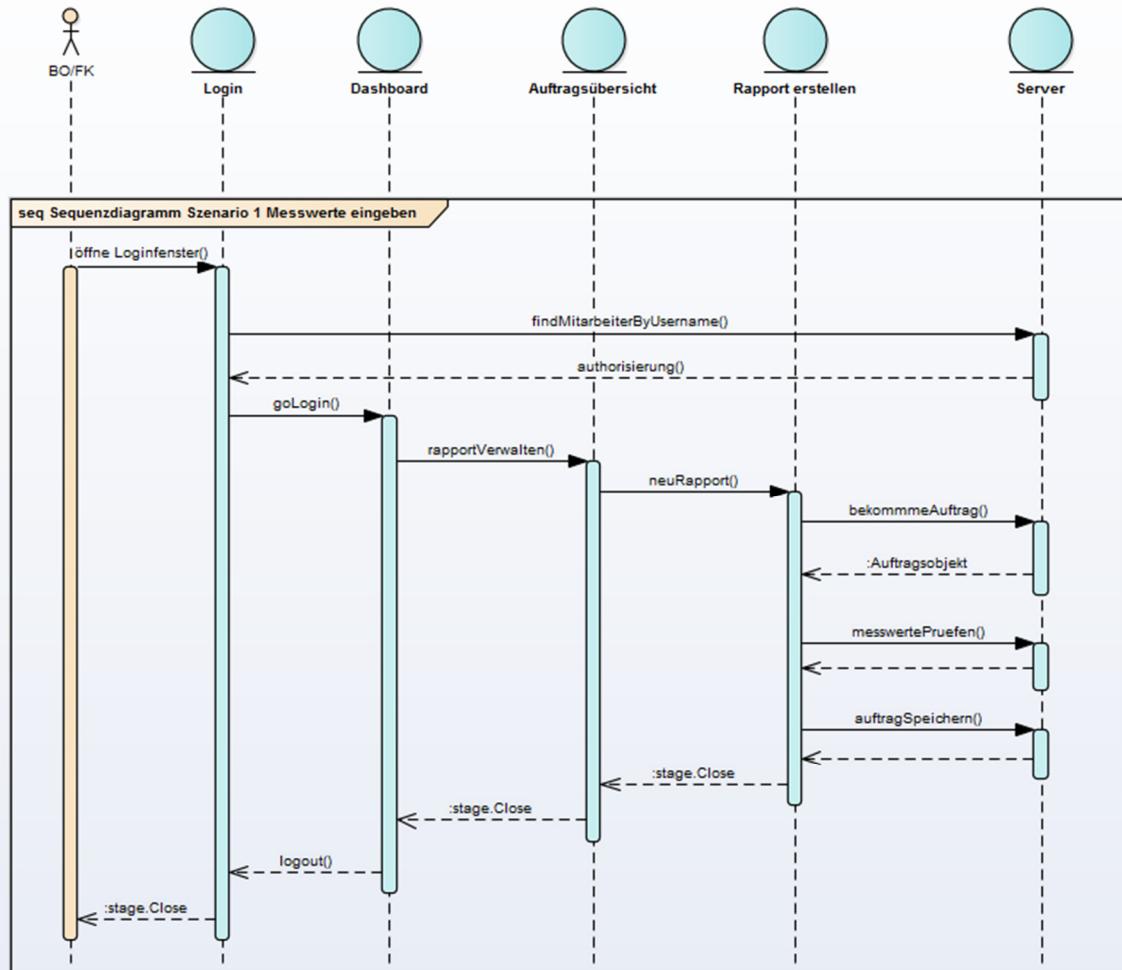


Abbildung 6 - Laufzeitszenario 1 Feukora Messwerte eingeben

SZENARIO	ZWECK UND KURZBESCHREIBUNG
Messwerte eingeben	Der Feuerungskontrolleur öffnet das Login Fenster der Applikation Feukora. Nachdem sich der Feuerungskontrolleur erfolgreich auf dem Applikations-Server eingeloggt hat, hat er die Möglichkeit via Terminübersicht seinen Termin zu finden. Ist dieser gefunden kann ein neuer Rapport eröffnet werden um die Kontrolldaten zu erfassen. Der Feuerungskontrolleur gibt alle Daten ein. Diese werden direkt überprüft, ob sie im zugelassenen Bereich sind. Das Ergebnis dieser Überprüfung wird visuell dem User mitgeteilt. Sind alle Daten erfasst, kann der Rapport gespeichert werden. Die Fenster werden anschliessend geschlossen und der Benutzer kann sich ausloggen.

6.2 Laufzeitszenario 2: Auftrag erstellen

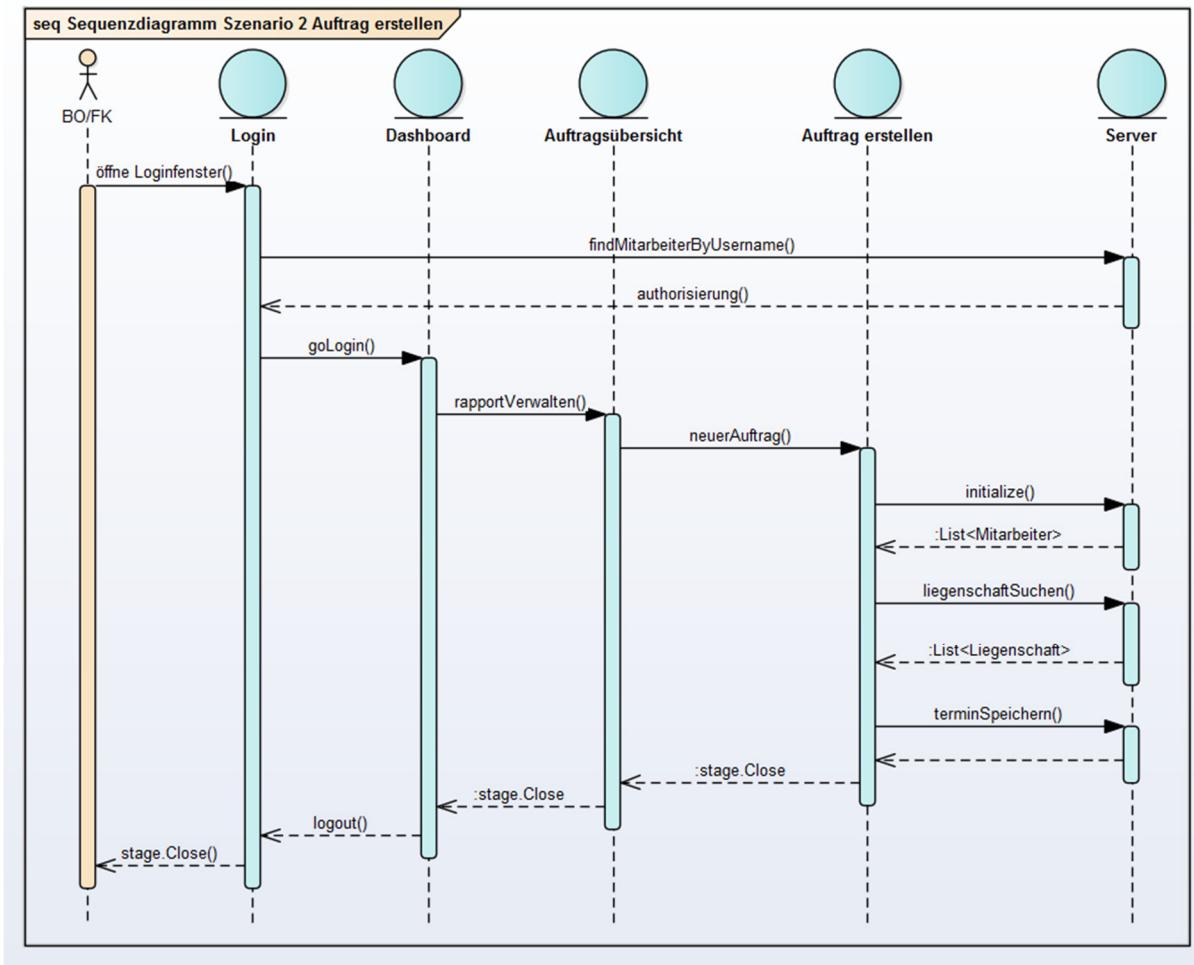


Abbildung 7 - Laufzeitszenario 2 Feukora Auftrag erstellen

SZENARIO	ZWECK UND KURZBESCHREIBUNG
Auftrag erstellen	<p>Der Feuerungskontrolleur öffnet das Login Fenster der Applikation Feukora. Nachdem sich der Benutzer erfolgreich auf dem Applikations-Server eingeloggt hat, hat er die Möglichkeit via Dashboard einen neuen Termin zu erstellen. Die Kunden sowie die Feuerungskontrolleure werden via Drop-Down Menü angezeigt. Erfasst ein Feuerungskontrolleur selber einen Termin ist das Mitarbeiterfeld automatisch mit seinem Namen ausgefüllt, da dieser nur Termine für sicher selber erstellen kann.</p> <p>Der Benutzer hat jetzt die Möglichkeit einer Liegenschaft, ein Datum mit Zeit Slot und ev. den auszuführenden Mitarbeiter auszuwählen. Wurden alle Daten richtig eingegeben, wird der Termin jetzt gespeichert. Dieser Termin ist nun neu in der Terminübersicht ersichtlich. Die Fenster werden anschliessend geschlossen und der Benutzer kann sich ausloggen.</p>

6.3 Laufzeitszenario 3: Terminübersicht drucken

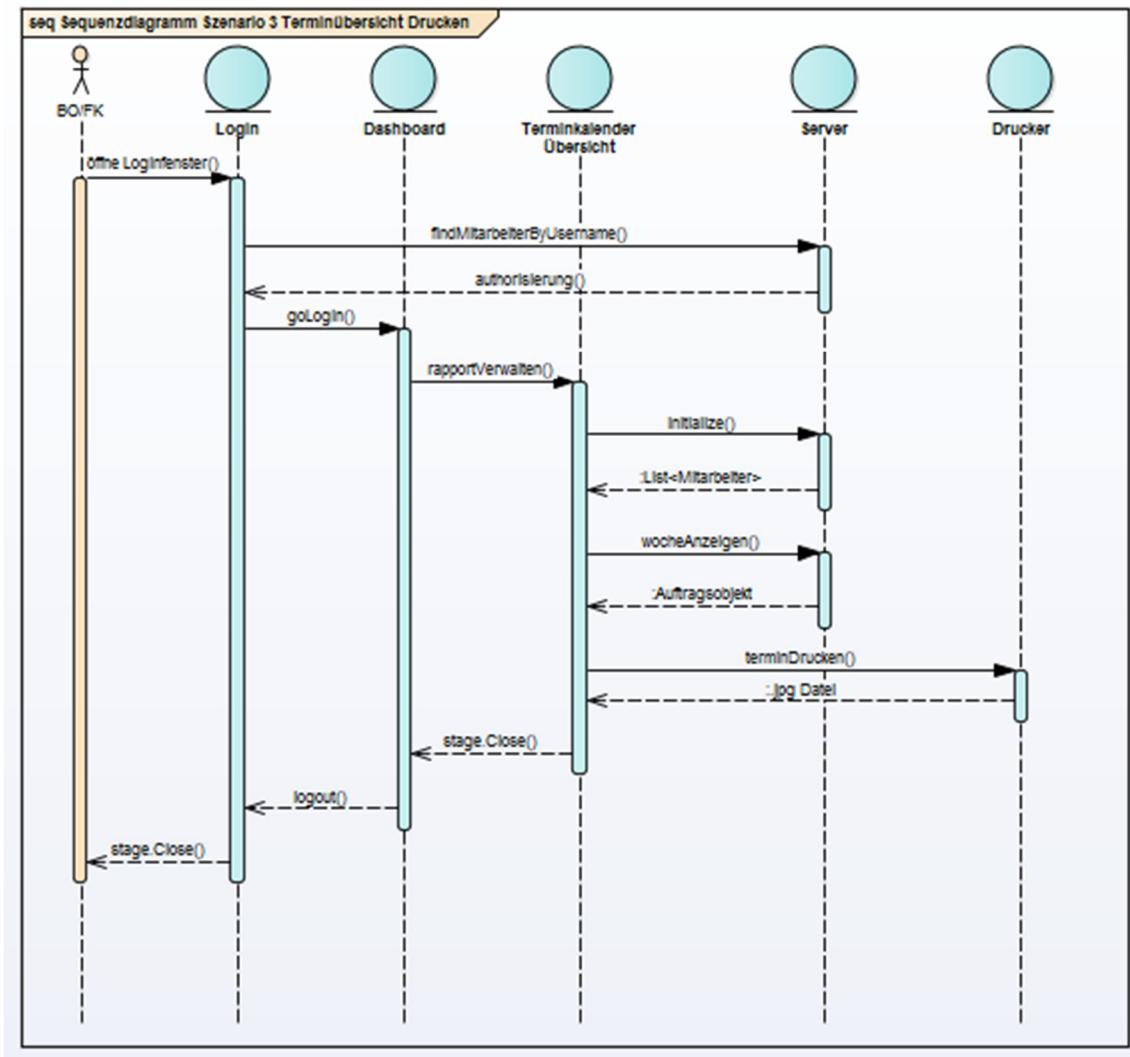


Abbildung 8 - Laufzeitszenario 3 Feukora Terminübersicht drucken

SZENARIO	ZWECK UND KURZBESCHREIBUNG
Terminübersicht drucken	Der Feuerungskontrolleur öffnet das Login Fenster der Applikation Feukora. Nachdem sich der Benutzer erfolgreich auf dem Applikations-Server eingeloggt hat, hat er die Möglichkeit via Dashboard seine Terminübersicht anzeigen zu lassen. Möchte ein Feuerungskontrolleur seine Terminübersicht drucken, ist das Mitarbeiterfeld automatisch mit seinem Namen ausgefüllt, da dieser nur Termine für sicher anzeigen lassen kann. Die gewünschte Woche, welche gedruckt werden soll ist anzuwählen, indem das Startdatum des Montags der Woche eingegeben wird. Die Terminübersicht kann nun bei Bedarf gedruckt werden. Dazu wird ein Printscrean erstellt, welcher dann bequem gedruckt werden kann. Die Fenster werden anschliessend geschlossen und der Benutzer kann sich ausloggen.

7 Verteilungssicht

7.1 Infrastruktur Ebene 1

7.1.1 Verteilungsdiagramm

Die Verteilungssicht gestaltet sich im Projekt Feukora relativ simpel. Die Applikation ist generell auf drei Geräte verteilt. Einmal auf dem Laptop des Feuerungskontrolleurs mit dem Applikationsbaustein «Client extern». Zum zweiten auf dem PC der Sachbearbeiter im Back-Office, mit dem Applikationsbaustein «Client intern». Ausserdem ist sie auf dem Applicationserver der Heizfirma (auf dem der Kern der Applikation läuft), mit den groben Bausteinen Webservice, Serverapplikation und Datenbank.

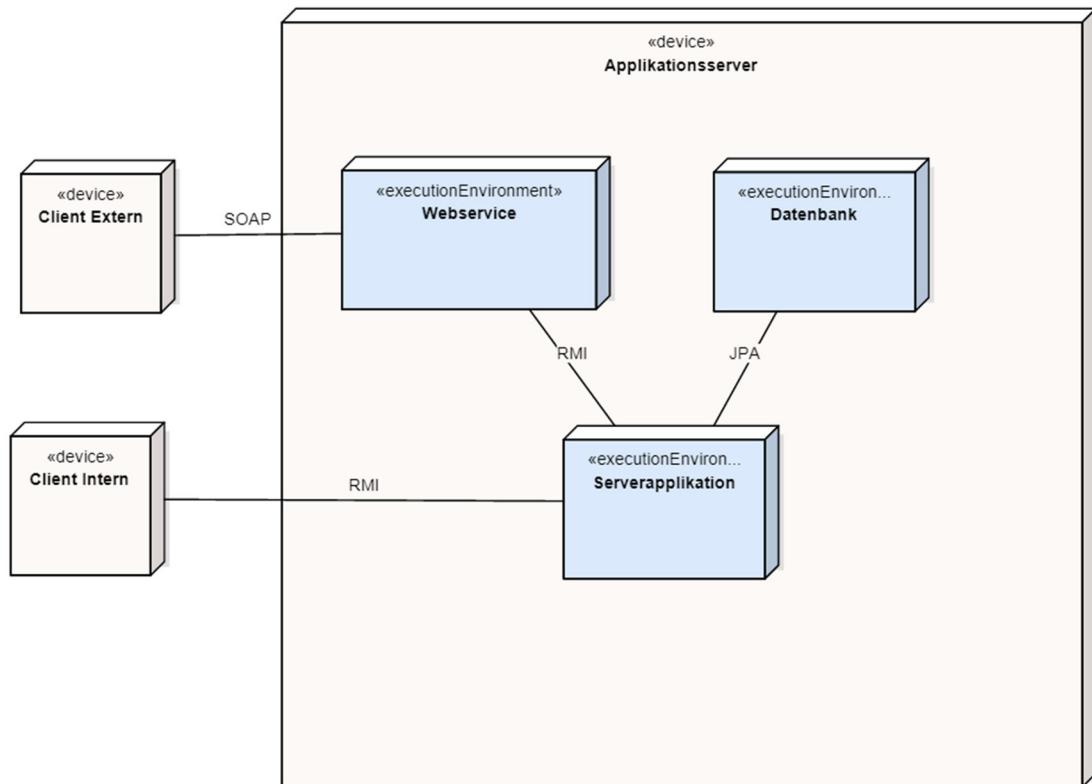


Abbildung 9 - Verteilungssicht Feukora

7.1.2 Verteilungsartefakte

VERTEILUNGSARTEFAKT	BAUSTEIN
Applikationsserver	Serverapplikation Webservice Datenbank
Client Extern	Client Extern
Client Intern	Client Intern

Schlussendlich können drei Verteilungsartefakte aus den Bausteinen definiert werden. Die einzelnen Bausteine werden im [Kapitel 5 der Bausteinssicht](#) erläutert.

7.1.3 Spezifikation der Knoten

7.1.3.1 Laptop Feuerungskontrolleur

KNOTEN ELEMENT	BESCHREIBUNG
NAME	Laptop/Tablet Feuerungskontrolleur
BESCHREIBUNG, MOTIVATION	Der Feuerungskontrolleur braucht ein portables Gerät welches er auf seine Ausseneinsätze für die Kontrollen / Service mitnehmen kann. Dafür eignet sich ein Laptop/Tablet. Eine zweite Variante wäre ein Smartphone, diese Lösung wurde jedoch vom Auftraggeber gestrichen. Auf dem Laptop muss Windows7 installiert sein. Genauere Leistungsmerkmale sind keine vorgeschlagen oder vorgeschrieben. Natürlich verändert sich die Performanz je nach Leistungsmerkmalen des Knoten.
LEISTUNGSMERKMALE	
ZUGEORDNETE VERTEILUNGSSARTEFAKTE	Client Extern
SONSTIGE VERWALTUNGSDATEN	Die Applikation feukora-gruppeB-clientExtern und ihre Komponenten werden durch die Heizfirma verteilt und kann dann installiert werden. Anschliessend ist die Applikation verfügbar.

7.1.3.2 PC Back-Office

KNOTEN ELEMENT	BESCHREIBUNG
NAME	PC Back-Office
BESCHREIBUNG, MOTIVATION	Die Sachbearbeiter brauchen ein stabiles Gerät welches sich für viel Administrative Arbeit im Büro eignet. Dafür eignet sich eine Workstation bestens. Eine zweite Variante wäre ebenfalls ein portable Computer – Laptop – diese Variante wäre jedoch nicht so ergonomisch für lange Büroarbeit und darum weniger empfohlen. Auf dem PC muss Windows7 installiert sein. Genauere Leistungsmerkmale sind keine vorgeschlagen oder vorgeschrieben. Natürlich verändert sich die Performanz je nach Leistungsmerkmalen des Knoten.
LEISTUNGSMERKMALE	
ZUGEORDNETE VERTEILUNGSSARTEFAKTE	Client Intern
SONSTIGE VERWALTUNGSDATEN	Die Applikation feukora-gruppeB-clientIntern und ihre Komponenten werden durch die Heizfirma verteilt und kann dann installiert werden. Anschliessend ist die Applikation verfügbar.

7.1.3.3 Applikationsserver Heizfirma

KNOTEN ELEMENT	BESCHREIBUNG
NAME	Applikationsserver Heizfirma
BESCHREIBUNG, MOTIVATION	Die Applikation mit Ihren Komponenten soll auf einem leistungsfähigen Server lauffähig sein. Damit ist die durchgehende Verfügbarkeit der Applikation für die Clients gewährleistet. Auf dem PC muss Windows7 installiert sein. Genauere Leistungsmerkmale sind keine vorgeschlagen oder vorgeschrieben. Natürlich verändert sich die Performanz je nach Leistungsmerkmalen des Knoten.
LEISTUNGSMERKMALE	
ZUGEORDNETE VERTEILUNGSSARTEFAKTE	Serverapplikation, Webservice, Datenbank

SONSTIGE VERWALTUNGSDOKUMENTATIONEN

Sämtliche Verteilungsartefakte werden vom Projektteam geliefert und schliesslich von der Heizfirma verwaltet und betrieben.

7.2 Fachliche Strukturen und Modelle

Die fachliche Struktur kann in ihren Grundsätzen aus der folgenden Grafik entnommen werden. Dabei zählt das Grafical User Interface (GUI) zur View, nach dem Model-View-Controller Prinzip. Der Controller hat die gleiche Funktion wie im ursprünglichen Modell und handelt Benutzer-Events und Eingabekontrolle sowie die Weitergabe der Zustandsänderungen. GUI und Controller sind zusammen im Baustein Client (Intern oder Extern) vereint. Der Bereich Model ist in unserem Fall etwas differenzierter und setzt sich aus zwei Instanzen zusammen.

Die erste Instanz setzt sich aus einem Prozess von RMI, Modelklassen und Persister zusammen durch welche Änderungsmeldungen oder Zustandsänderungen hindurchlaufen. Diese gehen anschliessend weiter in die Datenbank (DB) und werden dort gespeichert resp. aufgerufen.

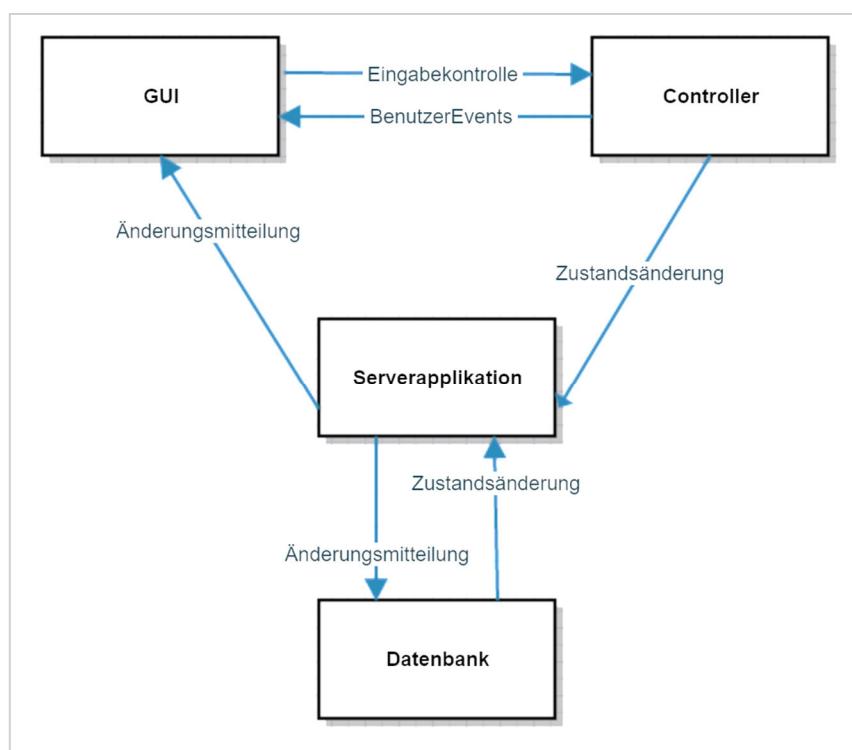


Abbildung 10 - Fachliche Strukturen Feukora

7.3 Typische Muster und Strukturen

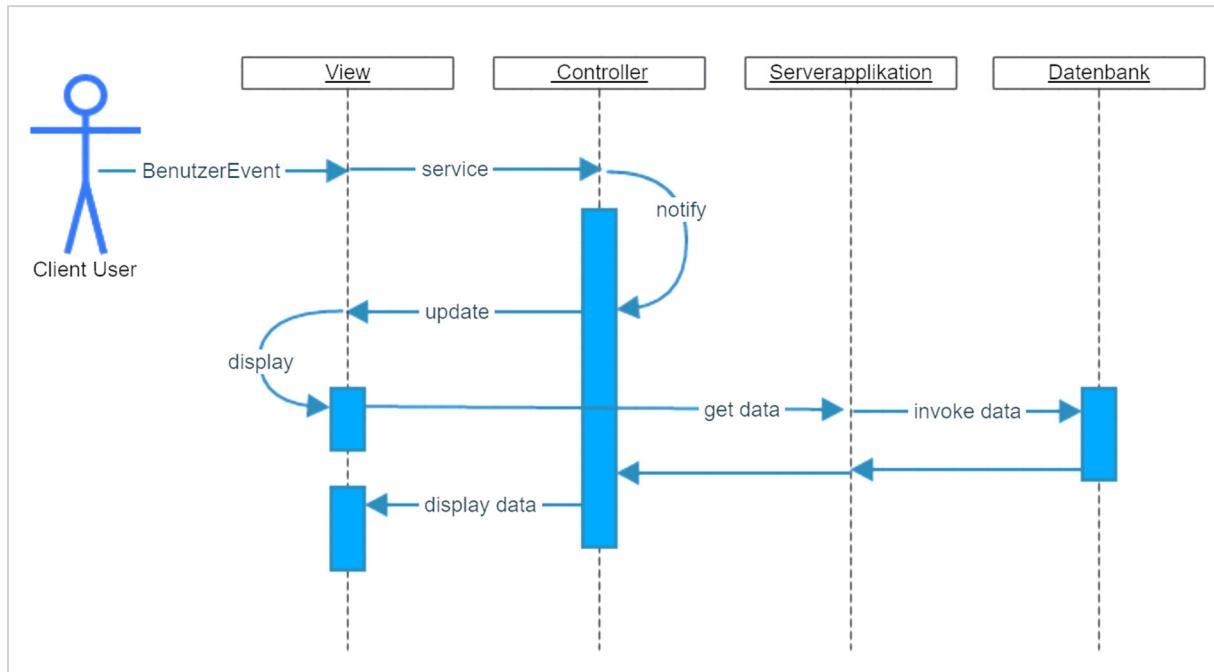


Abbildung 11 - Typische Muster und Strukturen Feukora

8 Technische Konzepte

8.1 Persistenz

Die Daten werden in eine PostgreSQL Datenbank abgespeichert. Dies geschieht mittels Java Persistence API (JPA). Die Entitäten dieser Datenbank werden direkt in den Modelklassen mit Annotationen festgelegt. Das heisst, ORM (Object Relational Mapping) wird als Grundlage der Persistierung benutzt. Mit dieser Variante ist es sowohl mit Java JEE und JSE verwendbar und es wird keine spezielle Middleware mehr benötigt. Ein weiterer Vorteil ist, dass ORM die SQL Statements übernimmt und nichts mehr manuell mit menschlicher Fehlerquote geschrieben werden muss.

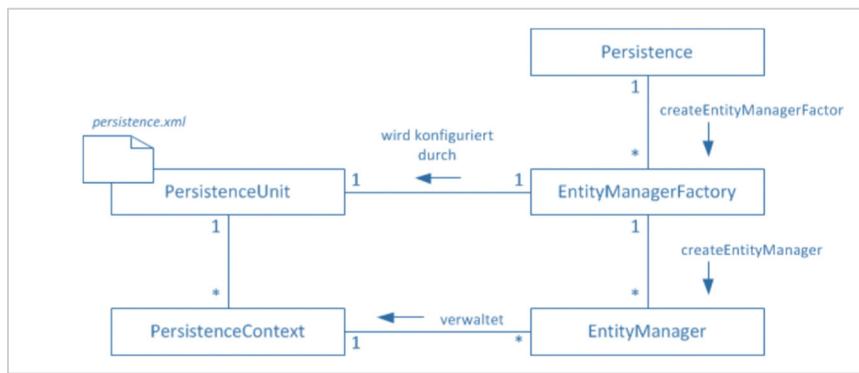


Abbildung 12 - Persistenz Prozess Aufbau

8.2 Benutzungsoberfläche

Die Grafical User Interfaces werden mit Scene Builder modelliert und mit JavaFX ausprogrammiert. Aufgrund der verschiedenen Benutzerrechte ergeben sich im Projekt Feukora zwei grafische Oberflächen.

8.2.1 Feuerungskontrolleur (Client Extern)

Das GUI des Baustein Client Extern wird vom Feuerungskontrolleur benutzt. Dieser muss während seinen Aufträgen, Feuerungskontrolldaten, Wärmeerzeuger und Brenner verwalten können. Zudem muss er neue Termine mit Kunden vereinbaren können und diese mithilfe einer Terminübersicht wieder auslesen können.

8.2.2 Backoffice (Client Intern)

Ergänzend zu den Funktionalitäten des Client Extern kann die Benutzeroberfläche des Backoffice ebenfalls Mitarbeiter verwalten. Der Client Intern hat somit die grösste Berechtigung unter den Mitarbeitern einer Heizfirma. Als einzige Instanz über ihm, ist der Systemadministrator, dieser kann zusätzlich Benutzerdaten für die Mitarbeiter erstellen. Er benutzt dafür ein abgespecktes GUI auf dem Client Intern welches nur diese Funktion anbietet. Mit Ausblick in die Zukunft könnte man dies sehr gut und relativ simpel erweitern.

8.3 Ergonomie

Die Applikation ist sehr kompakt und auf die wichtigste Funktionalität beschränkt. Die bestehenden Möglichkeiten sind übersichtlich und möglichst benutzerfreundlich umgesetzt, sodass die Arbeit mit der Feukora App keine Mühe mehr bereitet.

8.4 Ablauf- Dialog und Workflowsteuerung

Das Grafical User Interface bietet viele Funktionen, um einen Überblick zu schaffen bieten sich folgende Flussdiagramme an. Zum einen für den Client Intern mit voller Funktionalität und Client Extern mit einer etwas reduzierteren Funktionalität. Des Weiteren gibt es ein GUI für den Administrator, dieses wird jedoch hier nicht weiter aufgezeigt, da es rein zur Benutzererstellung dient.

8.4.1 Flussdiagramm Client Intern



Abbildung 13 - Workflow Client Intern

8.4.2 Flussdiagramm Client Extern

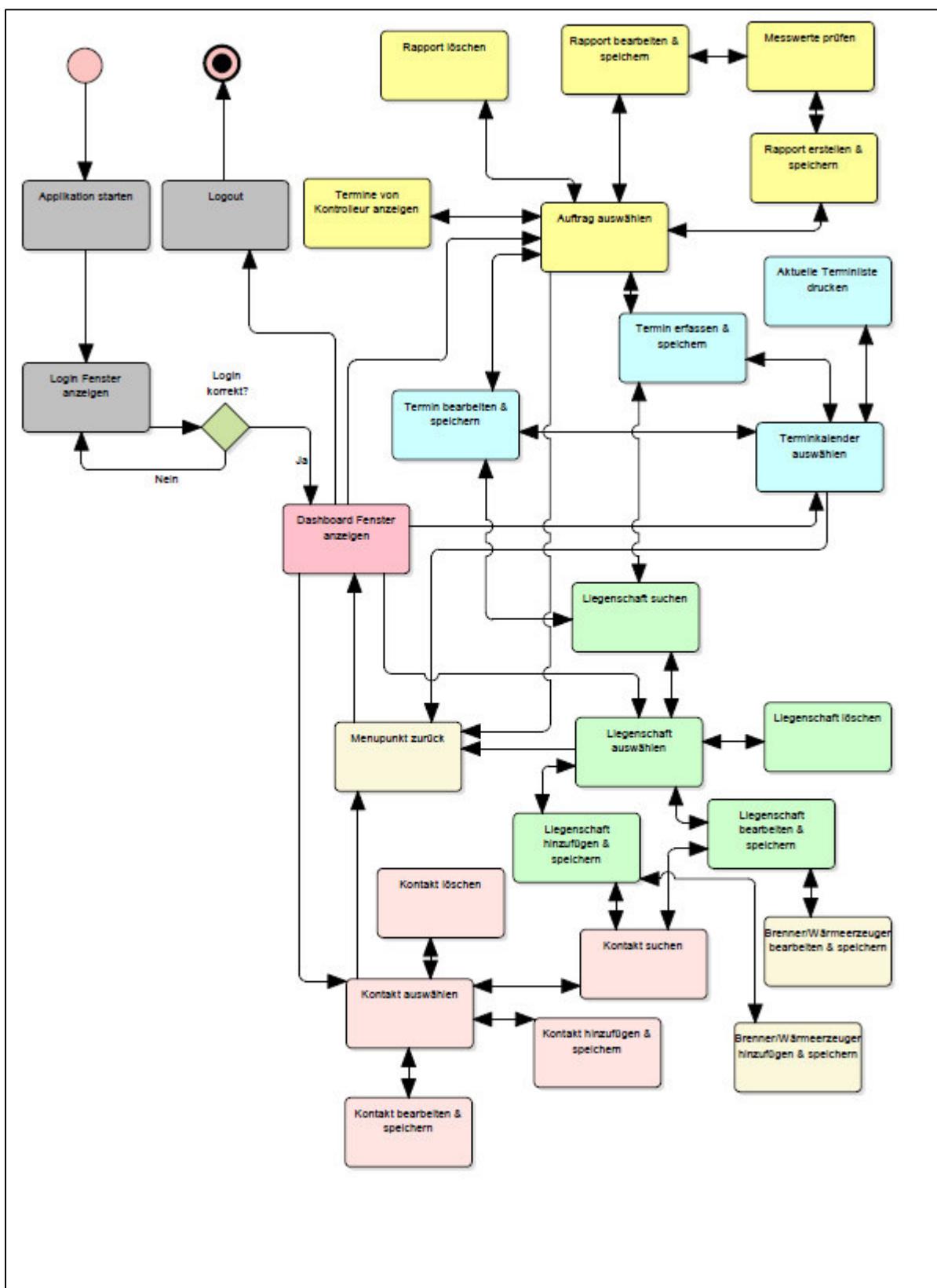


Abbildung 14 - Workflow Client Extern

8.5 Geschäftsregeln

Die gesetzlichen Grenzwerte zählen zu den Regeln für die Unternehmung bei den Feuerungskontrollen. Diese Regeln werden in der Businesslogik abgehandelt, so werden überschrittene Grenzwerte dementsprechend markiert und gespeichert. Der Feuerungskontrolleur im Einsatz kann darauf direkt einen neuen Termin vereinbaren um die Überschreitung zu behandeln oder erneut zu überprüfen.

Jedoch muss nach dem Deployment der Applikation, in der jeweiligen Heizungsfirma, sicher eine Form des Change Management vorgenommen werden. Die Applikation wird vorgestellt und anschliessend muss sich jede Firma für sich selbst klar werden, welche Regeln sie in Verbindung mit der Applikation aufstellen.

8.6 Plausibilisierung und Validierung

Daten welche ins User Interface eingegeben werden müssen vollständig sein, dies wird jeweils von den Controller Klassen im Client kontrolliert. Ebenfalls werden Zahlen oder Buchstabenwerte identifiziert und bei falscher Eingabe gemeldet. Im GUI wird dann jeweils mit roten und grünen Felder signalisiert, ob die Eingabe erlaubt ist. Das ACID Prinzip (atomic consistent, isolated, durable) sollte hier beachtet werden. Da das Projekt jedoch in der Objekt orientierten Sprache Java programmiert ist, ist die Datenstruktur nicht zu 100% normalisiert. Dadurch wurde aber eine performantere Applikation erreicht. Nach dem erfolgreichen Eingeben der Daten werden anschliessend die nachkommenden Prozesse ausgeführt.

Für die Korrektheit der Angaben sind jeweils die Benutzer der Applikation verantwortlich. So müssen Namen, Adressen, Nummern korrekt eingetragen werden, denn sie werden meist eins zu eins so in die Datenbank gespeichert.

8.7 Transaktionsbehandlung

Der Benutzer kann mit der Feukora Applikation eine Vielzahl an Transaktionen vornehmen und zwar immer dann wenn er ein Datensatz speichert, anzeigt, updatet oder löscht. Die Transaktion wird vom Client über die jeweilige Client Server Verbindung zur Businesslogik und von da zum Persister geleitet dieser setzt die Transaktionen schliesslich um. Siehe auch 8.6 (Plausibilisierung und Validierung)

8.8 Sessionsbehandlung und Zustandsbehandlung

Wir brauchen für unsere Software keine dauerhaften Sessions, da wir nur einzelne und vor allem getrennte Kommandos an die DB haben. Damit können wir problemlos das HTTP als Übertragungsprotokoll gebrauchen. Die Verbindung zur Datenbank muss aber auf jeden Fall wieder beim Entity Manager geschlossen werden(siehe Bild).

Einführung in Java Persistence API mit EclipseLink Grundbegriffe und Klassen – VII

- Transaktion "starten":

```
em.getTransaction().begin();
```
- Objekte erstellen, Speichern ...

```
Adresse adr = new Adresse ( ... );  
em.persist(adr);
```
- Änderungen in die Db speichern:

```
em.getTransaction().commit();
```
- EntityManager schliessen:

```
em.close();
```

Abbildung 15 - Grundlagen Entity Manager

8.9 Caching

Alle Daten werden direkt nach Eingabe und deren Speicherung in die Datenbank geschrieben. Somit besteht hier kein Grund Daten auf irgendeine Weise in Caches zu halten und effizienter aufzubereiten.

8.10 Sicherheit

Da unser System keine hochsensiblen Daten behandelt, wird die Sicherheit mit der Authentifizierung geprüft. Das heisst, die Daten werden bei der HTTP Übertragung nicht zusätzlich verschlüsselt und sind mit einfachen Tools wie Wireshark problemlos lesbar. Durch die Authentifizierung wird festgestellt, ob es sich um einen Feuerungskontrolleur oder um ein Backoffice Mitarbeiter handelt. Diese haben verschiedene Rechte bei der Benutzung der Software. Unterschieden wird zwischen Backoffice Mitarbeiter und Feuerungskontrolleuren, wobei erstere mehr Zugriffsrechte haben (z.B. Mitarbeiterverwaltung). Zudem gilt, nur der Administrator soll Benutzerlogins erstellen und ändern können. Die Verantwortung des Backups liegt auf der Kundenseite.

8.11 Kommunikation und Integration mit anderen IT Systemen

Die Applikation Feukora ist in sich eigenständig. Das heisst, es sind keine externen Systeme angeschlossen.

Einzig beim Drucken der Terminübersicht wird ein Dokument durch ein PrintScreen Befehl erzeugt. Dieses kann dann durch ein geeignetes Bildbearbeitungstool ausgedruckt werden.

8.12 Verteilung

Das Programm liegt ganz beim Applikationsserver. Der Externe Client muss seine Informationen per SOAP (Stub von Webservice) zuerst an die Webservicekomponente schicken, welche dann per RMI die Daten an die Businesslogik weiterleitet. Diese leitet es an die Persister Komponente weiter. Die URL und Port-Nummer von der Datenbank werden in vier Property-Files beim Entity Manager codiert. RMI und Webserver beinhalten ebenfalls Property-Files mit den jeweiligen Portnummern und Hostadresse

8.13 Ausnahme- und Fehlerbehandlung

Wir haben folgende Exceptions welche auch in einem Logfile protokolliert werden:

- IO Exception bei den ServerSockets
- Remote Exceptions beim RMI (siehe Grafik)

```
import java.rmi.*;
import java.rmi.server.*;

public class AdderImpl extends UnicastRemoteObject
implements Adder {

    public AdderImpl() throws RemoteException {
    }

    public int add(int x, int y) throws RemoteException {
        return x + y;
    }
}
```

Abbildung 16 – Exception-Handling

Die Fehler sollten alle eine printStackTrace() Methode aufrufen damit wir feststellen können in welchem Teil der Fehler aufgetreten ist.

Sämtliche Fehler aus dem System aus den Bausteinen Persister, Businesslogik, Modelklassen werden mit throws Exception weitergeleitet über RMI bis zum Client welcher sie abfängt und behandelt. So kann die Applikation benutzt werden ohne abzustürzen.

8.14 Systemmanagement und Administrierbarkeit

Für das System wird eine Administrator Rolle erzeugt. Dieser Administrator kann ein ausgebildeter BackOffice Mitarbeiter sein oder der Kunde kann auf Wunsch die Betreuung des Systems unserer Firma übergeben. Der Administrator bewirtschaftet das System nach empfinden und schaltet neue Benutzer(Kontrolleure/BackOffice-Mitarbeiter) auf.

8.15 Logging, Protokollierung und Tracing

Alle Vorgänge aus dem Benutzerinterface werden mithilfe von „log4j“ in einer Logdatei festgehalten und protokolliert. Darunter zählen Exceptions, modifizierte Datensätze um nur einige zu nennen. Die Logdatei kann vom Administrator bei Bedarf abgeholt werden.

8.16 Konfigurierbarkeit

Da sich die Systemlandschaft nach Einrichten beim Kunden nicht verändert sind alle Konfigurationen in Properties Dateien hart codiert. Diese werden bei der Übergabe durch unser Deployment-Team festgelegt. Weitere Informationen sind in [Kapitel 17](#) (Deployment Informationen).

8.17 Parallelisierung und Threading

RMI verfügt von Haus aus über die Funktion des Multithreading. Wir brauchen keine parallelen Prozesse zu programmieren.

8.18 Internationalisierung

Unser System wurde ausschliesslich für den schweizerischen Dachverband realisiert. Eine Internationalisierung wurde nicht gewünscht und wird deshalb auch nicht umgesetzt. Dies ist jedoch zu einem späteren Zeitpunkt technisch realisierbar. Kleine Anpassungen an internationale Tastaturen wurden vorerst beispielsweise bei der Namensgebung der Klassen gemacht (Vermeidung von Umlauten). Kommentare und sonstiger Code sind aber ausschliesslich für deutschsprachige Regionen geschrieben.

8.19 Skalierung

In Punkt Skalierung unterscheiden wir zwischen vertikaler und horizontaler Skalierung. Bei der vertikalen Skalierung (scale up) denken wir mit dem Aufrüsten von vorhandenen Ressourcen. Eine horizontale Skalierung (scale out) ist im Umfang dieses Projekts nicht beabsichtigt.

8.20 Hochverfügbarkeit & Desaster-Recovery

Webserver und Datenbank sollen außer in Wartungszeiten für 24 Stunden verfügbar sein. Die Verantwortung der Hochverfügbarkeit liegt aber beim Kunden. Wie in grossen Systemen üblich, soll regelmäßig ein Backup der Datenbank erstellt werden. Dies wird aber nicht durch das Projektteam zur Verfügung gestellt, womit der Kunde dies selbst übernehmen muss. Auf Wunsch kann aber eine solche Funktion implementiert werden.

8.21 Migration

Da keine vorhergehenden DB oder Altsysteme bekannt sind müssen die Daten vom Backoffice migriert werden. Sie müssen die Mitarbeiter, die Liegenschaften und die Personen in die DB selber eintippen. Falls vorhanden und dies gewünscht können die Daten aus alten Skripts oder Tabellen eingelesen werden.

8.22 Testbarkeit

Die Testbarkeit wird durch umfängliche Tests mit J-Unit 4 gewährleistet. Die Tests sind in einzelnen Tests Suits zusammengefasst für einzelne Bausteine und so einfach zu testen. Für weitere Informationen siehe Testklassen in [Kapitel 18 \(TDD und JUnit\)](#).

8.23 Codegenerierung

Für das Projekt wurde einzig SceneBuilder benutzt, welcher die GUI Codes in XML-Sprache generiert. Ansonsten ist sämtlicher Code eigens vom Projektteam programmiert.

8.24 Build-Management

Grundsätzlich werden die Bausteine der Struktur von [Kapitel 7 \(Verteilungsschicht\)](#) entnommen. Jedoch können hier im Build Management folgende Richtlinien festgelegt werden. Die genauen Verteilungsartefakte fürs Deployment sind nochmal im [Kapitel Deployment](#) beschrieben.

	Quellcode	Konfigurationsdateien (properties)	Testfälle /-daten	Build Skripte	Verteilungsartefakte
Client Intern	vorhanden	vorhanden			<ul style="list-style-type: none"> - Feukora-gruppeB-appCI.jar - Run-feukora-gruppeB-appCI.bat - Lib Folder
Client Extern	vorhanden	vorhanden			<ul style="list-style-type: none"> - Feukora-gruppeB-appCE.jar - Run-feukora-gruppeB-appCE.bat - Lib Folder
Webservice	vorhanden	vorhanden	vorhanden	vorhanden (Tomcat)	<ul style="list-style-type: none"> - Run-feukora-gruppeB-webService.bat - Feukora-gruppeB-webService.jar - Lib Folder
RMI	vorhanden	vorhanden			<ul style="list-style-type: none"> - Run-feukora-gruppeB-applikationsserver-bat (rmi)
Modell-klassen	vorhanden				<ul style="list-style-type: none"> - Run-feukora-gruppeB-init.bat - Lib Folder
Business-Logik	vorhanden		vorhanden		
Persist & Datenbank	vorhanden		vorhanden	vorhanden (dbSQL)	

9 Entwurfsentscheidungen

Viele Entwurfsentscheide wurden wie es sich gehört in der Software Architektur und Planungsphase gemacht. Vieles wurde jedoch auch aus der Aufgabenstellung vorgegeben. Somit beschränkte sich der Spielraum auf ein Minimum. Gestartet mit diesem, sind dann während dem Projekt noch einige Entscheidungen hinzugekommen zu unserem Applikationsentwurf einige aus gewollten Änderungen, andere durch Kompromisse um ein vollständiges Deployment machen zu können am Abgabetermin.

ENTSCHEIDUNG	Datum und Entscheider	Gründe, Konsequenzen, Alternativen
JPA anstatt von JDBC	Gruppe 25.04.16	<p>Grund: Wir müssen die SQL Statements nicht mehr selber programmieren, was uns effizienter macht.</p>
Klassendiagramm mit einzelnen Manager Klassen und nichtzusammengefasst	Gruppe 02.05.16	<p>Grund: Mit dem vorhandenen Wissen war es die vielversprechendste Lösung und vor allem entstehen so nicht ellenlange Files.</p> <p>Alternative: Zusammenfassen der Businesslogik für Client Intern, Client Extern und Basisfunktionen welche beide benötigen (=3 Interfaces).</p>
Keine Entity Adresse	Olivia Wassmer 12.05.16	<p>Grund: Die Zusammenstellung in Betracht der Datenbankstruktur gestaltet sich ohne Adresse viel einfacher. Zusammen mit</p>

		<p>unseren geringen Kenntnissen sprach dies für eine Entwurfsänderung.</p> <p>Konsequenz: Attribut Strasse zur jeweiligen Person hinzugefügt, und nur Ort inkl. PLZ als eigene Entität</p>
Druckersystem wird nicht implementiert	Alexandra Lengen 12.05.16	<p>Grund: Einen Druckertreiber direkt einzubinden übersteigt unsere momentanen Fähigkeiten.</p> <p>Konsequenz: Es wird ein PrintScreen gespeichert auf dem lokalen Computer welcher weiterbenutzt werden kann.</p>
Keine Annotation CASCADE	Olivia Wassmer 18.05.16	<p>Grund: Mit der Annotation CASCADE funktionieren unsere Tests nicht mehr trotz korrektem Code auch nach Experten-Meinung.</p> <p>Konsequenz: Um trotzdem den Projekt-Abgabetermin einzuhalten haben wir nicht viel Zeit verschwendet und direkt die Alternative ohne CASCADE ausgearbeitet.</p>
Einregulierungs-Checkbox in Auftrag Entity	Pascal Steiner, Olivia Wassmer 21.05.16	<p>Grund: Im Auftrag wird zentral entschieden ob eine Einregulierung möglich ist oder nicht. Darum macht es keinen Sinn diese Attribute wie zuerst geplant in der Messungs-Entity zu belassen.</p>
minimales Admin-Benutzer-Interfaces	Pascal Steiner, Alexandra Lengen 24.05.16	<p>Grund: Die GUI's sind simpel, eine Komplettversion des GUI für den Administrator hätte es nur unnötig kompliziert.</p> <p>Konsequenz: Ein Administrator-Login wird mit dem Deployment übergeben. Mit kleinen Programmierkenntnissen und Datenbankwissen kann der Administrator einfach im Backend alles tun was er muss. Für seine Hauptaktivität nämlich Benutzererstellung steht ein minimales GUI zur Verfügung welches auf dem Baustein Client Intern läuft.</p>
Vorerstes Ausklammern des Grenzwerte-Checks	Pascal Steiner, Olivia Wassmer 25.05.16	<p>Grund: Die Abgabe des Projektes naht um eine funktionierende Lösung deployen zu können muss das Problem zu seite geschoben werden.</p> <p>Konsequenz: Die Werte müssen manuell geprüft werden bis in einem folgenden Release die Check-Methode behoben werden konnte. Unterdessen senkt auch dieser Kompromiss die Erfüllung unseres Qualitätsziels Usability.</p>

Natürlich wurde während der Arbeit eine Vielzahl von Entscheidungen getroffen, die notierten hatten jedoch am meisten Impact auf das Projekt.

10 Qualitätsszenarien

In diesem Kapitel wird näher auf die Qualitätsziele eingegangen, welche zu Beginn im [Kapitel 1.3 \(Stakeholder\)](#) erwähnt sind.

10.1 Qualitätsbaum

Der folgende Qualitätsbaum visualisiert die Zuordnung der einzelnen Qualitätsziele im Detail.

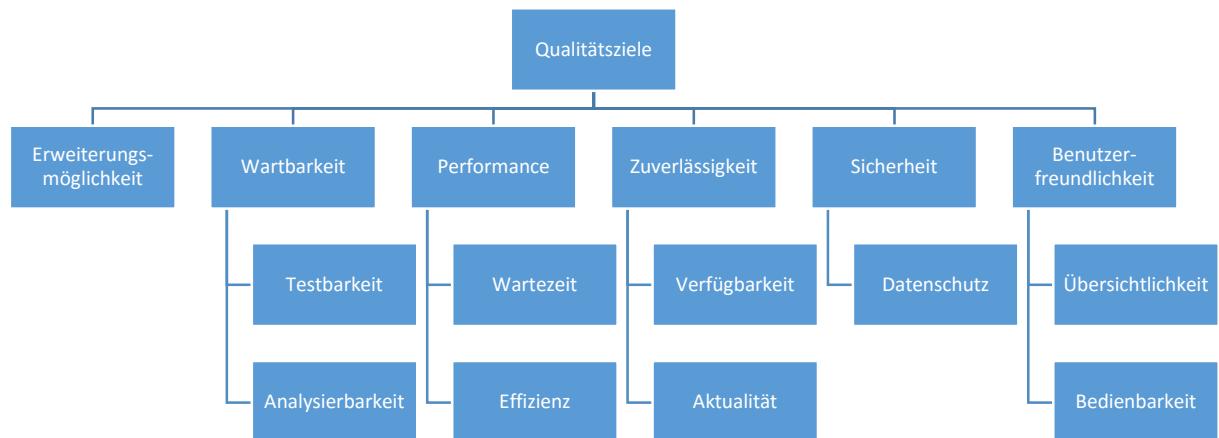


Abbildung 17 - Qualitätsbaum Feukora

Fokus liegt auf den Bereichen Zuverlässigkeit, Benutzerfreundlichkeit, Performance und Sicherheit. Doch wie im Qualitätsbaum festgelegt, geht es in diesen Bereichen noch weiter.

10.2 Bewertungsszenarien

10.2.1 Erweiterungsmöglichkeiten:

Sehr wahrscheinlich ist, dass die Software in ein paar Jahren nicht mehr dem Standard der Zeit entspricht. Jegliche Features, welche die Verbesserung der Qualitätsziele betreffen, sollten integriert werden können.

10.2.2 Wartbarkeit:

Wenn Fehler an der Software entdeckt werden nachdem diese veröffentlicht wurde, muss man Änderungen vornehmen können, damit diese behoben werden. Die Änderungen werden von einem Entwickler der lizenzierten Unternehmung sichergestellt.

10.2.3 Testbarkeit:

Der Softwareentwickler will die Systemkomponenten auf ihre Funktionalität prüfen. Der Entwickler gehört zur lizenzierten Unternehmung. Dieser Test wird auf einem Testsystem durchgeführt. Als Hilfe dient ihm den im arc42 enthaltenen TDD mit JUnit Test.

10.2.4 Analysierbarkeit:

Ein Interessent, welcher über Grundkenntnisse in Softwareentwicklung verfügt, möchte sich über das System informieren. Mithilfe der arc42-Dokumentation kann er ohne fremde Hilfe seine Fragen klären.

10.2.5 Wartezeit/Effizienz:

Der Endbenutzer der Software sollte ohne längere Ladezeiten oder Ladehemmungen die Funktionen des Programms zur Verfügung haben. Sei dies bei der Betätigung der Log-in Schaltfläche oder beim Einlesen von Messwerten. Jeglicher Zugriff soll flüssig und ohne Unterbruch sichergestellt werden.

10.2.6 Verfügbarkeit:

Sofern der Feuerungskontrolleur eine Internetverbindung hat, kann er von überall aus auf die Applikation und dementsprechend seine persönlichen Daten zugreifen. Dieser Zugriff wird per Log-in betätigt.

10.2.7 Aktualität:

Die Daten und die Termine die der Feuerungskontrolleur angezeigt erhält sind immer die aktuellen Daten. Falls der Sachbearbeiter etwas ändert, wird der Feuerungskontrolleur bei erneutem Abfragen der Daten direkt die aktuellen Daten erhalten.

10.2.8 Datenschutz (Sicherheit):

Der Zugriff auf die Datenbank ist für Aussenstehende nicht erlaubt. Jeder Mitarbeiter erhält sein persönliches Log-in.

10.2.9 Übersichtlichkeit/Bedienbarkeit:

Der Benutzer soll sich schnell mit der Software vertraut und sicher fühlen. Er weiss nach geringer Zeit wie das Programm funktioniert und ist auf keine Hilfe angewiesen. Das Menü ist möglichst einfach zu halten. Je nach Rolle des Benutzers erhält er mehr oder weniger Zugriff.

11 Risiken

Nachstehend eine Liste der Risiken, welche während dem Projektbeginn oder zu einem späteren Zeitpunkt aufgetaucht sind. Für die Eintrittswahrscheinlichkeit und die Schadenshöhe wird in drei folgende Kategorien unterschieden.

1 gering

2 mittel

3 hoch

Risikobeschreibung	Eintrittswahr-scheinlichkeit	Schadenshöhe	Risikominderung
Entwicklung falscher Funktionalität und Eigenschaften	hoch	mittel	Laufendes Überprüfen von Projektzielen. Klares Ausformulieren der bestehenden Ziele.
Entwicklung falscher Benutzeroberfläche	gering	hoch	Projektleitung definiert klare Ziele und Anforderungen.
Vergolden von Anforderungen	mittel	mittel	Abspecken von gedachten Ideen, Funktionen und Tools zur Unterstützung der Bedienbarkeit
Ständiges Verändern von Anforderungen	gering	hoch	Prioritäten setzen
Überschätzung der eigenen Fähigkeiten	hoch	hoch	In Teamsitzung mit Gruppe besprechen ob realisierbar ist. Einschätzen von Zeitaufwand.
Unerfahrenheit des Teams bezüglich IT-Projekte.	hoch	hoch	Setzen von Deadlines und priorisieren.
Installationsproblem der Software bei den Kunden	hoch	hoch	Durch einen Testlauf auf neuen (unkonfigurierten) Geräten kann das Deployment simuliert werden.
Inkompatibilität bei nicht Windows Betriebssystemen.	gering	mittel	Alle Entwicklungen (Code, Test, Umgebung) werden auf Windows Plattformen

			entwickelt. Somit besteht kein Koordinationsbedarf
Falscheingaben der künftigen Benutzer. Der Benutzer könnte Zahlenwerte statt String eingeben.	mittel	gering	Wie in Kapitel 8.6 festgehalten, werden Eingaben auf ihre Vollständigkeit überprüft. Ebenfalls werden Zahlen und Buchstabenwerte identifiziert und bei falscher Eingabe visuell angezeigt.
Sicherheit der Applikation. Unerlaubter Zugriff von aussen.	gering	hoch	Durch ein Login wird, wie in Kapitel 8.10 beschrieben, der Zugriff auf die Applikation gesteuert.
Verlust von Daten	mittel	gross	Auf eine Datensicherung wird verzichtet.

12 Glossar

BEGRIFF	ERKLÄRUNG
JPA	Java Persistence API Schnittstelle für Java-Anwendungen, die die Zuordnung und die Übertragung von Objekten zu Datenbankeinträgen vereinfacht.
MIDDLEWARE	Komponente zwischen zwei Anwendungskomponenten, sie erleichtert die Interaktion dazwischen.
PERSISTENZ	Bezeichnet die Fähigkeit, Daten oder logische Verbindungen über lange Zeit bereitzuhalten (über Programmabbruch hinaus).
GUI	Grafical User Interface Ist die Schnittstelle zum Benutzer, also die Oberfläche wo er mit dem System interagiert.

13 Arc42 Dokumentationen und darüber hinaus

13.1 Vorschläge und Anregungen für die Applikation Feukora

Die Implementierung eines komplexen Systems benötigt viele Ressourcen unter anderem Wissen, Geld und Zeit. Die Planung des Projektes verlief wie gewünscht doch wie es meistens der Fall ist, ist auch während diesem Projekt nicht alles glatt gelaufen und darum sind hier Verbesserungsvorschläge, oder Ausbaumöglichkeiten aufgeführt:

THEMA	VORSCHLAG
GRENZWERTE	Momentan werden Grenzwerte manuell vom Feuerungskontrolleur kontrolliert. Die Automatisierung wurde bereits vorbereitet musst jedoch aus Zeitgründen verändert werden (siehe Entwurfsentscheide). Es wäre eine enorme Erleichterung für die Mitarbeiter wenn dies wieder aufgenommen werden würde in einem nächsten Release.
GUI USABILITY	Es wurde darauf geachtet das GUI so übersichtlich wie möglich zu machen. Schönheitsfehler und falsche Tab Sprünge haben sich eingeschlichen. Diese wären mit etwas Aufwand lösbar und würden die Usability steigern. Auch dies sollte in einem kommenden Release beachtet werden.
DATENBANK SPEICHERPROZESS	Durch Cascade.ALL resp. Cascade.Persist könnten gewisse Abläufe im speichern vereinfacht werden. Da zu viele Fehler durch die Caskadierung geworfen wurden wurde diese entfernt. Falls die Datenbank vergrössert werden sollte könnte dies die Performance einschränken und sollte darum in Zukunft beachtet werden.
CODE OPTIMIERUNG	Die schmalen Kenntnisse der Teammitglieder führen leider zu, nicht optimalem Code. Dieser solle bei Gelegenheit optimiert werden solange die Funktionalität nicht beeinträchtigt wird.

13.2 Arc42 Dokumentation

Die Dokumentation ist nach dem Leitfaden von Arc42 erstellt und der Fokus liegt auf den Wichtigsten Bereichen der Dokumentation, so wie sie hier beschrieben sind. Es gibt keine Ergänzenden Arc42 Dokumentationsteile.

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



14 Requirements Dokumentation

Folgende Requirements sind aus der Tabelle “2016_FS_INM21_Requirements_Feukora.xlsx” entnommen.

NR.	ANFORDERUNG	STAKEHOLDER	FUNKTIONALE TESTS
REQ001	Das System muss die FeuerungskontrolleurIn-Daten verwalten können (CRUD)	Backoffice	T010
REQ002	Das System muss die Firmendaten (Name, Adresse, CEO etc.) von Eigentümer- oder Verwalter/Hauswart verwalten können (CRUD)	Backoffice	T003
REQ003	Das System muss die Wärmeerzeuger (Firma, Fabrikat/Typ, Baujahr etc.) aller Kunden verwalten können	Backoffice	T007
REQ004	Das System muss die Brenner (Firma, Fabrikat/Typ, Baujahr etc.) aller Kunden verwalten können	Backoffice	T008
REQ005	Das System muss alle Messergebnisse verwalten können	FeuerungskontrolleurIn	T006
REQ006	Das System muss alle Ergebnisse des Feuerungs-Rapports verwalten können	FeuerungskontrolleurIn	T009
REQ007	Das System muss alle abgemachten Termine mit Kunden verwalten können (CRUD)	FeuerungskontrolleurIn / Backoffice	T004
REQ008	Ein FeuerungskontrolleurIn soll noch vor Ort alle gemessenen Daten in ein GUI eingeben können	FeuerungskontrolleurIn	T009
REQ009	Ein FeuerungskontrolleurIn hat die Möglichkeit noch vor Ort Folgetermine erfassen zu können (CR)	FeuerungskontrolleurIn	T004
REQ010	Nichterfasste Wäremeerzeuger/Brenner müssen vor Ort in der Datenbank erfasst werden können (CRU)	FeuerungskontrolleurIn	T007 / T008
REQ011	Ein FeuerungskontrolleurIn kann seine eigene Terminliste selber ausdrucken (R)	FeuerungskontrolleurIn	T005
REQ012	Das Back-Office kann alle erledigten und nicht erledigten Kontrollen in einer Wochen-Übersichtsliste ausdrucken (R)	Backoffice	T005
REQ013	Der Zugriff auf die Applikation erfolgt über ein Anmelde-Verfahren	Backoffice	T001
REQ014	Die Business-Logik muss in einer separaten Komponente realisiert werden, um Drittssysteme anbinden zu können.	GFK	Keine Funktionale Tests
REQ015	Das System muss den Versand und Empfang des Feuerungsraports über eine externe Verbindung sicherstellen	GFK	T009
REQ016	Durch einen Administrations-Account können Daten erfasst und gepflegt werden (CRUD)	Administrator	T010

15 UseCase Dokumentation

15.1 Login

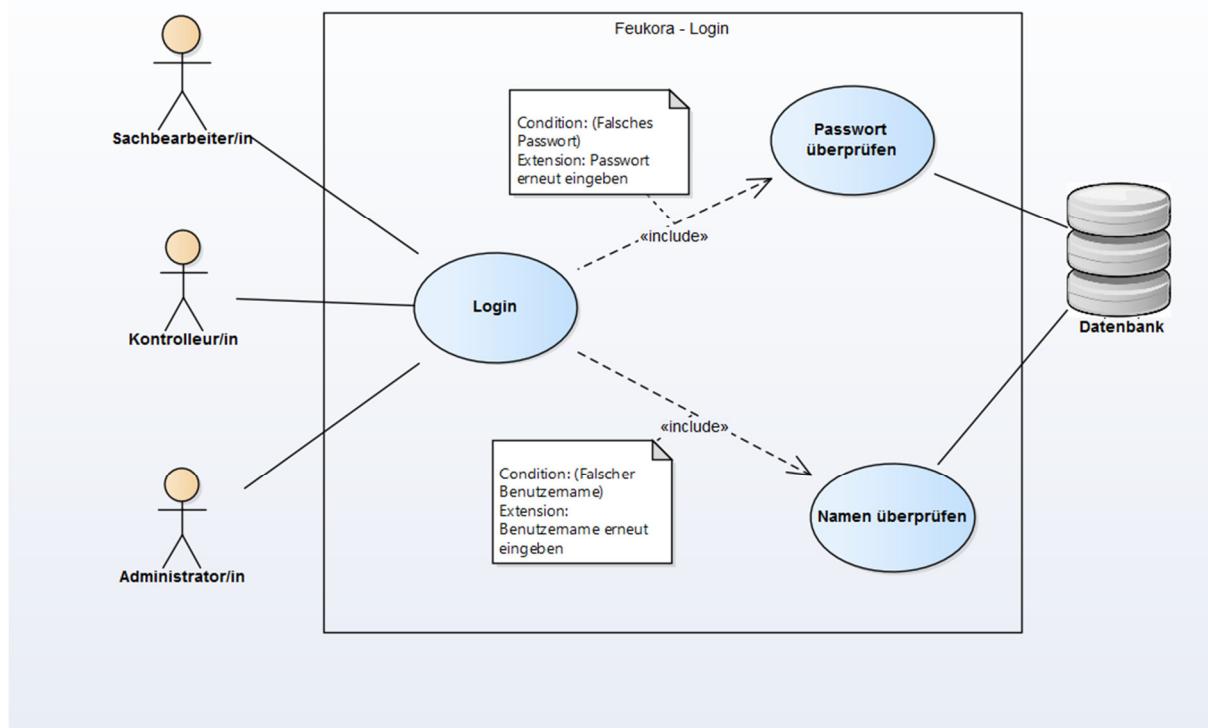


Abbildung 18 - UseCase Login

15.1.1 Benutzernamen eingeben

Use Case:

Benutzernamen eingeben

▪ Use Case-Name	Nr. 01 Benutzernamen überprüfen
▪ Requirement	Req012
▪ UseCase Nummer	UC_01
▪ Kurzbeschrieb	Der eingegebene Benutzername wird überprüft
▪ Akteur	Keiner
▪ Voraussetzungen	Applikation gestartet und funktionstüchtig. User hat einen persönlichen Benutzernamen durch den Administrator zugewiesen bekommen.
▪ Normal-Verlauf	<ol style="list-style-type: none">1. User gibt persönlichen Benutzernamen in vorgesehenes Login-Feld ein.2. Benutzername wird mit der Datenbank abgeglichen
▪ Abschlusskriterien	Benutzername wurde akzeptiert
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Wird der Benutzername abgelehnt, muss der Admin kontaktiert werden.
▪ Szenarien	Keine
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Alexandra Lengen

15.1.2 Login

Use Case:	Login
▪ Use Case-Name	Nr. 02 Login
▪ Requirement	Req012
▪ UseCase Nummer	UC_02
▪ Kurzbeschrieb	User logt sich auf seinem Konto ein
▪ Akteure	Sachbearbeiter / Kontrolleur / Administrator
▪ Voraussetzungen	Applikation gestartet. User hat Benutzernamen und Passwort eingegeben. Diese wurden durch UC_16 und UC_17 überprüft.
▪ Normal-Verlauf	<ol style="list-style-type: none">1. User meldet sich an die Applikation an2. User wird für die Applikation zugelassen3. User kann die Applikation gem. Zugriffsrechte nutzen
▪ Abschlusskriterien	Benutzer erhält Zugriff auf die Applikation
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Wird der Zugriff verweigert, muss der Administrator kontaktiert werden.
▪ Szenarien	Keine
▪ Spezielle Anforderungen	Keine
▪ Autor	Alexandra Lengen

15.1.3 Passwort eingeben

Use Case:	Passwort eingeben
▪ Use Case-Name	Nr. 03 Passwort überprüfen
▪ Requirement	Req012
▪ UseCase Nummer	UC_03
▪ Kurzbeschrieb	Das eingegebene Passwort wird überprüft
▪ Akteur	Keiner
▪ Voraussetzungen	Applikation gestartet und funktionstüchtig. User hat ein persönliches Passwort durch den Administrator zugewiesen bekommen.
▪ Normal-Verlauf	<ol style="list-style-type: none">1. User gibt persönliches Passwort in vorgesehenes Login-Feld ein.2. Passwort wird mit der Datenbank abgeglichen
▪ Abschlusskriterien	Passwort wurde akzeptiert
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Wird das Passwort abgelehnt, muss der Admin kontaktiert werden.
▪ Szenarien	Keine
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Alexandra Lengen

15.2 Liegenschafts- und Eigentümerverwaltung

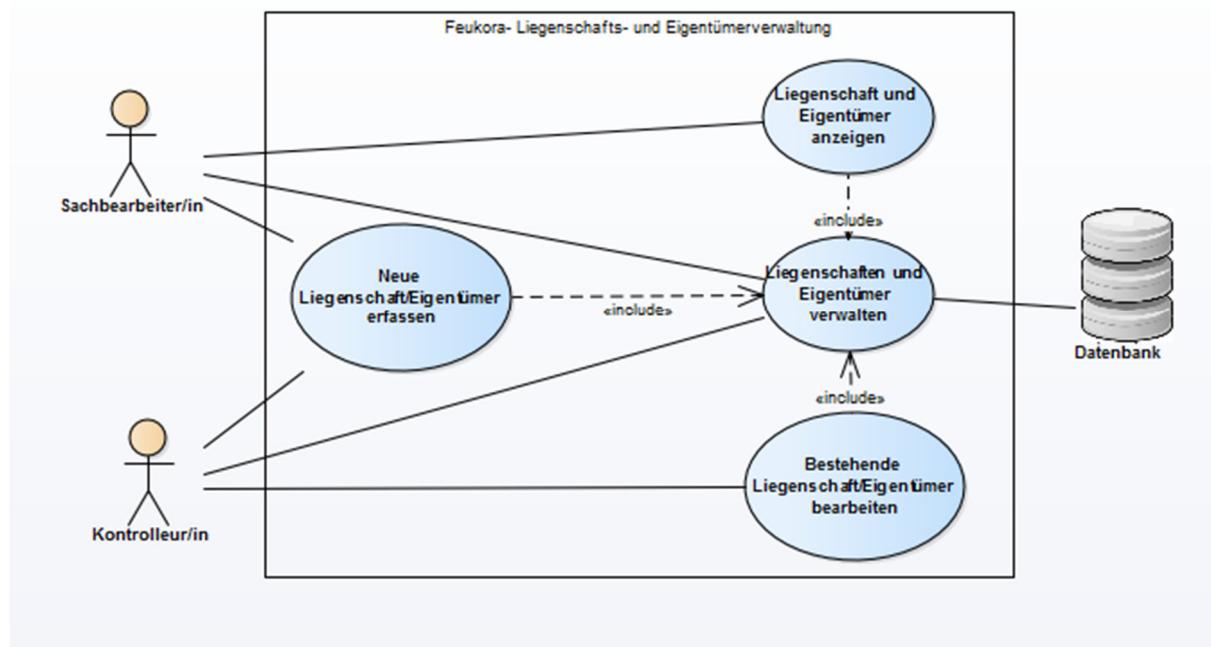


Abbildung 19 - UseCase Liegenschafts- und Eigentümerverwaltung

15.2.1 Liegenschaften und Eigentümer anzeigen

Use Case:	Liegenschaften und Eigentümer anzeigen
▪ Use Case-Name	Nr. 04 Liegenschaften und Eigentümer anzeigen
▪ Requirement	Req002
▪ UseCase Nummer	UC_04
▪ Kurzbeschrieb	Im System werden Liegenschaften und Eigentümer angezeigt.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Applikation funktioniert. Liegenschaften und Eigentümer sind vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Applikation wird gestartet. 2. Liegenschaften und Eigentümer werden im System angezeigt.
▪ Abschlusskriterien	Daten werden erfolgreich im System angezeigt.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	<ul style="list-style-type: none"> - Falls bereits Daten im System angezeigt werden und diese gefiltert oder die Anforderungen geändert werden, zeigt das System die neuen Daten an.
▪ Szenarien	keine
▪ Spezielle Anforderungen	Laufende Verbindung zum Server der Heizungsfirma sowie zur Datenbank.
▪ Autor	Dominik Stirnimann

15.2.2 Liegenschaften und Eigentümer verwalten

Use Case:	Liegenschaften und Eigentümer verwalten
▪ Use Case-Name	Nr. 05 Liegenschaften und Eigentümer verwalten
▪ Requirement	Req002

▪ UseCase Nummer	UC_05
▪ Kurzbeschrieb	Das System verwaltet die bestehenden Liegenschaften und Eigentümer. Durch die Verwaltung können mehrere Aktionen wie Bearbeiten, Löschen, Hinzufügen aufgerufen werden.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Applikation funktioniert. Liegenschaften und Eigentümer sind vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Applikation wird gestartet. 2. Liegenschaften und Eigentümer werden im System angezeigt. 3. Aktionen werden im System angezeigt.
▪ Abschlusskriterien	Daten werden im System angezeigt.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Über Netzzugriff kann der Feuerungskontrolleur auf das System zugreifen und somit die Daten verwalten. - Der Sachbearbeiter greift über das interne Netz auf das System zu und verwaltet so die Daten.
▪ Spezielle Anforderungen	Laufende Verbindung zum Server der Heizungsfirma sowie zur Datenbank.
▪ Autor	Dominik Stirnimann

15.2.3 Neue Liegenschaft/Eigentümer erfassen

Use Case: Neue Liegenschaft/Eigentümer erfassen

▪ Use Case-Name	Nr. 06 Neue Liegenschaft/Eigentümer erfassen
▪ Requirement	Req002
▪ UseCase Nummer	UC_06
▪ Kurzbeschrieb	Im System können neue Liegenschaften und Eigentümer erfasst werden.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Applikation funktioniert. User haben Berechtigung Liegenschaften und Eigentümer zu erfassen. User
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Applikation wird gestartet. 2. Neue Daten werden erfasst.
▪ Abschlusskriterien	Daten können erfolgreich erfasst werden.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	<ul style="list-style-type: none"> - Daten wurden falsch erfasst und müssen anschliessend wieder bearbeitet und berichtigt werden. - Der Feuerungskontrolleur erfasst neue Eigentümer und Liegenschaften vor Ort beim Kunden. Dies kann jedoch nur für weitere Liegenschaften erfolgen. - Der Sachbearbeiter fügt die neuen Daten hinzu wenn dies durch den Kunden telefonisch kommuniziert wird.
▪ Szenarien	
▪ Spezielle Anforderungen	Laufende Verbindung zum Server der Heizungsfirma sowie zur Datenbank.
▪ Autor	Dominik Stirnimann

15.2.4 Bestehende Liegenschaften und Eigentümer bearbeiten

Use Case: Bestehende Liegenschaften und Eigentümer bearbeiten

▪ Use Case-Name	Nr. 07 Bestehende Liegenschaften und Eigentümer bearbeiten
▪ Requirement	Req002
▪ UseCase Nummer	UC_07
▪ Kurzbeschrieb	Im System bestehende Liegenschaften und Eigentümer können bearbeitet werden.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Applikation funktioniert. Liegenschaften und Eigentümer sind vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Applikation wird gestartet. 2. Liegenschaften und Eigentümer werden im System angezeigt. 3. Gewünschter Daten werden gefiltert. 4. Daten werden bearbeitet.
▪ Abschlusskriterien	Daten wurden erfolgreich bearbeitet und abgespeichert.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	<ul style="list-style-type: none"> - Sofern die Daten bereits schon aktualisiert wurden, werden diese so belassen und nicht erneut geändert.
▪ Szenarien	Keine
▪ Spezielle Anforderungen	Bestehende Verbindung zum Server der Heizungsfirma und Datenbank.
▪ Autor	Dominik Stirnimann

15.3 Terminverwaltung

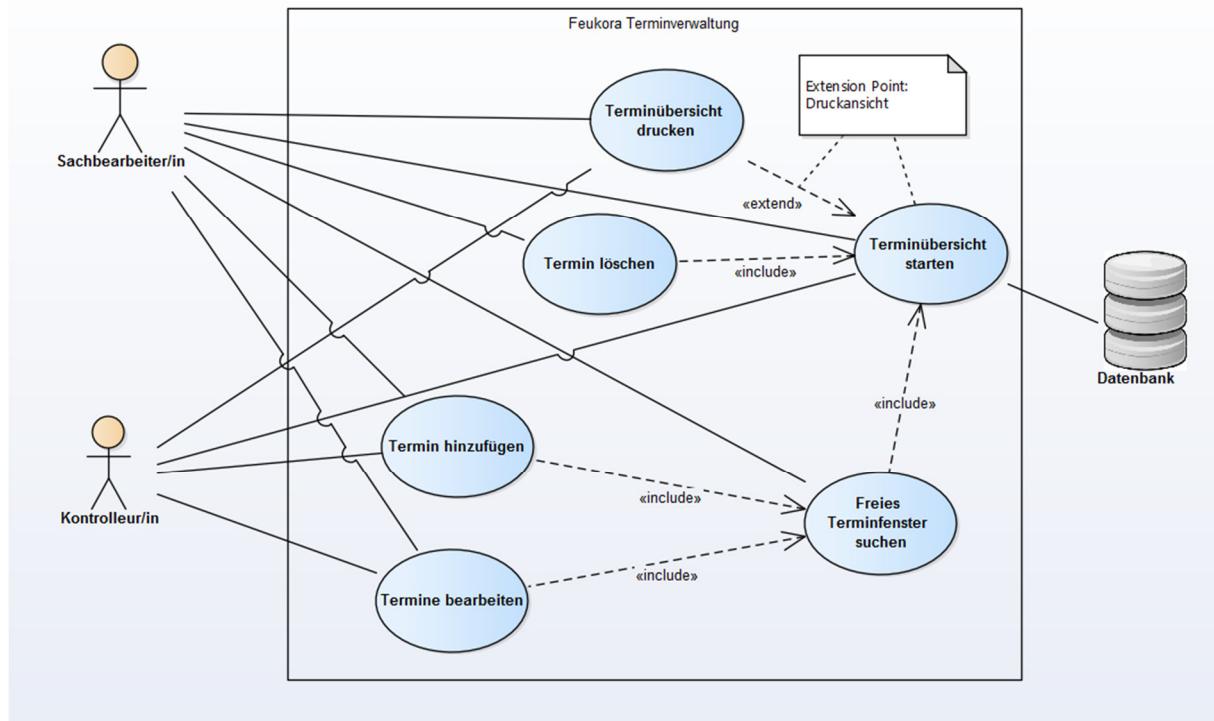


Abbildung 20 - UseCase Terminverwaltung

15.3.1 Termin hinzufügen

▪ Use case-Name	Nr. 08 Termin hinzufügen
▪ Requirement	
▪ UseCase Nummer	UC_08
▪ Kurzbeschrieb	Ein geplanter Termin soll im System hinzugefügt werden können durch den Kontrolleur/Sachbearbeiter.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Terminübersicht muss funktionieren(UC_04). Termin muss noch frei sein.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Terminübersicht wird gestartet. 2. Termin wird in das System hinzugefügt.
▪ Abschlusskriterien	Termin wurde erfolgreich ins System hinzugefügt.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Admin muss kontaktiert werden. Fehler im System.
▪ Szenarien	<ul style="list-style-type: none"> - Sachbearbeiter fügt einen Termin in den Kalender eines Kontrolleurs ein. - Kontrolleur fügt einen Termin in seine eigene Wochenansicht.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Matthias Perrollaz

15.3.2 Termin löschen

▪ Use case-Name	Nr. 09 Termin löschen
▪ Requirement	Req007

▪ UseCase Nummer	UC_09
▪ Kurzbeschrieb	Vorhandene Termine können vom Sachbearbeiter gelöscht werden.
▪ Akteure	Sachbearbeiter
▪ Voraussetzungen	Geplanter Termin muss im System vorhanden sein(UC_01).
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Terminübersicht wird gestartet. 2. Termin wird gelöscht durch Sachbearbeiter.
▪ Abschlusskriterien	Termin wurde erfolgreich aus dem System gelöscht.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Admin muss kontaktiert werden. Fehler im System.
▪ Szenarien	<ul style="list-style-type: none"> - Sachbearbeiter bemerkt einen Termin im Kalender eines Kontrolleurs, der nicht vorhanden sein sollte. Er löscht diesen Termin.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Matthias Perrollaz

15.3.3 Termine bearbeiten

▪ Use case-Name	Nr. 10 Termine bearbeiten
▪ Requirement	Req007
▪ UseCase Nummer	UC_10
▪ Kurzbeschrieb	Ein vorhandener Termin im System soll bearbeitet werden.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Terminübersicht muss funktionieren(UC_04). Termin muss vorhanden sein(UC_01).
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Terminübersicht wird gestartet 2. Termin wird geändert
▪ Abschlusskriterien	Termin wurde im System erfolgreich bearbeitet.
▪ Bemerkungen	-
▪ Alternative-Verläufe	Admin muss kontaktiert werden. Fehler im System.
▪ Szenarien	<ul style="list-style-type: none"> - Sachbearbeiter bemerkt, dass ein Termin in der Wochenansicht nicht korrekt eingegeben wurde. Sachbearbeiter ändert den Termin, damit er nun korrekt eingetragen ist. - Kontrolleur bemerkt, dass er einen Termin falsch eingegeben hat in seine Wochenansicht. Kontrolleur ändert nun diesen Termin.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Matthias Perrollaz

15.3.4 Terminübersicht starten

▪ Use case-Name	Nr. 11 Terminübersicht starten
▪ Requirement	Req007
▪ UseCase Nummer	UC_11
▪ Kurzbeschrieb	Terminübersicht enthält alle vorhandenen Termine und freie Terminfenster. Termine werden auf Wochenbasis dargestellt. Es sind

▪ Akteure	maximal 4 Termine pro Tag pro Kontrolleur möglich. Sachbearbeiter kann Termine bearbeiten, hinzufügen oder löschen.
▪ Voraussetzungen	Sachbearbeiter und Kontrolleur müssen im System erfasst sein. Applikation muss Terminübersicht implementiert haben.
▪ Normal-Verlauf	1. Terminübersicht wird gestartet.
▪ Abschlusskriterien	Applikation mit Terminübersicht ist erfolgreich gestartet.
▪ Bemerkungen	-
▪ Alternative-Verläufe	Admin muss kontaktiert werden. Fehler im System.
▪ Szenarien	- Sachbearbeiter kontrolliert, wann Kontrolleur noch einen freien Termin hat. Sachbearbeiter öffnet die Terminübersicht und sieht die Wochenpläne der Kontrolleure.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Matthias Perrollaz

15.3.5 Terminübersicht drucken

▪ Use case-Name	Nr. 12 Terminübersicht drucken
▪ Requirement	Req010, Req011
▪ UseCase Nummer	UC_12
▪ Kurzbeschrieb	Die Terminübersicht wird für den Kontrolleur in Papierform gedruckt.
▪ Akteure	Sachbearbeiter, Kontrolleur
▪ Voraussetzungen	Termine müssen eingetragen sein(UC_01). Applikation muss Druckfunktion implementiert haben.
▪ Normal-Verlauf	1. Terminübersicht wird gestartet. 2. Via Option Terminübersicht drucken für Kontrolleur.
▪ Abschlusskriterien	Terminübersicht wurde erfolgreich in Papierform ausgedruckt.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Sachbearbeiter muss Terminplan ohne Applikation ausfüllen und drucken.
▪ Szenarien	- Kontrolleur will eine Übersicht über seine Termine die nächste Woche. Sachbearbeiter druckt diese ihm in Papierform aus.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank.
▪ Autor	Matthias Perrollaz

15.3.6 Freies Terminfenster suchen

▪ Use case-Name	Nr. 13 Freies Terminfenster suchen
▪ Requirement	Req007
▪ UseCase Nummer	UC_13
▪ Kurzbeschrieb	Es soll ein Terminfenster gefunden werden, welches noch nicht belegt ist.
▪ Akteure	Sachbearbeiter
▪ Voraussetzungen	Terminübersicht muss erfolgreich starten(UC_04).
▪ Normal-Verlauf	1. Terminübersicht wird geöffnet 2. Freier Termin wird gefunden und mit einem Termin besetzt

▪ Abschlusskriterien	Freies Terminfenster wurde erfolgreich gefunden.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Admin muss kontaktiert werden. Fehler im System.
▪ Szenarien	<ul style="list-style-type: none"> - Sachbearbeiter soll für Kunden einen Termin im Kalender des Kontrolleurs hinzufügen. Sachbearbeiter sucht einen freien Termin in seinem Kalender. Freies Terminfenster wird gefunden und dem Kunden zugewiesen.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Matthias Perrollaz

15.4 Messdatenverwaltung

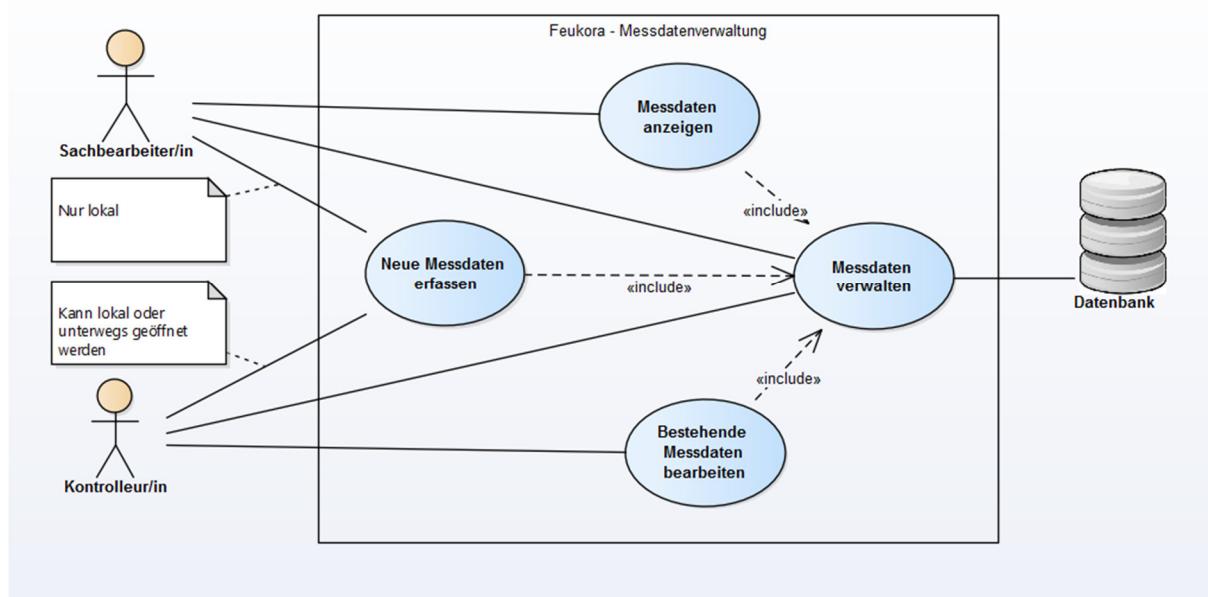


Abbildung 21 - UseCase Messdatenverwaltung

15.4.1 Messdaten anzeigen

▪ Use case-Name	Nr. 14 Messdaten anzeigen
▪ Requirement	Req005, Req008
▪ UseCase Nummer	UC_14
▪ Kurzbeschrieb	Vorhandene Messdaten werden im System korrekt angezeigt.
▪ Akteure	Sachbearbeiter
▪ Voraussetzungen	Messdaten wurden korrekt im System erfasst. Applikation zeigt die Daten korrekt an.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Sachbearbeiter öffnet den Rapport 2. Die Messdaten werden angezeigt
▪ Abschlusskriterien	Messdaten werden korrekt ausgegeben.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Sachbearbeiter muss die Daten aufrufen und überprüfen
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.

▪ Autor	Luca Raneri
---------	-------------

15.4.2 Neue Messdaten erfassen

▪ Use case-Name	Nr. 15 Neue Messdaten erfassen
▪ Requirement	Req005, Req008
▪ UseCase Nummer	UC_15
▪ Kurzbeschrieb	Neue Messdaten werden im System erfasst. Dies kann lokal im Büro oder unterwegs erledigt werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Messdaten sind vorhanden bzw. die Messung hat stattgefunden. Applikation funktioniert.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Messdaten werden vom Kontrolleur abgelesen 2. Messdaten werden im System eingegeben 3. Das System speichert die Daten in die Datenbank ab
▪ Abschlusskriterien	Messdaten wurden erfolgreich im System erfasst und abgespeichert.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerkontrolleur erfasst vor Ort neue Messungswerte in das System. - Die Messdaten werden parallel auf ihre Anforderungen überprüft.
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.
▪ Autor	Luca Raneri

15.4.3 Bestehende Messdaten bearbeiten

▪ Use case-Name	Nr. 16 Bestehende Messdaten bearbeiten
▪ Requirement	Req005, Req008
▪ UseCase Nummer	UC_16
▪ Kurzbeschrieb	Vorhandene Messdaten können im System bearbeitet werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Messdaten wurden zuvor im System erfasst und wurden erfolgreich in der Datenbank abgespeichert. Die Benutzer haben Zugriff auf Daten.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Benutzer öffnet den Rapport 2. Die Messdaten werden angezeigt 3. Die Messdaten werden bearbeitet 4. Änderungen werden gespeichert
▪ Abschlusskriterien	Messdaten werden erfolgreich bearbeitet und abgespeichert.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur bemerkt, dass er bei der Messung falsche Daten eingegeben hat und muss diese updaten. - Der Sachbearbeiter muss noch laufende Änderungen vornehmen.
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.
▪ Autor	Luca Raneri

15.4.4 Bestehende Messdaten verwalten

▪ Use case-Name	Nr. 17 Bestehende Messdaten verwalten
▪ Requirement	Req005
▪ UseCase Nummer	UC_17
▪ Kurzbeschrieb	Die vorhandenen Daten von den Messungen können in der Applikation verwaltet und gespeichert werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Messdaten wurden im System erfasst und sind schon in der Datenbank abgespeichert. Benutzer haben Zugriff auf die Daten.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Benutzer öffnet den Rapport 2. Die Daten werden geöffnet und angezeigt
▪ Abschlusskriterien	Mit der Applikation können die Messdaten verwaltet werden.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur kann die Messdaten über seinen Laptop via externen Netzzugriff verwalten. - Die Sachbearbeiter können die Messdaten via internen Netzzugang auf ihrem Computer verwalten.
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.
▪ Autor	Luca Raneri

15.5 Wärmeerzeugerverwaltung

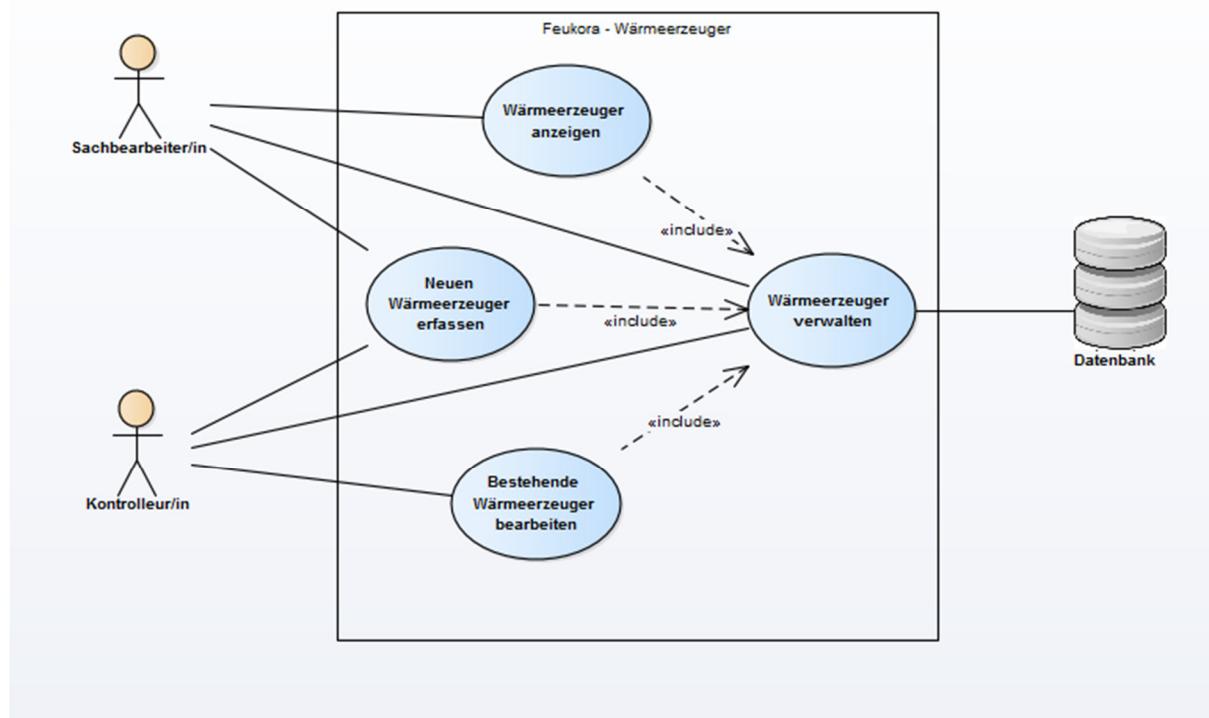


Abbildung 22 - UseCase Wärmeerzeugerverwaltung

15.5.1 Neuen Wärmeerzeuger erfassen

▪ Use case-Name	Nr. 18 Neuen Wärmeerzeuger erfassen
▪ Requirement	Req003, Req009
▪ UseCase Nummer	UC_18
▪ Kurzbeschrieb	Neuer Wärmeerzeuger wird vom User erfasst und im System erfasst.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Wärmeerzeuger sind im System erfassbar.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. User erfasst neuen Wärmeerzeuger 2. Wärmeerzeuger wird im System hinzugefügt 3. Noch nicht bestehender Wärmeerzeuger wird durch das System in die Datenbank gespeichert
▪ Abschlusskriterien	Wärmeerzeuger ist im System erfolgreich erfasst worden.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	<ul style="list-style-type: none"> - Wenn Wärmeerzeuger bereits vorhanden sind, werden diese nicht mehr hinzugefügt.
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur erfasst einen neuen Wärmeerzeuger wenn er vor Ort beim Kunden während der Kontrolle/Service merkt, dass dieser noch nicht erfasst ist. - Der Sachbearbeiter fügt einen neuen Wärmeerzeuger hinzu wenn dies durch den Kunden telefonisch kommuniziert wird. - Der Sachbearbeiter erfasst ebenfalls einen neuen Wärmeerzeuger wenn ein neues Heizungsmodell in der Firma angeschafft wurde welches später verkauft wird.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank und Verbindung zum Server der Heizfirma
▪ Autor	Olivia Wassmer

15.5.2 Wärmeerzeuger anzeigen

▪ Use case-Name	Nr. 19 Wärmeerzeuger anzeigen
▪ Requirement	Req003, Req009
▪ UseCase Nummer	UC_19
▪ Kurzbeschrieb	Wärmeerzeuger können im System angezeigt werden
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Wärmeerzeuger sind in der Datenbank vorhanden und durch die Applikation erreichbar.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Wärmeerzeuger sind im System vorhanden. 2. User tätigt eine Suchabfrage nach Wärmeerzeugern 3. Wärmeerzeuger wird angezeigt.
▪ Abschlusskriterien	Suchergebnisse werden angezeigt
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	<ul style="list-style-type: none"> - Wärmeerzeuger müssen erfasst werden, da keine Wärmeerzeuger vorhanden sind im System.
▪ Szenarien	<ul style="list-style-type: none"> - Feuerungskontrolleur ist vor Ort beim Kunden und sammelt/überprüft alle Daten über den Wärmeerzeuger und lässt

	<p>sich den Wärmeerzeuger des einzelnen Kunden anzeigen um die Angaben zu überprüfen</p> <ul style="list-style-type: none"> - Sachbearbeiter macht eine Systemabfrage aller erfassten Wärmeerzeuger
▪ Spezielle Anforderungen	Laufende Verbindung des Clients zum Server der Heizfirma sowie Zugriff auf die Datenbank
▪ Autor	Olivia Wassmer

15.5.3 Wärmeerzeuger verwalten

▪ Use case-Name	Nr. 20 Wärmeerzeuger verwalten
▪ Requirement	Req003, Req009
▪ UseCase Nummer	UC_20
▪ Kurzbeschrieb	Alle Wärmeerzeuger können im System verwaltet werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Wärmeerzeuger sind im System vorhanden oder werden eingefügt.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Wärmeerzeuger werden in der Datenbank des Systems gespeichert. 2. Sachbearbeiter wie auch Kontrolleure können sie anzeigen lassen
▪ Abschlusskriterien	Keine
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur kann die Wärmeerzeuger über seinen Laptop via externem Netzzugriff verwalten - Die Sachbearbeiter können die Wärmeerzeuger via ihrem internen Netzzugang via PC verwalten
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Olivia Wassmer

15.5.4 Bestehende Wärmeerzeuger bearbeiten

▪ Use case-Name	Nr. 21 Bestehende Wärmeerzeuger bearbeiten
▪ Requirement	Req003, Req009
▪ UseCase Nummer	UC_21
▪ Kurzbeschrieb	Wärmeerzeuger kann im System bearbeitet werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Wärmeerzeuger sind im System vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Wärmeerzeuger wird im System angezeigt. 2. Wärmeerzeuger wird bearbeitet 3. Upgedateter Wärmeerzeuger wird abgespeichert.
▪ Abschlusskriterien	Bearbeiteter Wärmeerzeuger wurde erfolgreich im System abgespeichert.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	<ul style="list-style-type: none"> - Falls Wärmeerzeuger noch nicht erfasst wurde, wird dieser neu im System erfasst - Falls Wärmeerzeuger bereits von einem anderen User angepasst wurde wird dieser so belassen

■ Szenarien	<ul style="list-style-type: none"> - Feuerungskontrolleur bemerkt einen neuen Wärmeerzeuger beim Kunden, er überprüft diesen mit den Kundendaten und sieht, dass es tatsächlich eine Änderung gab. Diese Änderung des Wärmeerzeugers passt er schliesslich im System an - Sachbearbeiter erhält per Telefon die Information vom Kunden, dass sich der Wärmeerzeuger der Liegenschaft geändert hat. Dieser wird dann im System angepasst
■ Spezielle Anforderungen	Laufende Verbindung zum Server der Heizfirma sowie zur Datenbank
■ Autor	Olivia Wassmer

15.6 Brennerverwaltung

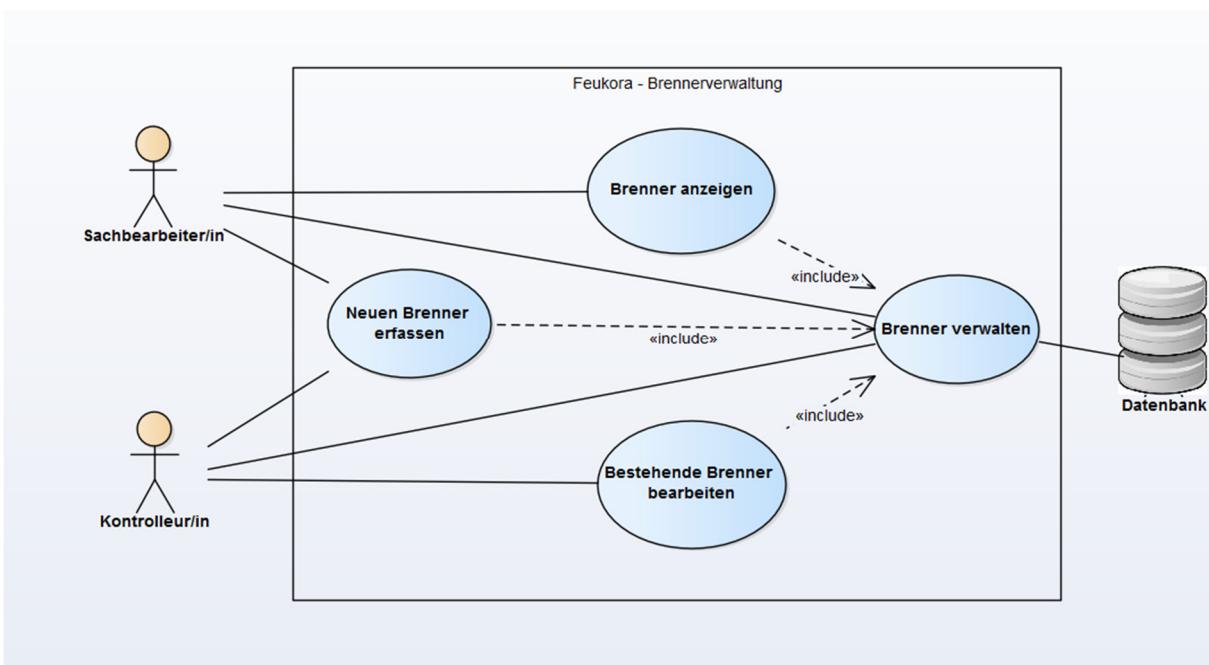


Abbildung 23 - UseCase Brennerverwaltung

15.6.1 Brenner anzeigen

▪ Use case-Name	Nr. 22 Brenner anzeigen
▪ Requirement	Req004, Req009
▪ UseCase Nummer	UC_22
▪ Kurzbeschrieb	Brenner wird im System angezeigt.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Brenner sind im System vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Brenner sind im System vorhanden. 2. Brenner wird durch User gesucht 3. Brenner wird angezeigt.
▪ Abschlusskriterien	Gesuchter Brenner wird richtig angezeigt.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Brenner müssen erfasst werden, da keine Brenner vorhanden sind im System.
▪ Szenarien	<ul style="list-style-type: none"> - Feuerungskontrolleur ist vor Ort beim Kunden und sammelt/überprüft alle Daten über den Brenner und lässt sich den Brenner des einzelnen Kunden anzeigen um die Angaben zu überprüfen - Sachbearbeiter macht eine Systemabfrage aller erfassten Brenner
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Pascal Steiner

15.6.2 Brenner verwalten

▪ Use case-Name	Nr. 23 Brenner verwalten
▪ Requirement	Req004, Req009

▪ UseCase Nummer	UC_23
▪ Kurzbeschrieb	Brenner kann im System bearbeitet werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Brenner sind im System vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Brenner wird im System gesucht und angezeigt. 2. Brenner wird bearbeitet und wieder abgespeichert.
▪ Abschlusskriterien	Bearbeiteter Brenner wurde erfolgreich im System abgespeichert.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Falls Brenner noch nicht erfasst wurde soll eine Fehlermeldung angezeigt werden.
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur kann die Brenner über seinen Laptop via externem Netzzugriff verwalten - Die Sachbearbeiter können die Brenner via PC verwalten
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Pascal Steiner

15.6.3 Neuen Brenner erfassen

▪ Use case-Name	Nr. 24 Neuen Brenner erfassen
▪ Requirement	Req004, Req009
▪ UseCase Nummer	UC_24
▪ Kurzbeschrieb	Neuer Brenner wird vom User erfasst und im System erfasst.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Brenner sind im System erfassbar.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. User erfasst Brenner 2. Brenner wird im System hinzugefügt 3. Noch nicht bestehender Brenner wird durch das System in die Datenbank gespeichert
▪ Abschlusskriterien	Brenner ist im System erfolgreich erfasst worden.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Wenn Brenner bereits vorhanden, wird dieser nicht mehr hinzugefügt.
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur erfasst vor Ort einen neuen Brenner falls dieser noch nicht erfasst ist. - Der Sachbearbeiter fügt einen neuen Brenner hinzu, wenn dies durch den Kunden telefonisch kommuniziert wird - Der Sachbearbeiter erfasst ebenfalls einen neuen Brenner, wenn ein neues Heizungsmodell auf dem Markt erscheint.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Pascal Steiner

15.6.4 Bestehende Brenner bearbeiten

▪ Use case-Name	Nr.25 Bestehende Brenner bearbeiten
▪ Requirement	Req004, Req009
▪ UseCase Nummer	UC_25
▪ Kurzbeschrieb	Brenner kann im System bearbeitet werden.

▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Brenner sind im System vorhanden.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Brenner wird im System angezeigt 2. Brenner wird bearbeitet und abgespeichert
▪ Abschlusskriterien	Bearbeiteter Brenner wurde erfolgreich im System abgespeichert.
▪ Bemerkungen	Keine
▪ Alternative-Verläufe	Falls Brenner noch nicht erfasst wurde, wird dieser neu im System erfasst.
▪ Szenarien	<ul style="list-style-type: none"> - Feuerungskontrolleur bemerkt einen neuen Wärmeerzeuger beim Kunden, er überprüft diesen mit den Kundendaten und sieht, dass es tatsächlich eine Änderung gab. Diese Änderung des Wärmeerzeugers passt er schliesslich im System an - Das Backoffice ändert einen Brenner Aufgrund Kundeninformationen.
▪ Spezielle Anforderungen	Laufende Verbindung zur Datenbank
▪ Autor	Pascal Steiner

7. Rapportdatenverwaltung

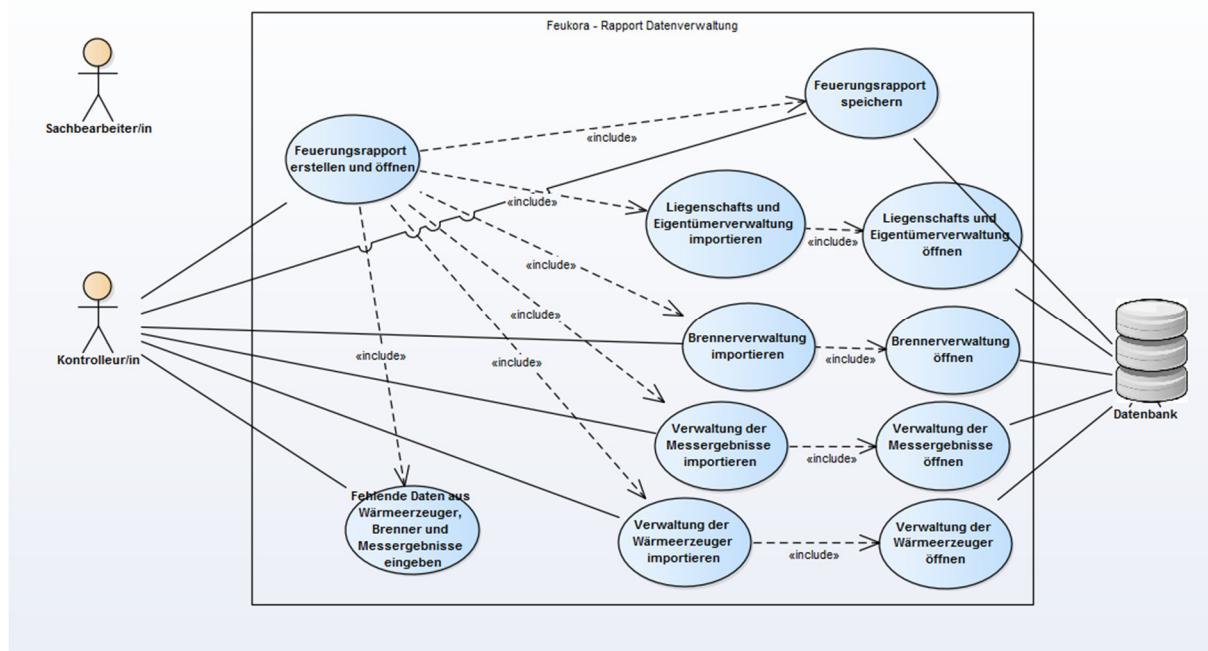


Abbildung 24 - Rapportdatenverwaltung

15.6.5 Neuer Feuerungsrapport erstellen und öffnen

▪ Use case-Name	Nr. 26 Neuer Feuerungsrapport erstellen
▪ Requirement	Req002, Req003, Req004, Req005, Req006, Req008
▪ UseCase Nummer	UC_26
▪ Kurzbeschrieb	Ein neuer Feuerungsrapport wird im System erfasst. Dies kann unterwegs erledigt werden.
▪ Akteure	Kontrolleur

▪ Voraussetzungen	Applikation funktioniert. Ebenfalls muss ein Termin zum Auftrag erstellt sein um ein Feuerungsrapport zu erstellen.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Feuerungsrapport wird vom Kontrolleur erstellt. 2. Feuerungsrapport steht im System bereit.
▪ Abschlusskriterien	Feuerungsrapport wurde erfolgreich im System erfasst und abgespeichert.
▪ Bemerkungen	Die weiteren Used Cases (Brennerverwaltung, Messergebnisse Verwaltung, Wärmeerzeuger Verwaltung und Eigentümer/Liegenschaftsverwaltung werden von den Used Case Kapiteln 15.2, 15.4, 15.5 und 15.6 ergänzt.
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerkontrolleur erfasst vor Ort einen neuen Feuerungsrapport in das System.
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.
▪ Autor	Luca Raneri

15.6.6 Feuerungsrapport bearbeiten

▪ Use case-Name	Nr. 28 Feuerungsrapporte bearbeiten
▪ Requirement	Req002, Req003, Req004, Req005, Req006, Req008
▪ UseCase Nummer	UC_28
▪ Kurzbeschrieb	Vorhandene Feuerungsrapporte können im System bearbeitet werden.
▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Feuerungsrapport wird zuvor im System erfasst und wird erfolgreich in der Datenbank abgespeichert. Die Benutzer haben Zugriff auf die Daten.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Benutzer öffnet den Feuerungsrapport 2. Der Feuerungsrapport wird angezeigt 3. Der Feuerungsrapport wird bearbeitet 4. Die Änderungen werden gespeichert
▪ Abschlusskriterien	Feuerungsrapport wird erfolgreich bearbeitet und abgespeichert.
▪ Bemerkungen	Die weiteren Used Cases (Brennerverwaltung, Messergebnisse Verwaltung, Wärmeerzeuger Verwaltung und Eigentümer/Liegenschaftsverwaltung werden von den Used Case Kapiteln 15.2, 15.4, 15.5 und 15.6 ergänzt.
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur bemerkt, dass im Feuerungsrapport falsche Daten eingegeben wurden und muss diese updaten. - Der Sachbearbeiter muss noch laufende Änderungen vornehmen.
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.
▪ Autor	Luca Raneri

15.6.7 Feuerungsrapport speichern

▪ Use case-Name	Nr. 29 Feuerungsrapport verwalten
▪ Requirement	Req002, Req003, Req004, Req005, Req006, Req008
▪ UseCase Nummer	UC_29
▪ Kurzbeschrieb	Die vorhandenen Feuerungsrapporte können in der Applikation verwaltet werden.

▪ Akteure	Sachbearbeiter / Kontrolleur
▪ Voraussetzungen	Die Feuerungsrapporte wurden im System erfasst und sind schon in der Datenbank abgespeichert. Benutzer haben Zugriff auf die Daten.
▪ Normal-Verlauf	<ol style="list-style-type: none"> 1. Benutzer öffnet den Feuerungsrapport 2. Die Daten werden geöffnet und angezeigt
▪ Abschlusskriterien	Mit der Applikation können die Feuerungsrapporte verwaltet werden.
▪ Bemerkungen	Die weiteren Used Cases (Brennerverwaltung, Messergebnisse Verwaltung, Wärmeerzeuger Verwaltung und Eigentümer/Liegenschaftsverwaltung werden von den Used Case Kapiteln 15.2, 15.4, 15.5 und 15.6 ergänzt.
▪ Alternative-Verläufe	Keine
▪ Szenarien	<ul style="list-style-type: none"> - Der Feuerungskontrolleur kann den Feuerungsrapport über seinen Laptop via externen Netzzugriff verwalten. - Die Sachbearbeiter können den Feuerungsrapport via internen Netzzugang auf ihrem Computer verwalten.
▪ Spezielle Anforderungen	Bestehende Verbindung zur Datenbank.
▪ Autor	Luca Raneri

16 Klassendiagramme

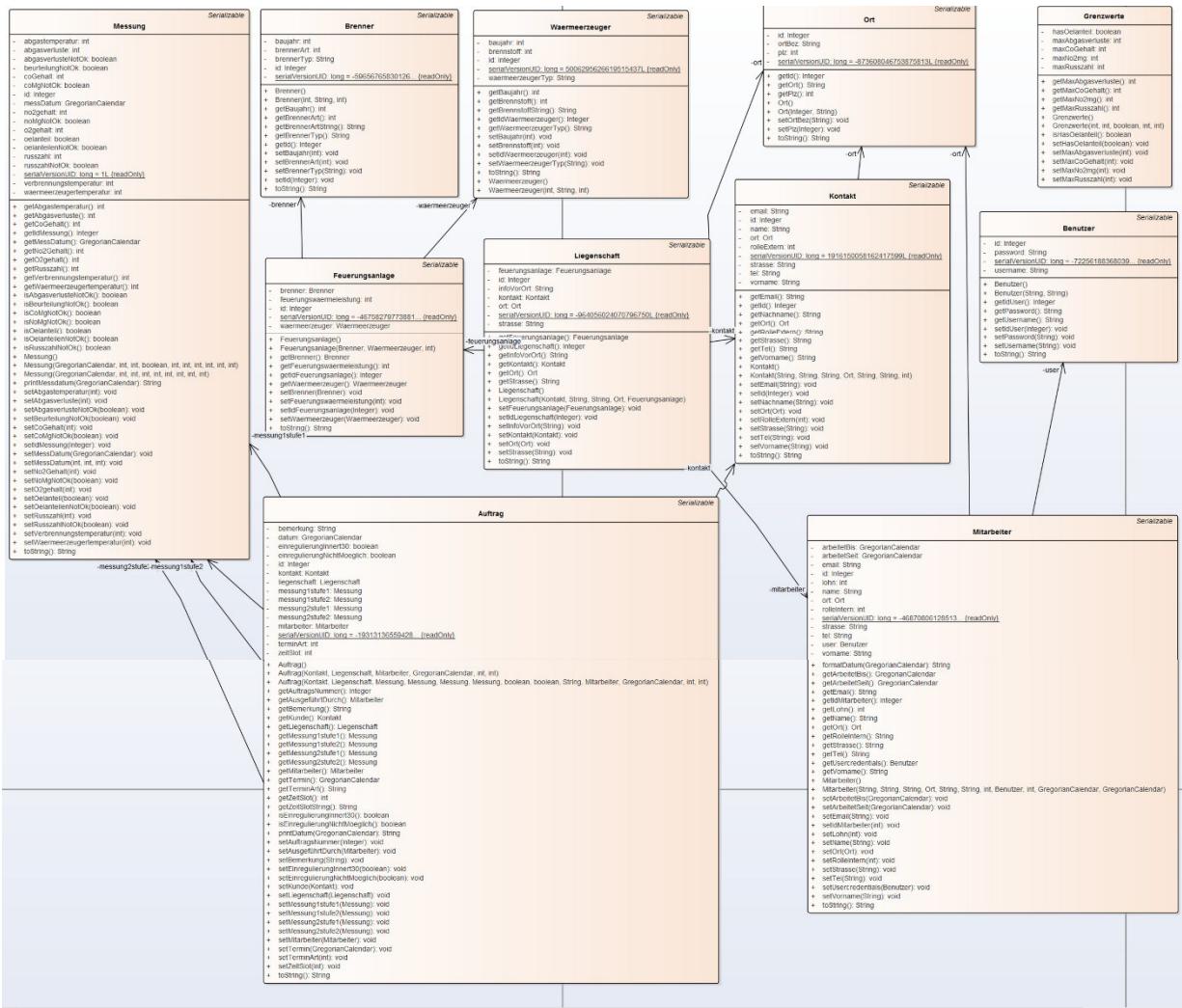


Abbildung 25 - Klassendiagramm Feukora

17 Deployment Informationen

Für das Deployment betreffend benötigen wir vier Komponenten des Feukora Projekts. Dazu gehören der Client Intern, Client Extern sowie der Applikationsserver und Webservice. Zudem wird die vorhandene PostgreSQL Datenbank für die Speicherung benutzt sowie Tomcat 8.x um die Webservice Datei verfügbar zu machen..

Der Client Intern wird von dem Sachbearbeiter im Office benötigt. Der Client Extern wird von dem Feuerungskontrolleur mit seinem Laptop benötigt. Webservice wird mithilfe von Apache Tomcat bereitgestellt um eine externe Verbindung zu ermöglichen. Und schlussendlich die Serverapplikation läuft auf der Serverinfrastruktur der jeweiligen Heizfirma und beinhaltet die Bausteine der Feukora Applikation.

Die gesamte Verteilungssicht kann in [Kapitel 7](#) nachgelesen werden.

17.1 Deployment Dateien

Folgende Ressourcen werden mit dem Deployment geliefert und werden zur Installation der Applikation benötigt:

KOMPONENTE	RESSOURCEN
CLIENT EXTERN	<ul style="list-style-type: none">- run-feukora-gruppeB-appCE.bat- libs Folder:<ul style="list-style-type: none">o feukora-gruppeB-appCE.jaro feukora-gruppeB-modelklassen.jaro log4j-1.2.17.jar
CLIENT INTERN	<ul style="list-style-type: none">- run-feukora-gruppeB-appCI.bat- resources Folder<ul style="list-style-type: none">o clientintern.properties- lib Folder:<ul style="list-style-type: none">o feukora-gruppeB-appCI.jaro feukora-gruppeB-rmi.jaro feukora-gruppeB-modelklassen.jaro log4j-1.2.17.jar
WEBSERVICE SERVER	<ul style="list-style-type: none">- Feukora-gruppeB-webservice.zip- run-feukora-gruppeB-server.bat- run-feukora-gruppeB-dbInit.bat- resources Folder<ul style="list-style-type: none">o rmi.propertieso META-INF<ul style="list-style-type: none">▪ persistence.xml- lib Folder:<ul style="list-style-type: none">o feukora-gruppeB-businesslogik.jaro feukora-gruppeB-persistier.jaro eclipselink.jaro javax.persistence_2.0.4.jaro postgresql-9.1-902.jdbc4.jaro feukora-gruppeB-modelklassen.jaro log4j-1.2.17.jar

17.2 Installationsanleitung

Die folgende Anleitung führt durch die Installation und Konfiguration der Applikation Feukora welche bis zum Start der Applikation führt.

17.2.1 Installation Datenbank «Feukora»

Für den Zugriff auf die Datenbank benutzten wir die Applikation „pgAdmin3“. Es wird erwartet dass die Datenbank und Benutzersoftware bereits auf der Serverinfrastruktur installiert ist.

Falls PGAdmin3 nicht vorhanden ist kann die Software gleich mitinstalliert werden. Jedenfalls muss dann zuerst die gestellte Datenbank darauf konfiguriert werden.

17.2.1.1 Installations-Anleitung

1. Neue Rolle mit dem Namen «feukora-B» erstellen. Das Passwort wird im Tab „Definition“ auf «passwd-B» gesetzt.
2. Die Datenbank muss mit rechtsklick konfiguriert werden. Und der Owner auf die eben hinzugefügte „Login role“ gesetzt werden.

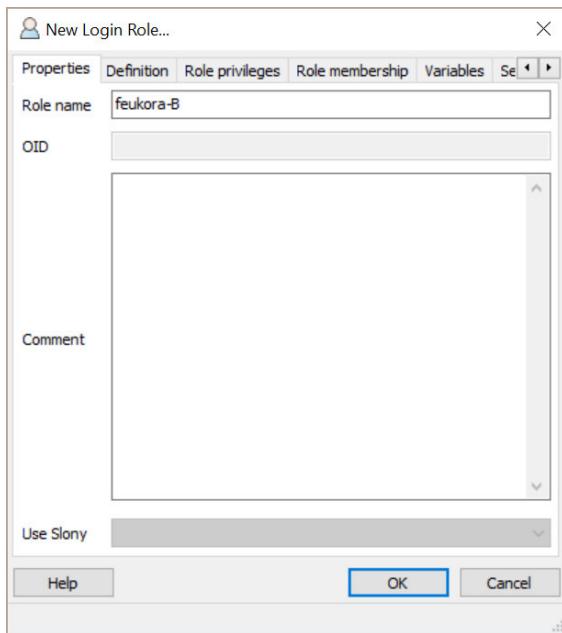


Abbildung 26 - Neue Login Rolle 1

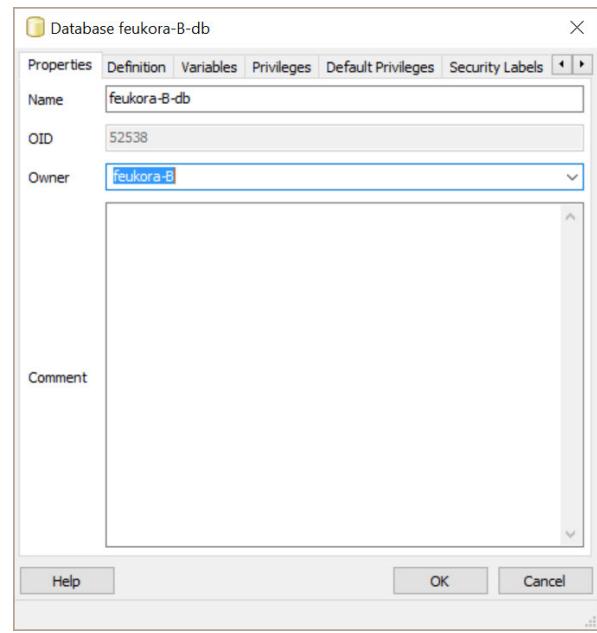


Abbildung 27 - Neue Login Rolle 2

17.2.1.2 Datenbank Initialisierung – Testdaten einlesen

Mit dem Deployment wird auch eine Datenbank Initialisierungs-Datei geliefert. Diese kann via run-feukora-gruppeB-dblInit.bat gestartet werden. Anschliessend ist die Datenbank mit Testwerten befüllt. Alle weiteren Datensätze werden dann über die Applikation hinzugefügt.

Die Struktur der Datenbank wird durch die Architektur mithilfe von EclipseLink aufgebaut und nimmt somit viel Arbeit ab.

17.2.1.3 Restore bzw. Backup der Datenbank

Um ein Backup der vorhandenen Datenbank zu erstellen, geht man wie folgt vor.

Rechtsklick auf die zu sichernende Datenbank und Backup auswählen. Nun kann man den Speicherort, Dateiname und diverse weitere Einstellungen bestimmen.

Um eine Datenbank einzulesen klickt man wieder mit einem Rechtsklick auf die Datenbank und wählt den Punkt Restore. Nun wählt man das gewünschte Backup-File aus.

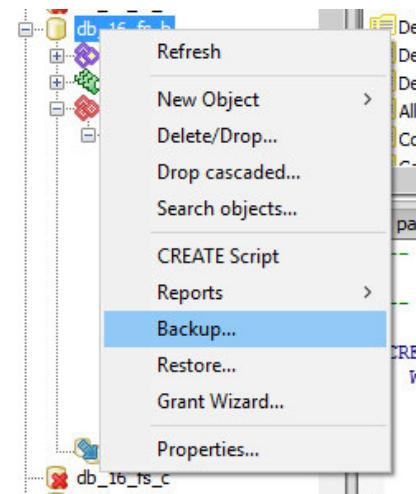


Abbildung 28 - Backup erstellen

17.2.2 Installation und Start Serverapplikation

Der Server kann durch Ausführen der „run-feukora-gruppeB-server.bat“ –Datei gestartet werden. Dabei sollte die Rückmeldung im cmd-Fenster „Objekt beim Namensdienst angemeldet. Server bereit.“ lauten. Das Dateienverzeichnis sollte in der gleichen Struktur wie in den [Deployment Daten](#) beschrieben, kopiert werden.

Falls ein neuer Server oder Client konfiguriert wird sollte ebenfalls die „rmi.properties“–Datei überprüft werden auf Port und Host Informationen. Falls diese noch nicht stimmen, müssen Sie in der Datei angepasst werden und gespeichert werden. Anschliessend kann der Server erfolgreich gestartet werden und mit Clients verbunden werden.

17.2.3 Installation und Start Webservice

Für die Installation des Webservice wird Tomcat 8.x auf der Serverinfrastruktur erwartet. Wenn die Infrastruktur wie angekündigt vorhanden ist kann die Datei „feukora-gruppeB-webservice.zip“ entzippedt werden. Via Eclipse JEE sollte ebenfalls Tomcat als Server bereits konfiguriert resp. installiert sein.

Bevor das Projekt aufgeschaltet wird, sollte überprüft werden ob die property Angaben für den RMI Server noch übereinstimmen (port, ip). Die Property-Datei findet sich im Projektordner „resource“. Wenn ja kann dann In Eclipse das Projekt „feukora-gruppeB-webservice“ dem Tomcat Server hinzugefügt werden durch „Add and Remove“.

Ist dies geschehen soll der Server in Eclipse gestartet werden. Nach dem Start kann weitergefahren werden mit den Clients.

17.2.4 Client Intern starten

Der Interne Client kann sehr komfortabel gestartet werden. Zuerst sollte die Datenbank installiert und initialisiert sein und die Serverapplikation gestartet werden. Anschliessend muss nur noch die „run-feukora-gruppeB-appCI.bat“ Datei ausgeführt werden und der Client startet und bietet das Login-Fenster an um sich direkt anzumelden.

Falls sich Property-Informationen ändern können diese im Ordner „resources“ angepasst werden (rmiport, rmiip).

17.2.5 Client Extern starten

Der Externe Client lässt sich noch komfortabler starten. Zuerst sollte die Datenbank installiert und initialisiert sein und die Serverapplikation gestartet werden. Anschliessend muss nur noch die „run-feukora-gruppeB-appCE.bat“ Datei ausgeführt werden und der Client startet und bietet das Login-Fenster an um sich direkt anzumelden.

Beim Externen Client gibt es keine Propertydateien welche sich ändern. Die Verbindung läuft über das konfigurierte WSDL File und müsste im Sourcecode angepasst werden wenn nötig.

Hiermit sollten sämtliche Bausteine funktionieren und ihre Arbeitsabläufe digitalisieren.¹

¹ Der Client Extern ist Stand 26.05.16 nicht vollständig lauffähig aufgrund eines für das Team unlösbar Defekts. Es wird vermutet, dass eine die Metro Version einen Defekt hat oder eine falsche Version welche nicht upgedatet werden konnte.

18 TDD und JUnit

Mit dem Test-Driven Development ist es möglich die neu geschriebenen Codes während der Realisierung auf ihre Funktionalität zu testen. Dies ermöglicht zusätzlich, dass nur lauffähige Codeabschnitte in die Applikation integriert werden. So werden Fehler frühzeitig erkannt. Die Codeabschnitte bilden eine gute Basis für die zukünftigen testbaren Klassen.

18.1 Erfahrung mit TDD

Die Zeitplanung für unser Projekt hat TDD mit einer Woche eingeplant. Als das Projekt so weit vorangeschritten war, mussten jedoch vorzeitig mit TDD abgebrochen werden. Aufgrund der noch schmalen Informatikkompetenzen in Projektteam, wurden wir gezwungen unsere Tests erst nach der Fertigstellung der Klassen zu realisieren. Test Driven Developpement konnte im Projekt somit nicht integriert werden. Um dies zu kompensieren wurden in den Bausteinen Persister, Businesslogik sowie Webservice umfangreiche Tests eingebaut. Diese JUnit Tests testen sämtliche Funktionen und konnten einwandfrei abgeschlossen werden.

18.2 JUnit for the Win

Um die geschriebenen Klassen auf Ihre Funktionalität zu testen entschieden wir uns für JUnit 4 Testing. JUnit4 gilt zum Best Practice für die Projektgruppe somit fiel der Entscheid nicht schwer. Sämtliche Tests sind in einer Testsuite zusammengefasst um die erneute Überprüfung mit einem einzigen Klick durchzuführen.

Wieso haben wir alle Methoden getestet? Das Defizit welches ohne TDD entstanden ist, musste ausgeglichen werden. Die einzige für das Team realistische Möglichkeit waren umfängliche JUnit Tests, nur so konnte mehr oder weniger sichergestellt werden ob die Methoden ohne TDD richtig implementiert wurden und die Verbindung zwischen den verschiedenen Schichten reibungslos funktioniert.

Natürlich funktioniert im Testing nicht von Beginn an alles reibungslos. Die Arbeitsgruppe ist auf viele Probleme gestossen. Und die Testklassen mussten im Verlauf des Projektes oft angepasst werden. Das 20/80 Prinzip wurde während der Projektarbeit nochmal definitiv untermauert. Das Test-Fixing beanspruchte enorm viel Zeit und ohne professionelle Hilfe wäre die Behebung dieser Fehler noch in weiter Ferne. Zusammenfassen haben sich die JUnit Tests auf jeden Fall bewährt und haben auch sichergestellt dass alle Funktionen funktionieren, bevor der Client auf Errors gestossen ist.

18.3 Testklassen

18.3.1 Persister

Mit unseren Testklassen versuchten wir als erstes das Persistieren in die Datenbank vorzunehmen. Somit konnten wir überprüfen ob die Daten richtig und ohne Probleme in die Datenbank abgelegt oder gelöscht werden. Eine Datenbank, nur für Testzwecke haben wir uns Lokal erstellt.

Klasse	JUnit Testklasse
AuftragDAOImpl	AuftragDAOTest
BenutzerDAOImpl	BenutzerDAOTest
BrennerDAOImpl	BrennerDAOTest
FeuerungsanlageDAOImpl	FeuerungsanlageDAOTest
KontaktDAOImpl	KontaktDAOTest
LiegenschaftDAOImpl	LiegenschaftDAOTest
MessungDAOImpl	MessungDAOTest
MitarbeiterDAOImpl	MitarbeiterDAOTest
OrtDAOImpl	OrtDAOTest
WaermeerzeugerDAOImpl	WaermeerzeugerDAOTest

Die erwähnten Testklassen testen jeweils immer die gleichnamigen Impl Klassen. Die Klasse AuftragDAOImpl wird beispielsweise getestet von der Testklasse AuftragDAOTest und befindet sich auf der Data Access Layer (DAO), welche die Schnittstelle zur Datenbank sicherstellen.

Für die Business-Logik Layer und für den Webservice haben wir das gleiche Prinzip angewendet. Für jede Impl Klasse wurden jeweils alle Methoden getestet welche benötigt werden. Dabei wurde sichergestellt ob die Verbindung mit dem Persister resp. RMI funktioniert.

18.3.2 Business-Logik Layer

Klasse	JUnit Testklasse
AuftragManagerImpl	AuftragManagerTest
BenutzerManagerImpl	BenutzerManagerTest
BrennerManagerImpl	BrennerManagerTest
FeuerungsanlageManagerImpl	FeuerungsanlageManagerTest
KontaktManagerImpl	KontaktManagerTest
LiegenschaftManagerImpl	LiegenschaftManagerTest
MessungManagerImpl	MessungManagerTest
MitarbeiterManagerImpl	MitarbeiterManagerTest
OrtManagerImpl	OrtManagerTest
WaermeerzeugerManagerImpl	WaermeerzeugerManagerTest

18.3.3 Web-Service

Klasse	JUnit Testklasse
FeuerungsrapportServiceImpl	FeuerungsrapportServiceAuftragTest

	FeuerungsrapportServiceBenutzerTest
	FeuerungsrapportServiceBrennerTest
	FeuerungsrapportServiceFeuerungsanlageTest
	FeuerungsrapportServiceKontaktTest
	FeuerungsrapportServiceLiegenschaftTest
	FeuerungsrapportServiceMessungTest
	FeuerungsrapportServiceMitarbeiterTest
	FeuerungsrapportServiceOrtTest
	FeuerungsrapportServiceWaermeerzeugerTest

18.4 Durchführung der JUnit Tests

Unsere JUnit Testklassen sind alle einheitlich aufgebaut. Zu Beginn werden immer die zu testenden Objekte instanziert und die Testwerte in die Datenbank initialisiert. Dann werden alle Methoden getestet.

Abschliessend werden initialisierten Daten wieder von der Datenbank gelöscht.

Folgend noch eine Darstellung wie ein JUnit Test ausgeführt wird:

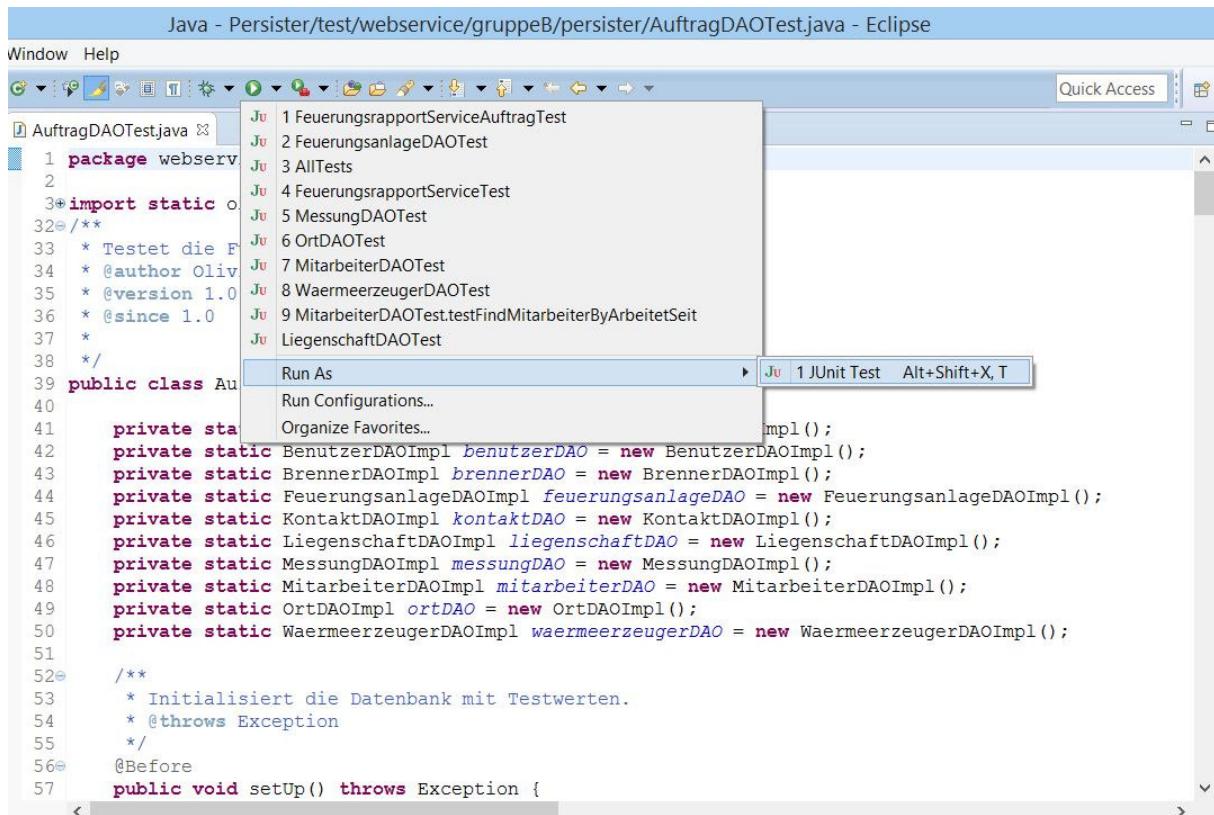


Abbildung 29 - Ausführen eines JUnit Test

19 Funktionale Tests

In diesem Kapitel wird die Feukora-Applikation ausgiebig getestet.

Zu jedem Use Case (siehe Kapitel 15) wurde ein Test Case erstellt.

19.1 Testumgebung

Es wurde auf folgenden Geräten getestet:

Während der Entwicklung: Pascal PC, Alexandra PC

Während der Testphase: Olivia PC

19.2 Testzeitpunkt

Der vollständige Applikationstest fand am 26.05.2016 statt.

Durchführende Testpersonen: Pascal Steiner, Olivia Wassmer und Alexandra Lengen.

19.3 Testergebnis

Die Feukora-Applikation hat alle notwendigen Tests bestanden.

19.4 Testbeschreibungen

Test Case 001: Login

Bewertung: **BESTANDEN**

Vorbedingung	Dem User wurde ein Benutzernamen inkl. Passwort zugeteilt
Testschritte	Applikation starten, Benutzername und Passwort eingeben, Login-Button wählen
Erwartetes Resultat	Zugriff auf das Dashboard
Resultat	Das Resultat war wie erwartet

Test Case 002: Liegenschaftsverwaltung

Bewertung: **BESTANDEN**

Vorbedingung	User (BO) ist eingeloggt und auf dem Dashboard
Testschritte	Liegenschaftsübersicht öffnen, Liegenschaft hinzufügen/bearbeiten/löschen, speichern
Erweitertes Resultat	Mutation auf der Datenbank, Veränderung in der Übersicht
Resultat	Das Resultat war wie erwartet

Test Case 003: Kontaktverwaltung

Bewertung: **BESTANDEN**

Vorbedingung	User (BO) ist eingeloggt und auf dem Dashboard
Testschritte	Kontaktübersicht öffnen, Kontakt hinzufügen/bearbeiten/löschen, speichern
Erweitertes Resultat	Mutation auf der Datenbank, Veränderung in der Übersicht
Resultat	Das Resultat war wie erwartet

Test Case 004: Terminverwaltung

Bewertung: **BESTANDEN**

Vorbedingung	User ist eingeloggt und auf dem Dashboard
Testschritte	Auftragsübersicht öffnen, Auftrag hinzufügen/bearbeiten/löschen, speichern
Erweitertes Resultat	Mutation auf der Datenbank, Veränderung in der Übersicht
Resultat	Das Resultat war wie erwartet

Test Case 005: Terminübersicht drucken		Bewertung: BESTANDEN
Vorbedingung	User ist eingeloggt und auf dem Dashboard	
Testschritte	Terminübersicht öffnen, Eingaben tätigen, drucken-Button wählen, printscreen auf dem Desktop öffnen, Druckauftrag ausführen	
Erweitertes Resultat	Terminübersicht als Hardcopy	
Resultat	Das Resultat war wie erwartet	
Test Case 006: Messdatenverwaltung		Bewertung: BESTANDEN
Die Messdaten sind Bestandteil eines Rapportes. Das anzeigen/hinzufügen/bearbeiten/löschen wurde mit den Test Case rapportdatenverwaltung bereits getestet.		
Test Case 007: Wärmeerzeugerverwaltung		Bewertung: BESTANDEN
Die Daten zum Wärmeerzeuger sind Bestandteil einer Liegenschaft. Das anzeigen/hinzufügen/bearbeiten/löschen wurde mit den Test Case Liegenschaftsverwaltung bereits getestet.		
Test Case 008: Brennerverwaltung		Bewertung: BESTANDEN
Die Daten zum Brenner sind Bestandteil einer Liegenschaft. Das anzeigen/hinzufügen/bearbeiten/löschen wurde mit den Test Case Liegenschaftsverwaltung bereits getestet.		
Test Case 009: Rapportdatenverwaltung		Bewertung: BESTANDEN
Vorbedingung	User ist eingeloggt und auf dem Dashboard	
Testschritte	Auftragsübersicht öffnen, Auftrag anwählen, Rapport erfassen/bearbeiten, speichern	
Erweitertes Resultat	Mutation auf der Datenbank, Veränderung in der Übersicht	
Resultat	Das Resultat war wie erwartet	
Test Case 010: Rapportdatenverwaltung		Bewertung: BESTANDEN
Vorbedingung	User ist eingeloggt und auf dem Dashboard	
Testschritte	Mitarbeiterübersicht öffnen, Mitarbeiter erfassen, bearbeiten, löschen, speichern	
Erweitertes Resultat	Mutation auf der Datenbank, Veränderung in der Übersicht	
Resultat	Das Resultat war wie erwartet	

20 Datenbank Dokumentation

20.1 ERD/Attributliste

Das Projekt wurde unter anderem mit der Erstellung der Datenbank begonnen. Bereits in der ersten Sitzung wurde eine erste Skizze als ERD und eine Attributliste dazu erstellt ([siehe Anhang](#)). Später während des Projekts haben wir dann bemerkt, dass unserer Überlegungen teils nicht korrekt waren, bzw. dass wir diese programmiertechnisch so nicht umsetzen konnten. Das ERD und die dazugehörige Attributliste wurde somit laufend mehrmals angepasst.

Folgend das aktuelle ERD nach der Projekt fertigstellung, welches mit MS Visio erstellt wurde:



Abbildung 30 - ERD Feukora

20.2 Datenbank-Server Informationen

Die Datenbank selber läuft unter dem Database Namen «feukora-B-db» und ist durch den Auftraggeber gestellt. Beim Deployment wird die Software bereits auf dem Server erwartet um die Datenbank konfigurieren zu können.

Gruppe B

Datenbank: feukora-B-db

User: feukora-B

Passwort: passwd-B

20.3 Test-Datenbank

In der Testing-Phase haben wir im Projekt mit einer Test-Datenbank gearbeitet. Die Datenbank wurde mit dem «pgAdmin III» erstellt. Die Datensätze und Tabellen, welche in der Datenbank initialisiert wurden, wurden in den Testklassen mit einer selbstgeschriebenen init-Methode realisiert.

21 Projektmitglieder Beiträge

21.1 Alexandra Lengen

21.1.1 Individuelles Management Summary zum Projekt Feukora

Im Projekt Feukora war ich für das Requirements Engineering und das GUI verantwortlich. Mein Hauptfokus war aber das Programmieren. Zum einen weil es mir viel Spass bereitet und das persönliche Lernpotential enorm gross war zum anderen aber auch weil unsere Gruppe leider kein Gruppenmitglied hat, welches bereits vertiefte Erfahrungen im Programmieren mit sich bringt. Die Requirement-Rolle hat zu anfangs etwas Zeit beansprucht, war aber auf das ganze Projekt gesehen minimal, zumal die Beschreibung der UseCases durch alle Mitglieder gleichermassen erfolgt ist. Die Rollen im Team haben sich schnell durchmischt und ich habe mich zu Beginn des Projektes stärker mit der konzeptionellen Vorgehensweise (Bausteinsicht, Laufzeitsicht, Klassendiagramm) befasst. Diese Erarbeitung erfolgte mit Olivia.

Eine folgenschwere Entscheidung wurde vor Beginn des Programmierens gefällt. Unerfahren wie wir waren, haben wir den einzelnen Gruppenmitglieder Programmier-Felder zugeordnet (Model, Persister, Business Logik, RMI, Client intern und extern). Dass diese isoliert betrachtet nicht/schwer realisiert werden können, war uns zu diesem Zeitpunkt zu wenig klar. Mit der Absicht die grösste Herausforderung (programmier-technisch) an mich zu nehmen, habe ich mich dem Client extern gewidmet. Die Abhängigkeiten und der systematische Aufbau des Programmes wurden mit der Zeit immer klarer und hat dementsprechend auch zu Mehraufwand und Verzögerungen (weniger aber beim Client) geführt. Da sich der externe Client nur minimal (Verbindung Business Logik, Berechtigungen etc.) vom Client intern unterscheidet, wurde der ganze Client durch Pascal und mich programmiert. Teilweise durch Parallelprogrammierung, teilweise durch Paarprogrammierung. Eine tiefere Trennung ist hier nicht möglich, da wir immer wieder am gleichen Code gearbeitet haben. In dieser Zusammenarbeit wurde auch die RMI-Verknüpfung erfolgreich programmiert. Die Anknüpfung an den Webserver erfolgte dann mit Unterstützung von Olivia. Das GUI-Design wurde hauptsächlich durch Dominik erstellt. Wie in der Aufgabenstellung verlangt, habe ich mich einen halben Tag, nach Einführung durch Dominik, mit dem JUnit-Testing beschäftigt. Das Ziel ein lauffähiges Programm abzuliefern hat vor allem in der Schlussphase nochmals viel Zeit und Energie gebraucht und hat meine Ressourcen bis ganz am Schluss beansprucht.

Zusammenfassend blicke ich auf eine intensive aber erfolgreiche Projektphase mit vielen Hochs und Tiefs zurück. Der zeitliche Aufwand war enorm, soziale wie auch andere fachliche Bereiche haben darunter aber sicher gelitten. Von der zeitlichen Einteilung des Projekts sehe ich kein Verbesserungspotential, durch mangelnde Erfahrung würde ich bei einem weiteren Projekt aber einiges anders lösen, was definitiv zu Zeiteinsparungen führen würde. Ganz klar unterschätzt habe ich aber den sozialen Austausch und die Abstimmung in der Gruppe. Scrum-ähnliche Besprechungen haben sich ab Mitte des Projektes automatisch eingestellt, was zu einer Verbesserung führte. Aber die Aufgabenverteilung und Zusammenarbeit hat definitiv noch Verbesserungspotential.

Wichtigste Aktivität(en) im Projekt Feukora :

Client intern und extern inkl. Verknüpfungen (70%),

Weitere Aktivität(en):

Software Architektur (15%), UseCases (5%), Requirement (5%), JUnit4 Test (3%), Dokumentation (2%)

21.1.2 Individuelles Portfolio zum Projekt Feukora

DATUM EVENTUELL PHASE	DAUER AUFWAND	TÄTIGKEIT / AUFGABE	BEMERKUNG/ERKENNTNIS
09.03.2016	2.0 Std.	Erstellung des Dokuments Projekt Feukora - Requirements	Einige Punkte sind unklar und müssen mit dem Kunden noch näher spezifiziert werden.
19.03.2016	2.0 Std.	Use Case Beschreibung 11, 12 und 13 erstellt	Use Case gemäss Vorlage erstellt und kommentiert
16.04.2016	1.0 Std.	Konzept UseCase Beschreibung überdenkt und angepasst (grafisch)	Die UseCase Beschreibung wurde im ersten Anlauf falsch erstellt. Pro Requirement wurde fälschlicherweise ein UseCase beschrieben. Neu werden die Requirements den UseCases zugeordnet
24.04.2016	6.0 Std.	Bausteinsicht und Laufzeitschicht gezeichnet	Grosse Schwierigkeiten mit dem Erstellen der Laufzeitschichten. Feukora-Prozesse waren noch zu unklar um diese festzuhalten. Sicher auch wegen mangelnder Erfahrung. Was ist wie realisierbar beim Programmieren resp. Im GUI.
29.04.2016	4.0 Std.	Grober Rahmen des Klassendiagramm erstellt	In der Anfangsphase fiel es uns schwer, einen geeigneten Weg zur Erstellung zu finden. Wir haben uns nachher sehr stark an das Referenzprojekt gehalten und konnten so viele Lücken resp. offene Fragen klären. Der ganze Aufbau und die Zusammenarbeit von Model/Persister-Klassen sowie der Businesslogik sind nun um einiges klarer. Das definitive Klassendiagramm wird am Ende des Projektes mit dem EA erstellt.
02.05.2016	2.0 Std.	Szenarien für das Laufzeitmodell beschrieben	Man hat eine Vorstellung wie die verschiedenen Szenarien aussehen könnten. Ob diese sich so umsetzen lassen wird sich im Laufe des Projektes zeigen

06.05.2016	3.0 Std.	UseCase Beschreibung 11, 12 und 13 erstellt, Grafiken mussten überarbeitet werden	Persönlich einige Schwierigkeiten beim Erstellen der Grafiken. Werden in UML 2 kompakt nur oberflächlich beschrieben und online sind mehrere Meinungen vertreten. Z.b. stellt man eine Datenbank auch als Akteur dar?
09.05.2016	8.0 Std.	Client intern mit Pascal ausprogrammiert	Viel Zeit ging verloren um zu realisieren, dass eine zusätzliche EclipseFX Version benötigt wird. Die Einbindung der FXElemente war dann nicht mehr schwer. Allgemeint lässt sich festhalten, dass der Anfang jeweils harzig war, durch viele Wiederholungen im Code aber rasch vorangegangen werden konnte.
10.05.2016	5.0 Std.	Client intern mit Pascal ausprogrammiert	
11.05.2016	7.0 Std.	Client intern mit Pascal ausprogrammiert	Unterschätzt habe ich den Austausch mit den anderen Gruppenmitgliedern. Viele interne Diskussionen über Vorangehens Weisen und Erläuterungen um alle Mitglieder auf den gleichen Stand zu bringen.
12.05.2016	3.0 Std.	Client intern mit Pascal ausprogrammiert	GUI-Handling
13.05.2016	8.0 Std.	Client intern ausprogrammiert	Terminübersicht
14.05.2016	7.0 Std.	Client intern ausprogrammiert	printscreen erstellen, Terminübersicht anzeigen
16.05.2016	9.5 Std.	Client intern, MA ändern	Zeitdruck steigt, kriege das erste mal Bedenken über die Realisierung bis zum Abgabetermin
17.05.2016	7 Std.	Client intern ausprogrammiert, JUnit4 Test durchgeführt	FeuerungsrapportServiceBrennerTest, FeuerungsrapportServiceOrtTest
23.05.2016	10.0 Std.	Client intern wird mit RMI verbunden, Zusammenarbeit mit Pascal	

24.05.2016	6.0 Std.	Client intern wir mit RMI verbunden, Zusammenarbeit mit Pascal, Laufzeitschicht angepasst	Die Laufzeitschichten Modelle mussten überarbeitet werden da sich einige Prozesse anders gestaltet haben als vorerst angenommen.
25.05.2016	6.0 Std.	Management Summary, individuelles Portfolio anpassen, Funktionale Tests	
26.05.2016	5.0 Std.	Schlussspurt, Verknüpfung mit Webserver	Bin froh das Projekt abzuschliessen, bin aber auch stolz auf das Resultat und die Lerneffekte.
28.05.2016	1.5 Std.	Deployment (ungefähre Angabe)	Deployment @ HSLU mit Alessio
28- 29.05.2016	2.0 Std.	Präsentationsvorbereitung (ungefähre Angabe)	Ich weiss noch nicht ob und was ich präsentiere aber ich werde mich bestimmt darauf vorbereiten.
TOTAL	105.0 Std.		

21.2 Dominik Stirnimann

21.2.1 Individuelles Management Summary zum Projekt Feukora Dominik Stirnimann

Zu meinen Hauptaktivitäten zählte neben der Projektleitung vor allem das Erstellen von allem was mit Grafiken zu tun hat. Hin von GUI (10%) zu Diagramme (10%) über Modelle. (10%) Da ich mich schon früh mit Java FX und Enterprise Architect befasste, ging es für mich mit der Zeit relativ einfach diese in einer passablen Zeit zu erstellen. Mit Java Scene Builder kannte ich mich zwar vom ersten Semester noch etwas aus, musste aber einiges Investieren um einigermassen klar zu kommen. Zum Glück war das Tool relativ einfach aufgebaut und somit mit ausprobieren zu bewältigen. Beim EA hingegen war es anders. Das Programm war viel komplexer und verschachtelter. Es kann einfach sehr viel und es war schwierig die richtigen Einheiten für die Diagramme zu finden und diese zu zeichnen. Im Nachhinein hätte ich besser mit einem einfacheren Tool(z.B. MS Visio) gearbeitet. Im Punkt Projektleitung (5%) habe ich zu Beginn des Projektes die Zügel ein wenig in die Hand genommen und unsere Teamsitzungen vorbereitet, To-Do's erstellt und die Aufgabenverteilung Koordiniert. Im späteren Verlauf aber hat Olivia, welche mit der Dokumentation sich befasste, mehr oder weniger übernommen. Dies, da Sie durch die Doku eine gute Übersicht hatte. Neben der Projektleitung befasste ich mich aber auch mit dem erstellen, betrieb und testing des Webservers (50%). Das Webserver Interface und Implementation wurden fast vollständig von mir erstellt. Beim Testing habe ich Teile übernommen und die Grundlage geliefert (Struktur und init Methode).

Im Schlussspurt habe ich auch noch an der Dokumentation (10%) mitgearbeitet und vereinzelte Kapitel überarbeitet und ergänzt. Ich habe Teile des Anhangs (5%) erstellt und eingefügt. Ebenfalls werde ich noch die Dokumentation endgültig formatieren und drucken.

Fazit zum Projekt:

Zusammenfassend bin ich froh, dass ich so ein Projekt realisieren konnte. Was mich aber lange gestört hat war, dass wir in der Gruppe keinen Crack hatten, der vielleicht schon einmal ein solches Projekt realisiert hat oder zumindest im Programmieren mehr Erfahrung hat. Dies hätte uns wahrscheinlich viel Frustration und mühsames erarbeiten erspart.

Wichtigste Aktivität(en) im Projekt Feukora :

GUI (10%), Diagramme (10%), Modelle (10%), Webserver (50%), Projektleitung (5%)

Weitere Aktivität(en):

Dokumentation (10%), Anhang (5%)

21.2.2 Individuelles Portfolio zum Projekt Feukora Dominik Stirnimann

Datum eventuell Phase	Dauer Aufwand	Tätigkeit / Aufgabe	Bemerkung/Erkenntnis
14.03.2016	1.5 Std	Vorbereiten 1. Teamstzung und erstellen der Taskliste. Bearbeiten der Rollenverteilung und erstellen von genaueren Beschreibungen. Management der allgemeinen Tasks.	Vorbereitung nahm mehr Zeit in Anspruch als zuerst gedach. Eine durchdachte Taskliste macht aber die Teamsitzung speditiver und entlastet so alle Teammitglieder.
14.03.2016	0.5 Std	Nachbearbeitung 1. Teamsitzung plus erstellen der Use Case Vorlage	Durch das Template haben wir alle eine einheitliches Aussehen der UseCases und am Schluss müssen diese nicht nochmals mühsam von einer Person formatiert und in eine einheitliche Form gebracht werden
14.03.2016	1.0 Std	1. Teamsitzung: Besprechen der vorbereiteten Punkte, festlegen der Deadlines für Used Cases und	Speditiv geführte Sitzung, da Taskliste. Gute Mitarbeit der Gruppenmitglieder

		weiteres Vorgehen. Klären von Unklarheiten sowie Standortbestimmung.	
16.03.2016	0.1 Std	Verfassen des E-Mails betreffend der Aufgabenverteilung sowie anpassen des Dokumentes. Organisatorische Arbeiten (Dropbox, Trello)	keine Besonderheiten
20.03.2016	1 Std	UseCase Modellierung: Erstellen der zwei UsedCases für Requirement Nr. 9 und Nr. 10.	Die Vorlagen auf Ilias sind vom Umfang des Inhalts nur teilaussagend. Nicht genau sicher, ob Used Case beschreibung genau genug ist bzw. alle Varianten miteinbezogen wurden.
18.04.2016	0.2 Std	Vorbereiten 2. Teamsitzung. Erstellen von Übersicht und Bulletliste	keine Besonderheiten
21.04.2016	3 Std	Used Cases mit EA zeichnen und überarbeiten	EA ist ein komplexes Tool was ein bisschen Einarbeit benötigt. Sobald man den Dreh aber raus hat, geht es einfach von der Hand.
23.04.2016	1 Std	Wiederinbetriebnahme von JavaFX und SceneBuilder für spätere GUI Oberflächen. Erstellen von ersten Vorlagen für GUI Oberflächen wie Loginfenster oder Eingabe der Werte.	keine Besonderheiten
25.04.2016	3.5 Std	Zeichnen von Baustien, Laufzeit und Verteilungsschicht mit EA Teil I.	EA ist ein komplexes Tool was ein bisschen einarbeit benötigt. Sobald man den Dreh aber raus hat, geht es einfach von der Hand. Gewisse Dinge müssen noch mit dem Team abgesprochen werden um Unklarheiten für aussen zu vermeiden
26.04.2016	1.5 Std.	Zeichnen von Baustien, Laufzeit und Verteilungsschicht mit EA Teil II und überarbeiten.	Zeichnen mit EA geht immer besser
02.05.2016	1. Std	Überarbeiten von Used Cases mit EA aufgrund von Inputs von Teamsitzungen und Änderungen in Struktur der Used Cases. Diese werden nicht mehr generell erfasst, sondern spezifisch für jede Aktion (Used Case Aktion) der EA Zeichnung	Siehe 26.04.16

07.05.2016	6 Std	GitHub	Wir hatten grosse Probleme beim Einrichten von Github. Die Anleitung welche zur Verfügung stand war leider quasi nie hilfreich. In Absprache mit anderen Teams welche dasselbe Problem hatten. Nach langen Googeln, Tutorialvideos schauen und Trial&Error ist es irgendwie gelungen das ganze einzurichten. Ich hätte es toll gefunden, wäre im Unterricht das ganze kurz eingerichtet oder zumindest eingeführt worden. Dies hätte uns einige Stunden an Arbeit erspart.
08.05.2016	2 Std	GitHub	siehe 07.05.2016
09.05.2016	8 Std	GitHub für andere Einrichten, Tests, Interfaces und Implementation von Webserver beginnen. Starten von Tomcat und Festlegen von Projekten. Mit Apache-Ant auseinandergesetzt - wieder verworfen. GitHub abgeglichen und Probleme behoben.	Durch das Referenzprojekt und die Vorlesungsfolien konnte ich mir ein gutes Bild verschaffen wie der Webservice etwa aussehen muss. Was mich jedoch verwirrte war, dass in den Folien plötzlich was über Apache-Ant stand. Dessenwegen habe ich kurz die Theoretische Grundlage erarbeitet bis ich eingesehen habe, dass dieses gar nicht nötig ist.
10.05.2016	2 Std	Webserver Implementation und Interface weiter geführt, Methoden von Modellklassen ergänzt.	Von Businesslogik her weitere implementation der Methoden. Nichts besonderes
11.05.2016	6 Std	Interface und Implementation von Webserver fertiggestellt überarbeiten von GUI aufgrund Inputs von Client-programmierung	Dank eines Rats von Alessio die GUI-Fenster zu testen ist uns aufgefallen, dass die Auflösung zu hoch war. Dessenwegen mussten wir noch Fenster überarbeiten und gewisse Slider einbauen. Danke Alessio
12.05.2016	2.5 Std	Erneute GitHub Probleme. Konflikte mit Dateien von Teammitglied Luca Raneri. Upload von Persister Webserver Dateien einfach auf den Stand vor 2 Tagen zurückgesetzt. Probier, dies wieder in Ordnung zu bringen. Mehrfaches Probieren von neuem Upload, immer wieder Konflikte und gegenseitige Rücksetzung von Dateien. Neue Projekte erstellt. Vorläufige Lösung war, dass Persister momentan nur noch Lokal gespeichert wird und Webserver neu geuploadet wurde.	GitHub bereitet uns einige Probleme, welche wir leider nicht vermeiden konnten.
14.05.2016	2 Std	Schreiben von ersten Tests und überlegen bzw. schreiben der init Methode (nicht fertig).	Übernehmen der Struktur von Persister von Tests und init().

16.05.2016	7 Std	Anpassen der Methoden und des Interfaces aufgrund neuer Methoden. Abgleichen mit RMI, erstellen und Import von .jar files für Modellklassen neuorganisation	
17.05.2016	3 Std	Schreiben von Tests, einführen von Alexandra zur unterstützung, löschen der bestehenden Init Methode, da diese zu fehlerhaft war bzw. nicht vollständig. Erstellen und anpassen der neuen Init Methode. Probieren die Webserver-Dateien zu parsen. Dieses ist leider nicht gelungen. Fehlersuche gestartet, mit Alessio Di-Lascio ebenfalls nach fehlern gesucht. Diesen aber noch nicht gefunden.	Das Überprüfen der konfiguration von property Datei und der Konfiguration des Webservers erbrachte leider kein Ergebnis. Problem muss weiter verfolgt werden. Evtl. Mit J. Succur überprüfen.
18.05.2016	1.5 Std	Schreiben von Tests und abschliessen von Tests. Überarbeiten von Methoden zum schreiben in die Datenbank	keine Besonderheiten
19.05.2016	2.5 Std	Erste Tests mit Verbindung zu RMI. Problem war, dass Webserver nur mit Fehler gestartet werden konnte. Starte Fehlersuche. Mit Hilfe von J. Succur diverse Fehler gefunden und behoben. Server gestartet und wsdl Datei erzeugt. Die .jar files von RMI und Modellklassen ebenfalls zur Verwendung vom Server zur Verfügung gestellt.	Langes hin und her. Debugger konnte nur bedingt weiterhelfen. Schlussendlich konnte mit direktimplementation der .jar files und Namensgebungsänderungen die Fehler behoben werden.
21.05.2016	4.5 Std	LiegenschaftsTesting fertigstellen. Fehler beim Aufruf von RMI Server sowie Webserver. Fehlersuche gestartet. Problem war mit Property Datei, dass nicht richtig ausgelesen wurde. RMI läuft, Webserver ist wirf leider wieder Fehler.	Schreiben der Testklassen lief bisher ohne probleme. Warscheinlich, da diese noch nicht richtig getestet werden konnten, da webserver nicht lief
22.05.2016	0.4 Std	Erneutes zeichnen von Bausteinschicht. Diese war leider veraltet bzw. nicht optimal. Neu mit Visio statt mit EA erstellt. Visio bot mir ein wenig mehr freiheit, was das Zeichnen nach meinen Vorstellungen anbelangte	Visio ist etwas einfacher zu bedienen als EA. Somit ging das zeichnen relativ einfach.
23.05.2016	8 Std	Überarbeiten, ergänzen und querlesen von Used Cases. Einfügen der diversen UC-Diagramme in Dokumentation	keine Besonderheiten

		sowie einfügen von gestern erstellten Dias (Bausteinschicht und Applikation Feukora). Für Webserver erneut .jar Datei für Client Extern erstellt.	
24.05.2016	7.5 Std	Anhang für Projektleitung erstellt und zusammengetragen (Sitzungsprotokolle und Entscheidungen, Präsentationen), Used Cases überarbeitet/gecheckt, Diagramme von Laufzeitschicht überarbeitet und neu gezeichnet. Beschreibung der Laufzeitschicht angepasst. Dokumentation Teile von Kapitel 8 geschrieben und ergänzt. Dokumentation Kapitel 17 TDD und Deployment Queergelesen, ergänzt und korrigiert. Weitere Tasks für Gruppenmitglieder koordiniert.	keine Besonderheiten
25.05.2016	9 Std	GUI Fenster überarbeiten für Bedienbarkeit,BPMN Diagramme mit EA Zeichnen, ManagementSummary schreiben, Dokumentation Kapitel 6, 7 und 11 überarbeitet und ergänzt. Anhang für GUI erstellt. Dokumentation Kapitel 21 ergänzt und eingefügt.	Durch diverse Entscheidungen und Anpassungen von Alexandra und Pascal musste das GUI wieder angepasst werden um die Ansicht und Bedienbarkeit sicherzustellen.
26.05.2016 - 30.05.2016	ungefähr 15 Std	Formatieren und Ergänzen von Anhang, Dokumentation und Source Code. Drucken von Source Code. Erstellen von Präsentation mit Prezi.	Geschätzte Zeitmenge
TOTAL	99.7		

21.3 Luca Raneri

21.3.1 Individuelles Management Summary zum Projekt Feukora Alexandra Lengen

Zu Beginn des Projekts habe ich ein mögliches ERD erstellt, welches ich im Verlauf des Projektes immer weiter ausgebaut habe. Für die Erstellung musste ich mich vertieft mit den Requirements auseinandersetzen und vermehrten den zur Verfügung gestellten Rapport unter die Lupe nehmen. Die Dazugehörige Attributliste, welche ich ebenfalls laufend erweitert habe, setzte einen Grundstein für die Erstellung der Modellklassen fest. Die zu bearbeitenden Use Cases wurden fair aufgeteilt, so dass jedes Gruppenmitglied in etwa ähnlich viel Aufwand hat.

Die zu Beginn festgehaltenen Rollen der einzelnen Gruppenmitglieder haben sich im Verlauf des Projektes vermischt. Es wurde jeder mit jedem Thema konfrontiert und hat außerhalb seiner Rolle etwas beigetragen. Das Programmieren war jedoch meine Haupttätigkeit während dem Projekt. Unsere Gruppe bestand aus Kameraden, welche keine vertieften Programmierkenntnisse haben. Wir waren somit auf uns alleine gestellt und mussten mit sehr viel Kopfzerbrechen uns das meiste noch selbstständig aneignen. Betreffend der Programmierung übernahm ich die Verantwortung des Persisters und programmierte die Interfaces und die dazugehörigen Impl-Klassen. Dazu gehörten auch die JUnit-Tests, an welchen ich mitgeholfen habe.

In der Dokumentation wurden vereinzelte Kapitel von mir persönlich bearbeitet. Auch sonst versuchte ich mitzuhelfen wo Probleme aufgetaucht waren. Dadurch bekam ich einen Einblick in alle Bereiche des Projektes. Meine eigenen Primäraufgaben litten daran nicht. Gegen Ende des Projektes half ich bei dem Webservice aus und erweiterte das dazugehörige Interface und die Impl-Klasse dort wo es mir als notwendig erschien. Diese Erweiterungen brauchte ich folglich für die JUnit-Tests des Webservices, an welchem ich auch mithalf. Ich konnte dort schon viel profitieren aufgrund der Erfahrung mit dem Persister-Testing.

Fazit zum Projekt:

Insgesamt bin ich sehr stolz auf mich und auf meine Kameraden. Wir passten sehr gut als Gruppe zusammen und hatten während der gesamten Projektzeit eine tolle Zusammenarbeit. Trotz der fehlenden Informatikkenntnisse habe ich das Gefühl, dass wir verhältnismässig eine grandiose Leistung erbracht haben.

Wichtigste Aktivität(en) im Projekt Feukora :

JUnit4 Tests (45%), Datenbank (20%), Persister-Programmierung (20%)

Weitere Aktivität(en):

Dokumentation (5%), Use Cases (5%), Webservice-Programmierung (5%)

21.3.2 Individuelles Portfolio zum Projekt Feukora Luca Raneri

Datum eventuell Phase	Dauer Aufwand in Stunden	Tätigkeit / Aufgabe	Bemerkung/Erkenntnis
21.03.2016	1.00	Datenbank Entwürfe erstellt mit Pascal Steiner und Matthias Perollaz. Entitäten mit ihren zugehörigen Attributen notiert. Verbindungen mit den verschiedenen Entitäten markiert und weiteres Vorgehen besprochen.	Durch das Zusammensitzen und Besprechen wurde uns die ganze Komplexität der Datenbank viel klarer. Wir erkannten, dass diese uns noch viele Probleme bereiten wird.
21.03.2016	2.50	Use Case zu der Verwaltung der Messergebnisse erstellt und geschrieben.	Bevor ich die Use Cases schreiben konnte musste ich mich noch einmal genauer mit der Materie auseinandersetzen. Anfangs hatte ich Schwierigkeiten, weil ich nicht genau wusste was ich schreiben soll.
08.04.2016	2.50	Die Datenbankentwürfe habe ich mit Pascal Steiner und Matthias Perollaz erweitert und verbessert. Wir haben neue Ideen einbezogen und sind verschiedene Möglichkeiten durchgegangen.	Die befürchteten Probleme sind teils schon aufgetaucht. Die Schwierigkeit liegt darin, dass es verschiedene Möglichkeiten gibt, die Datenbank zu modellieren, wir uns aber noch nicht sicher sind welche wirklich am meisten Sinn macht. Wir haben das Gefühl, dass wir schon sehr gut vorangekommen sind.
15.04.2016	3.00	Ich habe die Datenbank entwürfe erneut verbessert. Die Aufgetretenen Probleme vom letzten Mal teils ganz behoben. Zusätzlich habe ich erneut ein ERD gezeichnet und die dazugehörigen Attribute erfasst. Ebenfalls habe ich mich mit der Datenbankprogrammierung mit JDBC auseinandergesetzt inklusive Data Access Object (DAO).	Das DAO wir noch einige Zeit in Anspruch nehmen sowie die ganze Programmierung. Nur schon das Einarbeiten in die Thematik wird für mich anspruchsvoll. Das erstellte ERD wird den restlichen Gruppenmitglieder behilflich sein.
22.04.2016	3.00	Unsere heutige Aufgabe war es das ERD der Datenbank erneut zu verbessern und dazu die Attributliste zu erstellen. Dies habe ich mit Pascal gemacht.	Wir mussten dies tun, weil wir die Entitäten letztes Mal zu wenig ausgelagert haben und dadurch nur unnötige Redundanzen entstanden wären. Bei vereinzelten Entitäten müssen wir uns noch einmal Zeit nehmen und überlegen wie wir die noch übrigen Aufgaben bestreiten wollen.
25.04.2016	1.00	Ich habe einen ersten Entwurf für das Kapitel 10 und das Kapitel 11 gemacht. Dabei habe ich vorerst ein paar Qualitätsszenarien definiert und erläutert. Zusätzlich dazu dann den Qualitätsbaum erstellt. Des Weiteren habe ich die Risiken, welche uns erwarten könnten, festgelegt.	Ich musste meine Kreativität freien Lauf lassen.

04.05.2016	2.50	Da ich den Persister Teil der Software übernehme, habe ich mich mal an das Referenzprojekt gewendet und dies durchstudiert. Ebenfalls habe ich dann angefangen zwei Interfaces und die zugehörigen Klassen zu programmieren um das Gelernte gleich anzuwenden.	Ich hatte keine Ahnung was ich tun musste, da ich zum ersten Mal in der Form mit dem Programmieren konfrontiert wurde. Wir haben ebenfalls niemanden in unsere Gruppe der programmieren kann. Das ist ein riesen Problem, wenn man nicht weiter kommt. Ich orientierte mich deshalb am Referenzprojekt.
06.05.2016	3.50	Ich habe alle zehn Interfaces und zehn Klassen im Eclipse für die Programmierung der Persister vorbereitet. Der grösste Teil der Interfaces habe ich zusätzlich schon programmiert. Des Weiteren habe ich das JpaUtil und die Generic Persister Klasse erstellt und programmiert.	Dies war eine enorme Fleissarbeit und brauchte trotz kleinem Fortschritt viel Zeit.
09.05.2016	6.00	Das ERD, die Attributlisten und die Modellklassen habe ich verändert, da programmiertechnisch umgedacht werden musste.	Ich merkte, dass unser geplantes ERD so nicht stimmt. Somit mussten im Team Änderungen an den Modellklassen und an den Attributen getätigt werden. Ebenfalls wusste ich nicht wie die Methoden von den Persisterklassen mit der Modellklasse zusammenhängen. Mir war nicht klar welche Methoden in dem Persister implementiert werden müssen.
11.05.2016	4.00	Ich habe am Persister weiter Programmiert. Dabei habe ich die Methoden, welche ich im Interface vordefinierte, angefangen zu implementieren.	Ich war mir nicht sicher ob das was ich tue richtig oder falsch war. Daher musste ich erneut das Referenzprojekt zur Hilfe nehmen. Da ich nicht verstand was der Code genau bewirkt, brauchte ich zusätzlich Zeit diesen zu verstehen.
12 & 13.05.2016	14.50	Persister-Programmierung: Der ganze Persister unseres Projekts konnte ich soweit fertigstellen bzw. alle zehn Klassen fertig programmieren. Die dazugehörigen Testklassen, welche ich teilweise schon vorbereitet habe, konnte ich ebenfalls schon auf ihre Funktionalität prüfen.	Als ich begriff wie der Persister funktioniert, kam ich in einen Arbeitsfluss und konnte diese Klassen fertigstellen. Während dem Programmieren kam es zu Schwierigkeiten mit dem GitHub. Ich konnte meine Projekte nicht mehr hochladen und aktualisieren. Ich entschied mich daher den Persister vom GitHub zu entfernen und das Java Projekt lokal weiterzuführen. Teils schon zuvor codierte Teile der Klassen musste ich erneut hinzufügen, da GitHub diese nicht geschluckt hat. Die Testklassen des Persister brachten mich an meine Grenzen. Ich kapierte nicht wie ich meine Fehler beheben kann und musste mir Hilfe holen.

17. bis 19.05.2016	16.00	Mithilfe von Olivia konnte ich alle Testklassen für das TDD für den Persister bzw. DAO-Klassen fertigstellen.	Durch die Zusammenarbeit mit Olivia konnte ich sehr schnell verstehen wie die ganzen Tests aufgebaut werden. Ebenfalls verstand ich den Zusammenhang mit der Datenbank viel besser. Ich wusste nun wann Daten in die Datenbank gespeichert und wann sie wieder gelöscht werden. Wie die Reihenfolge des Erstellaublaufs und des Löschablaufs der Objekte sein muss, begriff ich auch.
20.05.2016	10.00	Ich übernahm den Web-Service und habe die FeuerungsrapportServiceTest-Klassen überarbeitet. Ebenfalls habe ich das Interface und die Klasse überarbeitet und auf Fehler überprüft.	Ich habe das Testing vom Webservice übernommen und bestehende Tests erweitert, überarbeitet und Klassen und Interfaces mit Methoden fürs Testing ergänzt. Da ich durch die viele Arbeit schon an Kenntnisse mit dem Persister gewonnen habe und den gesamten Zusammenhang zwischen Modellklassen, Persister und Businessklassen verstand, wusste ich, was ich zu tun hatte. Dank diesem Vorteil kam ich sehr schnell vorwärts.
21.05.2016	4.00	Use Case zu der Verwaltung der Messergebnisse überarbeitet. Dokumentation auf Fehler überprüft und Ergänzungen getätigter.	Wir waren programmiertechnisch schon ziemlich fortgeschritten. Ich nutzte die Zeit daher schon mal um die Dokumentation zu ergänzen und die Use Cases zu aktualisieren.
23.05.2016	5.00	Kapitel 18 in der Dokumentation fertig gestellt und Use Cases zu Feuerungsrapport bearbeitet.	Bei den Use Cases haben wir bemerkt, dass wir für den Feuerungsrapport an sich noch keine erstellt und beschrieben haben. Deshalb habe ich diese noch ergänzt (2h). Bezuglich des Test-Driven Development müssen wir zugeben, dass wir dort unsere Schwierigkeiten hatten und somit einen Abstrich machen müssen. Nebenbei half ich die Werte in die DB einzurichten um den Client zu testen. Ich probierte die Probleme beim RMI und Webservice heraus zu finden, jedoch lag das ausserhalb meiner Fähigkeiten diesen zu beheben (1h). (StackOverflow Error)

24.05.2016	4.00	Datenbank Dokumentation mit Matthias geschrieben. Überprüfung und Anpassung der veralteten Attributliste des ERD. Dann im Visio eine Datenbank für das Kapitel erstellt.	Die Attributliste stimmte nicht mehr mit den programmierten Modellklassen überein. Somit mussten wir Änderungen vornehmen (0,5h). Für die Erstellung des neuen ERD haben wir uns für Visio entschieden, weil wir für die Dokumentation noch ein übersichtliches ERD haben wollten. Das Erstellen nahm unerwartet viel Zeit in Anspruch, da das Visio technische Probleme bereitete (1,5h). Aber es gelang uns das ERD so wie wir es wollten zu erstellen.
25.05.2016	3.00	Den Code des Projektes exklusive Client durchgelesen und auf Fehler überprüft beim JavaDoc. (Persister, RMI, Web Server, Business-Logik)	Die JavaDoc-Dokumentation hatte noch Fehler vorhanden, die wir noch nicht behoben hatten.
25.05.2016	3.00	Ich habe das Management Summary geschrieben und das Individuelle Journal fertiggestellt.	keine Bemerkungen
26.05.2016	5.00	Endkontrolle der Doku und Abschlussarbeiten	Keine Bemerkungen
Total	93.5 Std.		

21.4 Matthias Perrollaz

21.4.1 Individuelles Management Summary zum Projekt Feukora

Meine Hauptaufgabe im Projekt Feukora bestand aus der Mitarbeit in Thema Datenbankentwurf und Leitung Datenbank erstellen. Ausserdem war mir die Programmierung der Schnittstelle RMI(vor allem RO Klassen und RMI-Server-Klasse) zugeteilt.

Zu Beginn des Projekts Feukora habe ich zwei UseCase zugeteilt erhalten und diese erstellt. Später stellte sich heraus, dass wir dies falsch angegangen waren. Daher gab es nochmals eine Neuverteilung. Nun habe ich alle UseCase für die Terminverwaltung erstellt.

Beim Datenbankentwurf habe ich mit Luca und Pascal zusammengearbeitet. Gestartet haben wir mit dem Erstellen des ERD und vergeben der Attribute. Ich habe dann die Aufgabe gefasst, das ERD mit MS Visio zu zeichnen.

Meine Hauptaufgabe bei der Datenbank Erstellung war das Einholen der Informationen für unsere Projekt-Datenbank und Erstellen einer Test-Datenbank für die Tests.

Programmiertechnisch habe ich mich dem RMI angenommen. Da wir eine etwas speziellere Aufgabenteilung beim Programmieren hatten, startete ich erst später effizient an dieser Schnittstelle zu arbeiten. Zu Beginn habe ich mehrheitlich bei Teammitgliedern ausgeholfen. Sei dies an den Modelklassen, dem Persister oder der Businesslogik.

Als dann diese Programm-Teile abgeschlossen waren, habe ich mich wie mehrere andere Teammitglieder auf die Dokumentation konzentriert. Dort konnte ich noch in verschiedene Teile Einblicke gewinnen.

Zusammenfassend habe ich positive, wie auch eher negative Erfahrungen in diesem Projekt gemacht. Was jedoch ganz sicher ist, dass ich mich weiterentwickeln konnte. Sei es in der Programmierung, der Software-Architektur oder auch nur im sozialen Umfeld. Ich würde sicherlich nicht alles nochmals genauso machen wie bei diesem Projekt, jedoch war der Lerneffekt da. Und aus den Fehlern konnte man lernen, wie man es nicht mehr tun sollte. Zum Schluss bin ich auch froh und ein bisschen stolz darauf, dass wir dieses Projekt gemeistert haben.

Wichtigste Aktivität(en) im Projekt Feukora :

RMI RO-Klassen und RMI-Server (40%), Datenbank (35%)

Weitere Aktivität(en):

Dokumentation (15%), UseCases (6%), JUnit4 Test (4%)

21.4.2 Individuelles Portfolio zum Projekt Feukora

DATUM EVENTUELL PHASE	DAUER AUFUND	TÄTIGKEIT / AUFGABE	BEMERKUNG/ERKENNTNIS
20.03.2016	1 Std.	Die 2 konkrete UseCase-Fälle die mir zugewiesen wurden habe ich erstellt.	Diese werden an der Sitzung vom 21.03.2016 nochmals besprochen.
21.03.2016	1 Std.	Wir haben mit der Arbeit betreffend der Datenbank begonnen. Es wurden erste Ideen betreffend der Tabellen erstellt.	Wird an der Sitzung vom 21.03.2016 besprochen bzw. die Ideen mit den Kollegen und Kolleginnen geteilt.
08.04.2016	2,5 Std.	Datenbankmodell wurde verfeinert. Die Entitätsmengen wurde gezeichnet und verglichen.	Es gab verschiedene Ansichten betreffend der einzelnen Tabellen. Differenzen wurden jedoch geklärt.
22.04.2016	1,5 Std.	Aktuelles Datenbankmodell wurde im Visio gezeichnet mit den Beziehungen.	Probierte zuerst das Modell mit dem EA zu zeichnen, leider ohne Erfolg.
24.04.2016	2.0 Std.	Diverse UseCase gezeichnet, müssen jedoch noch besprochen werden am 25.04	Wir hatten zuerst eine falsche Vorstellung von den UseCase, daher mussten diese nochmals überarbeitet werden.
29.04.2016	3 Std.	Klassendiagramm wurde mit den Mitglieder erstellt	Der Beginn war etwas langsam. Nach einer Stunde ca. waren wir dann produktiver.
01.05.2016	1 Std.	DB-Informationen wurden eingeholt. Mit pgAdmin getestet	
02.05.2016	2 Std.	Git Hub wurde mit Eclipse verbunden.	Dauerte ein Moment bis alles funktionierte. Mussten bei einer anderen Gruppe nachfragen betreffend div. Einstellungen
04.05.2016	1 Std.	UseCase überarbeitet und in Ordner abgelegt.	Habe die UseCase rund um die Terminverwaltung erstellt.
08.05.2016	2 Std.	Einarbeiten ins Thema RMI bezüglich des Projekts. Erste Schritte in Zusammenhang mit der Programmierung im Projekt.	Habe mich im Skript und Internet nochmals über das Thema RMI schlau gemacht. Daher 1 1/2h Lernzeit.
09.05.2016	2 Std.	Attributliste kontrolliert und überarbeitet	Mit Luca die Attributliste kontrolliert
09.05.2016	6 Std.	Programmierpraxis betreffend des Projektes. Gearbeitet wurde im Gebiet vom RMI(2) und den Modelklassen(4)	Hatten Startschwierigkeiten, bzw. mussten dazwischen unser Klassendiagramm umstellen.

11.05.2016	7 Std.	BusinessLogik mit Olivia(3), Attributsliste(1,5), Modelklassen(2) , DBTest erstellen(0,5)	Habe Olivia bei der BusinessLogik ausgeholfen. Datenbank fürs Testen wurde erstellt
12.05.2016	3 Std.	Verbesserungen Modelklassen mit Olivia	Bin nun mit Olivia ein gut eingespieltes Team, daher helfe ich Ihr bei den Modelklassen.
13.05.2016	2 Std.	Modelklassen mit Olivia beendet	Diverse Probleme mit GitHub und der aktuellen Version auf dem Laptop der einzelnen Dateien benötigte Zeit die unnötig verschwendet wurde.
15.05.2016	3 Std.	Skript für TestDB schreiben, korrigieren.	Hatte Probleme mit den ForeignKeys. Daher benötigte ich mehr Zeit als geplant.
15.05.2016	2 Std.	RMI begonnen	Habe nun mit meinem RMI Teil begonnen. Andere Klassen sind nun plus minus so weit.
16.05.2016	2 Std.	Skript für TestDB erweitert, getestet. Daten in TestDB füllen	Konnte durch einige Tipps das Skript für die Datenbank verbessern. Tabellen lönnen nun ohne weitere Probleme erstellt werden.
16.05.2016	4 Std.	RMI weiter programmiert.	
17.05.2016	3 Std.	RMI Fehler korrigiert, Projekt im Eclipse neu verbunden, betreffend Sync. Problemen	Unter anderem neue, bzw. wiederkehrende Probleme mit dem Git. Musste das komplette Projekt neu importieren. Hatte Hilfe durch Olivia, da Sie das Problem bereits auch hatte.
17.05.2016	3 Std.	Testing Persister OrtDAOTest(2), Probleme mit PK(0,5), Referenzen(0,25), BugFixing(0,25)	Erste Versuche beim Tests schreiben. (erste Projektbezogene Versuche). Etwas holpriger Start, da ich mit dem Vorgehen nicht mehr ganz vertraut war. Danach weiterarbeiten mit der Datenbank und dem Skript.
18.05.2016	2 Std.	Testing Persister bearbeiten	Testklassen weiter bearbeitet vom Vorabend.
18.05.2016	1 Std.	RMI Kontrolle	
18.05.2016	1,5 Std.	RMI Verbindungstest mit Pascal	Hatten diverse Probleme wegen Der Firewall, HSLU Netz usw. Konnten erst später die wirklichen Probleme angehen betreffend der Verbindung.
19.05.2016	2 Std.	Tests schreiben(BenutzerManagerTest), Hilfe von Luca erhalten	Luca war schon ziehmlich fit beim Tests schreiben. Konnte bei Ihm nachfragen und so Hilfestellung bekommen.

19.05.2016	2,5 Std.	RMI File überarbeiten bezüglich Properties	Nun neu nicht mehr fix codiert der Port. Jedoch sind Probleme betreffend des Codes entstanden. Problem konnte noch nicht gelöst werden, da String übergeben wird, aber INT erwartet.
20.05.2016	3 Std.	RMI Verbindungsfehler probieren zu beheben.	Leider immer noch Probleme mit der Property Datei. Sehe bislang noch keinen Weg, das Problem beheben zu können. Recherchiere im Internet darüber, da ich selber hier an meine Grenzen stosse.
20.05.2016	1 Std.	Verbindungstest mit Pascal	Neue Tests mit Pascal, diesmal konnten wir fast direkt zu den richtigen Problemen bzw. Tests gehen, da wir nun genug Erfahrung gemacht haben mit unnötigen Überlegungsfehlern bei der Verbindung.
20.05.2016	1 Std.	JavaDoc nachführen	JavaDoc Korrekturen bzw nachführen in einzelnen Klassen.
20.05.2016	2 Std.	GUI Korrekturen	Habe Erfahrungen mit dem SceneBuilder gemacht. Habe schon einem Moment nicht mehr damit gearbeitet. Konnte daher nicht direkt zu 100% arbeiten. Pascal erklärte mir zu Beginn dann schnell das Wichtigste.
20.05.2016	3 Std.	Dokumentation durchlesen, Rechtschreibung und Satzstellung kontrollieren. Änderungen vornehmen. Texte erweitern	Habe freie Zeit im Moment, daher lese ich die Dokumentation und bearbeite Sie teils schon. Vorarbeit, damit danach weniger zu tun ist.
23.05.2016	2 Std.	Kapitel 18 Dokumentation Deployment einarbeiten und begonnen zu schreiben	Habe mit dem Kapitel 18 in der Doku begonnen. Musste mich zuerst darauf vorbereiten. Habe dann noch Informationen darüber von Herrn Sucur erhalten (hatten eine Sitzung bezüglich des Deployment) Arbeitszeit 1h, Lernzeit 1h
23.05.2016	1 Std.	Tests mit DB Einrichtung usw.	Notierte Schritte bezüglich des Deployment getestet.
24.05.2016	1 Std.	Visio DB überarbeiten, kontrollieren	Visio Version unsereres ERD war nicht mehr aktuell, habe dies überarbeitet. (später dann bemerkte ich dass wir das ERD nur oberflächlich gezeichnet hatten (ohne Attribute))

24.05.2016	1 Std.	Kapitel 18 Doku Deployment weiter bearbeiten	Rücksprache mit Olivia genommen bezüglich der *.jar und *.bat Dateien.
24.05.2016	4 Std.	Datenbankkapitel in Dokumentation (1h) schreiben(zuerst informieren und Informationen sammeln (1h)) und ERD im Visio mit Luca zeichnen (1h). Dann noch Dokumentation allgemein durchlesen (1h)	Attributliste wurde kontrolliert, Kapitel zusammen erfasst, nötige Informationen in die Dokumentation geschrieben. Visio hat einige Performance Probleme. Dies kostete einiges an Zeit. Auch waren am Ende zwei drei technische Hindernisse uns im Weg. (Qualität des Diagramms z.B.)
25.05.2016	1 Std.	Dokumentation Kapitel 10 überarbeitet	
25.05.2016	2 Std.	Fachliche Struktur und Modelle mit Dominik zeichnen	Zuerst auf ein Blatt, dann im EA
26.05.2016- ABGABE	10 Std.	Arbeiten an Summary, Doku und Präsi	Geschätzte Zeit
TOTAL	95 Std.		

21.5 Olivia Wassmer

21.5.1 Individuelles Management Summary zum Projekt Feukora

Im Projekt Feukora habe ich zu Beginn den Themenbereich Dokumentation übernommen. Ich habe die Dokumentation vorbereitet, sowie die ersten Kapitel begonnen zu schreiben (1, 3, 4) resp. zu verteilen ans ganze Team.

Zusätzlich habe ich mir die Software Architektur sowie das Deployment mit Pascal Steiner geteilt. Die Software Architektur war zu Beginn eine Herausforderung, mit Alexandra Lengen habe ich schwierigere Kapitel und Grafiken an einem Wochenende geplant und schliesslich erstellt (Bausteinsicht, Verteilungssicht). Schliesslich ging es ans Klassendiagramm. Dieses haben wir alle zusammen erarbeitet, ich habe es anschliessend nochmal in 3 Varianten skizziert damit wir uns schliesslich für eine Architektur entscheiden können.

Als es ans Programmieren ging, startete ich mit den Modelklassen da diese noch nicht verteilt waren. Das gestaltete sich schwieriger als erwartet da wir das noch nicht durchgedacht hatten. Anschliessend half ich beim Persister-Testing da Luca noch etwas Hilfe brauchte. Schliesslich konnte ich meinen Teil die Businesslogik beginnen. Zuerst das Interface dass alle weiterarbeiten konnten. Dann die Implementierung. Für die Grenzwerte-Überprüfung wollte ich in der Businesslogik eine Überprüfungsmethode erstellen. Ich plante alles so gut als möglich durch und begann die Struktur, für die Fertigstellung zog ich meinen Bekannten zur Hilfe bis schliesslich alles funktionierte. Die Hashmap fand ich besonders gut da wir diese nicht oft praktisch anwendeten im Unterricht und es alles vereinfacht. Die Überprüfungsmethode für die Terminbelegung konnte ich ganz alleine erstellen. Als das abgeschlossen war Testete ich die ganze Businesslogik mithilfe JUnit und schloss diese Komponente ab.

Anschliessend hatte ich keine klare Rolle mehr, durch das entwickelte Wissen durch die vielen Tests und durch den Überblick über die Applikationsstruktur, sprang ich überall ein. Es gab viel zu tun, ich half Exceptions zu lösen und mit Programmierlogik und Methodenaufbau. Ich unterstützte die Client Programmierer mit dem Wissen über Businesslogik und Modelklassen Inhalt. So ging es weiter bis fast zum Schluss. F

Kurz vor Schluss musste ich mich zurückziehen um die Dokumentation machen zu können. Gleichzeitig beschäftigte ich alle Mitglieder die gerade nichts zu tun hatten.

Dass gegen Schluss eine allgemeine Aufgabe-Stimmung herrschte wollte ich mir nicht gefallen lassen und setzt mich nochmal mit vollster Energie an das Projekt und bereite Deployment vor, und versuchte nochmal den Client Extern zum laufen zu bringen, obwohl alles aussichtslos erschien. Und so kamen wir oder besser ich, zum Schluss. Auch wenn die Applikation perfekt wurde, bin dich doch stolz. Ich haben meiner Meinung nach enorm viel geleistet und auch wenn es sich teils schlecht ausgeglichen anfühlte, war es eine gute Erfahrung..

Zu sagen ist noch, dass trotz der Vor-Definierung der Rollen, während des Projektes alles etwas verschwamm, vor allem für mich. Ich hatte neben meinen Rollen viel Organisatorisches zu tun, ich koordinierte zu Ende die Aufgaben für alle Teammitglieder und war an allen Enden und Ecken am Helfen um das Projekt möglichst fertig zu bekommen was gegen Schluss unmöglich erschien. Somit war ich schlussendlich eher Teamleiter als nur ein Mitglied. Und GitHub möchte ich hier auch noch erwähnen, dieses Tool bereitete uns allen sehr viele Probleme, die ganze „verschwendete Zeit“ habe ich bei mir nicht aufgeschrieben jedoch hielt es mich Minimum 10h auf!

Wichtigste Aktivität(en) im Projekt Feukora :

Businesslogik (20%), Testing Persister, Businesslogik, Webservice (40%), Troubleshooting und Organisatorisches (20%),

Weitere Aktivität(en):

Modelklassen (10%), Dokumentation inkl. UseCases (10%)

21.5.2 Individuelles Portfolio zum Projekt Feukora

Datum eventuell Phase	Dauer Aufand in h	Tätigkeit / Aufgabe	Bemerkung/Erkenntnis
15.03.2016	1.0 Std.	Template für die einheitliche Gestaltung der Rollenbeschriebe erstellt. Und die <u>Rollenbeschriebe für Deployment und Dokumentation</u> angefertigt. Welche ich dann an die Teamkollegen aus den gleichen Bereichen zur Ergänzung weiterleitete. Den <u>Rollenbeschrieb für die Softwarearchitektur</u> zusammen mit Pascal Steiners Anmerkungen und Vorschlägen vervollständigt und für unserem Projektlead für den Abgabetermin fertiggestellt.	Durch das Rollenbeschrieb Template wussten alle direkt welche Fragen sie für die Rollen beantworten sollten und konnten dies einheitlich gestalten. Die Rollenbeschriebe wurden so, nach meinen Vorschlägen von der Gruppe übernommen. Der SA Rollenbeschreib konnte pünktlich abgegeben werden.
20.03.2016	0.5 Std.	Beschrieb der mir zugeteilten <u>UseCases</u> anhand der erstellten Vorlage.	Meiner Meinung nach haben wir zu viele UseCases da jedes Requirement ein UseCase ist... Trotzdem versuchte ich die überschneidenden UseCases möglichst gut zu beschreiben zur Weiterbearbeitung.
20.03.2016	1.5 Std.	Einlesen ins freiwillige Buch "Effektive Softwarearchitekturen" um die <u>Arc42 Dokumentation</u> vollständig ausfüllen zu können. Ausführung des gesamten <u>Kapitels 1</u> .	(Einlesen zähle ich nicht zum Aufwand in Std.) Vieles war noch schwierig auszufüllen ohne das nötige Wissen resp. die Sicherheit. Einige Teile musste ich auch noch freilassen z.B. UseCases da die erst in der Teamsitzung am 21.3.16 abgegeben werden.
17.04.2016	0.5 Std.	Vorbereitung einer kleinen Präsentation mit allen Informationen und einem <u>Vorschlag für einen Zeitplan</u> damit wir rechtzeitig fertig werden. (Präsentation mit Zeitplan und grober Architektur im Anhang)	Übers Wochenende habe ich mich gut vorbereitet (Lernzeit ca. 3h) für die Teamsitzung. Somit habe ich das Thema zu diesem Zeitpunkt am besten verstanden und konnte allen weiterhelfen. Da mir spätestens da auch der Aufwand bewusst wurde habe ich einen Zeitplan erstellt und konnte so die dringlichkeit ans Team vermitteln.
18.04.2016	1.0 Std.	Während der Teamsitzung stand ich zur <u>Verfügung um Fragen</u> bezüglich Bausteinsicht zu klären. Zudem habe ich alle über die grobe Architektur resp. Bausteinsicht <u>aufgeklärt</u> .	
23.04.2016	1.0 Std.	<u>Überarbeitung meiner UseCases Beschreibung</u> die wir jetzt korrigiert und von den Logischen Aktivitäten	Schade das wir diesen ersten Schritt falsch verstanden haben, wir haben dadurch viel Zeit verloren, und

		und nicht von den Requirements abhängig gemacht haben. Zudem habe ich meine <u>UseCases gezeichnet</u> mit dem EnterpriseArchitect.	eigentlich wäre es logisch gewesen das etwas nicht stimmen kann im ersten Entwurf. Der Enterprise Architect ist sehr mächtig und wirklich nützlich, jedoch braucht es viel Zeit sich einzuarbeiten, zum Glück hatten wir Herr Stirnimann als Coach.
24.04.2016	0.5 Std.	<u>Vorbereitung der Teamsitzung.</u> Update der Zeitplanung. Übersicht über ToDo's.	
30.04.2016	4.0 Std.	<u>Zeichnen</u> 3 Varianten des <u>Klassendiagramms</u> zur Übersicht fürs Programmieren sowie schreiben der Kapitel 3-5	nur Entwurf für Programmieren, richtiges Diagramm wird am schluss generiert
01.05.2016	0.5 Std.	<u>Vorbereitung der Gruppensitzung,</u> kleine Übersicht über Status und Fortschritt sowie weiteren <u>Zeitplan.</u> Klassendiagramm gestern vorbereitet für Aufgabenverteilung.	
10.05.2016	2.0 Std.	<u>Programmierung der Modelklassen</u>	Da ich vom Mi-So krank war und ausfiel hat sich alles etwas verzögert, wir konnten jedoch alles gut untereinander vereinbaren da noch niemand so weit war um die Modelklassen zu brauchen.
11.05.2016	4.0 Std.	Programmierung, Anpassung, Kontrolle der Modelklassen zusammen mit Matthias	Nachdem ich am Vorabend alles alleine Programmiert hatte wollte ich noch eine Gegenkontrolle um sicher zu sein. Diese haben Matthias und ich zusammen gemacht und dann direkt noch Anpassungen gemacht.
11.05.2016	4.0 Std.	Programmierung Modelklassen Änderungen sowie Businesslogik angefangen, mit Clients abgeglichen	Die kontrollierte Version der Modelklassen mussten wir dann nochmal anpassen weil noch Konstruktoren, <code>toString</code> override und spezifische Annotationen etc. fehlten. Wir haben alles nochmal angepasst.
12.05.2016	2.0 Std.	Modelklassen nochmal neu überarbeitet und mit Persister überprüft	Irgendwie ergaben sich immer wieder Änderungen oder Ergänzungen mit Hinsicht auf Persister querys und Businesslogik also habe ich dies übernommen und angepasst.
13.05.2016	2.0 Std.	<u>Modelklassen fertiggestellt</u>	Schliesslich konnte ich die Modelklassen abschliessen vorerst natürlich Ergänzungen vorbehalten.
13.05.2016	1.0 Std.	Luca geholfen <u>Persister fertig zu stellen</u>	Langsam hat sich herauskristallisiert dass ich die Organisation des Projektes ein bisschen an die Hand nehme und allen helfe wo es gerade brennt, neben meinen eigenen Aufgaben. Darum habe ich dann auch

			erst beim Persister geholfen diesen fertig zu machen bevor ich weiter zur Businesslogik kam. So konnte der Persister effizienter abgeschlossen werden.
14.-15.05.16	8.0 Std.	<u>Programmierung des Businesslogik Interfaces & Implementation</u>	Direkt nach der Programmierung der Modelklassen musste ich so schnell wie möglich das Businesslogik-Interface liefern damit alle weiter arbeiten können. Dies war eine Herausforderung, da ich noch viel Zeit zum Verständnis und zur Herleitung benötigte. Ein anderes Mal wäre es vielleicht besser die Arbeiten nicht auf Bausteine sondern nach Priorität aufzuteilen so müsste niemand stressen und andere nicht warten. Für die Implementation hatte ich schliesslich mehr Zeit, schwierig blieb es da ich das alles mehr oder weniger zum ersten Mal so mache. Stolz bin ich auf die beiden Check-Methoden in der Businesslogik die habe ich mit etwas Hilfe aus meinem Umfeld implementiert. Vor allem die Repetition und Anwendung der Hashmap war cool, denn die haben wir letztes Semester nicht viel praktisch geübt.
16.05.2016	13.0 Std.	<u>Persister Testing & Bugfixing</u>	Der Persister war ja am 13.5 abgeschlossen nun gingen wir weiter zu den Tests. Ich habe den ersten Test und Aufbau (Logik) erarbeitet. Anschliessend haben mich Luca und Matthias unterstützt und die Implementation auf die anderen Testklassen übertragen.
17.05.2016	4.0 Std.	<u>Persister Testing & Bugfixing</u>	Das Testing nahm wie gelernt sehr viel Zeit in Anspruch, immer wieder fanden wir Fehler in den Modelklassen oder Fehler in der Reihenfolge der Testsequenz. Dazu kamen viele Exceptions die wir immer wieder googeln mussten oder mithilfe von Alessio untersuchen mussten. Irgendwann hatte ich für meinen TEil den Dreh etwas raus an welchen Ecken es liegen kann und wir konnten das Tempo etwas steigern.
18.05.2016	4.0 Std.	<u>Persister Testing & Bugfixing sowie allgemeine Hilfeleistung bei anderen Anliegen im Team</u>	Auch heute ging es weiter mit Testing und Bugfixing es zog sich ziemlich in die Länge um alle Errors zu lösen die

			Teils sehr schwer ersichtlich waren. Nebenbei stand ich allen zur Seite mit Fragen über die Logik oder Methodenstrukturen etc. ich hatte mich unterdessen sehr gut in das Thema und ins Programmieren eingearbeitet, durch die vielen Baustellen wo ich arbeitete und das Programmieren der Businesslogik. Überall wo ich nicht mehr weiter wusste haben wir professionelle Hilfe beigezogen.
19.05.2016	4.5 Std.	<u>Persist Testing & Bugfixing sowie allgemeine Hilfeleistung bei anderen Anliegen im Team</u>	Am Morgen sowie nach der Schule haben wir wie gestern weitergemacht.
20.05.2016	12.0 Std.	<u>Businesslogik Tests sowie mithilfe am GUI</u>	Endlich war der Persist getestet und alles hat vorerst funktioniert. Zum Abschluss habe ich eine Test Suite erstellt für den Persist. Anschliessend habe ich in der gleichen Logik die Businesslogik getestet leider hat die 3/4 des Tages in Anspruch genommen. Unteranderem weil ich wieder zwischen den verschiedenen Baustellen hin und her sprang. Alexandra war für einige Tage weg darum habe ich Pascal beim Programmieren der GUI Logik unterstützt und konnte ihm durch meine Kenntnisse über die bestehenden Logikmethoden unterstützen.
21.05.2016	6.0 Std.	<u>Fertigstellung Businesslogik Tests</u> vor allem wegen der zwei komplexen Überprüfungsmethoden (checkTermin und checkGrenzwerte. Anschliessend Unterstützung beim Client Intern programmieren und beim Abschluss des Webservices.	Abgesehen von den Tests, habe ich mehr mitgeholfen überall zu schauen was es noch braucht und wenn möglich direkt zu helfen.
23.05.2016	13.0 Std.	<u>Verbindung RMI und Webservice</u> fixen mithilfe von Herrn Sucur. Webservice Tests anpassen. Client Intern mithilfe beim Test zur Verfügung-Stellung der Server seite mit meinem Laptop. Anschliessend weiterarbeit an der <u>Dokumentation</u> .	

23.05.2016	2.0 Std.	<u>Troubleshooting am Client Intern</u> , erst Verbindung geflickt, dann GUI Bereich Mitarbeiter gefixt, Entwurf für restliche RMI Klassen vorbereitet	Die Verbindung zuerst gefixt dann konnte ich erste funktionale Tests machen und habe fehler im Client entdeckt. Mithilfe meines Bekannten konnte ich einige schwierigere Fehler lösen und RMI soweit vorbereiten dass meine Teamkollegen morgen übernehmen können und den Client Intern hoffentlich fertig stellen können.
24.05.2016	7.0 Std.	<u>Aufgabenverteilung</u> für Schlussspurt, Erklärung des Fix von RMI<>Client von gestern an Client Team. Und parallel weiterarbeit an der <u>Dokumentation</u> Kap. 5, Kap. 7 sowie 8.0-8.7	Underdessen bin ich vor allem mit der Projektleitung und Organisation beschäftigt. Es ist sehr stressig alles zu koordinieren da nicht jeder gleich schnell arbeitet oder gleich gut recherchiert. Trotzdem verfolge ich weiter mein Ziel die Dokumentation.
24.05.2016	3.0 Std.	<u>Bugfix Client Intern "Rapport anpassen"</u>	Heute nachdem alle zuhause waren habe ich zuhause noch mit meinem Bekannten versucht den ungelösten Fehler zu finden. Wiedereinmal hatte ich wie das ganze Team enorme probleme mit dem GIT dies kostete viel Zeit wie auch der Fehler. Schlussendlich funktionierte die von Pascal und Alex weitergegebene Methode jedoch. Nur die Grenzwertkontrolle aus meiner Businesslogik Implementation konnte nicht benutzt werden. Vielleicht müssen wir diese aus Zeitgründen nun weglassen.
25.05.2016	12.0 Std.	Schlussspurt bei der <u>Dokumentation</u> meinen Individuellen Dokumenten und beim <u>Korrekturlesen</u> , nebenbei Gedankenstütze für Pascal am Client Intern	Viele Dokumente mussten nochmal überarbeitet werden, dies beanspruchte nochmal viel Zeit. Zum Glück hatte ich Unterstützung vom Team für einige Kapitel so konnten wir doch alles Abschliessen.
26.05.2016	15.0 Std.	<u>Dokumentation</u> fertig stellen und somit alles Individuelle einsammeln sowie <u>Deployment</u> koordinieren, schlussendlich selber schreiben und Ressourcen erstellen sowie testen	Der letzte Tag, das Dokument musste gedruckt werden aber es war noch nichts fertig. Da die Teamstimmung langsam eher Aufgeben anstatt Arbeiten war und alle extrem enttäuscht waren für ihren enormen Arbeitseinsatz kein funktionierendes (Client Extern) Endprodukt zu sehen. Ich trieb alles an und verteilte nochmal alles was noch ausstehend war. Ich selber kontrollierte die Dokumentation nochmal. Vieles fehlte und auch der Client Extern war nicht fertig. Als das Dokument gedruckt war und wir „nur“ noch das

			Deployment vorbereiten mussten. Arbeitete ich fast alleine und stellte alles zusammen. Setzte mich nochmal an den Client Extern und gab alles. Und vielleicht hat es sich ja gelohnt.
27.05.2016	0.5 Std.	Projektarbeit abgeben (ungefähre Angabe)	Abgabe @ HSLU
28.05.2016	1.5 Std.	Deployment (ungefähre Angabe)	Deployment @ HSLU mit Alessio
28-29.05.2016	2.0 Std.	Präsentationsvorbereitung (ungefähre Angabe)	Ich weiss noch nicht ob und was ich präsentiere aber ich werde mich bestimmt darauf vorbereiten.
Total	133.0 Std.		

21.6 Pascal Steiner

21.6.1 Individuelles Management Summary zum Projekt Feukora

Mein Aufgabenbereich im Projekt Feukora der Gruppe B war sehr stark auf das technische ausgelegt. Zuerst war ich bei der Datenbankgruppe dabei. Wir erarbeiteten die Anforderungen an die Datenbank und erstellten das ERD. Dabei definierten wir zum grössten Teil die Entitäten und die Beziehungen zueinander. Durch das erhielten wir einen ersten Überblick. Danach setzte ich mich intensiv mit dem Referenzprojekt auseinander. Mit dem Verständnis des Referenzprojekts konnten wir ein erstes Konzept der Softwarearchitektur ausarbeiten. Beim Programmieren war ich nebst 2 Junit4 Test zum grössten Teil mit dem Client beschäftigt. Diesen habe ich in Zusammenarbeit mit Alexandra Lengen entworfen und ausprogrammiert.

Das Projekt war über alle Stufen hinweg sehr fordernd. Angefangen bei der Datenbank, bei welcher wir zu exakt vorgenommen und Stunden mit überarbeiten verbrachten. Schlussendlich hatten wir ein zu komplexes ERD welches nicht so umgesetzt werden konnte. Einer der grössten Fehler machten wir bei der Aufgabenaufteilung der Softwarearchitektur. Da wir alle gleichzeitig unsere Teile programmierten fehlte uns schlussendlich ein ganzheitlicher Überblick was bei den anderen festgelegt wurde. Ein Beispiel: Während wir im Client davon ausgingen das wir bei Aufruf eines Mitarbeiters nach Namen ein einzelnes Objekt bekommen, wurde uns schlussendlich vom RMI eine Liste zurückgeschickt. Ähnliche und weitere Probleme zogen sich durch das ganze Projekt durch.

Schlussendlich kann ich aber unser Projekt als gelungen betrachten, obwohl wir nicht alles so umsetzen konnten wie geplant. Dies war auch dem enormen Zeitaufwand zu verdanken welcher die Gruppe in dieses Projekt investierte. Wir haben aber dabei alle sehr viel gelernt und sind nun softwaretechnisch um einiges weiser geworden.

Wichtigste Aktivität(en) im Projekt Feukora :

Client intern und extern inkl. Verknüpfungen (70%),

Weitere Aktivität(en):

Software Architektur (15%), Datenbank(10%), Junit4 Test(5%)

21.6.2 Individuelles Portfolio zum Projekt Feukora

Datum eventuell Phase	Dauer Aufwand	Tätigkeit / Aufgabe	Bemerkung/Erkenntnis
19.03.2016	3.0 Std.	Dateien durchforstet und einen ersten Überblick über die Anforderungen gemacht	Es sind viele Fragen zu den Anforderungen aufgetaucht
21.03.2016	1.0 Std.	Datenbankentwürfe mit Luca und Matthias erstellt. Wir haben eine Attributliste und erste Entitäten erfasst.	Wir stellten fest, dass noch einige Unklarheiten herrschen. Diese sollten wir mal mit der Gruppe besprechen.
21.03.2016	2.0 Std.	Use Case erstellt und ausgearbeitet	Die Use Case anhand der vorgegeben Vorlage erstellt. Einige unklarheiten betreffend Definition eines Use Case aufgetaucht.

08.04.2016	2.5 Std.	Das Datenbank ERD überarbeitet und erweitert. Durch neue Inputs haben wir verschiedene Lösungen kreiert. Diese werden von Luca nochmals überprüft.	Durch fehlende Erfahrung wissen wir nicht genau welcher ERD Entwurf für unser Projekt der geeignete ist.
19.04.2016	1.5 Std	Use Case nach Rückmeldung verbessert	
24.04.2016	7.0 Std.	Das Referenzprojekt von Herr Succur analysiert. Zusätzlich habe ich den Enterprise Architect installiert. Ich habe versucht das Referenzprojekt in den Enterprise Architect zu importieren.	Ich sehe den Zusammenhang der verschiedenen Schichten noch nicht. Der Enterprise Architect ist sehr gross und auch sehr verwirrend.
01.05.2016	6.0 Std.	Dokumentation Kapitel 8 und 9 ausgearbeitet und die Use Cases überarbeitet	Es ist sehr schwierig verschiedene Teile auszufüllen, da ich mir das Endprodukt noch nicht richtig vorstellen kann. Ich glaube ich hatte aber einige gute Ideen und Inputs. Im Enterprise Architect habe ich ein Model-View-Controller Diagramm erstellt.
06.05.2016	3.0 Std.	UseCase Beschreibung 11, 12 und 13 erstellt, Grafiken mussten überarbeitet werden	Persönlich einige Schwierigkeiten beim Erstellen der Grafiken. Werden in UML 2 kompakt nur oberflächlich beschrieben und online sind mehrere Meinungen vertreten. Zb. stellt man einen Server auch als Akteur dar?
09.05.2016	8.0 Std.	Mit Alexandra angefangen den Client intern zu programmieren.	Wir hatten erhebliche Probleme nur schon das JavaFX zum laufen zu bringen. Schlussendlich brauchten wir ein neues Eclipse auf welchem das FX schon vorinstalliert ist. Danach haben wir angefangen das Login Gui mit dem DashboardGui zu verbinden. Wir mussten hier unsere FX Kenntnisse wieder auffrischen.
10.05.2016	5.0 Std.	MitarbeiterEventhandling programmiert	Ich musste sehr viel JavaFX wissen wieder auffrischen.
11.05.2016	10.0 Std.	Grosse fehler beim aufbau unseres EventHandlings entdeckt. Umstellen der gesamten aufbaustruktur des Clients	Wir haben das Eventhandling bis jetzt immer über eine switch gemacht. Damit hatten wir faktisch aber nur 1 Methode in der ganzen Klasse.
12.05.2016	5.0 Std.	Client Programmiert	Umstellen der grundstruktur des Clients
13.05.2016	10.0 Std.	Client Programmiert	Mitarbeiter EventHandling ausprogrammiert
17.05.2016	4.5 Std.	Am Client weiterprogrammiert	Fehler ausbessern (nullchecks einbauen)

18.05.2016	3.0 Std.	Client Programmiert	Eventhandling verbessert Gui ein wenig angepasst.
19.05.2016	4.0 Std.	Client Programmiert	Tableviews in den Übersichten programmiert. GUIs noch angepasst.
20.05.2016	12.0 Std.	Client Programmiert	Liegenschaftsbehandlung ausprogrammiert und Fehler ausgebessert. Kontaktbehandlung angepasst.
21.05.2016	11.0 Std.	Client Programmiert	Rapport erfassen und Rapport bearbeiten programmiert.
23.05.2016	9.0 Std.	Der Client versucht mit dem RMI zu verbinden	RMI wissen aufgefrischt und die RMI Implementationen in den verschiedenen Klassen programmiert und getestet
24.05.2016	6.0 Std.	RMI getestet und fehler am Eventhandling angepasst.	Da der RMI nicht funktioniert hatte, weiter getestet bis er funktionierte. Nachher noch weiter an den Eventhandlings programmiert. Als die Verbindung mit dem RMI funktionierte hatte ich viele weiter Fehler entdeckt
25.05.2016	11.0 Std	Funktionale Tests am Client durchgeführt und den Client verbessert	Hätten die RMI verbindung schon früher aufbauen sollen
26.05.2016	6.0 Std.	Management Summary geschrieben und Dokumentation verbessert	Bin froh das Projekt abzuschliessen, bin aber auch stolz auf das Resultat und die Lerneffekte.
TOTALI	112.5 Std		

22 Source-Code von selber beschriebenem Code

Der Sourcecode der Applikation Feukora ist aufgrund seines Umfangs im Anhang zu finden.

23 Anhang

23.1 Dokumentation Änderungen und Autoren

Verantwortlich für die Dokumentation ist laut Rollenverteilung Olivia Wassmer. Jedoch wurde das Dokument vom ganzen Team erarbeitet. Folgende Liste gibt Aufschluss über die Erst-Autoren und wer Gegengelesen hat. Schlussendlich wurde dann von der Verantwortlichen nochmal alles gegengelesen und in Druck gegeben.

Datum	Autor	Kapitel	Datum	Kontrolle
23.04.16	Olivia Wassmer	Kapitel 1	24.05.16	Matthias Perrollaz
20.03.16	Pascal Steiner	Kapitel 2	21.03.16	Olivia Wassmer
30.04.16	Olivia Wassmer	Kapitel 3	24.05.16	Matthias Perrollaz
30.04.16	Olivia Wassmer	Kapitel 4	24.05.16	Matthias Perrollaz
30.04.16	Olivia Wassmer	Kapitel 5	24.05.16	Matthias Perrollaz
30.04.16	Alexandra Lengen	Kapitel 6	24.05.16	Dominik Stirnimann
30.04.16	Olivia Wassmer	Kapitel 7	25.05.16	Dominik Stirnimann
25.04.16	Pascal Steiner	Kapitel 8	25.05.16	Olivia Wassmer (8.0-8.8) Dominik Stirnimann (8.9-8.24)
25.04.16	Pascal Steiner	Kapitel 9	25.05.16	Olivia Wassmer
25.04.16	Luca Raneri	Kapitel 10	25.05.16	Matthias Perrollaz
25.04.16	Luca Raneri	Kapitel 11	25.05.16	Dominik Stirnimann
25.05.16	Olivia Wassmer	Kapitel 12	24.05.16	Matthias Perrollaz
25.05.16	Olivia Wassmer	Kapitel 13	24.05.16	Matthias Perrollaz
20.03.16	Alexandra Lengen	Kapitel 14	25.05.16	Luca Raneri
div.	alle	Kapitel 15	23.05.16	Dominik Stirnimann
25.05.16	Olivia Wassmer	Kapitel 16	26.05.16	Alexandra Lengen
24.05.16	Matthias Perrollaz	Kapitel 17	25.05.16	Olivia Wassmer
23.04.16	Luca Raneri	Kapitel 18	25.05.16	Olivia Wassmer
25.05.16	Alexandra Lengen	Kapitel 19	25.05.16	Pascal Steiner
24.05.16	Matthias Perrollaz	Kapitel 20	25.05.16	Alexandra Lengen
div.	alle	Kapitel 21	-	keine
div.	alle	Kapitel 22	26.05.16	Olivia Wassmer (Vorbereitung für Druck)
25.05.16	Olivia Wassmer	Kapitel 23	25.05.16	Dominik Stirnimann

23.2 Anwesenheitsliste

INM21 - FS2016

Projektarbeit - Anwesenheitskontrolle

Gruppe B

Unterschrift Gruppenleiter: 

	Name und Vorname	02.05.2016	09.05.2016	12.05.2016	19.05.2016	23.05.2016	26.05.2016	30.05.2016
1	Lengen Alexandra							
2	Perrollaz Matthias	MP	MP	MP	MP	MP	MP	MP
3	Raneri Luca							
4	Steiner Pascal							
5	Stirnimann Dominik							
6	Vogel Silvan							
7	Wassmer Olivia							
8	Wunderer Carlos							

23.3 Genaue Aufgaben-Zeitverteilung im Projekt

23.4 Anhang Projektleitung

AW: Grobe Aufgabenverteilung Version 2

Mathis Erwin HSLU I

Do 17.03.2016 08:30

Au: Stirnimann Dominik W.BSCW/Z.1501 <dominik.stirnimann@stud.hslu.ch>

Sehr geschätzte Herr Stirnimann

Vielen Dank für Ihre Rückmeldung.

Zu Ihrer Frage:
«Sollte pro Requirement Ziel ca. 1-2 Used Cases erstellt werden oder für jedes Ziel vier Used Cases nach dem CRUD Prinzip aufgestellt werden?»

Es gibt dafür keine allgemeingültige Regel. Machen Sie es in Ihrem Projekt einfach konsequent immer gleich.
Die Use Cases nach dem CRUD-Prinzip zu notieren ist in vielen Fällen nicht sehr aussagekräftig,
weil oft nur «banale» Tatsachen notiert werden leben (CREATE, READ,...).
wenn aber z. B. bei einem dieser Vorgänge wichtige «andere Sachen» noch geschehen z. B.
(theoretische Annahme... nicht auf Anforderungen von unserem Projekt angepasst)
beim Löschen eines Kunden könnte es notwendig werden auch alle seine Brenner etc. zu
löschen und daraus auch alle Messdaten zu diesem Brenner etc. dann kann es SEHR wohl
sinnvoll UND notwendig sein einen oder mehrere CRUD-Vorjäge eines Use Cases sehr
genau zu dokumentieren ist.

Gehen Sie immer von der Annahme aus, dass Sie Ihre Use-Case-Dokumentation einem neuen Mitarbeiter in die Hände drücken (Betriebswirtin) und diese soll sich innerhalb einer Stunde einen guten Überblick über die abzudeckenden Geschäftsprozesse machen können.
Oft können auch UML-Notizen in den Use Cases die Diagramme noch klarer verständlich machen
oder Sie notieren diese «Zusatzerläuterungen» in der UseCase – Dokumentation als spezielle
Bemerkung (neben Namen, Szenario, Pre- und Post-Condition etc.).

Mit freundlichen Grüßen

Erwin Mathis

Hochschule Luzern
Wirtschaft

IWI (Institut für Wirtschaftsinformatik)
Erwin Mathis
Dipl. Inf. lic. phil. nat.
Lecturer
T direkt +41 41 228 42 69
erwin.mathis@hslu.ch

**AW: INM21: Projekt Feukora - Zusätzliche Erläuterungen /
Fragen beantwortet auf llas**

Stürmann Dominik W.BSCWIVZ.1501

Do 17.03.2016 08:49

An: Mathis Erwin HSLU I <erwin.mathis@hszu.ch>;

Sehr geehrter Herr Mathis

Gerne bestätige Ich Ihnen, dass wir (Gruppe B) die Information über die Projektänderungen/
Zusätzlichen Erläuterungen erhalten haben.

Vieelen Dank.

Freundliche Grüsse

Dominik Stürmann

Von: Mathis Erwin HSLU I

Gesendet: Donnerstag, 17. März 2016 08:41

An: Stürmann Dominik W.BSCWIVZ.1501; Willi Bettina W.BSCWIVZ.1501; Tomic Tamara W.BSCWIVZ.1501; Hug

Claudio W.BSCWIVZ.1501

Cc: Sucur Jordan HSLU I

Betreff: INM21: Projekt Feukora - Zusätzliche Erläuterungen / Fragen beantwortet auf llas

Sehr geschätzte Projektleiterinnen

Sehr geschätzte Projektleiter

Beachten Sie bitte die neuen Dokumente auf llas «04. Projektänderungen».

Es handelt sich dabei nicht um echte Änderungen sondern um
zusätzliche Erläuterungen/Erläuterungen auf Fragen von den Gruppen.

Ich bitte um die «Erhalts-Bestätigung» für dieses Mail!

Erwin Mathis

**Hochschule Luzern
Wirtschaft**

IWI (Institut für Wirtschaftsinformatik)

Erwin Mathis

Dipl. Inf. lic. phil. nat.

Lecturer

T direkt +41 41 228 42 89

erwin.mathis@hszu.ch

Zusätzliche Erläuterungen / Fragen beantwortet in File auf IlIAS

Mathis Erwin HSLU I

Mi 20.04.2016 21:25

A: n.Stürmann Dominik WBSCW/Z.1501 <dominik.stürmann@stud.hslu.ch>; Willi Bettina WBSCW/Z.1501 <bettina.will@stud.hslu.ch>; Tomi Tamara WBSCW/Z.1501 <tamara.tomic@stud.hslu.ch>; Hugo Claudio WBSCW/Z.1501 <claudio.hug@stud.hslu.ch>;

c: Sucur Jordan HSLU I <jordansucur@hslu.ch>;

✉ 1 Anlage

2016_F5_IMM21_protokoll_20160414.pdf

Sehr geschätzte Projektleiterinnen
Sehr geschätzter Projektleiter

Beachten Sie bitte das neue Dokument auf IlIAS «04. Projektänderungen»
(auch diesem Mail beigelegt).

Es enthält einige Fragen von Frau Willi, welche wir in einem
Gespräch geklärt haben.

Aus meiner Sicht KEINE NeuAnforderungen sondern einige «Präzisierungen».

Ich wünsche Ihnen auch von meiner Seite weiter viel Energie und
Abé: Es chund scho gueil!!

Erwin Mathis

Hochschule Luzern
Wirtschaft

IWI (Institut für Wirtschaftsinformatik)

Erwin Mathis

Dipl. Inf. lic. phil. nat.

Lecturer

T direkt +41 41 228 42 89

erwin.mathis@hslu.ch

Zentralstrasse 9 CH-6002 Luzern
T +41 41 228 41 70, F +41 41 228 41 71
www.hslu.ch/iwi

AW: Grobe Aufgabenverteilung Version 2

Zentralstrasse 9, CH-6002 Luzern
T +41 41 228 41 70, F +41 41 228 41 71
www.hslu.ch/iwi

Mathis Erwin HSLU I

Do 17.03.2016 08:30

An: Stimmann Dominik WBSCWIVZ.1501 <dominik.stimmann@stud.hslu.ch>

Sehr geschätzte Herr Stirnimann

Vielen Dank für Ihre Rückmeldung.

Zu Ihrer Frage:

«Sollte pro Requierment Ziel ca. 1-2 Used Cases erstellt werden oder für jedes Ziel vier Used Cases nach dem CRUD Prinzip aufgestellt werden?»

Es gibt dafür keine allgemeingültige Regel. Machen Sie es in Ihrem Projekt einfach konsequent immer gleich.

Die Use Cases nach dem CRUD-Prinzip zu notieren ist in vielen Fällen nicht sehr aussagekräftig, weil oft nur «banale» Tatsachen notiert werden (eben CREATE, READ,...). ... wenn aber z.B. bei einem dieser Vorgänge wichtige «andere Sachen» noch geschehen z.B. (theoretische Annahme... nicht auf Anforderungen von unserem Projekt angepasst), beim Lösen eines Kunden könnte es notwendig werden auch alle seine Brenner etc. zu löschen und daraus auch alle Messdaten zu diesem Brenner etc. dann kann es SEHR wohl sinnvoll UND notwendig sein einen oder mehrere CRUD-Vorgänge eines Use Cases sehr genau zu dokumentieren ist.

Gehen Sie immer von der Annahme aus, dass Sie Ihre UseCase-Dokumentation einem neuen Mitarbeiter in die Hände drücken (Betriebswirtin!) und diese soll sich innerhalb einer Stunde einen guten Überblick über die abzudeckenden Geschäftsprozesse machen können. Oft können auch UML-Notizen in den Use Cases die Diagramme noch klarer verständlich machen oder Sie notieren diese (z.B. «Zusatzaufmerksamkeiten») in der UseCase – Dokumentation als spezielle Bemerkung (neben Namen, Szenario, Pre- und Post-Condition etc.).

Mit freundlichen Grüßen

Erwin Mathis

Hochschule Luzern
Wirtschaft

IWI (Institut für Wirtschaftsinformatik)
Erwin Mathis
Dipl. Inf. lic. phil. nat.
Lecturer

T direkt +41 41 228 42 59
erwin.mathis@hslu.ch

Zusätzliche Erläuterungen / Fragen beantwortet in File auf Ilias

Mathis Erwin HSLU 1

MI 20.04.2016 21:25

Au: Stirnimann Dominik WBSCWIV/Z.1501 <dominik.stirimann@stud.hslu.ch>; Willi Bettina WBSCWIV/Z.1501

<bettina.willi@stud.hslu.ch>; Tomic Tamara WBSCWIV/Z.1501 <tamara.tomic@stud.hslu.ch>; Hug Claudio

WBSCWIV/Z.1501 <claudio.hug@stud.hslu.ch>; Sucur Jordan HSLU 1 <jordan.sucur@hslu.ch>;

0 1 Anlage

2016_FS_INM2_1_protokoll_20160414.pdf

Sehr geschätzter ProjektleiterInnen

Sehr geschätzte Projektleiter

Beachten Sie bitte das neue Dokument auf Ilias «04. Projektkänderungen»
(auch diesem Mail beigelegt).

Es enthält einige Fragen von Frau Willi, welche wir in einem
Gespräch geklärt haben.

Aus meiner Sicht KEINE Neuansforderungen sondern einige «Präzisierungen».

Ich wünsche Ihnen auch von meiner Seite weiter viel Energie und
Ärge: Es chund scho ghet!!!

Erwin Mathis

Hochschule Luzern
Wirtschaft

IWI (Institut für Wirtschaftsinformatik)
Erwin Mathis
Dipl. Inf. lic. phil. nat.

Lecturer

T direkt +41 41 228 42 89
erwin.mathis@hslu.ch

Zentralstrasse 9, CH-6002 Luzern
T +41 41 228 41 70, F +41 41 228 41 71
www.hslu.ch/iwi

Feukora

Wie geht es weiter?

Software Architekten - Diagramme

- Kapitel 1 (Aufgabenstellung, Qualitätsziele, Stakeholder)
- Kapitel 2 (Randbedingungen technisch, organisatorisch, Konventionen)
- Kapitel 3 (Kontextabgrenzung fachlich, technisch)
- Kapitel 4 (Lösungstrategie)

- Kapitel 5 (Bausteinsicht)
- Kapitel 6 (Laufzeitsicht)
- Kapitel 7 (Verteilungsicht)

Software Architekten - Entscheide

- Kapitel 8 (K erzählt wie wird was behandelt. Details wie z.B. Sicherheit "eher Kontrolle haben wir an alles gedacht, allenfalls ergänzen")
- Kapitel 9 (Entwurfsentscheidungen)
- Kapitel 10 (Qualitätszenarien "Qualitätsziele detailliert")
- Kapitel 11 (Risiken)

- Kapitel 16 (Klassendiagramme) Entwurf 3 Varianten!

Dokumentationen - individuell

- Kapitel 14 (Requirements Engineering)
- Kapitel 15 (UseCase Dokumentation inkl. Diagramm)

Was ist noch offen?



NEXTÜ Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
18.4	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
Heute	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

Schnittstellen (DB), TDD, Programmieren erledigt bis 15.5

NEXTü

- ▷ PROGRAMMIEREN!!!
- ▷ Kapitel 18 (TDD und J unit - min 2 Fälle TDD)
- ▷ Kapitel 20 (Datenbank Dokumentation)
- ▷ Kapitel 22 (SourceCode und Javadoc) >> bis 15.5

SOONü Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

Dokumentation erledigt bis 16.5

SOONü Dokumentation

- ▷ Kapitel 12 + 13 (Glossar, Diverses)
- ▷ Kapitel 21 (individuelle Beiträge zusammentragen)
- ▷ Kapitel 22 (Code zusammentragen und darstellen)
- ▷ Kapitel 23 (Anhang, evtl. Quellenverzeichnis)
- ▷ Kapitel 21 (Projektmitgliederbeiträge Portfolio und Management Summary) >> bis 16.5

< >> bis 16.5

SOONü Testing & Deployment

- ▷ Kapitel 19 (Funktional Tests)
- ▷ Kapitel 17 (Deployment Informationen) >> bis 22.5
- ▷ Spätestens bis 27.5 Deployment HSLU

Terminplan definitiv

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

Alle "Uhrzeiten bis 22.5
 ▷ SA 1 Diagramme bis 22.5
 ▷ SA 2 Erreichende bis 25.5
 Alle "TDD und Programmieren bis 15.5
 ▷ DS Schrittsystem programmieren bis 15.5
 ▷ DS Dokumentation von Modulen und individuelle Dokumentation von Modulen
 T & D: Funktional Tests und Deployment bis 22.5
 ▷ DS/F Dokumentation abschließen und Dokumentation vorbereiten
 ▷ Präsentation: "LATEST" Arbeit in HSLU
 ▷ Dokumentar bis 27.5
 ▷ All: Präsentation allen bis 29.5
 ▷ All: Dokumentar bis 30.5

Feukora

Wie geht es weiter?

Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
24.5						

Software Architekten - Diagramme

- § Kapitel 1 (Aufgabenstellung, Qualitätsziele, Stakeholder)
- § Kapitel 2 (Randbedingungen technisch, organisatorisch, Konventionen)
- § Kapitel 3 (Kontextabgrenzung fachlich, technisch)
- § Kapitel 4 (Lösungsstrategie)
- § Kapitel 5 (Bausteinsicht)
- § Kapitel 6 (Laufzeitsicht)
- § Kapitel 7 (Verteilungssicht)

↳ Bis 22.4

Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
24.5						

SA → Diagramme erledigt bis 22.4

Fachlicher Kontext

Technischer Kontext

Software Architekten - Entscheide

a Kapitel 8 (Konzepte wie wird was behandelt, Details wie z.B. Sicherheit / eher Konkrete haben wir an alles gedacht allenfalls ergänzen)
 a Kapitel 9 (Entwurfsentscheidungen)
 a Kapitel 10 (Qualitätszonen / Qualitätsziele detailliert)
 a Kapitel 11 (Risiken)

 a Kapitel 16 (Klassendiagramme)

<> bis 25.4

Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

SA : Entscheide erledigt bis 25.4

Dokumentationen - individuell

a Kapitel 14 (Requirements Engineering)
 a Kapitel 15 (UseCase Dokumentation inkl. Diagramm)
 <> bis 22.4

a Kapitel 21 (Projektmitgliederbeiträge Portfolio und Management Summary)
 >> bis 16.5

Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

UseCases gezeichnet und kontrolliert bis 22.4

TDD Programmieren & Schnittstellen

a Kapitel 18 (TDD und J unit - min 2 Fälle TDD)
 a Kapitel 20 (Datenbank Dokumentation)
 a Kapitel 22 (SourceCode und Javadoc)
 >> bis 15.5

Terminplan

Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

Schnittstellen (DB), TDD, Programmieren erledigt bis 15.5

Dokumentation						
Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

< >> bis 16.5

Dokumentation erledigt bis 16.5

Terminplan						
Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

Dokumentation erledigt bis 16.5

Testing & Deployment						
Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

< >> bis 22.5

Spättests bis 27.5 Deployment HSLU

Funktions Tests und Deployment erledigt bis 22.5
Präsentation erledigt bis 25.5

OBLIGATORISCHES DEPLOYMENT 27.5

Terminplan definitiv						
Mo	Di	Mi	Do	Fr	Sa	So
Heute	19.4	20.4	21.4	22.4	23.4	24.4
25.4	26.4	27.4	28.4	29.4	30.4	1.5
2.5	3.5	4.5	5.5	6.5	7.5	8.5
9.5	10.5	11.5	12.5	13.5	14.5	15.5
16.5	17.5	18.5	19.5	20.5	21.5	22.5
23.5	24.5	25.5	26.5	27.5	28.5	29.5
30.5						

Allgemeine Dokumentationen bis 22.5
SA-Diagramme bis 22.5
SA-Einschreibe bis 25.5
Allgemeine TD und Programmieren bis 15.5
DB-Schichtenstellen programmieren bis 15.5
DB-Dokumentationen erstellen und individuelle Präsentation bis 15.5
T & D - Funktional Tests und Deployment bis 22.5
D&K P - Dokumentation abschließen und Präsentation vorbereiten
Dokumentation bis 22.5 abgeschlossen
Präsentation bis 25.5
Allgemeine Präsentation ab 26.5
Vorstellung bis 27.5

Grüne Kategorie	
Betreff:	Projektbesprechung Teamitzug Feukora
Ort:	Wird noch bekannt gegeben
Beginn:	Mo. 14.03.2016 <input type="button" value="..."/> 16:30 <input type="button" value="..."/> <input checked="" type="checkbox"/> Ganztägiges Ereignis
Ende:	Mo. 14.03.2016 <input type="button" value="..."/> 17:30 <input type="button" value="..."/>
Tasklist	
XX Beschlussprotokoll 1. Sitzung (wer macht das?) - Olivia	
<ul style="list-style-type: none"> • XX Rollen [Architekten] 5.34 • XX Standortbestimmung, was wurde bisher gemacht? Datenbank, GUI - noch keine • Wie siehts mit Used Cases aus? -> Datum für eigene used cases festlegen+ Format beschreiben Vorlage? Mathis Fragen ob gleiche Used Cases oder immer unterschiedliche CRUD (Create, Read Update, Delete) • Wie weiter? • Szenarios? Mathis fragen • Kalender erstellen? 	
Trial und error vermeiden	
Reminder:	
<ul style="list-style-type: none"> • XX individuelles Portfolio (Vorlage auf ilias) führen • XX Tipps und Empfehlungen lesen (auf ilias) Gibt einige Anhaltspunkte welche gemacht werden müssen oder nicht gemacht werden sollen. 	
<ul style="list-style-type: none"> • Selbst: Mit Alexandra Requirements koordinieren. <ul style="list-style-type: none"> • Wann und wie? Warten? 	
Tasks:	
Vorlage Used Cases	
Mathis Mail – Used cases nur 1 ner oder nach Crud Prinzip	

Grüne Kategorie	
Betreff:	Teammeeting Feukora
Ort:	
Beginn:	Mo. 10.04.2016 <input type="button" value="..."/> 11:00 <input type="button" value="..."/> <input checked="" type="checkbox"/> Ganztägiges Ereignis
Ende:	Mo. 10.04.2016 <input type="button" value="..."/> 13:30 <input type="button" value="..."/>
Datenbank:	
<ul style="list-style-type: none"> • Wie sieht aus? • Was braucht ihr noch? • Weitere Schritte? 	
Dokumentation Arc42:	
<ul style="list-style-type: none"> • Wie weit? • Wo kann man helfen? • Was braucht ihr noch? • Was sind die Pläne? 	
 GUI: Anhand der Vorlagen Werte erstellen mithilfe von Datenbank	
Mögliche Fenster zum Erstellen:	
<ul style="list-style-type: none"> • LOGINFENSTER • EINGABEFENSTER NEUE DATEN • TERMINEINGABEFENSTER 	
Used Cases Modellierung	
<ul style="list-style-type: none"> • Wo steht die Pendenz? • Wie siehts mit dem Zeichnen aus? • Used Cases ergänzen... • Requirements neu definieren (mehr als bisher) 	
DOMINIK: USED CASES ZEICHEN	
PASCAL: Arc42 Kpt. 8-22	
OLIVIA & LEXI: Diagramm Bausteinschicht Arc42 Kpt. 5-7	
Arc42 Documentation	
Datenbank Luca, Bartli	
 Used Cases <ul style="list-style-type: none"> - Verbesserungen - Tipps - Ziel erfüllt? 	
Requirements:	
Tasks:	
Stakeholder bedürfnis:	
Was ist den Stakeholdern besonders wichtig?	
Was ist Feuerungskontrolleur besonders wichtig?	

Teilnehmer:	Alle Teammitglieder	
Betreff:	Festlegen der Rollen und Verteilung von Aufgaben	
<ul style="list-style-type: none"> - Abarbeiten von Taskliste - Festlegen von Architekten - Standortbestimmung und wichtige Infos für Gruppenmitglieder - Ausblick, wie weiter? 		
Wer	Was	Bis
Alexandra & Dominik	Requirements koordinieren	21.04.16
Andere	Befassen mit Used Cases	21.04.16

FEUKORA - Sitzungsprotokoll 21.03.2016		
Teilnehmer:	Alle Teammitglieder	
Betreff:	Standortbestimmung, Arc 42, Used Cases und Requirements	
<ul style="list-style-type: none"> - Kurze Erklärung der Arc 42 von Olivia - Was haben wir noch zu tun? Wo stehen wir? - Überprüfen Used Cases - Stakeholder Bedürfnisse festlegen 		
Wer	Was	Bis
Matthias & Luca	Sich mit Datenbank erfassen	-

FEUKORA - Sitzungsprotokoll 18.04.2016		
Teilnehmer:	Alle Teammitglieder	
Betreff:	Erklärung der groben Architektur und Projektplanung	
<ul style="list-style-type: none"> - Kurze Erklärung der Kontextabgrenzung von Olivia - Was haben wir noch zu tun? Wo stehen wir? - Aufgabenverteilung und Zeitplan 		
Wer	Was	Bis
Alexandra & Olivia	Kapitel 5-7 Dokumentation	25.04.16
Olivia	Kapitel 3-4	25.04.16
Dominik	UseCase Kontrolle (EA Zeichnungen)	25.04.16
Matthias	UseCase Kontrolle (Beschriebe)	25.04.16
Pascal & Luca	Kapitel 8-11 Dokumentation	25.04.16
Matthias & Pascal & Luca	Datenbank einrichten / Dokumentation beginnen	25.04.16

FEUKORA - Sitzungsprotokoll 25.04.2016		
Teilnehmer:	Alle Teammitglieder	
Betreff:	Software Architektur abschliessen	
<ul style="list-style-type: none"> - Präsentation Alex und Olivia was haben wir gemacht? <ul style="list-style-type: none"> o Kapitel 5-7 - Präsentation Olivia was habe ich gemacht? <ul style="list-style-type: none"> o Kapitel 3-4 - Präsentation Pascal und Luca was haben wir gemacht? <ul style="list-style-type: none"> o Kapitel 8-11 - Präsentation Matthias was habe ich gemacht? <ul style="list-style-type: none"> o UseCases - Fragen klären, nach der Sitzung Dokumentation fertigstellen - Next Step Klassendiagramm wann, wie, wo? 		
Wer	Was	Bis
Alexandra & Olivia	Kapitel 5-7 Dokumentation überarbeiten	02.05.16
Olivia	Kapitel 3-4 Dokumentation überarbeiten	02.05.16
Dominik	Kapitel 5-7 (EA Zeichnungen) Entwürfe GUI	02.05.16

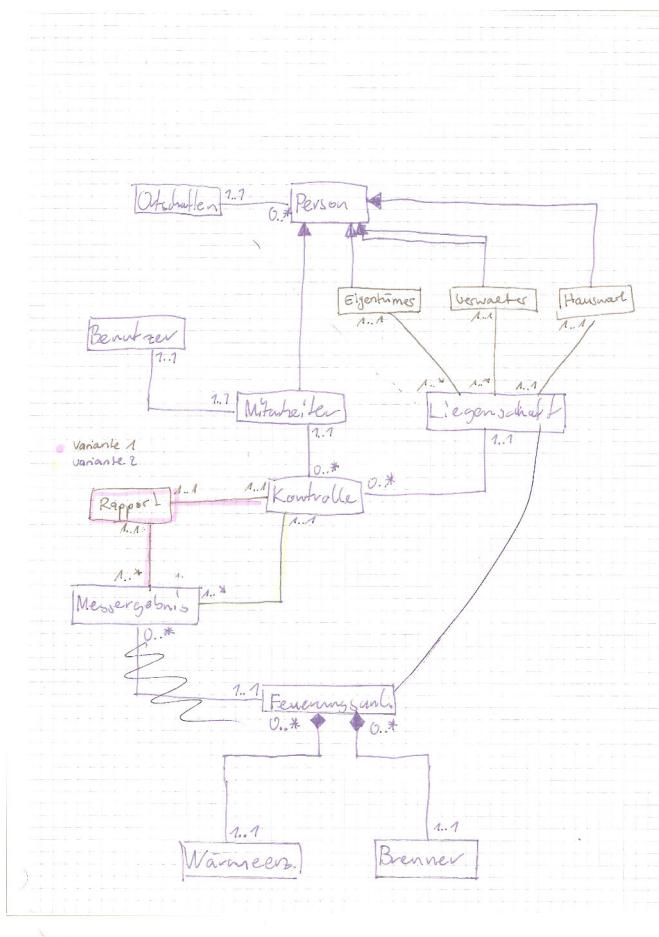
Alle	UseCase Beschriebe nach Kontrolle anpassen: <i>Nr. 1-6 Matthias</i> <i>Nr. 7-10 Dominik</i> <i>Nr. 11-14 Pascal</i> <i>Nr. 15-18 Alex</i> <i>Nr. 19-22 Olivia</i> <i>Nr. 23-26 Luca</i>	02.05.16
Pascal & Luca	Kapitel 8-11 Dokumentation überarbeiten	02.05.16
Matthias & Pascal & Luca & Alex & Olivia	Klassendiagramm machen	29.04.16 9:00 Uhr @ HSLU

FEUKORA - Sitzungsprotokoll 02.05.2016		
Teilnehmer:	Alle Teammitglieder	
Betreff:	Vorbereitung fürs Programmieren	
	<ul style="list-style-type: none"> - GUI besprechen stimmt alles mit ERD und Klassendiagramm überein? - 	
Wer	Was	Bis
Alexandra	Client_FK programmieren Laufzeidiagramm dokumentieren UseCases	15.5.16 5.5.16 mittags! 5.5.16 mittags!
Dominik	Webservice programmieren UseCases Diagramme in SA Ordner laden	15.5.16 5.5.16 mittags! 5.5.16 mittags!
Pascal	Client_BO programmieren Dokumentation 8-9 Kapitel UseCases	15.5.16 5.5.16 mittags! 5.5.16 mittags!
Olivia	Businesslogic programmieren Dokumentation überarbeiten Bausteinsicht zeichnen Visio	15.5.16 5.5.16 mittags! 5.5.16 mittags!
Luca	Persister UseCases	15.5.16 5.5.16 mittags!
Matthias	RMI programmieren UseCases	15.5.16 5.5.16 mittags!

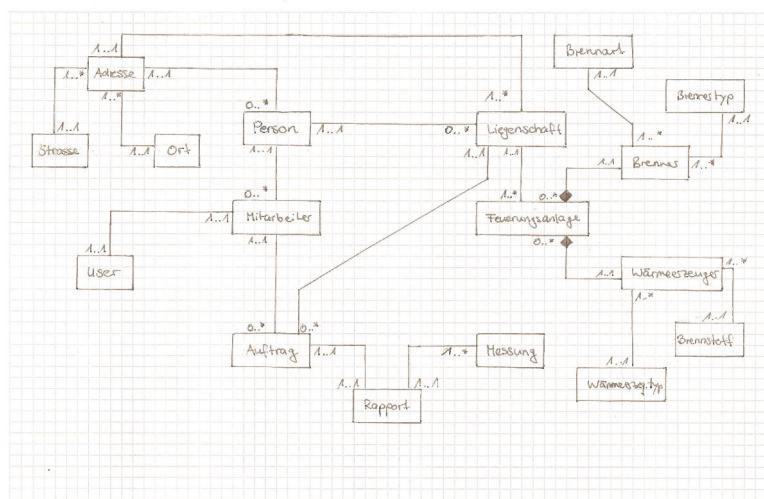
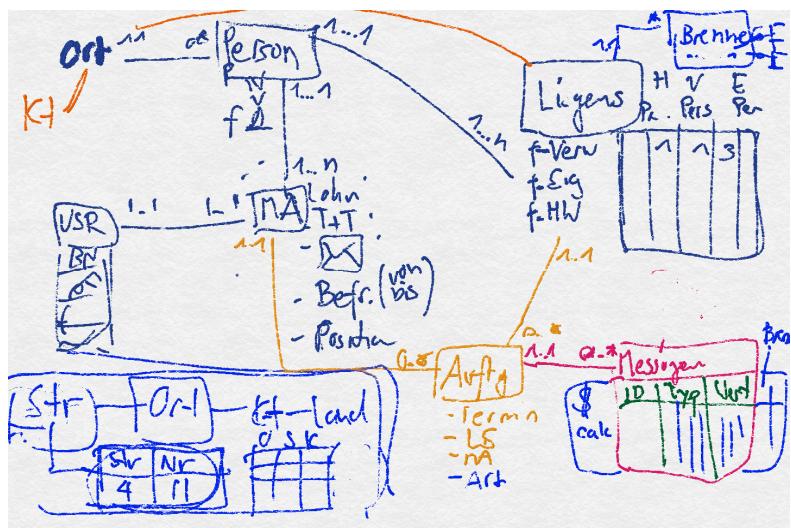
Entscheide:

- Wir trennen Mitarbeiter und Personen komplett. Wir machen je eine Tabelle. Auch in Ausblick auf die Zukunft könnten so noch ein HR System gemacht werden.
- Klassendiagramm V1 ausgewählt

23.5 Anhang Entwicklung



<u>Attributliste</u>	= PK	= FK
Person (PersNr, Vorname, Nachname, (Adresse),		
Mitarbeiter (PersNr, Telefonnr, E-mail, Position, Lohn, Befristung von, bis,		
User (PersNr, Username, Passwort, Berechtigung		
Adresse (PLZ, (ID Str))		
Ort (PLZ, Ortsbezeichnung)		
Strasse (ID, Strassenbez, Hausnummern)		
Auftrag (PersNr, Eigenschaft_V), AuftragNr, Termin, Art		
Liegenschaft (LgNr, Verwalt., Hausnr, Eigentimer, PersNr)		
Messung (RapportNr, Ress.zahl, Kohlenstoff, Sauerst., Stickstoffgehalt, AbgasTemperatur, WärmezeugerTemperatur, Feuerungsart, (LgNr), BrennerNr, Wärmeersatz)		
Brenner (BrennerNr, (Brennst.), Typnr, Baujahr,		
Wärmeersatz (WärmeersatzNr, (BrennstNr), (WetypNr), Baujahr,		
Rapport (RapportNr, Bemerkung, Grenzwert Ress.zahl, Grenzwert Kohlenstoff, Grenzwert Stickstoff, Grenz. Abgas-temperatur, Grenz. WärmezeugerTemp, Grenz. Verbr. LuftTemperatur, Grenz. O2 Gehalt, Grenz. Abgasverluste)		
Brennstoff (BrennstNr, Brennst.-bez)		
Brennstyp (TypNr, Typ-bez)		
Brennstoff (BrennstNr, Brennst.-bez)		
Wärmeersatztyp (WärmeersatztypNR, Wärmereszt.-bez)		

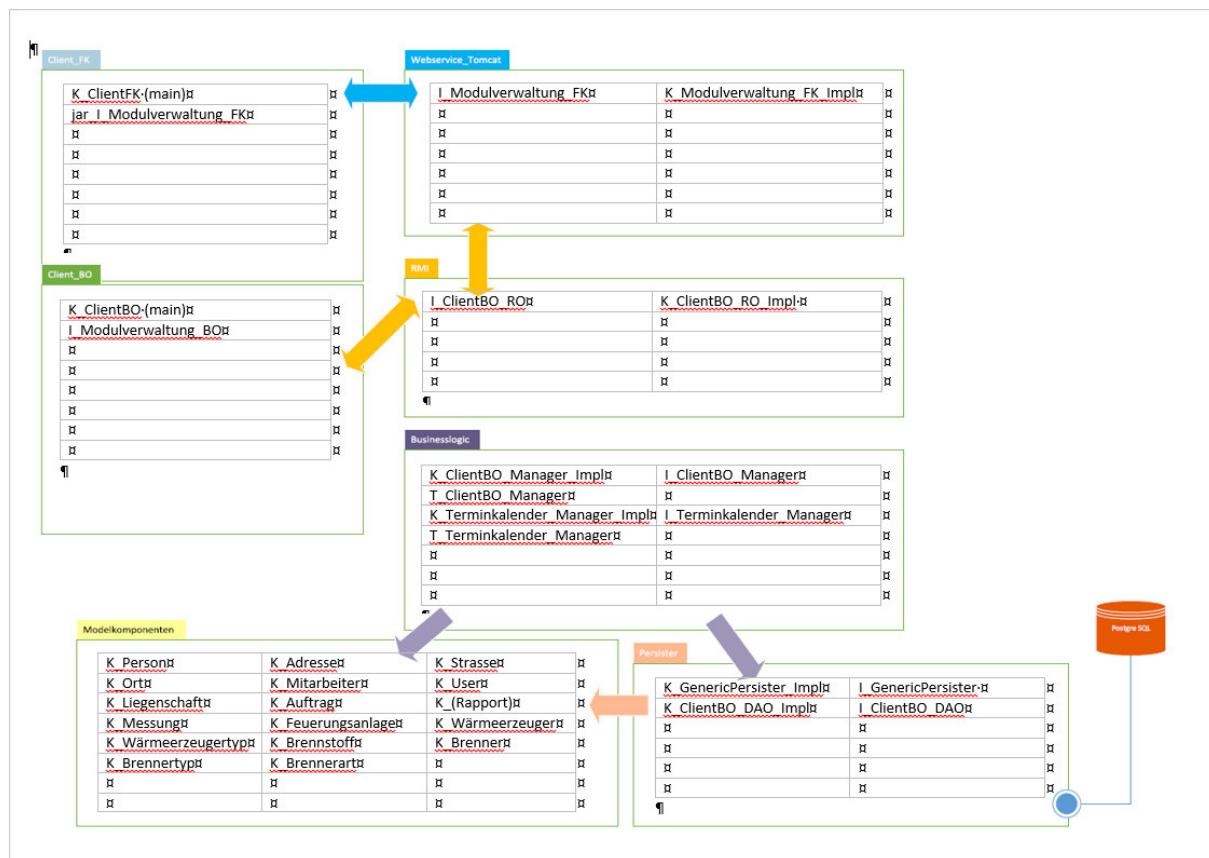
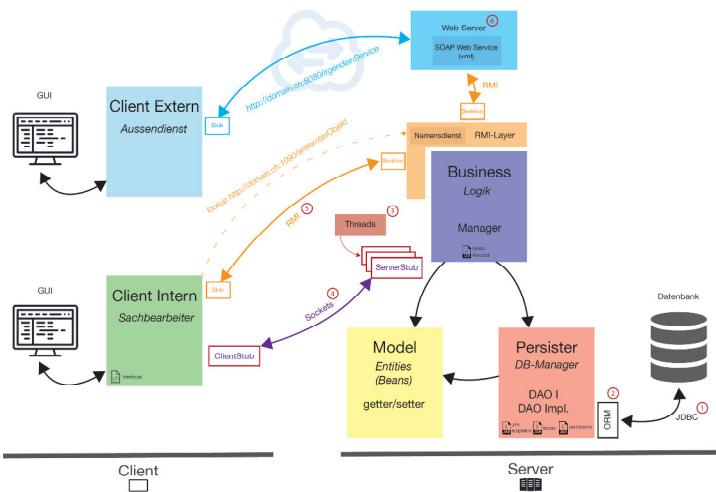


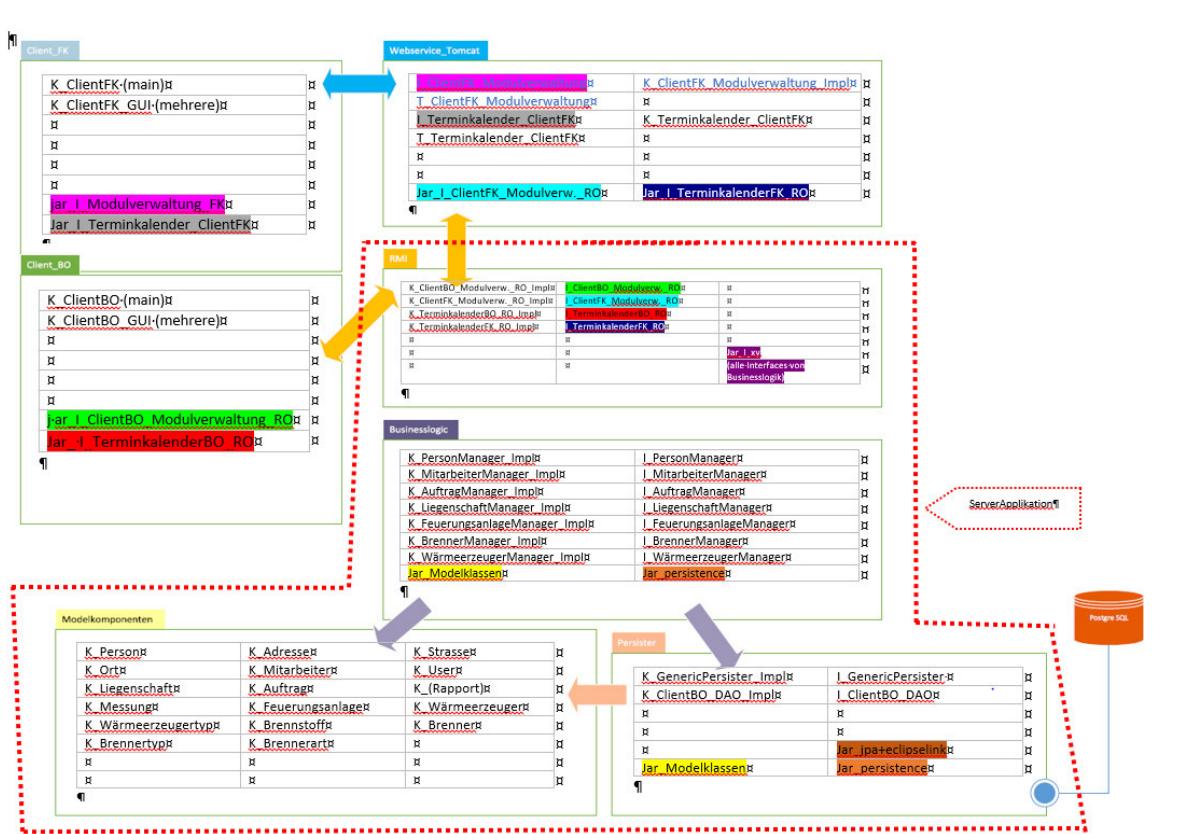
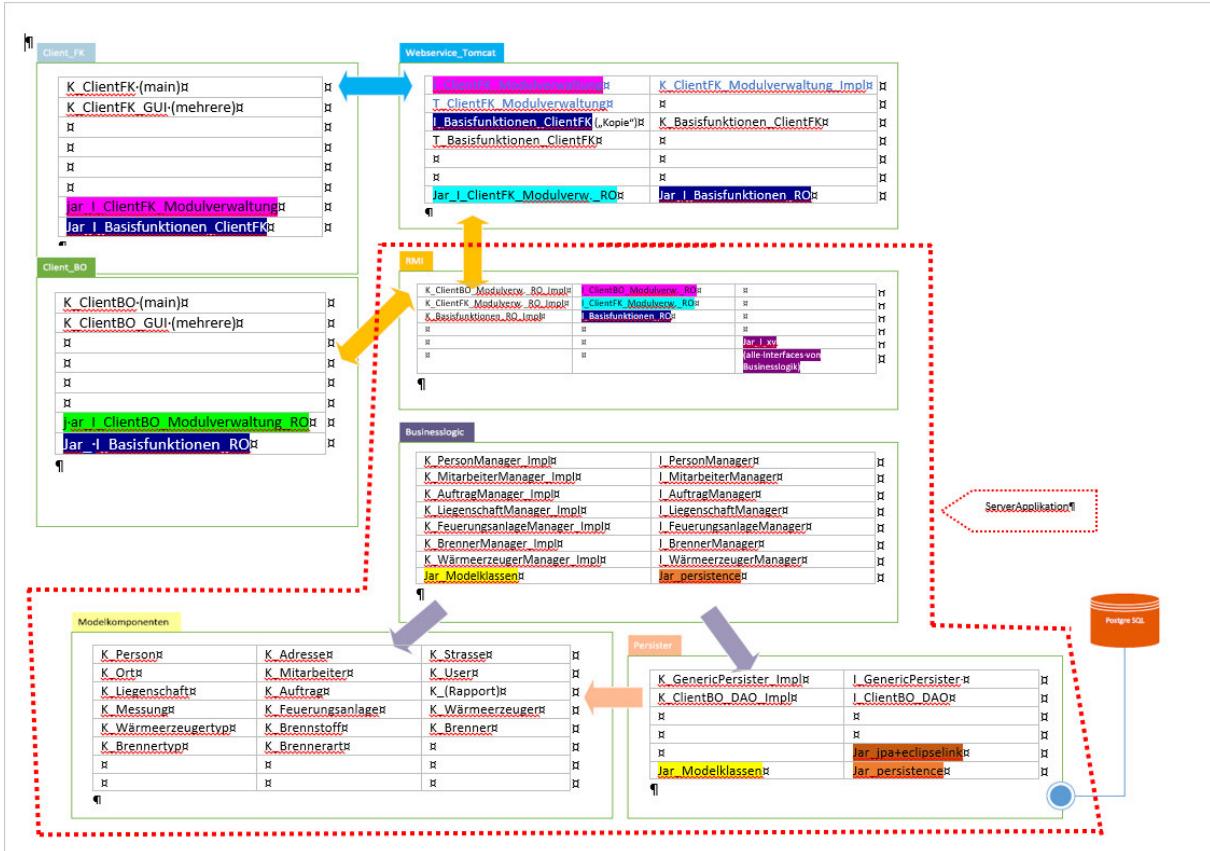
Attributliste

Primary Key	Foreign Key
Mitarbeiter	ID-Mitarbeiter, ID-Ort, ID-Benutzer, Strasse, Vorname, Nachname, Telefon, E-Mail, Lohn, Angestellt seit, Angestellt bis, RolleIntern
Kontakt	ID-Kontakt, ID-Ort, Strasse, Vorname, Nachname, Telefon, E-Mail, RollenExtern
Benutzer	ID_Benutzer, Username, Password,
Ort	ID-Ort, PLZ, Ortsbezeichnung
Liegenschaft	ID-Liegenschaft, ID-Feuerungsanlagen, ID-Kontakt, ID-Ort, Info vor Ort, Strasse
Auftrag	ID-Auftrags, ID-Messung, ID-MitarbeiterNr, ID-Liegenschaft, ID-Kontakt, Zeitslot, Terminart, Datum, EinregulierungInnert30, EinregulierungNichtMoeglich,
Messung	ID-Messung, Messdatum, Russzahl, Kohlenstoff bezogen auf 3% vol. O ₂ , Ölanteile, Stickstoffdioxid bezogen auf 3% vol. O ₂ , Abgastemperatur, WärmeerzeugerTemperatur, Verbrennungslufttemperatur, O ₂ - Gehalt, Abgasverluste, BeurteilungNotOk, RusszahlNotOk, OelAnteilenNotOk, AbgasverlusteNorOk, CoMgNotOk, NoMgNotOk
Feuerungsanlage	ID-Feuerungsanlagen, ID-Brenner, ID-Wärmeerzeuger, Feuerungswaermeleitung
Brenner	ID-Brenner, Brennertyp, Brennart, Baujahr
Wärmeerzeuger	ID-Wärmeerzeuger, Wärmeerzeugertyp, Brennstoff, Baujahr

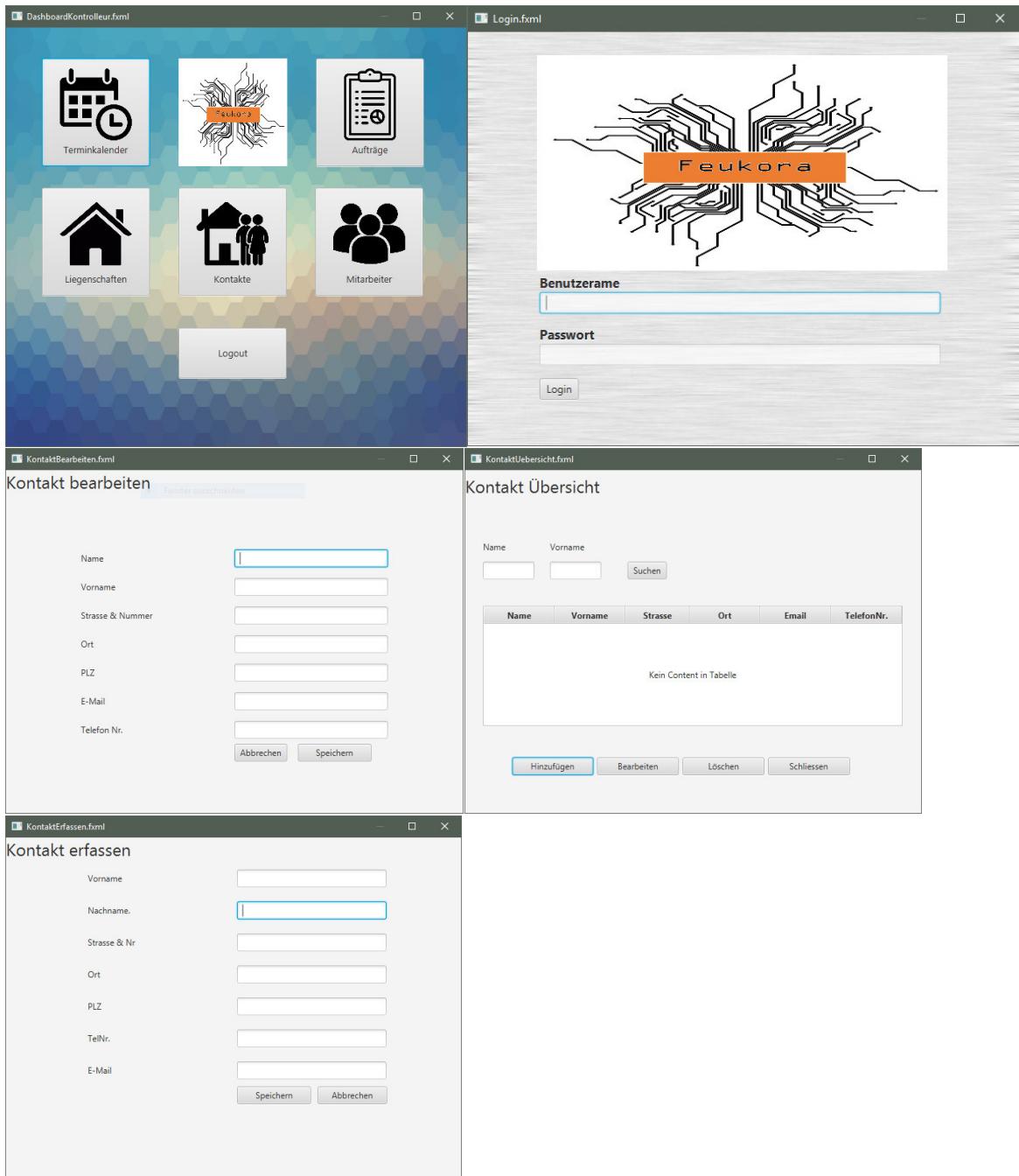
Erster Ablauf: Erstellen eines Studenten mit dem ReferenzProjekt von Herrn Succur.

Client extern





23.6 Anhang GUI



LiegenschaftÜbersicht.frm

Liegenschaft Übersicht

Info vor Ort	Strasse	Ort
Kein Content in Tabelle		

[Hinzufügen](#)

[Bearbeiten](#)

[Löschen](#)

[Schließen](#)

Suchen

X

Liegenschaft bearbeiten

Vorname	Nachname	Strasse	Ort	Telefon	E-Mail

Kontak:

Vorname:

Nachname:

Suchen

Speichern

Abrufen

Kein Content in Tabelle

Liegenschaft:

Strasse:

P.L.:

Feuergesamtheit

Brenner:

Type:

Warenzeuge:

Artikel:

Bemerk:

Bemerk:

Bugjahr:

Bugjahr:

Info vor Cr:

Cr:

Hilf:

Info vor Cr:

Cr:

Kein Content in Tabelle

Liegenschaft erfassen

Vorname	Nachname	Strasse	Ort	Telefon	E-Mail

Kontak:

Vorname:

Nachname:

Suchen

Speichern

Abrufen

Kein Content in Tabelle

Liegenschaft:

Strasse:

P.L.:

Feuergesamtheit

Brenner:

Type:

Warenzeuge:

Artikel:

Bemerk:

Bemerk:

Bugjahr:

Bugjahr:

Info vor Cr:

Cr:

Hilf:

Lohn

E-Mail

Telefon Nr.

Arbeitet seit:

Arbeitet bis:

Pl.Z.

Rolle

Speichern

Abrufen

Hilf:

LiegeMitarbeiterbearbeiten.frm

Mitarbeiter bearbeiten

Name	Vorname	Strasse & Nummer	Ort	Pl.Z.	Rolle

Speichern

Abrufen

■ MitarbeiterErfassen.form

Mitarbeiter erfassen

Ilegenschaft:	
Strasse & Nr.	<input type="text"/>
Ort:	<input type="text"/>
Kunde:	<input type="checkbox"/> Vorname <input type="checkbox"/> Nachname
Vorname	<input type="text"/>
Nachname	<input type="text"/>
Strasse	<input type="text"/>
PLZ	<input type="text"/>
Ort	<input type="text"/>
Zeitsct	<input type="text"/>
Datum	<input type="text"/>
Terminart:	<input type="checkbox"/> Feuerungskontrolleur <input type="checkbox"/> Feuerwehr
Feuerungskontrolleur	<input type="text"/>
■ Speichern Abbrechen	

■ MitarbeiterLehersicht.form

Mitarbeiter Übersicht

Name	Vorname	Suchen	
<input type="text"/>	<input type="text"/>	<input type="button" value="Suchen"/>	<input type="button" value="Zurück"/>

■ TerminÜbersicht.form

Terminübersicht - Kontrolleurin

Von		Bis		Anzeigen		Festtag	
Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Montag	Montag
8:00 - 10:00							
10:00 - 12:00							
Mittagspause							
13:00 - 15:00							
15:00 - 17:00							

■ Hinzufügen Bearbeiten Löschen Schliessen

RapportBearbeiten.fxml

Kundendaten

Name	<input type="text"/>	Strasse	<input type="text"/>
Vorname	<input type="text"/>	Ort	<input type="text"/>
Ort	<input type="text"/>	PLZ	<input type="text"/>
PLZ	<input type="text"/>	Info vor Ort	<input type="text"/>
TelefonNr	<input type="text"/>	Label Rueckmeldung!!!	

Brenner - Daten

Brenner	<input type="text"/>
Baujahr	<input type="text"/>
Brennstoff	<input type="text"/>

Wärmeerzeuger - Daten

Wärmeerzeuger	<input type="text"/>
Baujahr	<input type="text"/>
Brennstoff	<input type="text"/>

Kontrolle-Arten

Auftragsart	<input type="text"/>
Feuerungswärmeleistung	<input type="text"/>

Messergebnisse

1. Messung			
Russzahl	CO auf 3%vol O ₂	Abgastemperatur	Verbrenn. luft- Te...
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/> Abgasverl.
Ölanteile	NOx	Wärmeerz. Temp.	O2-Gehalt <input type="text"/>
<input type="checkbox"/> CheckBox	<input type="text"/>	<input type="text"/>	<input type="text"/>

2. Messung			
Russzahl	CO auf 3%vol O ₂	Abgastemperatur	Verbrenn. luft- Temp
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/> Abgasverl.
Ölanteile	NOx	Wärmeerz. Temp.	O2-Gehalt <input type="text"/>
<input type="checkbox"/> Ölanteile	<input type="text"/>	<input type="text"/>	<input type="text"/>

Beurteilung

<input type="checkbox"/> Beurteilung nicht Okey	<input type="checkbox"/> Abgasverluste	<input type="checkbox"/> Russzahl	<input type="checkbox"/> CO ₂ in mg/m ³
		<input type="checkbox"/> Ölanteil	<input type="checkbox"/> NO ₂ in mg/m ³
<input type="button" value="Messwerte Pruefen"/>			

Weiteres Vorgehen

<input type="checkbox"/> Anlage muss innerst 30 Tagen einreguliert werden	<input type="checkbox"/> Einregulierung auf einhalten der LRV-Grenzwerte nicht möglich
---	--

Bemerkungen

RapportUebersicht.fxml

Auftragsübersicht

Kontrolleur: Von: Bis:

Kunde	Liegenschaft	Datum	Zeitslot
Kein Content in Tabelle			

23.7 Projekt Feukora Source Code

Der Sourcecode ist nach Bausteinen sortiert. Die folgenden Register sind jeweils für einen Baustein. Darin befinden sich alle Klassen, Interfaces und weiterer Sourcecode. Sämtliche Dokumente sind jeweils mit dem Package Namen und Klassennamen erkennbar.

1.0 Modelklassen

1.1 *Entitys*

2.0 Persistier

2.1 *interfaces*

2.2 *persistier*

2.3 *Persistier.util*

3.0 Businesslogik

3.1 *managerInterfaces*

3.2 *managerKlassen*

3.3 *testManager*

4.0 RMI

4.1 *Rmi*

4.2 *Rmi.interfaces*

4.3 *Rmi.server*

5.0 Client intern

5.1 *Application*

5.2 *Application.view.admin*

5.3 *Application.view.dashboard*

5.4 *Application.view.kontakt*

5.5 *Application.view.liegenschaft*

5.6 *Application.view.login*

5.7 *Application.view.mitarbeiter*

5.8 *Application.view.rapport*

5.9 *Application.view.termin*

5.10 *Resource*

6.0 Webservice

6.1 *Feukora.webservice.rmi*

6.2 *Resource*

6.3 *Test.feukora.webservice.rmi.test*

7.0 Client Extern

7.1 *Application*

7.2 *Application.view.dashboard*

7.3 *Application.view.kontakt*

7.4 *Application.view.liegenschaft*

7.5 *Application.view.login*

7.6 *Application.view.rapport*

7.7 *Application.view.termin*