

TAU INF202 Software Engineering
Projektarbeit
Pflichtenheft

Projektdokumentation

Version: 2023.v1.0

Status: Abgegeben

SoftwareUnternehmen Informationssystem

Verantwortliche/r:

Hilal Ebru Duran, e200503037@stud.tau.edu.tr

Zeynep Örengül, e200503058@stud.tau.edu.tr

Beraterin: Fulya Yenilmez, fulya.yenilmez@tau.edu.tr

Stakeholder: DI. Ömer Karacan, omer.karacan@tau.edu.tr

Dokumentenverwaltung

Dokument-Historie

Version	Status *)	Datum	Verantwortlicher	Änderungsgrund
v0.9	Entwurf	07.04.2023	Hilal Ebru Duran Zeynep Örengül	Das Dokument zur Entwurfskontrolle freigegeben
v1.0	abgegeben	17.04.2023	Hilal Ebru Duran Zeynep Örengül	Das Dokument wurde fertiggestellt und abgegeben.

**) Sofern im Projekt nicht anders vereinbart, sind folgende Statusbezeichnungen zu verwenden (in obiger Tabelle und am Deckblatt):*

Dokument-Status: Entwurf / in Review / freigegeben (abgegeben)

Dokument wurde mit folgenden Tools erstellt:

Microsoft Office Word

Draw.io

Clipchamp

Inhaltsverzeichnis

1. Einleitung.....	4
2. Ausgangssituation und Ziele	5
3. Gesamtarchitektur	6
4. Funktionale Anforderungen	9
5. Nichtfunktionale Anforderungen	16
6. Abnahmekriterien	18
7. Projekt Meilensteine	19
8. Referenzen.....	20

1. Einleitung

Dieses Dokument soll das Projekt, nämlich SoftwareUnternehmen Informationssystem, mithilfe verschiedenen Techniken wie Klass-, Flussdiagrammen oder Skizzen im Detail erläutern. Bei der Entwicklung dieses Dokuments ist unser Ziel, uns als Projektteam während der Entwicklungsphase und dem Bewerter während der Bewertungsphase einen Fahrplan zu geben.

Im Einleitungsteil werden die wichtigsten Überschriften des Dokuments und deren Zwecke kurz angegeben. Das Dokument besteht einschließlich der Einleitung aus 8 Kapiteln.

- 2) **Ausgangssituation und Ziele:** In diesem Abschnitt werden die Gründe für die Erstellung des Projektthemas erläutert.
- 3) **Gesamtarchitektur:** In diesem Abschnitt werden wichtige Komponente des Projekts vorgestellt und Kommunikationsschnittstellen erwähnt.
- 4) **Funktionale Anforderungen:** In diesem Abschnitt werden die Anwendungsfälle (Use Cases), die während des Aufbaus des Systems erstellt wurden, detailliert definiert und erläutert. Anwendungsfälle können verwendet werden, um das Verhalten des Systems zu erklären. Bei der Definition wurden verschiedene Techniken wie Klassendiagramme und Schnittstellenzeichnungen verwendet.
- 5) **Nichtfunktionale Anforderungen:** In diesem Abschnitt werden die angestrebten Anforderungen bei der Auslegung des Systems genannt. Im Gegensatz zum vorherigen Abschnitt werden jedoch nichtfunktionale Anforderungen diskutiert.
- 6) **Abnahmekriterien:** Abnahmekriterien werden vom Stakeholder festgelegt. In diesem Abschnitt werden einige Dateien geteilt, um den Dokumentations- und Softwareteil nachzuweisen.
- 7) **Projekt Meilensteine:** Dieser Abschnitt beschreibt die wichtigsten Schritte während der Erstellung des Projekts. Die Schritte wurden vom Stakeholder zu Beginn des Projekts festgelegt.
- 8) **Referenzen:** Die bei der Erstellung des Dokuments verwendeten Literaturquellen werden aufgelistet.

2. Ausgangssituation und Ziele

Einleitung

In diesem Teil unserer Arbeit informieren wir Sie darüber, für welches Problem das von uns entwickelte System eine Lösung bietet, wer die Zielgruppe bildet, die von dieser Lösung profitieren wird, welche Infrastruktur und technische Details für den reibungslosen Betrieb erforderlich sind System und die Ziele, die wir erreichen wollen.

Problemstellung (Funktionalität)

Dieses System ermöglicht dem Benutzer den Zugriff auf alle Informationen von einem einzigen Bildschirm aus, wenn Informationen über Mitarbeiter oder Projekte benötigt werden. Dies bietet einen großen Komfort für den Benutzer. Gleichzeitig werden dadurch die meisten aus Papier bestehenden Dokumente eingespart und der Papieraufwand minimiert sowie der Platzbedarf der Akten gespart.

Stakeholder (Anwender):

Software Unternehmen, die ein Informationssystem für die Personalabteilung aufbauen und Projekte weiterbearbeiten möchten, können von diesem System profitieren.

Systemumfeld (Einsatzumgebung)

Das System wird von einem einzigen Computer aus betrieben und nur ein Benutzer hat Zugriff. Das System kann auf Windows-Betriebssystemen ausgeführt werden.

Rahmenbedingung (Einschränkungen)

- Als Database wird MySQL verwendet.
- In der Entwicklungsphase werden Java und JavaFX verwendet.
- Der API-Standard muss RESTful sein.
- Als IDE wurde NetBeans gewählt.

Ziele (Lösung)

Dieses System, das keine physische Komplexität aufweist, reicht aus, um das aktuelle Problem mit seinen geringen Kosten, seiner einfachen Verwendung und seiner Infrastruktur zu lösen, die die gewünschten Informationen in Bezug auf das Speichern von Informationen erfüllen kann.

3. Gesamtarchitektur

Einleitung

In diesem Kapitel wird die Systemarchitektur aus Sicht des Anwenders erwähnt, durch die Begründung der Entwurfsentscheidungen werden die Beziehungen zwischen den Anforderungen und dem zu etablierenden System definiert.

Gesamtarchitektur

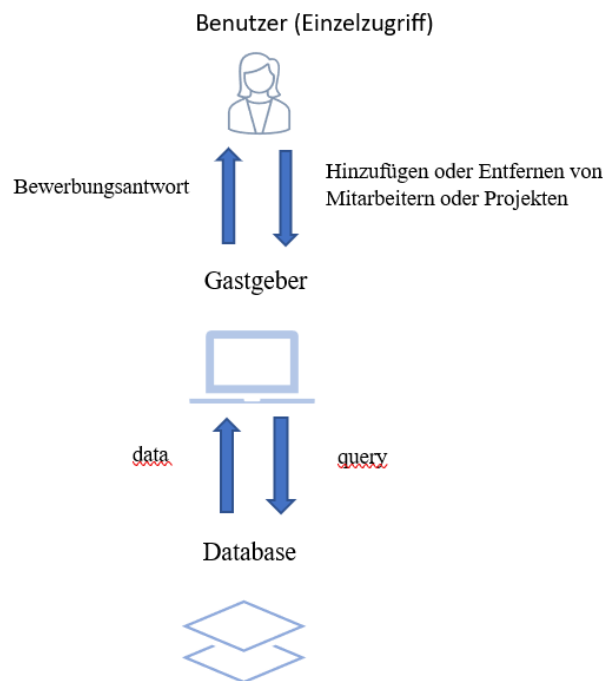


Abbildung 1

In der Abbildung 1 wird die grundlegende Architektur des Systems einfach vorgestellt.

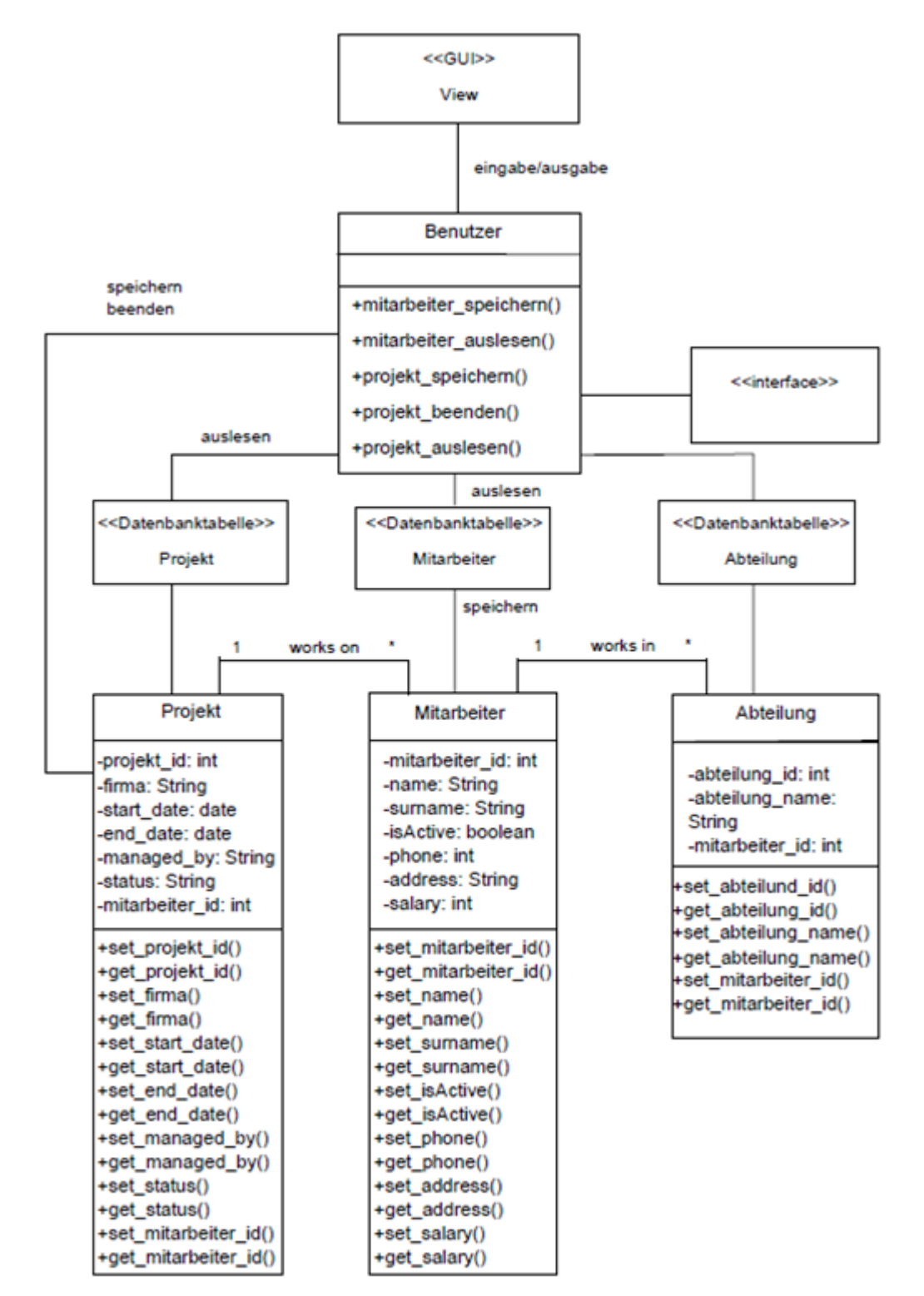


Abbildung 2

In der obigen Abbildung, nämlich Abbildung 2, sind die in der Entwicklung verwendeten Entitäten mit ihren Eigenschaften detailliert dargestellt, um die grundlegende Architektur des Systems zu erläutern.

Komponente <x>

Komponenten sind mit MVC- Architektur (Model-View-Controller) geplant. MVC Architektur wird im Projekt verwendet, um die verschiedenen Schichten des Systems zu trennen. Dies trägt auch dazu bei, eine flexiblere Entwicklungsumgebung bereitzustellen. Ein einfaches Schema zum Verständnis der Architektur ist in Abbildung 3 dargestellt.

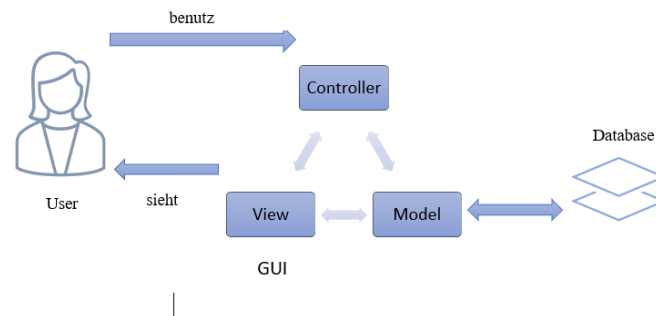


Abbildung 3

Externe Schnittstellen

In diesem Abschnitt wird beschrieben, wie das Projekt mit anderen Komponenten interagiert. Dieser Abschnitt gibt an, welche Technologien das Projekt zur Kommunikation mit anderen Systemen oder Komponenten verwendet und welche Datenformate es unterstützt.

- Das System ist mit dem Datenbanksystem verbunden, um die Informationen von Projekten und Mitarbeitern zu speichern. Zu diesem Zweck wird MySQL verwendet.
- RESTful API wird für die Datenintegration verwendet.
- Als Datenformat für den Datenaustausch wurde JSON festgelegt.
- API wird mit Spring Framework entwickelt.

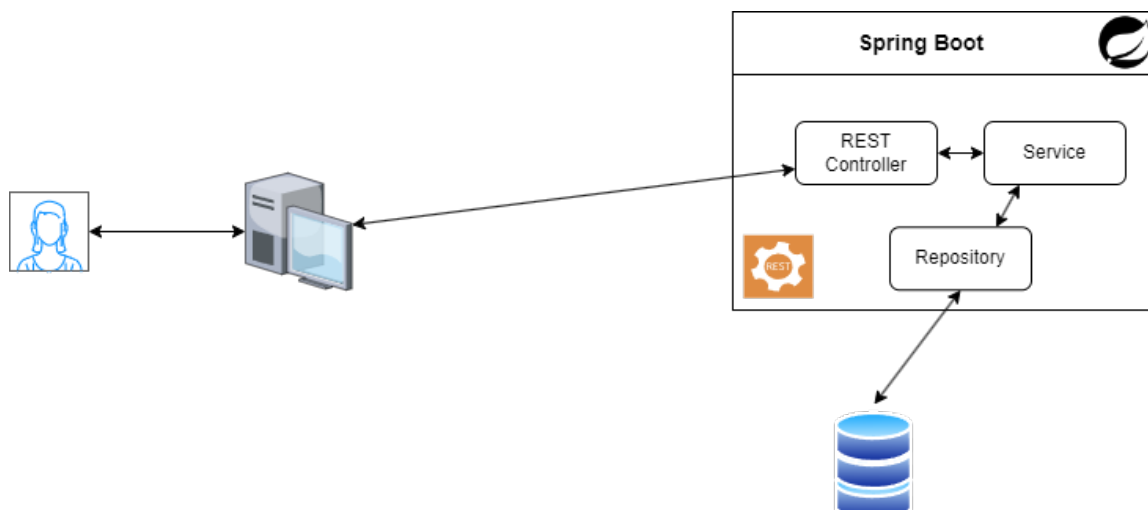


Abbildung 4

4. Funktionale Anforderungen

Einleitung

In diesem Abschnitt werden die funktionalen Anforderungen vorgestellt, die zum Verständnis des Verhaltens des Systems dienen. Es wurden mehrere Methoden verwendet wie Klassendiagramme, Benutzeroberflächen-Designs, Flussdiagramme um Anwendungsfälle zu identifizieren und sie leichter verständlich zu machen. Bevor wir ins Detail gehen, lassen sich die Use Cases aus dem folgenden Diagramm ableiten.

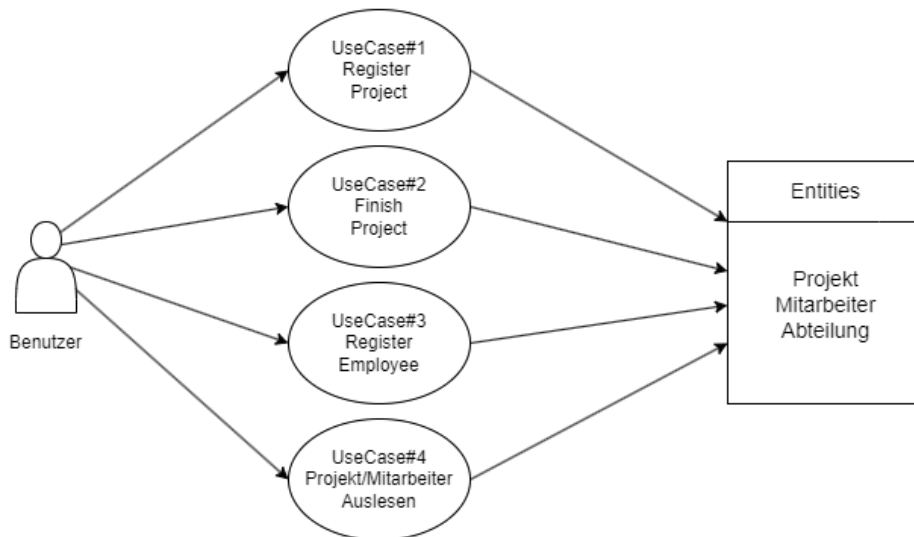


Abbildung 5

/UC_REG_PRJ/: Register Project

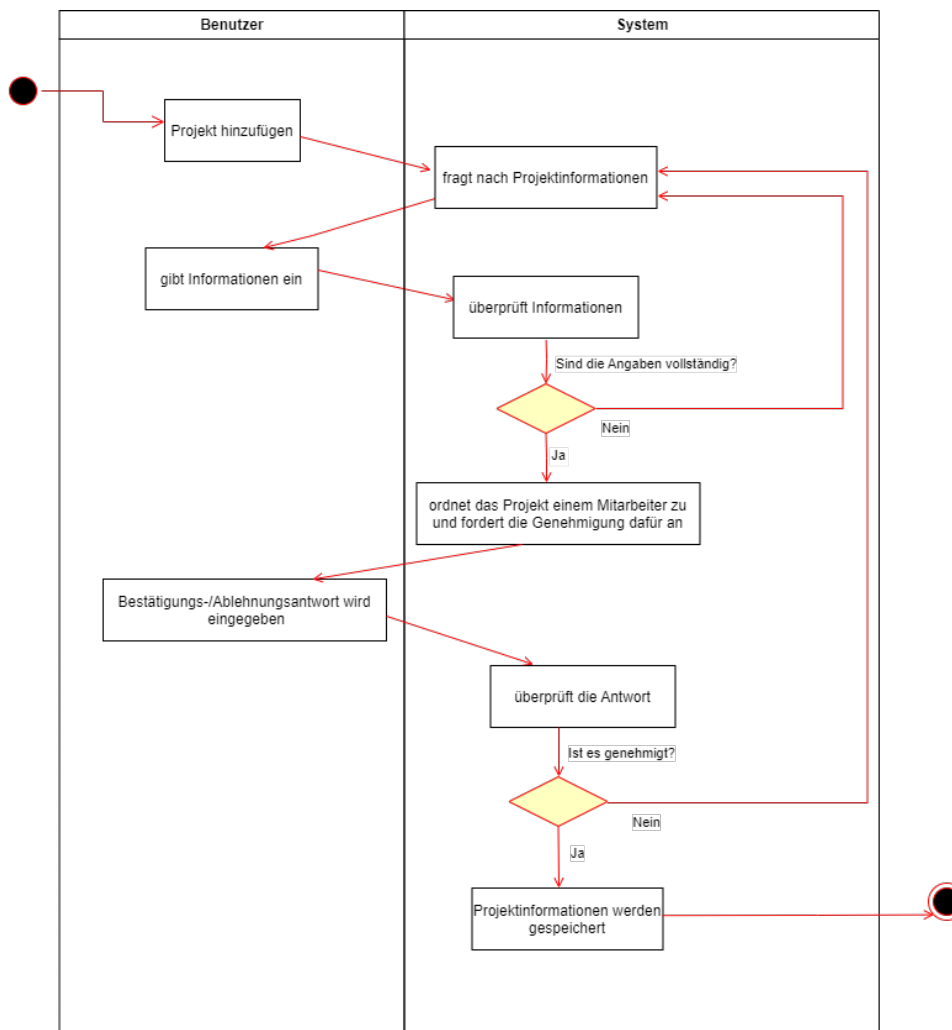


Abbildung 6: Flussdiagramm von Use Case#1

Use Case#1 Name:	Register Project
Summary:	Der Benutzer notiert die Einzelheiten über das Projekt.
Pre-Condition:	-
Basic Flow:	<ol style="list-style-type: none"> 1. Use Case#1 beginnt, wenn der Benutzer ein Projekt hinzufügen möchte. 2. Das System fragt nach dem Projektnamen, dem Start- und Enddatum des Projekts, der Anzahl der zu arbeitenden Personen nach Abteilungen und weitere Informationen. Einige der Informationen sind erforderlich. 3. Der Benutzer gibt die Informationen ein. 4. Je nach Personenzahl und Abteilung ordnet das System dem Projekt die passenden Mitarbeiter zu und druckt deren Namen und dann fordert den Benutzer zur Bestätigung auf. 5. Nach der Genehmigung speichert das System Informationen.
Alternative Flows:	<ol style="list-style-type: none"> 3. Wenn der Benutzer ein erforderliches Feld nicht ausfüllt, wird eine Meldung angezeigt und Use Case wiederholt Schritt 2. 5. Wenn der Benutzer nicht zustimmt, dann Use Case wiederholt Schritt 2.

Tabelle1: Use Case#1

/UC_FIN_PROJ/: Finish Project

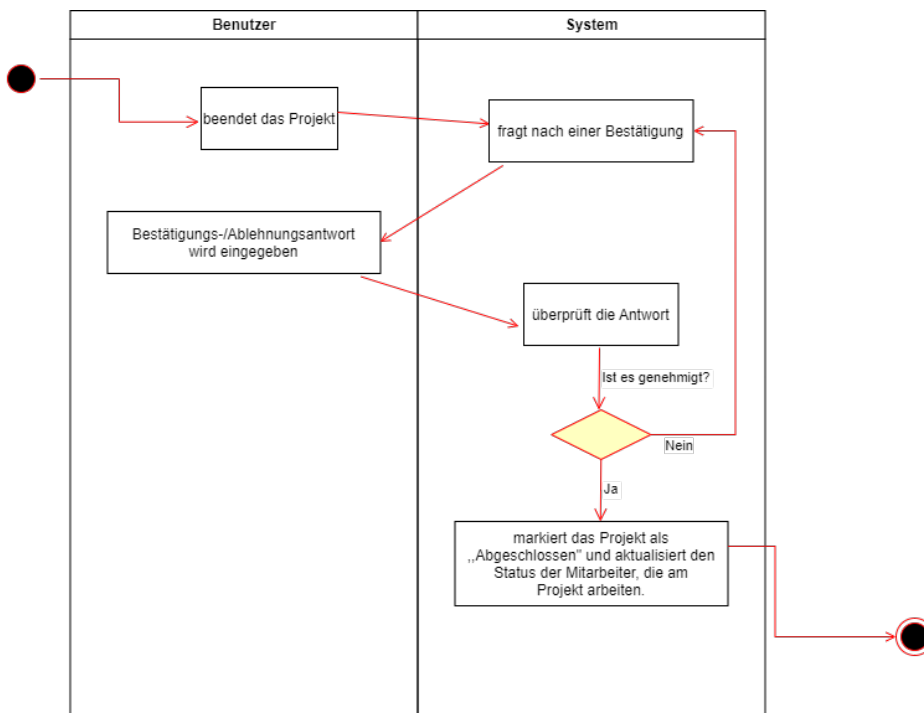


Abbildung 7: Flussdiagramm von Use Case#2

Use Case#2 Name:	Finish Project
Summary:	Der Benutzer aktualisiert den aktuellen Status des Projekts als „Abgeschlossen“.
Pre-Condition:	Das Projekt muss gespeichert und mit seiner eindeutigen ID gefunden werden.
Basic Flow:	<ol style="list-style-type: none"> 1. Use Case#2 beginnt, wenn der Benutzer den Status des abgeschlossenen Projekts aktualisieren möchte. 2. Das System fragt nach einer Bestätigung des Projekts, das der Benutzer abschließen möchte. 3. Nach der Genehmigung markiert das System das Projekt als „Abgeschlossen“. 4. Das System aktualisiert auch den Status der Mitarbeiter, die am Projekt arbeiten.
Alternative Flows:	<ol style="list-style-type: none"> 2. Wenn der Benutzer nicht zustimmt, wird Use Case#2 abgebrochen.

Tabelle2: Use Case#2

/UC_REG_EMP/: Register Employee

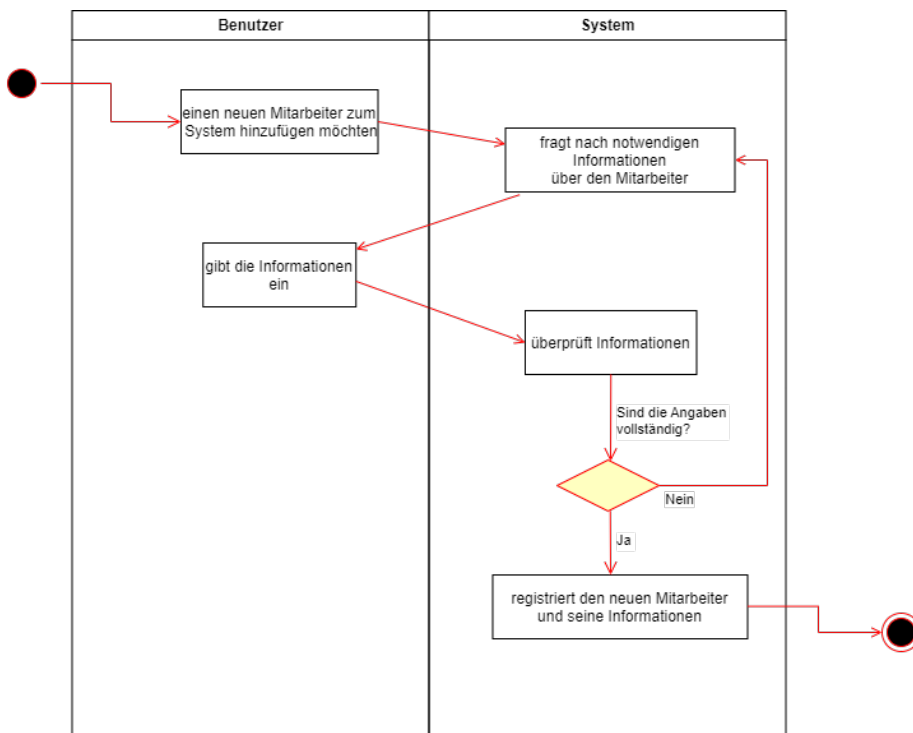


Abbildung 8

Use Case#3 Name:	Register Employee
Summary:	Der Benutzer möchte einen neuen Mitarbeiter hinzufügen.
Pre-Condition:	-
Basic Flow:	<ol style="list-style-type: none"> 1. Use Case#3 beginnt, wenn ein neuer Mitarbeiter eingestellt wird. 2. Das System fordert zur Eingabe der erforderlichen Informationen über den Mitarbeiter auf. 3. Der Benutzer gibt die Informationen ein. 4. Je nach Personenzahl und Abteilung ordnet das System dem Projekt die passenden Mitarbeiter zu und druckt deren Namen und dann fordert den Benutzer zur Bestätigung auf. 5. Das System speichert die Informationen.
Alternative Flows:	<ol style="list-style-type: none"> 3. Wenn der Benutzer ein erforderliches Feld nicht ausfüllt, wird eine Meldung angezeigt und Use Case wiederholt Schritt 2.

Tabelle3: Use Case#3

/UC_LIST_PROJ/ & /UC_LIST_EMP/: List Projekt/ Mitarbeiter

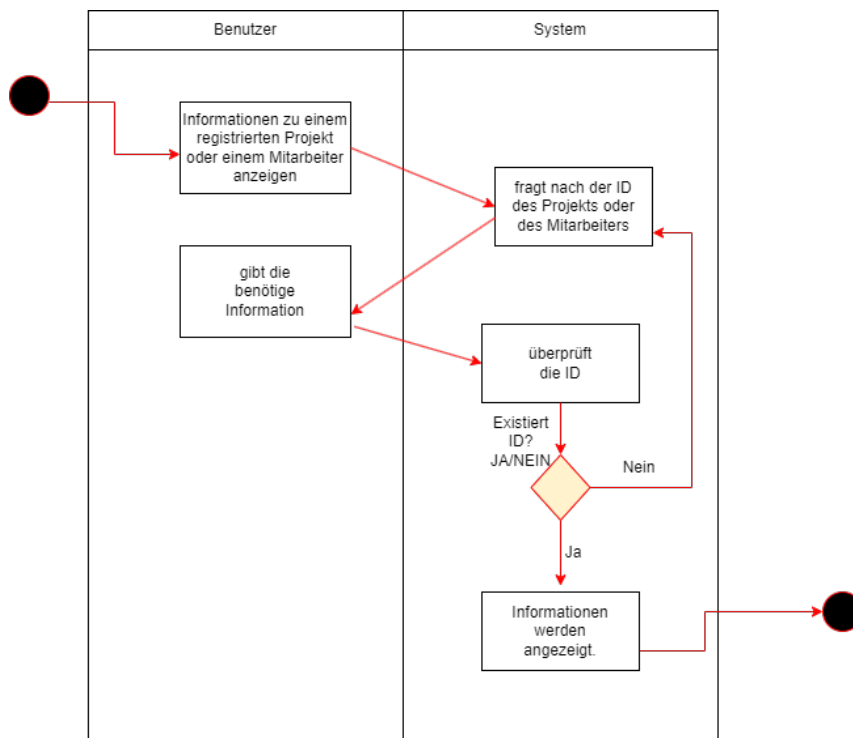


Abbildung 9: Flussdiagramm von Use Case#4

Use Case#4 Name:	Projekt/ Mitarbeiter Anzeigen
Summary:	Der Benutzer möchte Informationen zu einem registrierten Projekt oder einem Mitarbeiter anzeigen.
Pre-Condition:	Das Projekt oder der Mitarbeiter mit der eindeutigen ID gefunden werden.
Basic Flow:	<ol style="list-style-type: none"> 1. Use Case#4 beginnt, wenn der Benutzer Projekt- oder Benutzerinformationen anzeigen möchte. 2. Das System fragt nach der ID des Projekts oder des Mitarbeiters. 3. Der Benutzer gibt die Informationen ein. 4. Informationen werden angezeigt.
Alternative Flows:	<ol style="list-style-type: none"> 5. Wenn die eingegebene ID nicht im System gefunden wird, wird eine Fehlermeldung angezeigt.

Tabelle4: Use Case#4

UI Use Cases

Nun finden Sie einen einfachen Entwurf der Benutzeroberfläche des Systems gemäß unseren Anwendungsfällen.

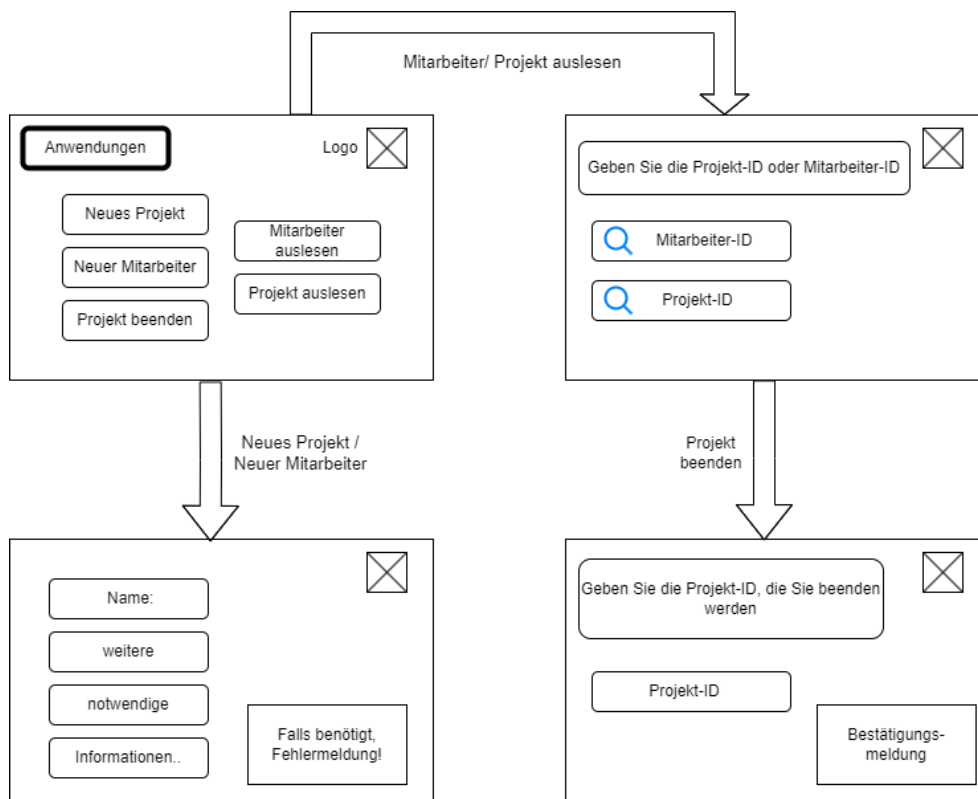


Abbildung 10: Skizzen

API Use Cases

Es werden RESTful-APIs verwendet, damit der Endbenutzer Anfragen an Datenbankressourcen stellen kann, wie in den Systemanforderungen angegeben.

Oben haben wir die Use Cases erläutert, die alle Benutzerinteraktionen des Systems abdecken. In diesem Unterabschnitt sprechen wir über die API-Use Cases, die wir in Übereinstimmung mit den allgemeinen Use Cases des Systems erstellt haben.

Zuerst ein einfaches Schema um die Position der API in der Systemarchitektur zu zeigen:

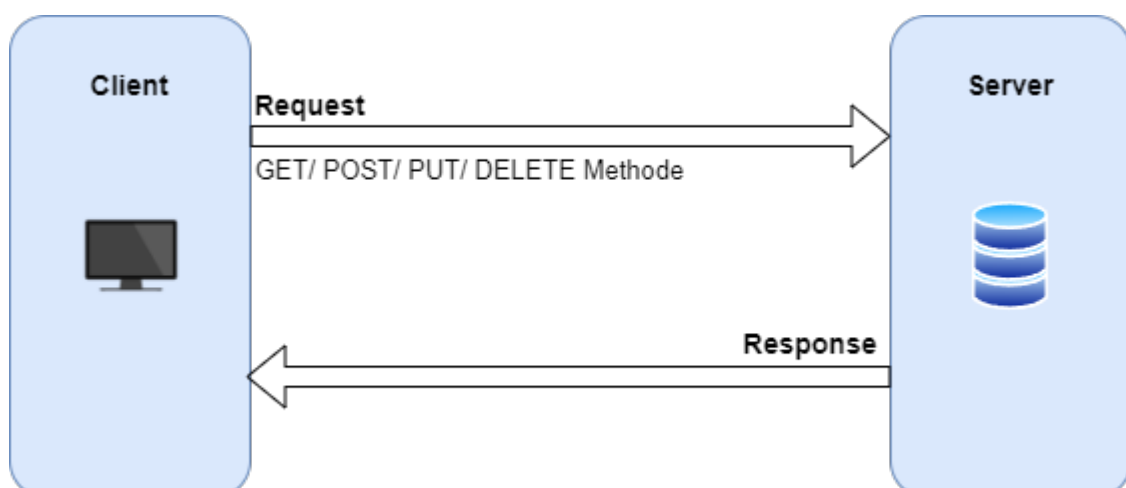


Abbildung 11

Die Methoden von RESTful-APIs sind unten aufgeführt:

•/API-1/ GET: /api/project	Liste aller Projekte
•/API-2/ GET: /api/project/{id}	Details eines bestimmten Projekts
•/API-3/ POST: /api/project	Neues Projekt hinzufügen
•/API-4/ DELETE: /api/project/{id}	Ein bestimmtes Projekt löschen
•/API-5/ GET: /api/employee	Liste aller Mitarbeiter
•/API-6/ GET: /api/employee/{id}	Details eines bestimmten Mitarbeiters
•/API-7/ POST: /api/employee	Neues Mitarbeiter hinzufügen

Tabelle 5: Methoden von APIs

Technischen und fachliche Anforderungen

- Das System wird als Desktop Application entwickelt.
- Als Database wird mySQL verwendet.
- In der Entwicklungsphase werden Java und JavaFX verwendet.
- Der API-Standard muss RESTful sein.

Datenmodell

Persistente Daten werden in einer Datenbank gespeichert. Das Entity-Relationship-Diagramm wird verwendet, um diesen Teil einfacher zu erklären.

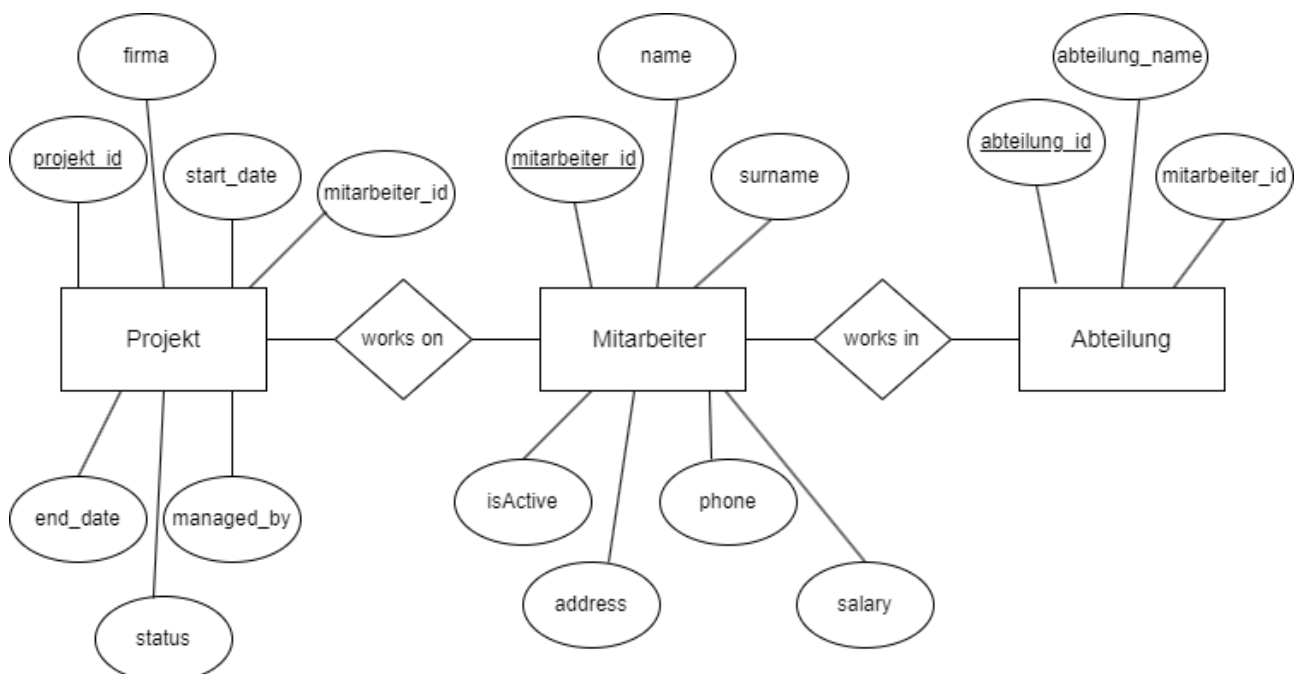


Abbildung 12

5. Nichtfunktionale Anforderungen

Einleitung

Der erfolgreiche Betrieb unseres Systems hängt nicht nur von der Erfüllung der Funktionen ab, die wir im vorherigen Abschnitt erwähnt haben. Gleichzeitig muss es andere Bedingungen wie Systemleistung, Benutzerkompatibilität, Zuverlässigkeit erfüllen. In diesem Abschnitt werden solche nichtfunktionalen Anforderungen beschrieben.

Systemarchitektur

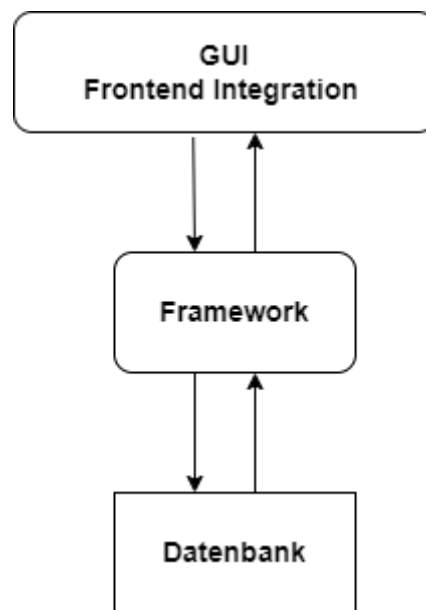


Abbildung 13

Hier steht ein einfaches Schema der technischen Systemarchitektur.

- **/SYS-1/** Der RESTful-API-Standard wird für die Kommunikation zwischen Systemkomponenten verwendet.

Entwicklungswerkzeuge

- **/TOL-1/** MySQL wird verwendet, um persistente Informationen zu speichern.
- **/TOL-2/** Bei der Entwicklung vom Backend Teil werden Java und Java Spring Boot Framework verwendet.
- **/TOL-3/** Der Frontend Teil wird mithilfe von JavaFX entwickelt.
- **/TOL-4/** Als IDE wurde NetBeans gewählt.

Teststrategie (Qualitätssicherung)

- **/TEST-1/** Unit-Tests werden mit JUnit durchgeführt, um sicherzustellen, dass jede Komponente unseres Systems korrekt funktioniert.

- **/TEST-2/** Black-Box wird verwendet, um alle Systemdienste zu steuern.

Zuverlässigkeit

- Das System sollte 99,99% des Jahres fehlerfrei laufen. Das Programm oder die Daten müssen vor unerwünschten Änderungen geschützt werden. Informationen sollten nicht verloren gehen.

Benutzerfreundlichkeit

- Die grafische Oberfläche sollte verständlich und einfach zu bedienen gestaltet sein. Der Benutzer sollte in der Lage sein, das System ohne zusätzliche Schulung zu verwenden.

Ergonomische Anforderungen

- Die im System zu verwendende Sprache ist Deutsch. Fehlermeldungen, Systemoberfläche etc. werden in deutscher Sprache verwendet.

6. Abnahmekriterien

Das Projekt wird mit den folgenden Artefakten abgegeben:

- Dokumentation:

Pflichtenheft: inf202-SoftwareUnternehmen-Informationssystem-pflichtenheft-2023.v1.0.doc

- Software:

Link zu GitHub Projekt: <https://github.com/orgs/Gruppe1-Fulya/teams/muzzy>

- Evidenz:

System/Software-Demo via Videoclip:

https://photos.google.com/share/AF1QipOymeDBnA7gkzvNRkCY5b_COjlrEQBlafpcmLzlw22xXJsiJ1D-BVjCh_UNBpIPAQ?key=MWU5ZEQ1TGRINFZyVWk0SEt2Z21nVUxWX0Nha0tB

Anm.: Die Abgabetermine der Projektartefakte werden durch den Stakeholder festgelegt!

7. Projekt Meilensteine

Die wichtigsten Schritte und Fristen, die vom Stakeholder zu Beginn des Projekts festgelegt wurden, sind unten dargestellt.

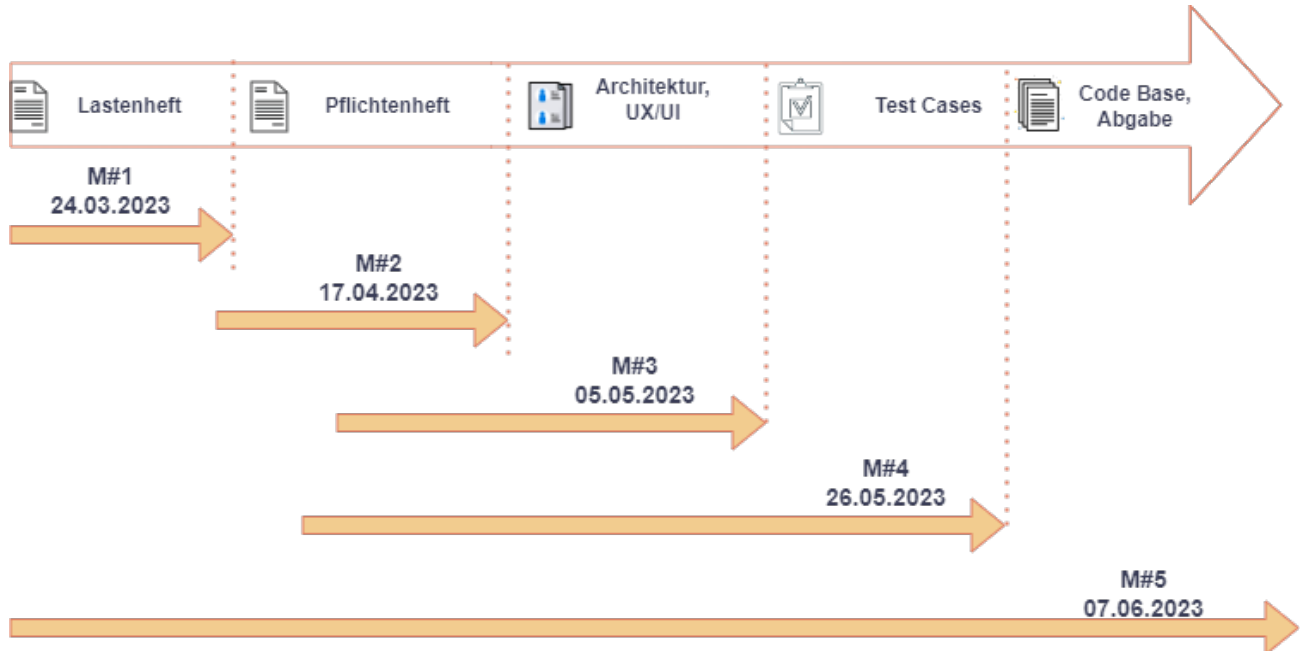


Abbildung 14

8. Referenzen

- Tabelaing, Peter, 2005, Softwaresysteme und ihre Modellierung, Springer