
BELEG

Saubere Kueche

Frau Juliet Petzold

im Rahmen des Moduls Softwaretechnik -
Grundlagen

Seminargruppe: IF18wS-B

Email: jpetzol1@hs-mittweida.de

Betreuer: W. Schubert, R. Beier

Mittweida, Sommersemester 2020

Inhaltsverzeichnis

Inhaltsverzeichnis	0
1 SWS-Skizze	2
2 Pflichtenheft	3
2.1.1 Ziel/Produktzwecke	3
2.1.2 Produkteinsatz	3
2.1.3 Produktübersicht	4
2.1.4 Produktfunktionen	4
2.1.5 Produktdaten	6
2.1.6 Produktleistungen	7
2.1.7 Qualitätsanforderungen nach ISO/IEC 25010:2011	7
2.1.8 Benutzeroberfläche	8
2.1.9 Nichtfunktionale Anforderungen	8
2.1.10 Produkt-Schnittstellen	8
2.1.11 Spezielle Anforderungen an die Entwicklungs-Umgebung	8
2.2 Use-Case-Diagramm	9
2.3 Use-Case-Beschreibung	9
2.4 User Stories	10
2.5 Aktivitätsdiagramm	11
2.6 Ausgewähltes Klassendiagramm	12
2.7 ERM (Entity-Relationship-Modell)	12
2.8 GUI-Prototyp	13
2.9 Persona Beschreibungen	14
2.10 Anforderungen an IT-Grundschutz	16
3 Dokumente zur Beschreibung und Begründung des Softwaredesigns ..	17
3.1 Blockdiagramm zur Softwarearchitektur	17
3.2 Fein-Design-Elemente	18
4 Zusammenfassung und Schluss	20
4.1 Rückblick auf Teil A (konzeptioneller Teil)	20
4.2 Rückblick auf Teil B (Praktischer Teil)	20

Selbstständigkeitserklärung	22
--	-----------

1 SWS-Skizze

Das Softwaresystem mit dem Namen „Saubere Küche“ soll dem Sortieren und Archivieren von Rezepten einer Großküche dienen. Zusammenfassend wird die Software eine Verwaltungssoftware für Rezepte mit ihren Bestandteilen sein. (Zutaten, benötigte Geräte, Zubereitung, ...) Es soll ebenfalls möglich sein, neue Rezepte einzutragen und bereits eingetragene Rezepte zum Lesen abzurufen und bei Bedarf zu bearbeiten.

Die Nutzer sind in zwei Gruppen unterteilt. Da wären zum einen die Köche, die die Rechte dazu haben neue Rezepte in das Programm einzutragen und bereits existente Rezepte anzuschauen und zu bearbeiten. Alle anderen Nutzer, also beispielsweise Beiköche und Küchenhilfen, können lediglich Rezepte und ihre Bestandteile aufrufen, d.h. dass sie nur lesen aber nicht schreiben können.

Die Nutzung der Software erfolgt auf Computern des Büros der Küche. Eine Home-Office Variante soll es nicht geben, da die Software nur für die Großküche geeignet ist und eine auswärtige Nutzung keinen Sinn machen würde. Es könnte jedoch eine Möglichkeit geben eine Home-Office Variante hinzuzufügen damit Köche in der Lage sind von zu Hause einige Rezepte hinzuzufügen, doch das ist nicht zwingend nötig und wird auch in diesem Beleg nicht weiter beleuchtet.

Die Nutzung des Programmes soll einfach sein. Dabei sind eindeutig beschriftete Schaltflächen, ein simples Design und unkomplizierte Benutzung sehr wichtig. Der Gedanke dahinter ist, dass auch jemand ohne Erfahrung im Umgang mit Computern in der Lage sein soll damit zu arbeiten. Da keine vertraulichen Daten gespeichert sind und Nutzer nur via Nutzernamen & evtl. einem Passwort auf die Rezepte zugreifen können sollte der Bereich der IT-Sicherheit ausreichend abgedeckt sein.

Das Softwaresystem soll nur lokal in der Großküche funktionieren. Das bedeutet, dass nur Computer des Büros darauf zugreifen können. Allerdings können im Büro mehrere Computer sein, was doch eine Funktion zum Synchronisieren im lokalen Netzwerk erfordert.

Für den Teil B des Belegs möchte ich gern mit Java arbeiten. Dazu stelle ich mir vor Eclipse für den Code und Scenebuilder für das GUI zu benutzen.

2 Pflichtenheft

2.1.1 Ziel/Produktzwecke

1.1 Musskriterien

- Abrufen Rezeptdatenbank
- Eintragung in Rezeptdatenbank
- Existenz verschiedener Rechte (Benutzerkonten)

1.2 Wunschkriterien

- Barrierefreie Benutzeroberfläche
- Version für das Home Office
- Synchronisation im lokalen Netzwerk

1.3 Abgrenzungskriterien

- Online-Funktion

2.1.2 Produkteinsatz

2.1 Anwendungsbereich

Das Programm soll in einer Großküche dem Archivieren und Sortieren von Rezepten dienen. Dabei soll es einer bestimmten Gruppe von Nutzern möglich sein, bereits bestehende Rezepte zu editieren.

2.2 Zielgruppen

Eine Person von egal welchem Alter oder Bildungsstand sollte dazu in der Lage sein das Programm zu bedienen.

2.3 Betriebsbedingungen

Es gibt keine besonderen Betriebsbedingungen. Das Programm erfordert einen Computer.

2.1.3 Produktübersicht

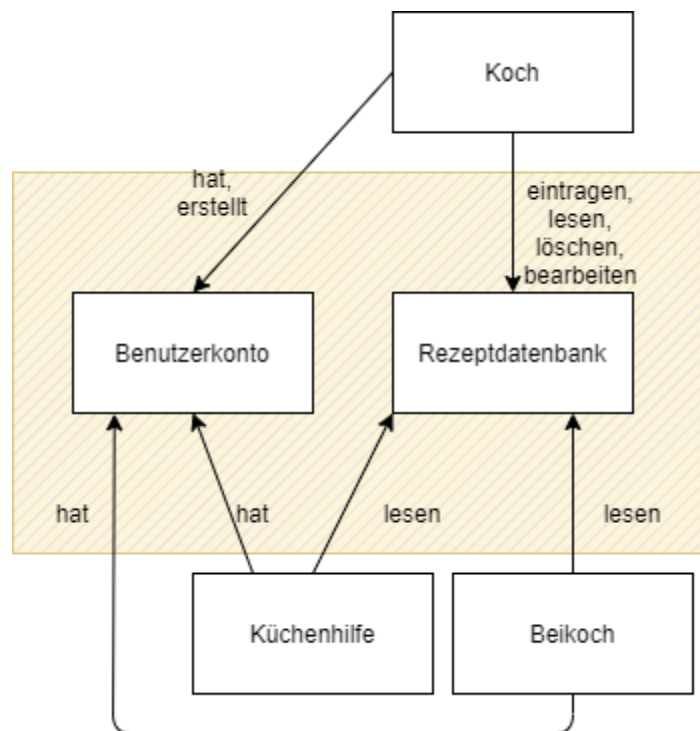


Abbildung 1: Produktübersicht

2.1.4 Produktfunktionen

- Erstellen und Ändern der Rezepte

/LF 010/

Funktion:

Rezept erstellen

Akteur:

Koch

Beschreibung:

Ein Eintrag für ein neues Rezept ist zu erstellen und zu erweitern.

/LF 020/

Funktion:

Rezept bearbeiten

Akteur:

Koch

Beschreibung:

Die bestehenden Daten für ein Rezept sind zu editieren.

- **Benutzerkonto erstellen**

/LF 030/	Funktion:	Benutzerkonto erstellen
	Akteur:	Koch
	Beschreibung:	Alle Benutzer bekommen festgelegte Daten, mit denen sie sich einloggen können.
/LF 040/	Funktion:	Berechtigungen festlegen
	Akteur:	Koch
	Beschreibung:	Verschiedene Benutzergruppen bekommen verschiedene Berechtigungen.

- **Abrufen von Rezeptdetails**

/LF 050/	Funktion:	Aufrufen eines Rezepts
	Akteur:	Koch
	Beschreibung:	Die Details (Zutaten, Zubereitung) zu einem Rezept werden angezeigt.
/LF 060/	Funktion:	Aufrufen eines Rezepts
	Akteur:	Beikoch
	Beschreibung:	Die Details (Zutaten, Zubereitung) zu einem Rezept werden angezeigt.
/LF 070/	Funktion:	Aufrufen eines Rezepts
	Akteur:	Küchenhilfe
	Beschreibung:	Die Details (Zutaten, Zubereitung) zu einem Rezept werden angezeigt.

- **Aktualisierung der Datenbank**

/LF 080/	Funktion:	Abgleich der Datenbank
	Akteur:	System bzw. Software
	Beschreibung:	Beim Hochfahren des Systems werden eventuelle Änderungen der Rezepte festgestellt.

/LF 090/	Funktion:	Aktualisierung der Daten
	Akteur:	System bzw. Software
	Beschreibung:	Die festgestellten Änderungen werden übernommen.

2.1.5 Produktdaten

/LD 010/	Name:	Rezept
	Max. Anzahl:	1000
	Beschreibung:	Ein Rezept umfasst alle Daten, um es erfolgreich auszuführen. Dazu gehören Zutaten, Zubereitung und evtl. Kommentare.

/LD 020/	Name:	Benutzerkonto
	Max. Anzahl:	3
	Beschreibung:	Jede Position hat ein eigenes Benutzerkonto zu dem jeweils eigene Rechte gehören.

2.1.6 Produktleistungen

/LL 010/

Alle Reaktionen auf Benutzereingaben müssen unter 2 Sekunden liegen.

/LL 020/

Beim Eintragen und Verändern der Rezepte sollten keine schwerwiegenden Unterschiede bis zur nächsten Aktualisierung vorliegen. Der Betriebsablauf soll durch diese Vorgänge nicht gestört werden.

/LL 030/

Es sollte möglich sein, dass mehrere Benutzer auf mehreren Computern die Rezepte gleichzeitig abrufen können.

/LL 040/

Die Sprache des Produktes soll einfach und unmissverständlich sein.

2.1.7 Qualitätsanforderungen nach ISO/IEC 25010:2011

Produkt-merkmale	Sehr gut	Gut	Normal	Nicht relevant
Funktionale Eignung		x		
Kompatibilität			x	
Sicherheit		x		
Zuverlässigkeit		x		
Benutzbarkeit	x			
Leistungs-effizienz			x	
Wartbarkeit			x	
Portabilität				x

Die sehr gute Benutzbarkeit ist nötig, damit jede Personengruppe dazu in der Lage ist das Produkt ohne Probleme zu benutzen.

Jeglicher Grad von Portabilität ist nicht nötig, da die Software lediglich in einem Betrieb an festen PC's zum Einsatz kommt und verwendet wird.

2.1.8 Benutzeroberfläche

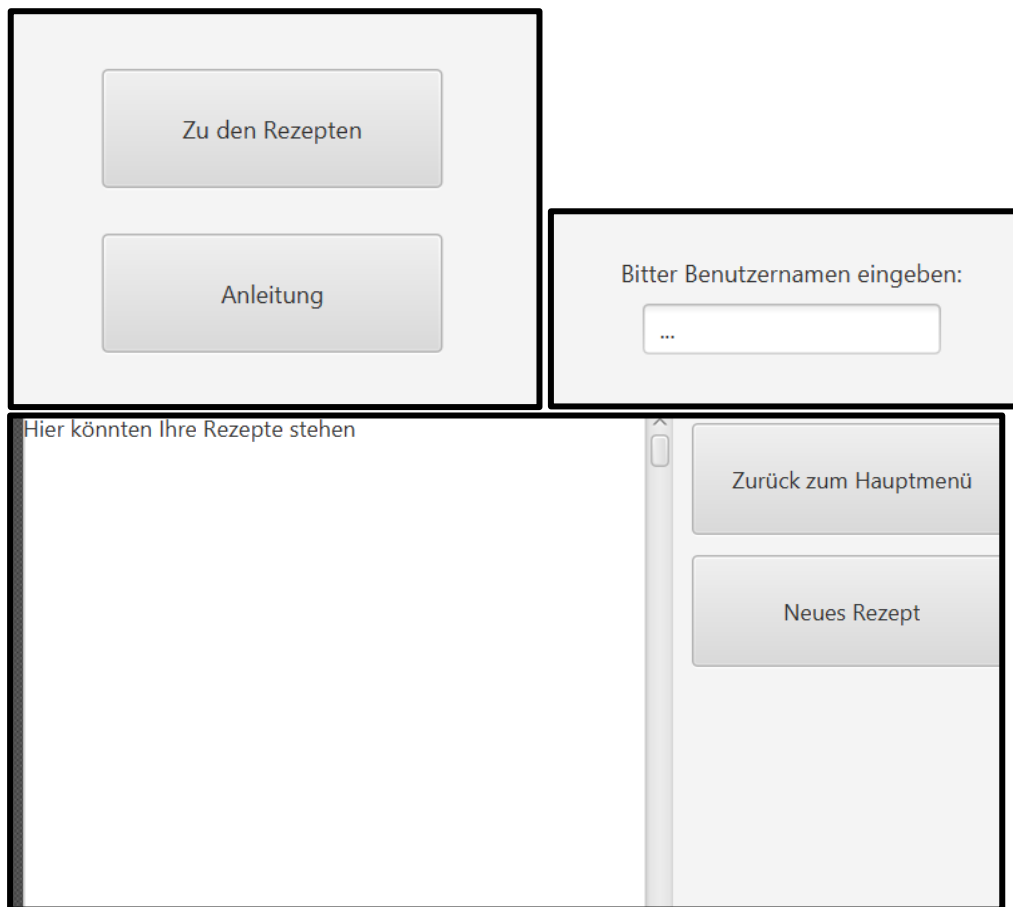


Abbildung 2: Screenshots Benutzeroberfläche

2.1.9 Nichtfunktionale Anforderungen

Es gibt keine nennenswerten nichtfunktionalen Anforderungen, die beachtet werden müssen.

2.1.10 Produkt-Schnittstellen

Die Kommunikation des Produktes/Programmes erfolgt über API's bzw. sql-Anweisungen mit einer Rezeptdatenbank.

2.1.11 Spezielle Anforderungen an die Entwicklungs-Umgebung

Es muss eine Möglichkeit geben eine grafische Benutzeroberfläche zu erstellen.

2.2 Use-Case-Diagramm

Als Koch/Beikoch/Küchenhilfe möchte ich mich einloggen, damit ich Zugriff auf die Rezeptdatenbank erhalte.

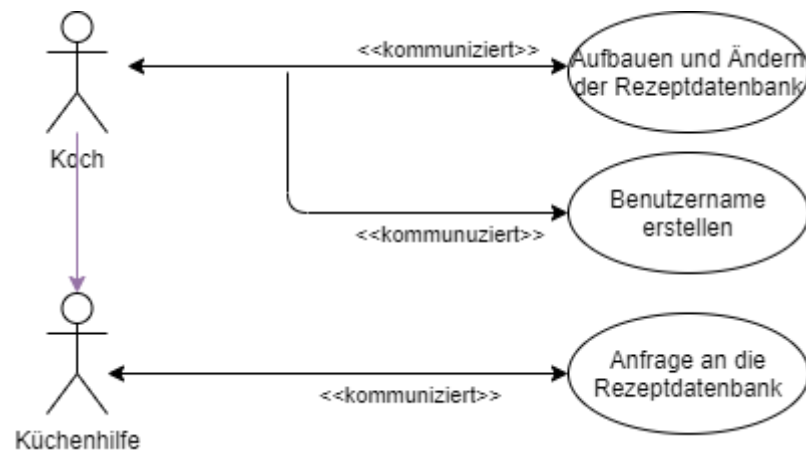


Abbildung 3: Use-Case-Diagramm

2.3 Use-Case-Beschreibung

Als Koch/Beikoch/Küchenhilfe möchte ich mich einloggen, damit ich Zugriff auf die Rezeptdatenbank erhalte.

1	Use Case-Titel	Zugriff auf die Rezeptdatenbank erhalten (Anfrage auf Rezept vom System bearbeitet und Zugriff darauf erteilen)
2	Ziel	Ziel ist es, Zugriff zum System zu erhalten um Rezepte einsehen zu können.
3	Vorbedingung	System muss mindestens ein vollständiges Rezept in der Datenbank haben und die Benutzernamen müssen bekannt sein.
4	Nachbedingung	Die Anfrage war erfolgreich und der Zugriff wurde erteilt.
4.1	Erfolg	
4.2	Nachbedingung Fehlschlag	Die Anfrage ist fehlgeschlagen und es wurde kein Zugriff erteilt. Es könnte daran liegen, dass der benutzte Nutzernamen nicht existiert.
5	Akteure	Koch, Beikoch, Küchenhilfe (jeder kann dies ausführen)
6	Auslösendes Ereignis	Zu Beginn der Systemnutzung, sobald eine Anmeldung erforderlich wird.
7	Eingabedaten	Benutzername

8	Beschreibung	1. bestehendes System aufrufen
8.1	(Standardablauf)	2. Benutzername in die Eingabeaufforderung eintragen 3. Login-Button betätigen 4. Erfolg oder Fehlschlag feststellen 5. bei Erfolg: gewünschtes Rezept aus der Datenbank auswählen und einsehen
8.2	Erweiterungen	/
8.3	Alternativen	/
	<i>Letzte Änderung</i>	10.08.2020
	<i>Autor</i>	Juliet Petzold

2.4 User Stories

Die Grundlegende Form einer User Story ist wie folgt aufgebaut: Als [Nutzer] möchte ich [Funktion], damit / um / weil [Wert]. Das heißt, eine User Story beantwortet die Fragen WER möchte WAS und WARUM.¹ Im Folgenden werden also User Stories aufgezählt.

Als Koch/Beikoch/Küchenhilfe möchte ich mich einloggen, damit ich Zugriff auf die Rezepte-Datenbank erhalte. [x]²

Als Koch möchte ich Rezepte eintragen, damit alle anderen Nutzer sie abrufen können.

Als Koch möchte ich fehlerhafte Rezepte bearbeiten, damit alle anderen Nutzer sie abrufen können.

Als Koch möchte ich Rezepte löschen, damit alle anderen Nutzer sie nicht mehr sehen können.

Als Koch möchte ich Benutzerkonten erstellen, damit ich Rechte an die anderen Nutzer vergeben kann.

Als Beikoch möchte ich Rezepte anschauen, damit ich weiß wie ich sie kochen muss. [x]

Als Küchenhilfe möchte ich Rezepte anschauen, damit ich weiß wie ich sie kochen muss.

¹ Andreas Diehl, „User Stories - Universelle Sprache in agilen Teams“, zuletzt aufgerufen am 01.08.2020 von <https://digitaleneuordnung.de/blog/user-stories/>

² Alle User Stories, die mit [x] markiert sind wurden in Teil B programmiert.

2.5 Aktivitätsdiagramm

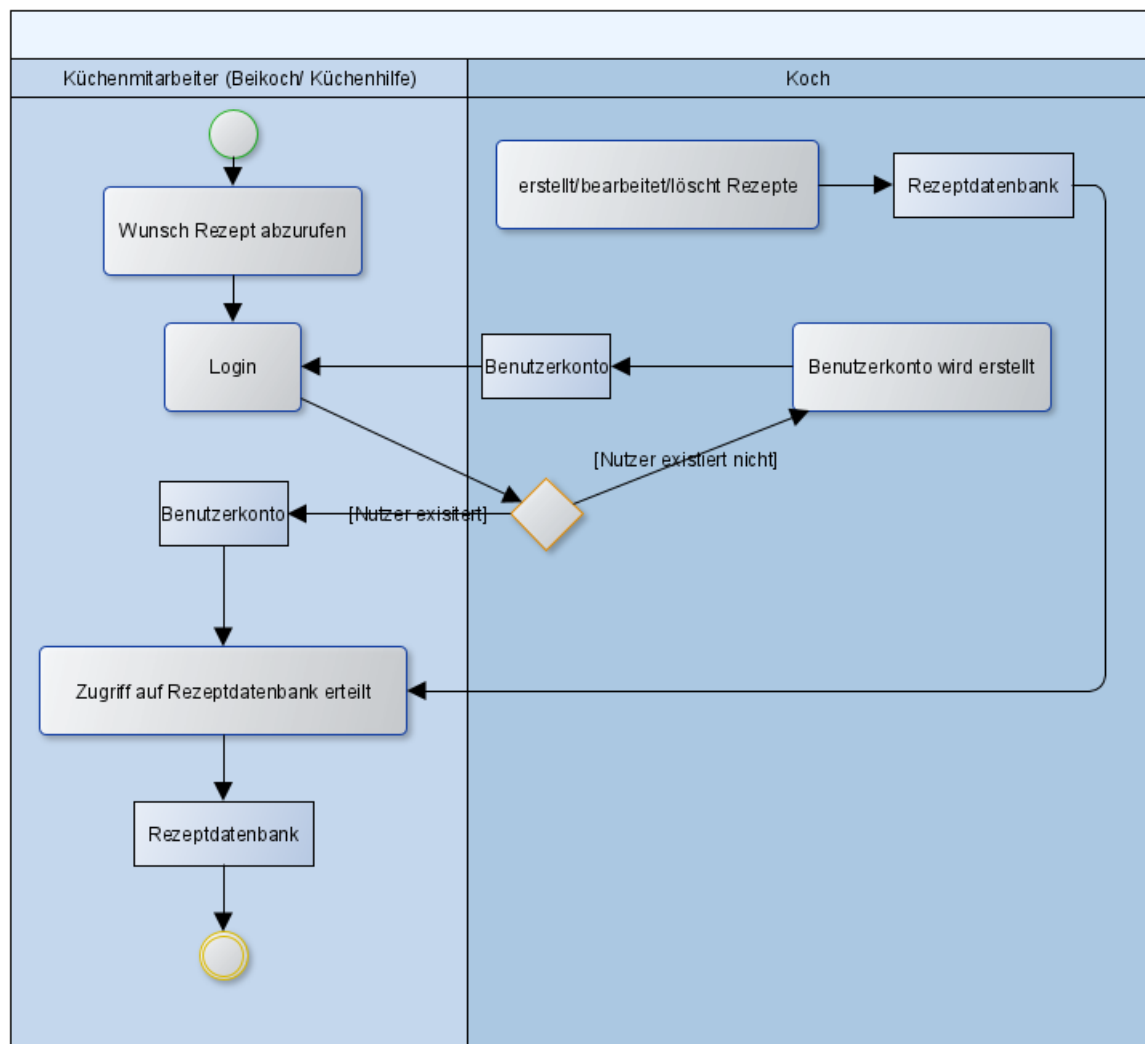


Abbildung 4: Aktivitätsdiagramm

2.6 Ausgewähltes Klassendiagramm

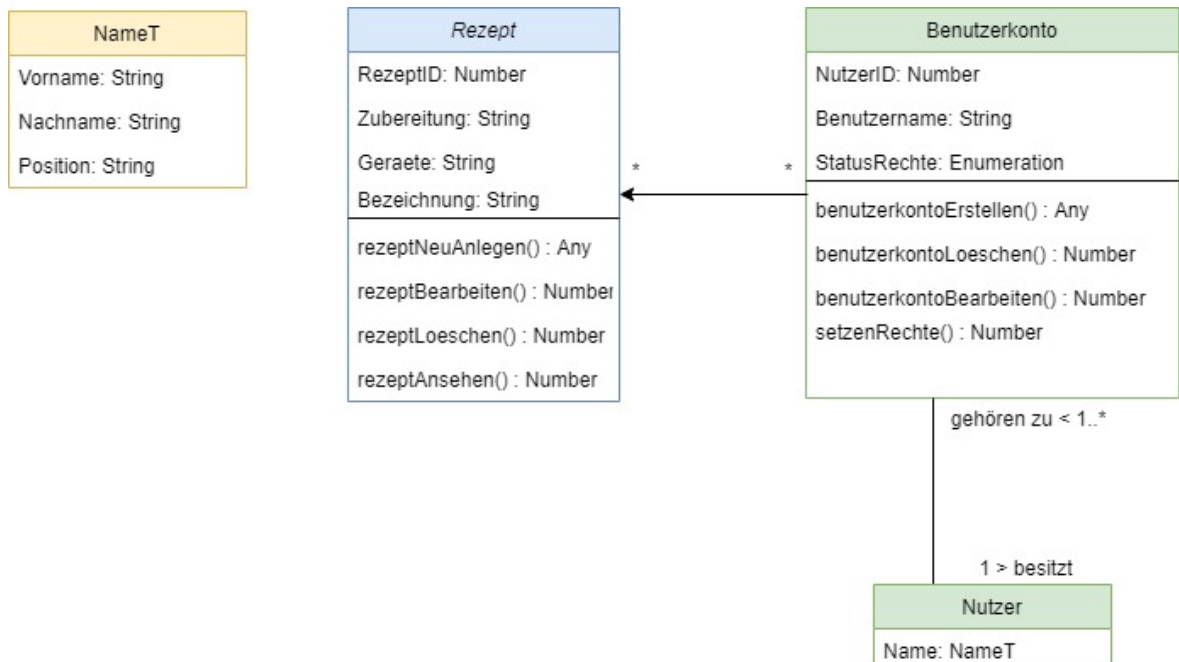


Abbildung 5: Klassendiagramm

2.7 ERM (Entity-Relationship-Modell)

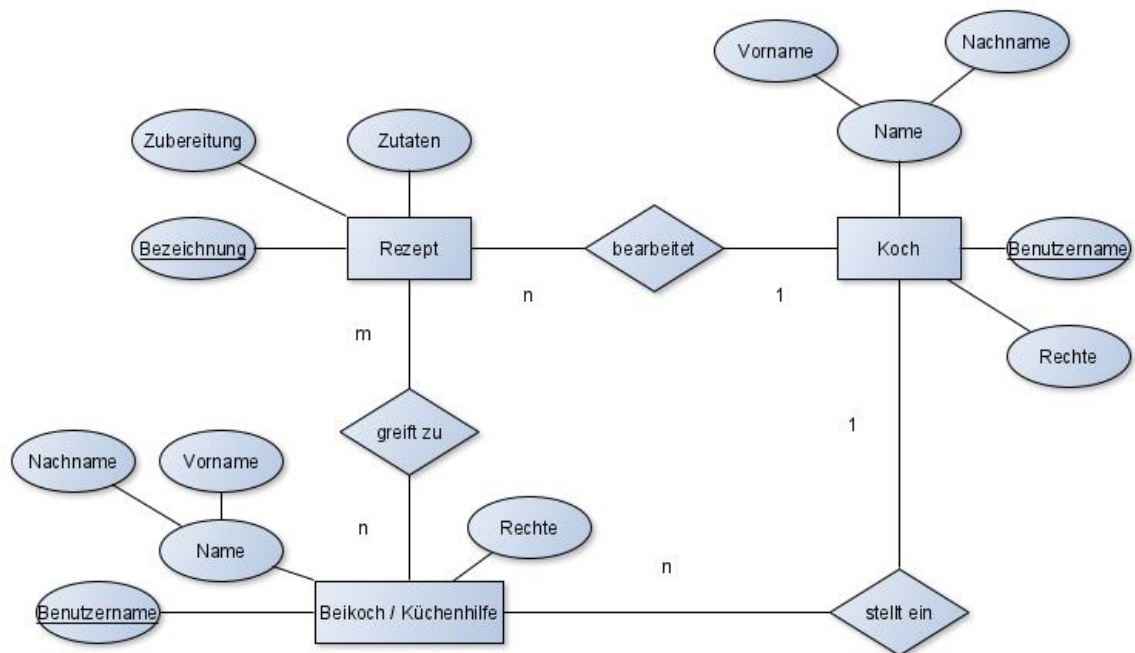


Abbildung 6: ERM

2.8 GUI-Prototyp

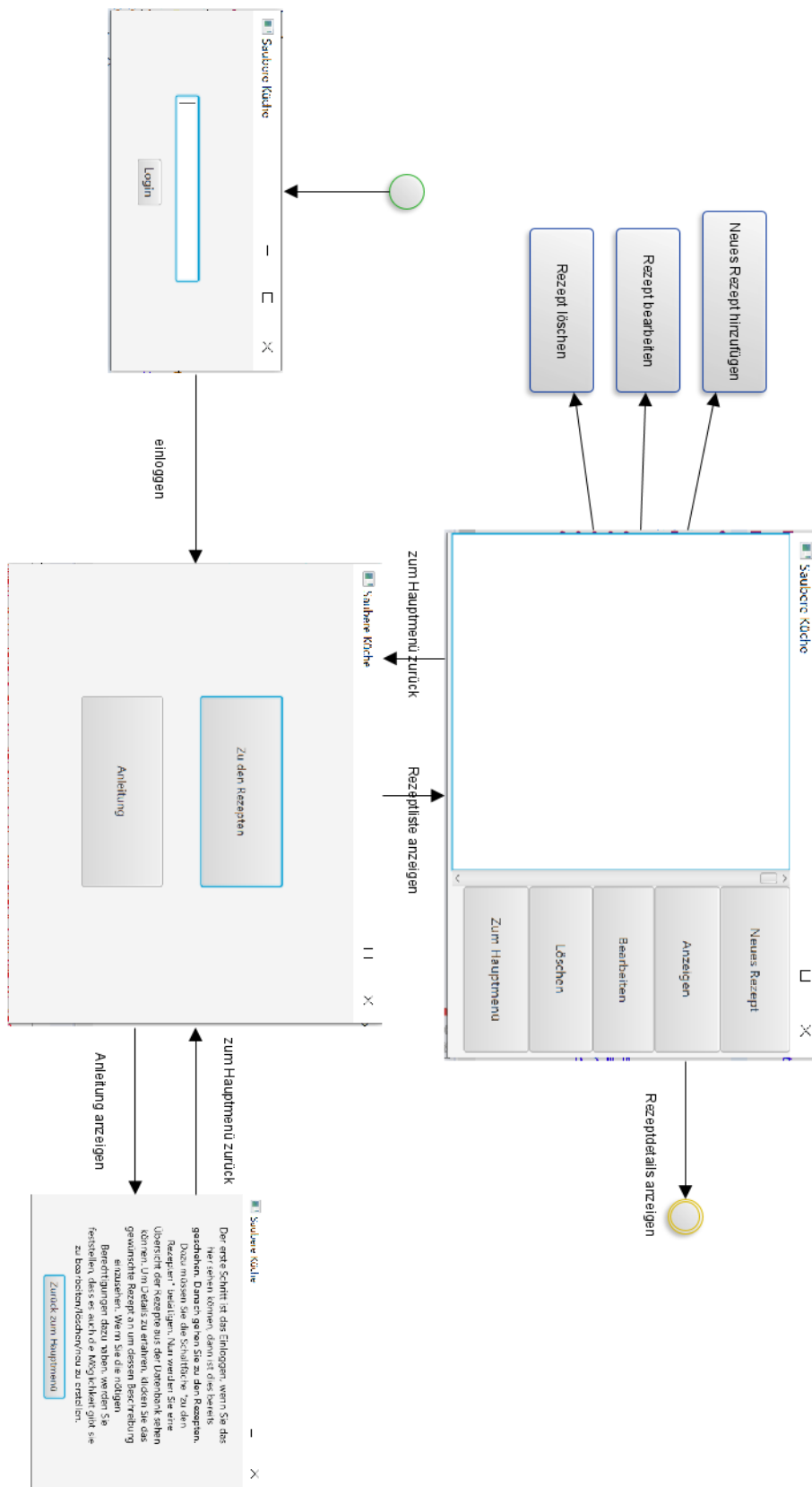



Abbildung 7: GUI-Prototyp

2.9 Persona Beschreibungen

Es folgen in diesem Kapitel Persona Beschreibungen, welche auf den verwendeten Person-Bezeichnungen der User-Stories aus Absatz 2.2 basieren. Sie wurden jeweils mit einer Schablone³ in PowerPoint erstellt und anschließend wurden Screenshots davon erstellt. Für die Fotos der Personas wurden Open-Access Stockfotos verwendet, welche exemplarisch für die Position der Person im Betrieb stehen. Es wurden dafür Piktogramme verwendet. Es gibt viele verschiedene Möglichkeiten eine Persona zu gliedern bzw. zu entscheiden welche Punkte man beleuchten will. Aufgrund der zu erhaltenden Übersichtlichkeit und ausreichender Komplexität hat man sich auf folgende Punkte festgelegt: Hintergrund zur Person; Demografie; Identifikatoren; Erwartungen, Ziele & Emotionen; Herausforderungen; Ideale Lösung und Häufige Einwände. In den folgenden drei Abbildungen/Screenshots sind diese Punkte für jeweils eine andere Position im Betrieb ausgefüllt und vertieft worden. Da es insgesamt nur drei Positionen gibt, welche die Anwendung nutzen werden, hat es sich angeboten jede von ihnen zu beleuchten.

Abbildung 8: Horst Schlemmer, Koch

Hintergrund zur Person: <ul style="list-style-type: none"> - Horst ist seit 20 Jahren Leiter seiner eigenen Großküche. - Er ist gelernter Koch. - Er möchte jedem eine Chance geben, um in seinem Betrieb erfolgreich zu sein. 	Demographie: <ul style="list-style-type: none"> - Männlich - 55 Jahre alt - Vorort von Leipzig - Haus, wohnt mit Frau und 2 Kindern 	Foto: 
Identifikatoren: <ul style="list-style-type: none"> - Horst ist in seiner Freizeit leidenschaftlicher Angler und erweitert sein Wissen über Fisch ständig. - Er hat kaum Kenntnisse mit einem Computer, da er Bücher oder Zeitschriften als Informationsquelle bevorzugt. - Er ist ein respektierter Chef, der es sich nicht erlauben kann, Fehler zu machen. 		
Erwartungen, Ziele & Emotionen: <ul style="list-style-type: none"> - Er möchte seinen Betrieb modernisieren und später gern weitergeben. - Er ist besorgt darüber, dass dies ohne baldige Digitalisierung von Daten nicht möglich ist. 	Herausforderungen: <ul style="list-style-type: none"> - Er hat bereits selbst versucht eine Digitalisierung seiner Rezepte durchzuführen. - Horst ist sehr ehrgeizig und es fällt ihm schwer Verantwortung abzugeben. 	
Ideale Lösung: <ul style="list-style-type: none"> - Horst braucht jemand, der ihn an die Hand nimmt. - Er braucht jemandem, dem er über die Schulter schauen kann um ggf. noch etwas zu lernen. - Er muss verstehen, welche Schritte für die Digitalisierung nötig sind. 	Häufige Einwände: <ul style="list-style-type: none"> - Mögliche Fehler. Fehler, diese nicht von Horst selbst stammen sondern durch die Entwicklung auftreten. - Er möchte seinen Betriebsablauf nicht stoppen um die Entwicklung voranschreiten zu lassen. 	

³ Die Schablone wurde heruntergeladen am 03.08.2020 von <https://www.netspirits.de/blog/personas-erstellen/>

Abbildung 9: Barbara Streuselkuchen, Beikoch



Hintergrund zur Person: <ul style="list-style-type: none"> Barbara arbeitet seit 10 Jahren im Betrieb. Sie hat als Küchenhilfe angefangen und wurde vor 2 Jahren befördert. Sie hofft auf weitere Aufstiegschancen. 	Demographie: <ul style="list-style-type: none"> Weiblich 34 Jahre alt Vorort von Leipzig Wohnung, sie wohnt allein 	Foto: 
Identifikatoren: <ul style="list-style-type: none"> In ihrer Freizeit beschäftigt sie sich damit neue Rezepte auszuprobieren und zu erlernen. Sie informiert sich weitestgehend im Internet. Sie hat einige Freunde, die ebenfalls in der Gastronomie arbeiten. Ihre Freunde erzählen ihr oft, wie modern andere Betriebe bereits sind. 		
Erwartungen, Ziele & Emotionen: <ul style="list-style-type: none"> Barbara wünscht sich eine Modernisierung. Sie würde ihre Arbeitsweise dafür umstellen. Sie möchte gern bei ihren Freunden mitreden. Sie strebt eine Beförderung als Koch an. 	Herausforderungen: <ul style="list-style-type: none"> Barbara ist sehr versiert und konzentriert, wenn sie arbeitet. Sie möchte nur ungern nach der Arbeitszeit eine neue Arbeitsweise erlernen. 	
Ideale Lösung: <ul style="list-style-type: none"> Sie kann sich vorstellen, mit den anderen Mitarbeitern ein Seminar zu besuchen um die neuen Techniken (das Programm) zu erlernen. Barbara sollte sich mit ihren Arbeitskollegen austauschen, um sich andere Meinungen anzuhören. 	Häufige Einwände: <ul style="list-style-type: none"> Alles ist zu kompliziert. Barbara neigt dazu Dinge nicht verstehen zu wollen. Es ist zu zeitaufwändig. Das Lernen beansprucht zu viel Zeit. 	

Abbildung 10: Paul Elefant, Küchenhilfe

Hintergrund zur Person: <ul style="list-style-type: none"> Paul arbeitet seit 2 Wochen im Betrieb. Er hat keine Vorkenntnisse im Gastronomiebereich. Er studiert Angewandte Informatik. Er plant, dass er nach den Semesterferien kündigt. 	Demographie: <ul style="list-style-type: none"> Männlich 22 Jahre alt Leipzig Wohnung, wohnt mit 3 Mitbewohnern 	Foto: 
Identifikatoren: <ul style="list-style-type: none"> Paul arbeitet im Betrieb um sich sein neues Auto zu finanzieren. Er verschwendet wenig Zeit mit der Arbeit und beschäftigt sich kaum damit. Viel lieber spielt er nach der Arbeit Computerspiele. Er ist in Social Media aktiv. Er erzählt kaum jemanden, dass er als Küchenhilfe tätig ist. 		
Erwartungen, Ziele & Emotionen: <ul style="list-style-type: none"> Paul möchte gern, dass seine Tätigkeit als Aushilfe mehr gewürdigt wird. Er findet es schade, dass die Buchhaltung in dem Betrieb kaum digital stattfindet. Er wünscht sich Änderungen in der Buchhaltung. 	Herausforderungen: <ul style="list-style-type: none"> Paul ist nur eine Küchenhilfe und wird wahrscheinlich nicht nach seiner Meinung gefragt. Das Gleiche gilt für seine Ideen, auch wenn er Informatik studiert. Paul wird in ein paar Monaten kündigen. 	
Ideale Lösung: <ul style="list-style-type: none"> Paul ist noch jung und hat durch sein Studium Kenntnisse mit Computern sowie Programmen. Er wird das Programm schnell verstehen und kann vielleicht seinen Kollegen beim Umgang damit helfen. 	Häufige Einwände: <ul style="list-style-type: none"> Keine Lust. Paul neigt dazu, sich nur mit Dingen zu beschäftigen, wenn er es wirklich muss. Keine Zeit. Er möchte sich nicht über die Arbeitszeiten hinaus mit seiner Arbeit beschäftigen. 	

2.10 Anforderungen an IT-Grundschutz

Um ein mittleres, angemessenes und ausreichendes Schutzniveau zu gewährleisten, müssen bestimmte Sicherheitsmaßnahmen getroffen und Schutzprogramme durchgeführt werden. Diese Maßnahmen lassen sich in vier grundlegende Punkte gliedern: technische, infrastrukturelle, organisatorische und personelle.⁴ Im Folgenden Absatz werden diese genannten Punkte weiterführend vertieft.

Die Mitarbeiter des Betriebes müssen in das Programm/System eingeführt werden. Dazu ist es nötig ausreichend vertiefende Schulungen zu halten, Broschüren an diese auszugeben oder Vorträge zu halten. Besonders wichtig ist es die Mitarbeiter darüber zu unterrichten, dass die Technologie nicht dazu in der Lage ist alle Probleme zu lösen. Das bedeutet, dass die Beteiligten ihre Passwörter (in diesem Falle Benutzernamen) nicht mit anderen teilen oder offen herumliegen lassen. Im Bereich der infrastrukturellen Maßnahmen sollte man eine Gebäudesicherung in Betracht ziehen. Es gibt zwar keine vertraulichen Daten im System, allerdings ist es bei den meisten Küchen der Fall, dass die Konkurrenz untereinander dafür sorgt, dass jemand die „Geheimzutat“ herausfinden möchte. Das mag sehr utopisch klingen, aber es ist nicht auszuschließen. Des Weiteren ist es wichtig, dass die Ebene der Leitung (also der Administrator bzw. der Koch/Chef) die Verantwortung für Informationssicherheit übernimmt. Dieser hält und steuert somit die Informationssicherheit aufrecht und dient als Vertrauensperson und agiert im gewissen Maße als Vorbild für alle anderen Mitarbeiter. Ein Mitarbeiter sollte der Leitung vertrauen können und sich bei Sicherheitsvorfällen an sie wenden können. Um den Bereich der technischen Sicherheitsmaßnahmen zu beleuchten sollte man die Rezeptdatenbank bzw. die Festplatten, auf welchen sie sich befindet, verschlüsseln. Bei dieser Art von Programm/System sollte es trotz aller Maßnahmen nicht dazu kommen, dass ein Sicherheitsvorfall auftritt.

⁴ Vgl. BSI - Broschüren - Leitfaden zur Basis-Absicherung nach IT-Grundschutz, 2017.

3 Dokumente zur Beschreibung und Begründung des Softwaredesigns

3.1 Blockdiagramm zur Softwarearchitektur

Die Software lässt sich gemäß 3-Schichten-Architektur in folgende drei Schichten aufteilen:

1. Die Präsentationsschicht, welche die gesamte Benutzeroberfläche repräsentiert. Demnach sind alle „Fenster“ bzw. Scenes, wie es in JavaFX genannt wird, darin inbegriffen.
2. Schicht ist die Funktionsschicht, welche das eigentliche Programm darstellt. Es beinhaltet Verarbeitungselemente und bildet die Schnittstelle zur 3. Schicht.
3. Und letzte Schicht ist die Datenschicht. In diesem Falle soll sie die Form einer relationalen Datenbank haben und der Zugriff auf diese erfolgt durch Datenbank API's bzw. SQL-Anweisungen ausgehend von der Funktionsschicht.

Diese Art und Weise der Aufteilung in der 3-Schichten-Architektur bezeichnet man auch als Daten-Server, da zwischen der 2. und 3. Schicht ein Austausch stattfindet, ähnlich wie bei einer Client-Server Anwendung. Um dies grafisch zu unterstützen kann man Abbildung 11 betrachten, welche unter diesem Absatz zu finden ist. Die Pfeile darin stehen für einen Austausch, der zwischen den Schichten stattfindet.

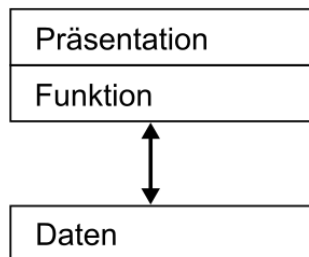


Abbildung 11: Daten-Server

Um die Anwendung dieser Form des 3-Schichten-Modells auf die spezifische Software des Belegs zu zeigen, soll das nachfolgende Blockdiagramm bei der Erklärung unterstützend wirken. Die verwendeten Funktionen stammen aus dem Klassendiagramm (vgl. 2.6), die Erwähnten Tabellen der Datenschicht stammen aus dem Entity-Relationship-Modell (vgl. 2.7) und die einzelnen Unterpunkte/"Fenster"/Szenen der Präsentationsschicht lassen sich im GUI-Prototyp (vgl. 2.8) wiederfinden.

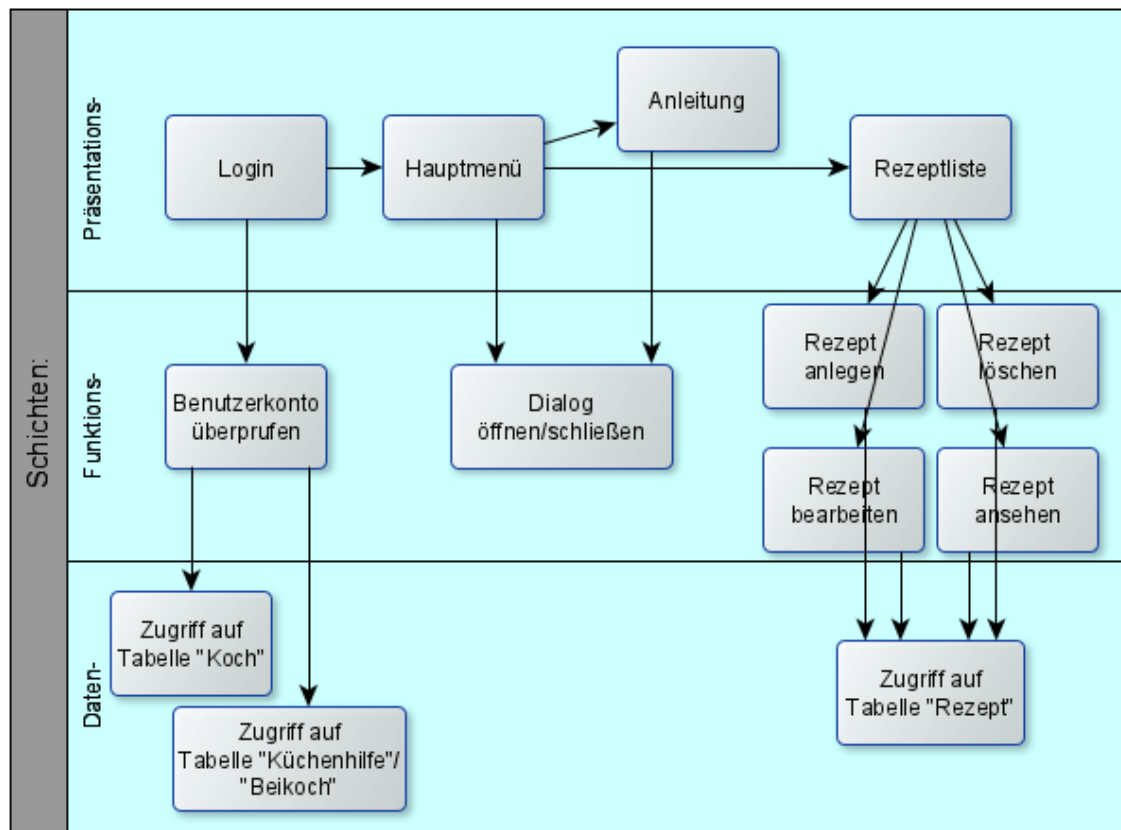


Abbildung 12: Blockdiagramm zur 3-Schichten-Architektur

3.2 Fein-Design-Elemente

Für das Feindesign des Softwaresystems werden einige Ziele gesetzt. Dabei soll die grundsätzlich mögliche Umsetzung der Vorstellungen im Vordergrund stehen. Man muss also, vorbereitend für die Programmierphase, bereits Systemklassen und Objektinteraktionen entwerfen. Mit Systemklassen sind folgende Dinge gemeint: die Kontroll-, Schnittstellen- und Datenklassen. Da hierfür lediglich Datenklassen zur Verwendung kommen, werden also folglich diese erstellt. Im folgenden Screenshot des Programmcodes wird eine dieser Datenklassen mit zugehörigem Objekt gezeigt.

```
/*NameT*/  
class NameT  
{  
    public String Vorname;  
    public String Nachname;  
    public String Position;  
  
    public NameT(String Vorname, String Nachname, String Position)  
    {  
        this.Vorname = Vorname;  
        this.Nachname = Nachname;  
        this.Position = Position;  
    }  
}
```

Abbildung 14: Datenklasse

```
/*NameT für alle Personas*/  
NameT Koch = new NameT("Horst", "Schlemmer", "Koch");  
NameT Beikoch = new NameT("Barbara", "Streuselkuchen", "Beikoch");  
NameT Kuchenhilfe = new NameT("Paul", "Elefant", "Küchenhilfe");
```

Abbildung 13: Objekte der Klasse

4 Zusammenfassung und Schluss

4.1 Rückblick auf Teil A (konzeptioneller Teil)

Die zu entwickelnde Software ist sehr einfach aufgebaut, weshalb es mir öfter schwerfiel Variation für die Grafiken und vor allem die Use Cases zu finden. Das Programm ist durchzogen von Wiederholungen, weshalb es mir an manchen Stellen so schien, als wäre der Umfang doch nicht ausreichend. Aus diesem Grund habe ich beispielsweise mehr als nur 2 User Stories in diesen Beleg eingefügt. An vielen Punkten war es nicht möglich die Grafiken in einer ordentlichen Auflösung bzw. Größe zu exportieren und einzufügen. Die darauffolgende Nicht-Lesbarkeit bzw. Unschärfe der Abbildungen bitte ich hiermit zu entschuldigen. Des Weiteren muss erwähnt werden, dass einige Schritte im Ablauf (wie z.B. die Kommunikation der Benutzernamen) nur auf analoge Basis stattfinden. Die Software bekommt somit nur die Benutzernamen eingespeist und muss sie selbst nicht an die individuellen Nutzer weitergeben bzw. zuteilen. Die Vorgänge wurden trotzdem in Use-Cases und weitere Diagramme aufgenommen, da sie als unverzichtbar eingestuft wurden.

Besonders interessant war die Ausarbeitung eines eigenen Pflichtenheftes. In der Art und Weise, wie man sich dabei Gedanken über das eigene Belegthema machen musste, empfand ich als sehr förderlich für das Verständnis. Ich hätte persönlich nicht erwartet, dass es möglich ist damit ein vollständig funktionsfähiges Konzept zu erstellen. Des Weiteren möchte ich erwähnen, dass generell diese Form der Ausarbeitung sehr zum Verständnis der Lehrveranstaltungen beigetragen hat. Dadurch, dass man sich selbstständig mit fast allen Themen nochmal auseinandersetzen musste, wurden auch die jeweiligen Vorgehensweisen, für z.B. die Erstellung einer Use-Case-Beschreibung, gefestigt.

4.2 Rückblick auf Teil B (Praktischer Teil)

Durch die technischen Voraussetzungen welche mir zu Hause gegeben sind erwies sich die Nutzung von diversen Programmen auf dem eigenen Computer als deutlich schwieriger als anfangs gedacht. Letztendlich wurde somit für die Modellierung von Daten, Übersichten oder Diagrammen auf online verfügbare Dienste, wie z.B. draw.io zurückgegriffen. Diese Entscheidung sorgte dafür, dass die Vielfalt bei den Design-Elemente welche mir zur Verfügung standen eher mangelhaft war. Ich war zwar in der Lage meine Diagramme und Übersichten logisch und verständlich zu erstellen, zufrieden bin ich damit allerdings nicht ganz. Die gleichen Probleme kamen bei der Nutzung von Programmierumgebungen wie eclipse für den Code und Scenebuilder für die Anwendungsfenster auf. Dabei hingen sich die Programme regelmäßig an vielen Stellen auf und es gab vergleichsweise häufig komplette Abstürze des selbst erstellten Programms durch Fehler beim Programmieren, welche mit besserer Ausstattung sicherlich vermeidbar gewesen wären. Dies führte zu steigender

Frustration und Motivationsverlust beim Bearbeiten des Beleges. Durch Mangel an Speicherplatz war es, wie bereits vorher erwähnt, schwierig jede benötigte Software bereitzustellen. So genügte es bereits, die Desktopversion von Git, Eclipse, Zusatzpakete, usw. zu installieren und einige Testprojekte zu speichern, um an technische Grenzen zu gelangen, von denen ich dachte, dass sie unmöglich zu erreichen sind. Darauffolgend war es also nur möglich flüssig zu arbeiten, wenn ich mit der geringsten Speicherplatznutzung und dem geringsten Prozessoraufwand arbeitete. Um mit den negativen Erfahrungen abzuschließen möchte ich nun noch auf Besonderheiten eingehen, welche mir besonders positiv in Erinnerung geblieben sind. Besonders interessant war die Erstellung des Blockdiagrammes zur Softwarearchitektur, da für die Informationen darin bereits vorherige Kapitel wiederverwendet werden mussten. Es war im Laufe der Belegbearbeitung öfter der Fall, dass dies getan werden musste, allerdings stach diese Form der Verarbeitung der bereits erarbeiteten Daten insbesondere heraus, da viel mehr Inhalte aus verschiedenen Kapiteln gebraucht worden sind.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Frohburg, den 31.08.2020



Juliet Petzold