

Diplomarbeit

## Entwicklung eines Prototyps für ein Kita Essenbestellungstool



Name: Perolini  
Vorname: Joshua  
Höhere Fachschule: Teko Bern  
Klasse: B-TIN-18-a

Datum: 30.10.2021

# Management Summary

## Aufgabenstellung

Einer meiner Freunde, der Teilhaber von NeoTropical Catering & Events ist, fragte mich, ob ich ein Tool programmieren kann. Die Firma verwendet aktuell keine Software. Das von mir zu entwickelnde Tool soll in vereinfachter weisse die Essenbestellungen aufnehmen und abspeichern. Es wird ein Bestelungstool für die Kitas sein. In dem Bestelungstool werden verschiedene Angaben getätigt, damit die Firma NeoTropical einfacher und schneller das Essen zubereiten kann.

Zu Beginn war ich skeptisch, doch mit der Zeit entdeckte ich, dass die Idee perfekt für meine Diplomarbeit ist, da ich selbstständig ein reales Problem lösen kann. Da das Bestelungstool dafür zu aufwendig ist, wird nur ein Prototyp vom Bestelungstool erstellt.

## Varianten

Für die Entwicklung eines Prototyps wurden nicht viele Varianten erarbeitet. Ich entschied mich schnell für eine Variante, die ich - mit ein paar Anpassungen - bis zum Schluss durchgezogen habe.

## Konzept

Es wurden verschiedene Konzepte für den Prototyp erstellt:

- Lösungskonzept
- Testkonzept
- GUI-Konzept

Weitere Details sind in der Dokumentation ersichtlich.

## Realisierung

Der Prototyp wurde gemäss der Themeneingabe zur Diplomarbeit erstellt und die Entwicklung wurde dokumentiert. Details zur Entwicklung sind im Abschnitt Realisierung ersichtlich.

## Testbericht

Für den Prototyp wurden einige Tests durchgeführt. Die Tests wurden erfolgreich bestanden und dokumentiert. Detailliert zu finden sind die Testfälle im Abschnitt Testprotokoll in der Dokumentation.

## Mittelbedarf

Der Prototyp wurde mit Visual Studio erstellt und als Programmiersprache wurde C# verwendet. Die Datenbank wurde mit Microsoft SQL-Server Management Studio erstellt. Für die Diagramme verwende ich ConceptDraw 7 Professional, dadurch wurde nur ein Laptop benötigt. Da dieses Gerät bereits vorhanden war, benötigte ich keine zusätzlichen Mittel.

## Fazit

Die Diplomarbeit verlief nicht ganz planmässig, jedoch konnte ich viel Neues lernen. Ich konnte am Schluss alle Ziele (muss und kann) im Essenbestelungstool realisieren. Damit dies möglich wurde, musste ich viel Zeit in Recherche stecken.

# Inhaltsverzeichnis

1	Initialisierung	8
1.1	Ausgangslage	8
1.2	Situationsanalyse	8
1.2.1	Ist-Zustand	8
1.2.2	Soll-Zustand	8
1.3	Wirtschaftlichkeit	8
1.4	Rahmenbedingung	9
1.5	Abgrenzung	9
1.5.1	Generelle Abgrenzung der Arbeit	9
1.5.2	Geleistete Vorarbeiten	9
1.6	Vorgehensmodell	10
1.7	Stakeholder-Analyse	11
1.8	Grobe Anforderungen an das neue System	12
1.9	Lösungskonzept	13
1.9.1	Architektur der Software	13
1.9.2	Variantenentscheid	14
1.10	Projektmanagement	15
1.10.1	Projektplanung	15
1.10.2	Risikoanalyse	16
1.10.3	Qualitätsmanagement	17
1.10.4	Konfigurationsmanagement	18
2	Konzept	19
2.1	Detailanforderungen an die neuen Systeme	19
2.1.1	Funktionale Anforderungen	19
2.1.2	Nicht-funktionale Anforderungen	20
2.1.3	Organisatorische Anforderungen	20
2.2	Kontextdiagramm	21
2.3	Use-Case	22
2.4	Sequenzdiagramme	30
2.5	Zustandsdiagramme	36
2.6	Klassendiagramm	38

2.7	Modellierung der Datenbank	39
2.7.1	Leistungsdiagramm	39
2.7.2	Entitäts-Matrix	41
2.7.3	Entity-Relationship-Modell	42
2.7.4	Entity-Relationship-Diagramm	43
2.8	Testkonzept	44
2.8.1	Anforderungsanalyse	45
2.9	GUI-Konzept	49
2.10	Endprotokoll Konzept	53
3	Realisierung	54
3.1	Programmierungsumgebung	54
3.2	Softwareaufbau	54
3.2.1	Kita	54
3.2.2	NeoTropical	59
3.3	Datenbankimplementierung und Datenbankanbindung	64
3.3.1	ADO.NET Framework	64
3.3.2	Programm realisieren	64
3.4	Testprotokoll	67
4	Schlussbericht	70
4.1	Vergleich Ist/Soll	70
4.2	Persönliches Fazit	70
5	Eigenständigkeitserklärung	71
6	Anhang	73

# Abbildungsverzeichnis

Abbildung 1: Hermes 5.1	10
Abbildung 2: Stakeholder-Analyse	11
Abbildung 3: Model-View-Controller	14
Abbildung 4: Grob Projektplan	15
Abbildung 5: Teilausschnitt Projektplan	15
Abbildung 6: Risikomatrix	16
Abbildung 7: Header	18
Abbildung 8: Kontextdiagramm Kita	21
Abbildung 9: Kontextdiagramm NeoTropical	21
Abbildung 10: Use Case Kita Login	22
Abbildung 11: Use Case Kita Essensbestellungen ausfüllen	23
Abbildung 12: Use Case Kita Archiv kontrollieren	24
Abbildung 13: Use Case NeoTropical Login	25
Abbildung 14: Use Case NeoTropical Neue Kita registrieren	26
Abbildung 15: Use Case NeoTropical Essensplan anpassen	27
Abbildung 16: Use Case NeoTropical Archiv kontrollieren	28
Abbildung 17: Use Case NeoTropical Endprotokoll erstellen	29
Abbildung 18: Sequenzdiagramm Kita Essensbestellungen ausfüllen	30
Abbildung 19: Sequenzdiagramm Kita Archiv kontrollieren	31
Abbildung 20: Sequenzdiagramm NeoTropical Essensplan anpassen	32
Abbildung 21: Sequenzdiagramm NeoTropical Neue Kita registrieren	33
Abbildung 22: Sequenzdiagramm NeoTropical Archiv kontrollieren	34
Abbildung 23: NeoTropical Endprotokoll erstellen	35
Abbildung 24: Kita Zustandsdiagramm	36
Abbildung 25: NeoTropical Zustandsdiagramm	37
Abbildung 26: Kita Klassendiagramm	38
Abbildung 27: NeoTropical Klassendiagramm	38
Abbildung 28: Leistungsdiagramm Kita	39
Abbildung 29: Leistungsdiagramm NeoTropical	40
Abbildung 30: Entity-Relationship-Modell	42
Abbildung 31: Entity-Relationship-Diagramm	43
Abbildung 32: Testkonzept	44
Abbildung 33: GUI-Login	49
Abbildung 34: GUI-Essensbestellung	50
Abbildung 35: GUI-Essensplan	50
Abbildung 36: GUI-Gericht	51
Abbildung 37: GUI-Registrieren	51
Abbildung 38: GUI-Archiv	52
Abbildung 39: Endprotokoll	53
Abbildung 40: Überprüfung Login	54
Abbildung 41: Essensplan anzeigen	55

Abbildung 42: Überprüfen des Ernährungskriteriums Besonderes	56
Abbildung 43: Durchläufe überprüfen	57
Abbildung 44: Passwort anzeigen	59
Abbildung 45: Login-Form	59
Abbildung 46: Datum ermitteln	60
Abbildung 47: Enter-Methode	61
Abbildung 48: Leave-Methode	61
Abbildung 49: Kita-List erstellen	62
Abbildung 50: Kita-List in CheckedListBox einfügen	62
Abbildung 51: DataTable erstellen für Endprotokoll	63
Abbildung 52: DB-auslesen	64
Abbildung 53: DB-schreiben	65
Abbildung 54: DB-löschen	65
Abbildung 55: Endprotokoll erstellen	66
Abbildung 56: Endprotokoll Zellen befüllen	66

## Tabellenverzeichnis

<i>Tabelle 1: Hermes 5.1</i>	10
<i>Tabelle 2: Stakeholder-Analyse</i>	11
<i>Tabelle 3: Risikoanalyse</i>	16
<i>Tabelle 4: Use Case Kita Login</i>	22
<i>Tabelle 5: Use Case Kita Essensbestellungen ausfüllen</i>	23
<i>Tabelle 6: Use Case Kita Archiv kontrollieren</i>	24
<i>Tabelle 7: Use Case NeoTropical Login</i>	25
<i>Tabelle 8: Use Case NeoTropical Neue Kita registrieren</i>	26
<i>Tabelle 9: Use Case NeoTropical Essensplan anpassen</i>	27
<i>Tabelle 10: Use Case NeoTropical Archiv kontrollieren</i>	28
<i>Tabelle 11: Use Case NeoTropical Endprotokoll erstellen</i>	29
<i>Tabelle 12: Kita Zustandsdiagrammanalyse</i>	36
<i>Tabelle 13: NeoTropical Zustandsdiagrammanalyse</i>	37
<i>Tabelle 14: Entitäts-Matrix</i>	41
<i>Tabelle 15: Anforderungsanalyse Kita</i>	46
<i>Tabelle 16: Anforderungsanalyse NeoTropical</i>	48
<i>Tabelle 17: Testprotokoll Kita</i>	68
<i>Tabelle 18: Testprotokoll NeoTropical</i>	69

# 1 Initialisierung

## 1.1 Ausgangslage

Zu entwickeln ist eine Applikation, welche in vereinfachter Weise erlaubt, die Essensplanung der Kitas zu erleichtern. Dabei sollen sowohl die Kitadaten als auch die effektiven Angaben über die Bestellung hinterlegt werden können. Die Applikation soll einerseits den Köchen einen besseren Überblick über die Essensbestellungen erlauben, andererseits auch die Grundlage für ein effizientes Buchhaltungssystem liefern. Die Kitaangestellten sollen in der Lage sein, Bestellungen in einem Archiv direkt in der Applikation zu sehen.

## 1.2 Situationsanalyse

### 1.2.1 Ist-Zustand

Die Firma NeoTropical schickt jeder Kita, welche Kunde ist, ihren Menüplan. Die Kitas legen ihren Menü Bestellungsplan als ausgefülltes Blatt Papier in die Essensboxen, welche die Firma NeoTropical jeden Tag zu den Kitas bringt und wieder abholt. Darauf steht ihr Menü Bestellungsplan der nächsten Woche oder des nächsten Monats.

### 1.2.2 Soll-Zustand

Es wurde ein Prototyp erstellt, in dem die Firma NeoTropical den Essensplan jede Woche anpassen kann. Die Essensbestellungen müssen wöchentlich von den Kitas geschickt werden, sonst ist keine Bestellung möglich. Die Kitas können auch die Essensbestellungen von zwei Wochen abschicken, wenn die Firma NeoTropical für beide Wochen schon einen Essensplan erstellt hat. Bei jeder Bestellung für jeden Tag können die Kitas verschiedene Ernährungskriterien auswählen, die zur Auswahl stehen: Vegetarisch, Vegan, Glutenfrei, Laktosefrei, Ohne Eier, Erdnüsse und Besonderes. Bei jedem Ernährungskriterium, das Sie bestellen möchten, müssen die Kitas eine Anzahl Bestellungen angeben.

## 1.3 Wirtschaftlichkeit

Durch die Erstellung eines vollumfänglichen Prototyps und einer klaren Struktur kann die geplante Applikation besser entwickelt werden. Weiter können Probleme früher behoben werden.

Durch ein effizientes Bestellsystem benötigt die Firma NeoTropical weniger Zeit für Arbeiten, welche das Tool übernimmt, wie zum Beispiel, um alle Essensbestellungen abzuarbeiten, da die Essensbestellungen für die nächste Woche in einem Excel-Dokument dargestellt werden können. Oder der Essensplan muss nicht jeden Monat separat zu jeder Kita gesendet werden.

Alle eingegangenen Essensbestellungen werden in einem Archiv gespeichert und können wieder betrachtet werden. Durch ein Archiv kann die Firma einfacher dokumentieren, wann Sie was gekocht hat und wie viel.

Mit einem Bestellsystem ist NeoTropical immer abgesichert, falls eine Kita reklamiert, dass NeoTropical zu wenig Bestellungen oder das Falsche geliefert hat.

## 1.4 Rahmenbedingung

In der Diplomarbeit wird erwartet, dass die gelernten Methoden und Instrumenten des Projektmanagements sowie des System Engineerings in die Praxis umgesetzt werden. Dazu gehören insbesondere (aber nicht nur) eine fundierte und detaillierte Planung der verschiedenen Aktivitäten sowie eine vollständige und nachvollziehbare Dokumentation des Projekts.

Das Projektmanagement umfasst nebst der bereits aufgeführten Planung noch ein entsprechendes Controlling, eine Risikoanalyse mit Massnahmen, ein Qualitätsmanagement sowie ein Konfigurationsmanagement in einer sinnvollen Granularität. Analyse und Design werden mittels UML vorgenommen.

Für die Diplomarbeit bestehen folgende Rahmenbedingungen, die von der TEKO bestimmt worden sind:

Rahmenbedingungen Administrativ der Projektarbeit

- Starttermin: Freitag, 17. September 2021
- Abgabetermin: Montag, 1. November 2021
- Dauer der Arbeit: ca. 6 Wochen
- Präsentationstermin: Freitag/Samstag, 12./13. November 2021

Rahmenbedingungen des Projektes

- Die Diplomarbeit ist eine Einzelarbeit und wird als Informatikprojekt durchgeführt.
- Die Diplomarbeit wird mit einem Textverarbeitungsprogramm im Format A4 verfasst, einseitig beschrieben, fortlaufend nummeriert, gebunden oder in einem Ordner.
- Ca. 150 – 250 Stunden Arbeitsaufwand
- Die Arbeit umfasst eine schriftliche Dokumentation.

## 1.5 Abgrenzung

### 1.5.1 Generelle Abgrenzung der Arbeit

Von der zu erstellenden Applikation wird nur ein Prototyp erstellt. Im Prototyp wird keine Historie erstellt werden. Der Prototyp kann eine Stand Alone Applikation sein und es müssen keine Sicherheitsaspekte beachtet werden.

### 1.5.2 Geleistete Vorarbeiten

Erstes Zusammentreffen mit Ueli Enggist in der Catering Firma. Beim Zusammentreffen wurde mir physisch gezeigt, wie sie in der Firma arbeiten. Es wurde mir erklärt, wie die Firma aktuell die Menüs plant. Mir wurde auch der Menüplan, den die Kitas bekommen, gezeigt sowie der Menü Beststellungsplan.

## 1.6 Vorgehensmodell

Für die Diplomarbeit wählte ich das Vorgehensmodell Hermes 5.1 aus. An der Teko habe ich schon oft mit Hermes 5.1 gearbeitet, deshalb war mir von Anfang an klar, dass ich mich beim Vorgehensmodell für Hermes entscheiden würde. Bei diesem Modell ist der Projektablauf klar strukturiert und in Phasen und Meilensteine unterteilt. Dies hilft bei der Planung und vereinfacht die Durchführung. Hermes ist ein offener Standard zur Führung und Abwicklung von IT-Projekten.

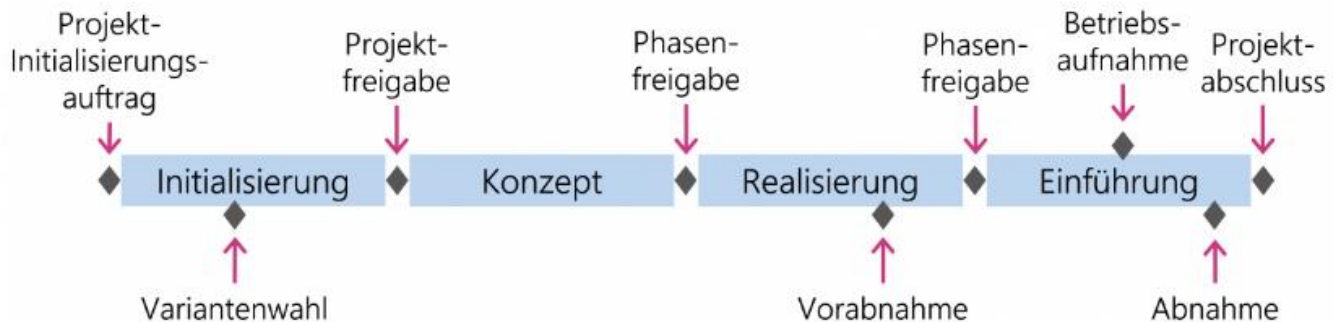


Abbildung 1: Hermes 5.1

Phase	Beschreibung	Ergebnisse
Initialisierung	In der Phase Initialisierung werden in der Studie die Ziele festgelegt und die Anforderungen so weit erarbeitet, dass Varianten gebildet und bewertet werden können. Auf der Grundlage der Variantenwahl wird der Projektauftrag erstellt.	Der Projektauftrag wird analysiert und eine Studie erstellt. In der Studie befinden sich die Ziele, Grobanforderungen und Varianten.
Konzept	In der Phase Konzept werden die in der Studie dokumentierten Grobanforderungen als Systemanforderungen konkretisiert und vervollständigt. In Detailstudien werden projektspezifische Lösungskonzepte erarbeitet. Sie stellen einen Teil der Systemarchitektur dar. Diese beschreibt das System mit den Prozessen, der Funktionalität, den Systemkomponenten und ihrer Einbindung über Schnittstellen ins Systemumfeld.	Aus der Studie der Initialisierungsphase wird eine Detailstudie erstellt. Am Ende der Konzeptphase sind die Systemanforderungen und die Systemarchitektur definiert. Wenn möglich ist auch ein Prototyp gebaut.
Realisierung	Auf der Grundlage der Systemarchitektur wird die Detailspezifikation erstellt und darauf basierend das System entwickelt. Dazu gehört auch das Testen als Voraussetzung für den Entscheid zur Vorabnahme.	So wie der Name Realisierung sagt, ist am Ende der Phase das System mit der Detailspezifikation entwickelt.
Einführung	In der Phase Einführung wird das System aktiviert und anschliessend der Entscheid zur Abnahme getroffen.	Am Ende der Phase Einführung ist das System aktiviert und betriebsbereit.

Tabelle 1: Hermes 5.1

### 1.7 Stakeholder-Analyse

Eine Stakeholder Analyse wird für eine systematische Ermittlung der Stakeholder erstellt. Ein Stakeholder (Interessenträger) ist eine Gruppe oder Person, welche an einem positiven Verlauf des Projekts interessiert ist.

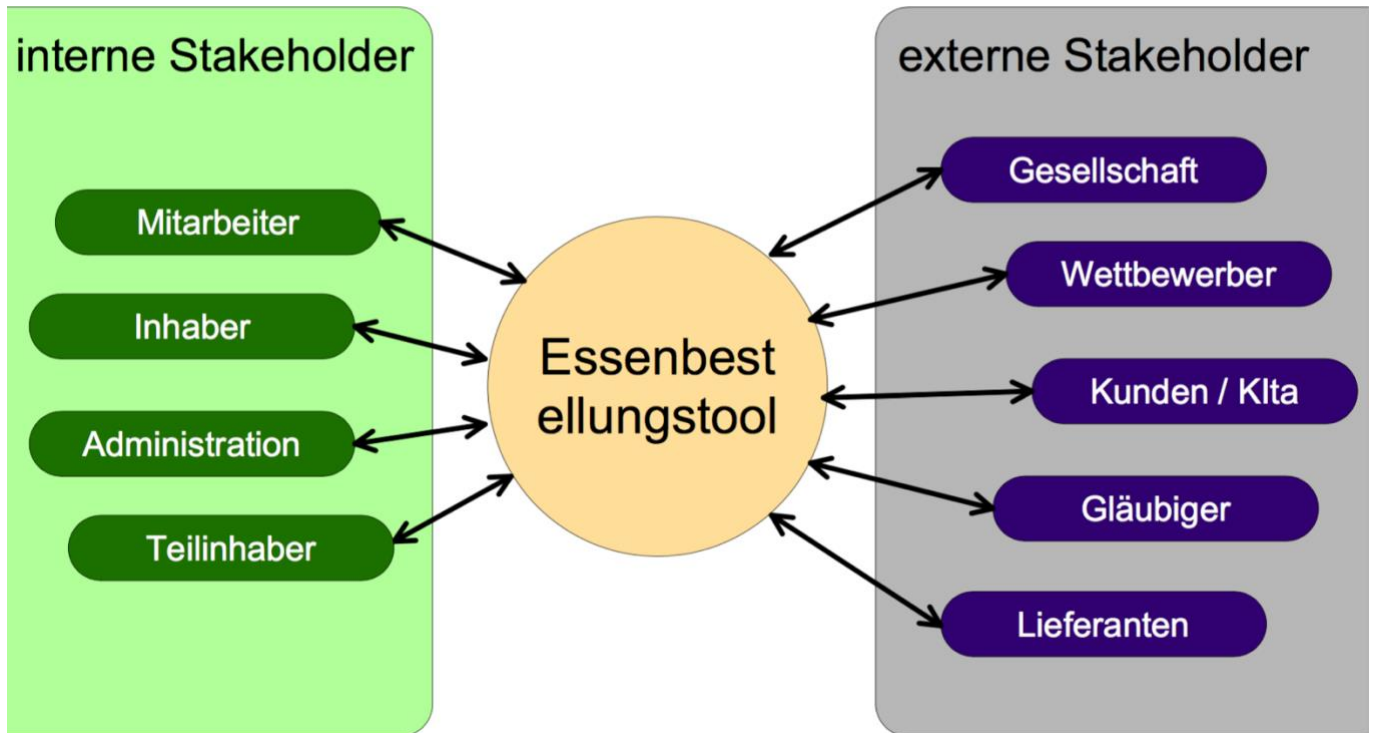


Abbildung 2: Stakeholder-Analyse

	Stakeholder	Interaktion
interne Stakeholder	Mitarbeiter	Prozessowner, technisches- und fachliches Wissen, Nutzen
	Teilinhaber	Steering Committee, Vorgaben
	Inhaber	Budget, Freigabe, Auftraggeber, Rahmenbedingungen, Ressourcen
	Administration	Support, Fachwissen
externe Stakeholder	Gesellschaft	Indirekter Einfluss
	Wettbewerber	Druck auf Prozessoptimierung und Differenzierung
	Kunden / Kita	Indirekte Interaktion
	Gläubiger	Indirekter Einfluss
	Lieferanten	Indirekter Einfluss

Tabelle 2: Stakeholder-Analyse

## 1.8 Grobe Anforderungen an das neue System

Der zu entwickelnde Prototyp für das Kita Essenbestellungstool hat zwei verschiedene Logins. Das eine Login ist für die Mitarbeiter der Firma NeoTropical. In dieser Applikation ist es möglich, das Tool zu verwalten. Das zweite Login ist für die Kita Kunden, die ihre Bestellungen absenden können.

In der Applikation für die Firma NeoTropical liegt das Hauptaugenmerk auf dem Menüplan. Den Menüplan muss NeoTropical jede Woche erneuern, sie können aber auch zukünftige Wochenpläne im Voraus anlegen. Im Archiv von NeoTropical können sie sehen, zu welcher Woche sie schon einen Wochenplan erstellt haben und welche Gerichte sie in den Wochenplan geschrieben haben.

Ein weiterer wichtiger Punkt ist, dass die Firma NeoTropical selbstständig in der Applikation neue Kitas registrieren kann. So kann die Firma Neo Tropical auf eine einfache, effiziente Art neue Kundenkonten anlegen und kann gleichzeitig vermeiden, dass es keine unerwünschten Accounts gibt, von welchen NeoTropical nichts weiss. Für die Anforderungen bezüglich des Registrierens musste ich von Ueli Enggist wissen, welche Daten einer Kita für sie relevant sind.

Sobald die Kitas ihre Bestellungen abgeschickt haben, können die Mitarbeiter der Firma NeoTropical ein Endprotokoll zu jeder Kita erstellen. Das Endprotokoll wird in Excel verfasst. Darin wird gut ersichtlich dargestellt, an welchem Tag wie viele Bestellungen eingegangen sind und welche Ernährungskriterien zu den Bestellungen ausgewählt worden sind.

In der zweiten Applikation, die für die Kitas ist, liegt das Hauptaugenmerk auf der Essensbestellung. Die Kitas müssen die genaue Woche auswählen können. Sobald dies die Kitas gemacht haben, können sie zu jedem Tag ihre Bestellung eingeben. Die Bestellungen werden in zwei Gruppen aufgeteilt, Erwachsene und Kinder, da dies für NeoTropical wegen der Grösse der jeweiligen Portion relevant ist. Bei der Bestellung ist es wichtig auf die Ernährungskriterien zu achten. Es gibt insgesamt sieben Ernährungskriterien: Vegetarisch, Vegan, Glutenfrei, Laktosefrei, Ohne Eier, Erdnüsse und Besonderes. Bei jedem Ernährungskriterium, dass Sie bestellen möchten, müssen die Kitas eine Anzahl an Bestellungen angeben.

Damit auch die Kitas sehen, wie viele Bestellungen Sie bezüglich eines jeweiligen Tages abgeschickt haben, wird es auch ein Archiv geben. Darin können die Kitas ihre eigenen Bestellungen detailliert ansehen.

## 1.9 Lösungskonzept

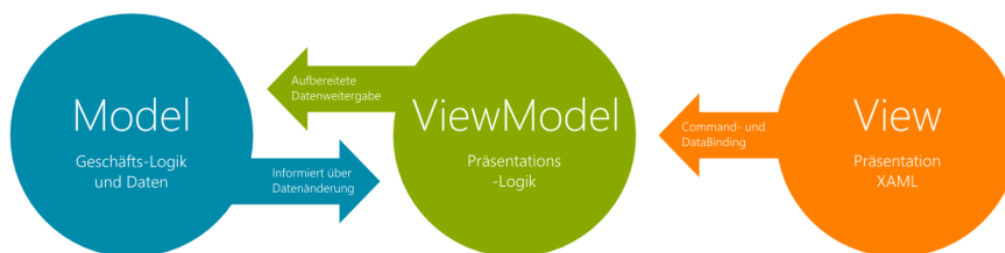
### 1.9.1 Architektur der Software

Für mich kamen für die Architektur der Software nur zwei Programm-Entwurfsmuster in Frage: Model View ViewModel und Model View Controller. Bevor ich mit der eigentlichen Diplomarbeit starten konnte, musste ich entscheiden, auf welche Softwarearchitektur ich setzen wollte. Beide Programm-Entwurfsmuster bieten Vor- und Nachteile.

#### 1.9.1.1 MVVM

MVVM (Model View ViewModel) ist ein Programm-Entwurfsmuster. Es sieht die Rollentrennung von UI-Designern (VIEW) und Entwicklern (MODEL und VIEWMODEL) vor. Durch diese Trennung können die Anwendungsschichten von verschiedenen Arbeitsgruppen entwickelt werden. Designer können einen Fokus auf das Benutzererlebnis und die UI legen (VIEW) und Entwickler unabhängig davon die UI- und Geschäftslogik schreiben (MODEL und VIEWMODEL). In der Regel sind keine aufwendigen UI-Tests nötig. Stattdessen genügen codebasierte Modultests des ViewModel.

- View : XAML, UX-Logik, Bindung zum ViewModel/Model und der Datenkonvertierung über IValueConverter
- ViewModel : Stellt Bindbare Propertys und Commands bereit und enthält UI-Logik.
- Model : Entität, Geschäftsregeln und Validierungs-Logik.



MVVM  
MODEL VIEW VIEWMODEL



### 1.9.1.2 MVC

Im Fach objektorientierte Programmierung (OOP) haben wir das Model View Controller (MVC) genauer erarbeitet. Das MVC-Muster ist vor allem in der Softwareentwicklung mit Programmiersprachen wie Java, Smalltalk, C# und C++ verbreitet.

- Das Modell (Model) wird verwendet, um die Daten von C# in eine Datenbank zu schreiben. Dieses Objektmodell enthält keinerlei Informationen über die Benutzeroberfläche.
- Die Präsentation (View) als eine Sammlung von Klassen, die die Elemente in der Benutzeroberfläche darstellen (alle Dinge, die Benutzer sehen und auf die sie auf dem Bildschirm reagieren können, wie Buttons, ListView und so weiter). Die Präsentation ist vor allem für die Darstellung der benötigten Daten aus dem Modell und die Entgegennahme von Benutzerinteraktionen zuständig.
- Eine Steuerung (Controller), die die Klassen repräsentiert und das Modell und die Präsentation verbindet. Die Steuerung wird für die Kommunikation zwischen den Klassen im Modell und in der Präsentation verwendet. Sie verwaltet eine oder mehrere Präsentationen, nimmt von ihnen Benutzeraktionen entgegen, wertet diese aus und agiert entsprechend.

## Model-View-Controller

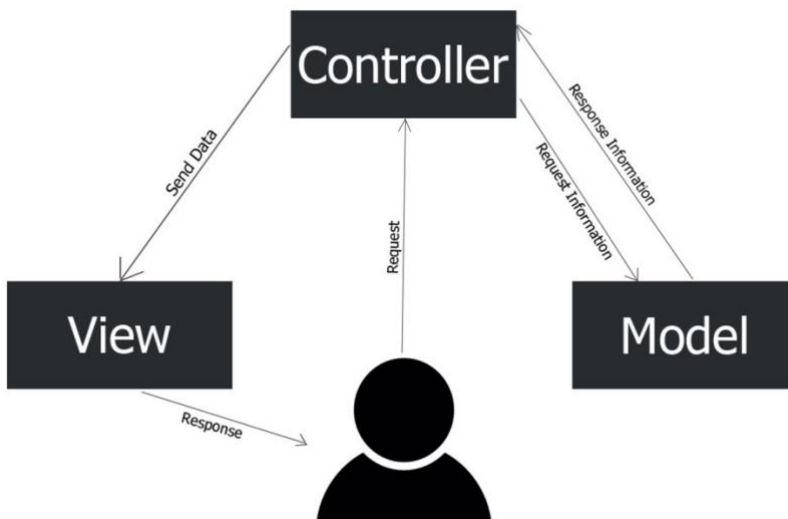


Abbildung 3: Model-View-Controller

## 1.9.2 Variantenentscheid

### 1.9.2.1 Architektur der Software

Für die Architektur der Software entschied ich mich für Model-View-Controller (MVC). MVVM hat auch viele Vorteile aber für mich war entscheidend, dass ich bis jetzt noch kein Projekt mit MVVM durchgeführt habe und bei der Diplomarbeit nicht eine neue Architektur ausprobieren möchte. Da wir einen Prototyp entwickeln, kennen wir die zukünftigen Anforderungen an Interaktion und Präsentation der Daten noch nicht. MVC hat den Vorteil, dass die Daten unabhängig von ihrer Repräsentation geändert werden können und umgekehrt.

### 1.9.2.2 Programmiersprache / Datenbank

Als Programmiersprache kam für mich nur Visual Studio C# in Frage, weil ich bis jetzt nur mit dieser Programmiersprache gearbeitet habe. In C# kann ich die komplette Semesterarbeit realisieren.

Ich entschied mich bei der Datenbank für Microsoft SQL-Server. Den SQL-Server hatten wir letztes Semester in der Schule behandelt. Mit dieser Datenbank kann ich die komplette Semesterarbeit realisieren

## 1.10 Projektmanagement

### 1.10.1 Projektplanung

Ich erstellte als erstes eine Grob Projektplanung der gesamten Diplomarbeit. Da ich mein Vorgehensmodell kannte, wusste ich welche Phasen meine Diplomarbeit haben würde. In der Grobplanung erstellte ich auch die Meilensteine der Diplomarbeit. Vor allem ist in der Grob Projektplanung wichtig, dass man die Daten für eine erfolgreiche Durchführung herausfindet. Hier muss man auf Kriterien achten, wie den Termin des Projektabschlusses und wie viele Aufgaben die einzelnen Phasen haben.

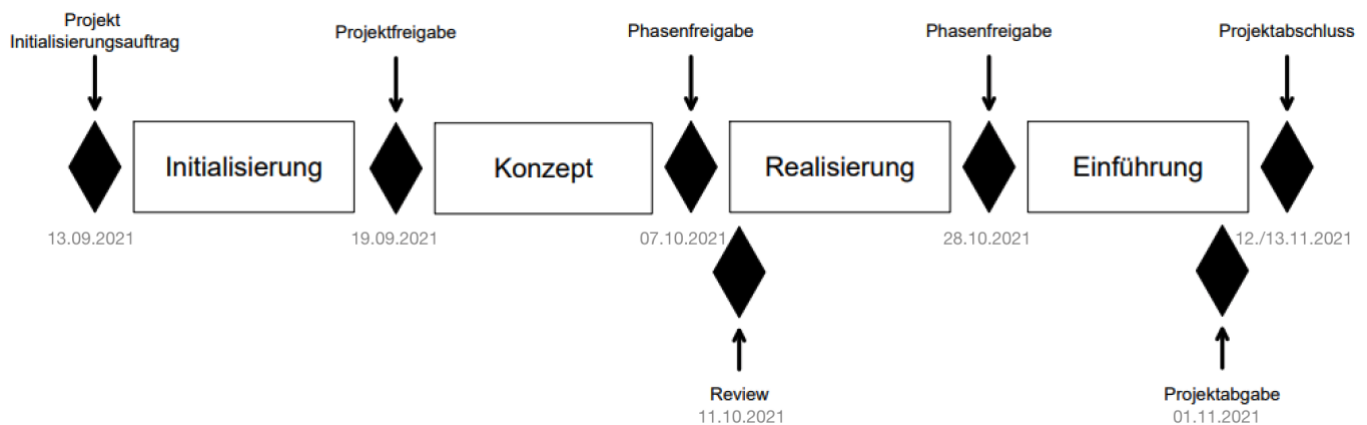


Abbildung 4: Grob Projektplan

Meine Projektplanung erstellte ich mit Excel, da ich dafür kein anderes Tool zur Verfügung hatte und mir Jürg Hess vor der Diplomarbeit mitgeteilt hat, dass für so ein kurzes Projekt Excel ausreichen würde. Ich plante mein Projekt nach dem Vorgehensmodell Hermes 5.1. Bis zur Phase Realisierung funktionierte alles ziemlich planmässig, ich war sogar schneller als geplant. Danach konnte ich nicht mehr strukturiert nach Plan arbeiten. Der Grund der unstrukturierten Arbeit war, dass bei der Realisierung schnell etwas Unerwartetes passieren kann. So kann eine Arbeit früher als geplant erledigt sein oder auch viel länger dauern. Deshalb hatte ich bei der Realisierung am meisten Abweichungen. Aber damit hatte ich gerechnet, da ich bis zum aktuellen Zeitpunkt nur bei der Semesterarbeit einen Zeitplan für ein Informatikprojekt erstellen musste. Damit ich beim Planen von Informatikprojekten besser werde, benötige ich mehr Praxiserfahrung. Da die Diplomarbeit während einer sehr kurzen Zeitspanne erarbeitet werden muss, konnte ich nicht viel Ersatzzeit einberechnen. Weil ich aber jeden Mittwoch frei hatte, da wir keine Schule mehr hatten, konnte ich an diesen Tagen immer mehr erarbeiten als geplant.

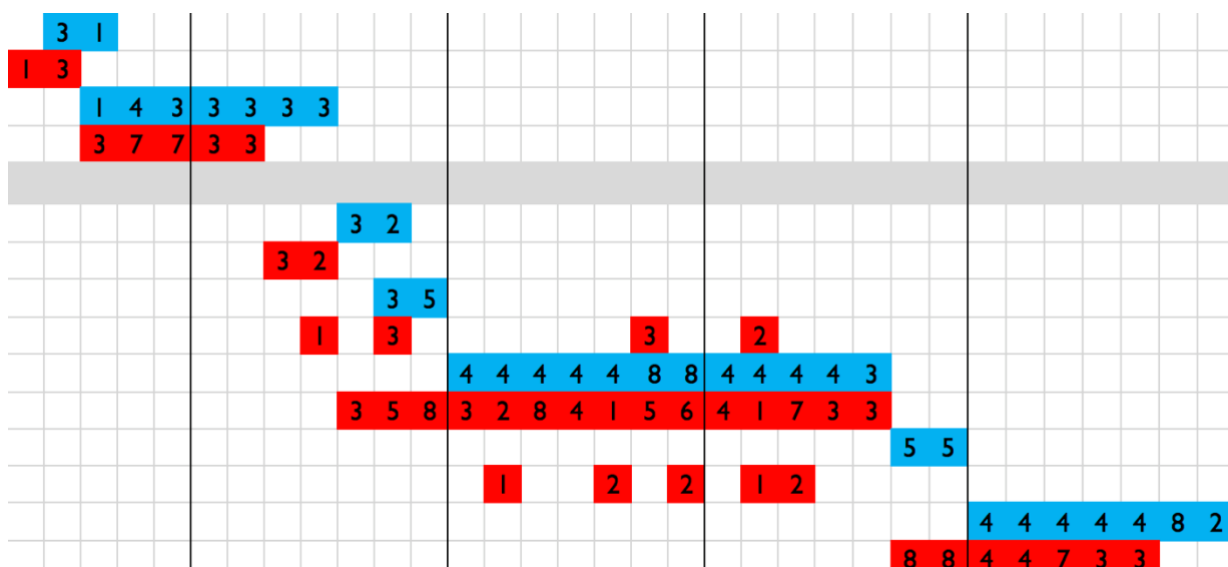


Abbildung 5: Teilausschnitt Projektplan

### 1.10.2 Risikoanalyse

Für die Risikoanalyse erstellte ich als erstes eine Tabelle im Excel. Darin führte ich alle Risiken auf, die mir bekannt waren. Zu den Risiken schrieb ich den Effekt des Risikos und die Massnahme, die ich ergreifen kann, damit das Risiko so gering wie möglich ist.

Risikoanalyse		Semesterarbeit System und Softwareengineering					
						Joshua Perolini	
Nr.	Risikobeschrieb	Ursache	Eintretens- wahrscheinlichkeit	Schadensausmass	Effekt	Massnahmen	Priorität
1	Absturz von Visual Studio	Fortschritt wurde nicht gespeichert.	gering	mittel	Durch einen Absturz von Visual Studio wird der aktuelle Fortschritt nicht gespeichert und alles muss nochmals erarbeitet werden.	In regelmässigen Abständen das Projekt speichern und Backup erstellen.	gering
2	Corona/Krankheit	Wenn ich Covid-19 positiv werde oder sonst krank werde, kann ich nicht mehr am Projekt arbeiten.	mittel	hoch	Wenn ich krank werde, kann ich nicht mehr am Projekt arbeiten. Dadurch kommt es zu einem Verzug und ich kann meinen Zeitplan nicht mehr einhalten.	BAG Massnahmen einhalten und gesund leben.	mittel
3	Ressourcenmangel	Da mein Wissen nur schulisch ist, sind meine Erfahrungen noch nicht so gross.	hoch	mittel	Um mein Wissen zu vergrössern, werde ich viel Zeit in Rechercharbeiten stecken müssen.	Ich muss bei der Planung mehr Zeit für die Realisierung einberechnen, da ich diese auch für die Recherche benötigen werde.	hoch
4	Vorgehensmodell Hermes 5.1	Bis zum Start der Diplomarbeit habe ich erst einmal mit Hermes 5.1 gearbeitet oder ein Projekt geplant.	hoch	gering	Ich muss bei der Planung Reservezeiten einberechnen. Dadurch kann es sein, dass ich mit meiner Planung nicht richtig liege.	Ich muss bei der Planung genügend Reservezeiten einberechnen.	mittel
5	Wegen meiner Lohnarbeit, habe ich nicht so viel Zeit für die Diplomarbeit, wie geplant.	Manchmal muss ich länger arbeiten. Dann kann es sein, dass ich weniger Zeit für die Diplomarbeit habe, als geplant.	mittel	mittel	Ich komme mit meinem Zeitplan in Verzug und es wäre deshalb möglich, dass ich die Diplomarbeit nicht erfolgreich abschliessen könnte.	Ich habe bei meiner Arbeitsstelle gemeldet, dass aktuell an der Diplomarbeit bin und es deshalb sein könnte, dass ich manchmal früher nach Hause muss.	mittel

Tabelle 3: Risikoanalyse

Mit dieser Tabelle konnte ich eine Risikomatrix erstellen. Diese hat den Vorteil, dass alle Risiken auf einen Blick ersichtlich sind. Auf der X-Achse liegt die Eintretenswahrscheinlichkeit und auf der Y-Achse das Schadensausmass. So kann ich alle Risiken eintragen und sehe sofort, was ich unbedingt beachten bzw. vermeiden sollte.

### Risikomatrix

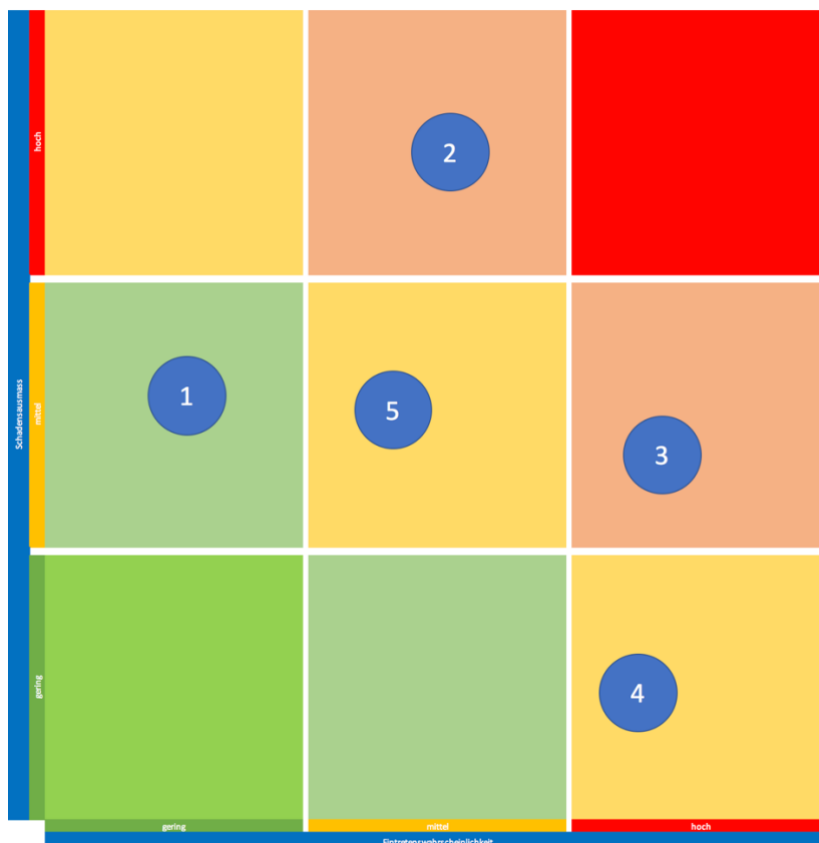


Abbildung 6: Risikomatrix

### 1.10.3 Qualitätsmanagement

Das Qualitätsmanagement hat die Aufgabe sicherzustellen, dass die entwickelten Softwaresysteme ihren Zweck erfüllen. Dies bedeutet, die Systeme sollten den Bedürfnissen ihrer Benutzer gerecht werden, effizient und zuverlässig funktionieren und innerhalb des Zeit- und Kostenrahmens fertiggestellt werden.

#### 1.10.3.1 Softwarequalität

Softwarequalität ist nicht nur eine Frage der korrekten Implementierung der Softwarefunktionalität, sondern hängt auch von den nichtfunktionalen Systemmerkmalen ab.

Damit die Softwarequalität nicht unter der Entwicklung leidet und die Software ihren angestrebten Zweck erfüllt, werden immer folgende Fragen beantwortet:

- Wurde die Software ordentlich getestet und wurde bewiesen, dass alle Anforderungen implementiert wurden?
- Ist die Leistung der Software beim normalen Einsatz akzeptabel?
- Ist die Software gut strukturiert und verständlich?
- Hat man sich im Entwicklungsprozess an Programmier- und Dokumentationsstandards gehalten?

Während der Diplomarbeit gab es ein Review für den Verifikationsprozess und den Validierungsprozess. Der Zweck der Review und Inspektionen ist es, die Softwarequalität zu verbessern und nicht die Leistung der einzelnen Personen zu bewerten. Im Review wird der Stand der Diplomarbeit angeschaut um wenn nötig Korrekturen anzubringen. Der Review fand am 11.10.2021 statt.

Besprechungspunkte:

- Standortbestimmung in Bezug auf die Zielsetzung
- Aufzeigen allfällig aufgetretener Probleme mit Lösungsvorschlägen
- Aufbau und Struktur der Dokumentation
- Fragen zur Vorbereitung der Präsentation und Onlinepublikation

#### 1.10.3.2 Softwarestandards

Für eine erfolgreiche Realisierung wurden Standards definiert. Standards müssen sich positiv in Form verbesserter Produktqualität niederschlagen.

Es wurden verschiedene Produktstandards definiert:

- Reviews werden dokumentiert
- Headerformat für die einzelnen Klassen
- MVC-Programmierstil wird angewendet
- Projektplan wird mit Hilfe von einem Format erstellt

## 1.10.4 Konfigurationsmanagement

Konfigurationsmanagement ist ein Managementprozess zur Herstellung und Erhaltung einer Übereinstimmung der Produktleistungen sowie der funktionalen und physischen Eigenschaften des Produktes mit den Anforderungen, dem Produktdesign und den operativen Informationen während des gesamten Produktlebenszyklus.

Ziel des Konfigurationsmanagements ist, den Grad der Erfüllung physischer und funktionaler Anforderungen an eine Konfigurationseinheit zu dokumentieren und diesbezüglich volle Transparenz herzustellen. Diese sollte zudem dazu führen, dass jede an einer Konfigurationseinheit beteiligte Partei die richtigen und zutreffenden Informationen verwendet

### 1.10.4.1 Änderungsmanagement

Nach jeder Änderung vom Programmcode wird im Header der jeweiligen Klasse die jeweilige Änderung oder die erstellte Methode aufgeschrieben.

```

/*****
/*
/* Schule:   Teko Bern
/* Projekt:  Diplomarbeit Kita-System
/*
/* Author:   Joshua Perolini
/*
/* Anpassungen
/*-----
/* Datum: 13.10.2021
/*
/* Login überprüfen und Überprüfen wurde ein Jahr und eine Kalenderwoche ausgewählt erstellen
/*
/*-----
/* Datum: 15.10.2021
/*
/* Bestellungen überprüfen, wurde schon eine Bestellung erstellt für die Kalenderwoche erstellen
/*
/*-----
/* Datum: 17.10.2021
/*
/* PKey_Jahr in Datenbank suchen , PKey_Kalenderwoche in Datenbank suchen , PKey_Tag in Datenbank suchen , PKey_Kita in Datenbank suchen und
/* Erstellte Essensbestellung abspeichern erstellen
/*
/*-----
/* Datum: 18.10.2021
/*
/* Wochenplan abspeichern , Überprüfen ist für jeden Wochentag eine Bestellung ausgeführt , Alle Bestellungen löschen mit Status neu und
/* Überprüfen ist schon eine Bestellung für den ausgewählten Tag abgesendet worden erstellen
/*
/*-----
/*****

```

Abbildung 7: Header

## 2 Konzept

### 2.1 Detailanforderungen an die neuen Systeme

Funktionale Anforderungen beschreiben gewünschte Funktionalitäten (was soll das System tun/können) eines Systems bzw. Produkts, dessen Daten oder Verhalten.

Nichtfunktionale Anforderungen sind Anforderungen an die "Qualität" in welcher die geforderte Funktionalität zu erbringen ist.

Funktionale Anforderungen unterscheiden sich von Produkt zu Produkt; das heisst das System Kita hat sehr wahrscheinlich andere funktionale Anforderungen als das System NeoTropical, obwohl beide Systeme zum gleichen Endprodukt gehören.

Nicht funktionale Anforderungen hingegen können für viele Produkte oder Systeme derselben Unternehmung vorgegeben/vorgeschlagen oder sogar genormt werden (z.B. einheitliches Erscheinungsbild und damit "ähnliche" Usability; Vorgaben von Betriebsmodi, welche von der Organisation unterstützt werden, etc.)

#### 2.1.1 Funktionale Anforderungen

##### 2.1.1.1 Kita

1. Jede Kita benötigt ihre eigenen Logindaten mit der die Kita sich in das Essenbestellungstool einloggen kann. Das ist wichtig, damit jede Kita ihre eigene Essensbestellung hat.
2. Kitas können für jede ausgewählte Kalenderwoche einmal die Essensbestellung ausfüllen und individuell gestalten. Nach dem Speichern der Essensbestellung kann diese nicht mehr angepasst werden.
3. In einer ausgewählten Kalenderwoche kann von Montag bis Freitag eine Essensbestellung erstellt werden. An den Wochenenden hat NeoTropical geschlossen.
4. Zu jeder Essensbestellung können Ernährungskriterien ausgewählt werden. Diese sind noch für Erwachsene und Kinder strukturiert. Die auswählbaren Ernährungskriterien sind:
  - Vegetarisch
  - Vegan
  - Glutenfrei
  - Laktosefrei
  - Ohne Eier
  - Erdnüsse
  - Besonderes
5. Sobald ein Essensplan gespeichert und abgeschickt worden ist, kann der Essensplan unter dem ausgewählten Jahr und der Kalenderwoche im Archiv angesehen werden.

##### 2.1.1.2 NeoTropical

1. NeoTropical hat ein eigenes Login, mit dem sie auf das Essenbestellungstool zugreifen können. NeoTropical braucht ein Login damit nur ausgewählte Personen Zugriff auf das Essenbestellungstool haben und nicht jede Person einen Essensplan erstellen kann.
2. NeoTropical kann ein Jahr und in diesem eine Kalenderwoche auswählen. In dieser kann sie einen Essensplan mit fünf Gerichten erstellen. Falls zur ausgewählten Kalenderwoche schon ein Essensplan erstellt worden ist, kann die Kalenderwoche nicht ausgewählt werden.
3. Sobald ein Essensplan erstellt worden ist, kann der Essensplan im Archiv angesehen werden. Falls im Archiv eine Kalenderwoche ausgewählt wird, für die noch kein Essensplan gespeichert worden ist, kann die Kalenderwoche nicht angezeigt werden.
4. NeoTropical muss jede Kita selbstständig registrieren, damit sie unkompliziert neue Kunden registrieren können und dabei die volle Kontrolle haben, wer ihre Kunden sind.

5. Am Ende einer Woche kann ein Endprotokoll erstellt werden. Darin sind alle notwendigen Daten damit die Köche-innen die Gerichte der nächsten Woche für die ausgewählten Kitas vorbereiten können.

### 2.1.2 Nicht-funktionale Anforderungen

1. In der Dokumentation befindet sich die gesamte Projektplanung mit dem entsprechenden Controlling.
2. In der Dokumentation befindet sich eine Risikoanalyse mit Massnahmen zur Risikoverkleinerung.
3. In der Dokumentation befindet sich ein Qualitätsmanagement.
4. In der Dokumentation befindet sich das Konfigurationsmanagement in einer sinnvollen Granularität.
5. Die Analyse und das Design werden mittels UML vorgenommen und in der Dokumentation beschrieben.

### 2.1.3 Organisatorische Anforderungen

1. Die Diplomarbeit wird in einem Textverarbeitungsprogramm im Format A4, einseitig beschrieben, fortlaufend nummeriert
2. Die Dokumentation wird in elektronischer Form abgegeben.
3. Die Mindestschrift in der Dokumentation ist elf.
4. Alle Quellcodes werden in elektronischer Form abgegeben.
5. Ca. 150 – 250 Stunden Arbeitsaufwand.

## 2.2 Kontextdiagramm

Kontextdiagramme dienen zur Abgrenzung von dem entwickelnden System, jedoch dienen sie nicht zur Darstellung von Details. Im Kontextdiagramm sieht man alle Prozesse und Um Systeme eines Systems. Das Kontextdiagramm ist kein echtes UML-Diagramm.

Beim Erstellen der Kontextdiagramme ist mir aufgefallen, dass das Kita Essenbestellungstool aus zwei Systemen entsteht. Ein System ist für die Kita und ein System ist für NeoTropical selbst.

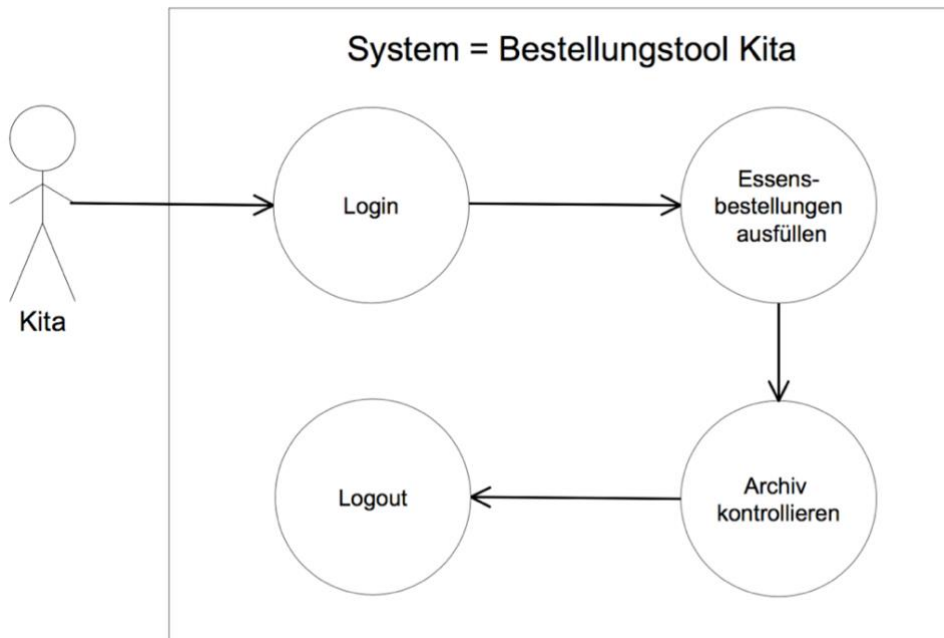


Abbildung 8: Kontextdiagramm Kita

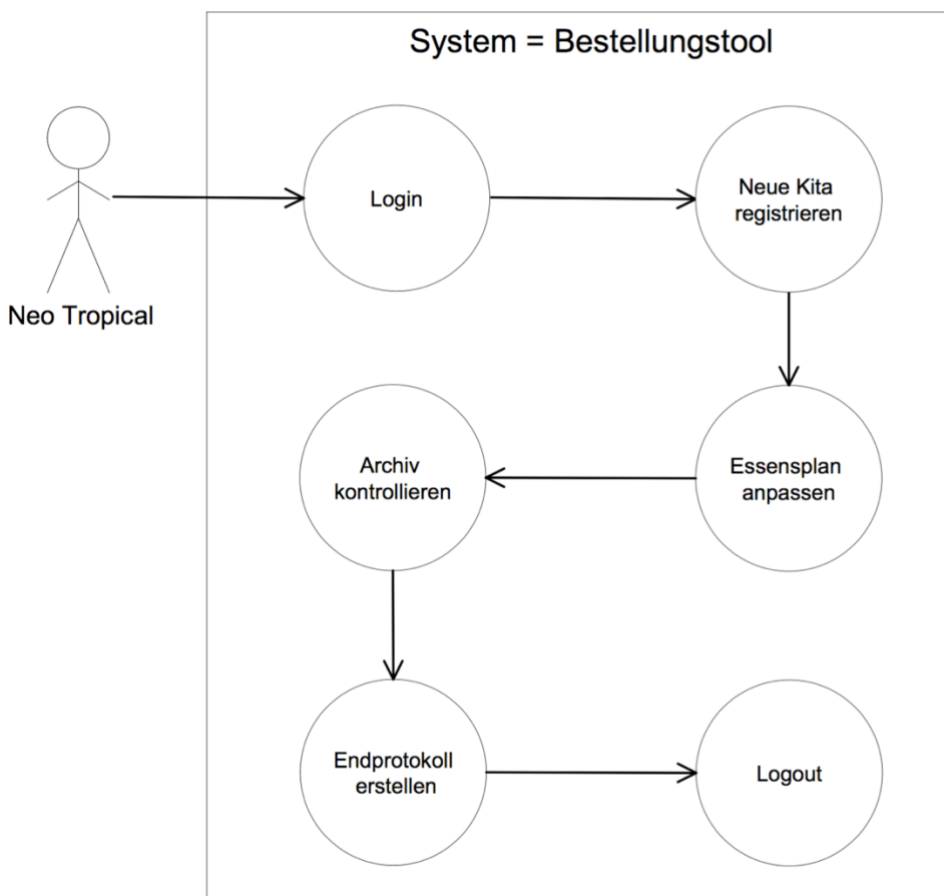


Abbildung 9: Kontextdiagramm NeoTropical

### 2.3 Use-Case

Use-Case ermöglicht, Interaktionen zwischen Benutzer und dem Essensbestellungstool mittels grafischer Modelle und strukturiertem Text zu beschreiben. Use-Case werden immer in der Unified Modeling Language (UML) erstellt.

Ich erstelle zu jedem nötigen Prozess, welcher im Kontextdiagramm ermittelt worden ist, ein Use-Case und zum Use-Case eine passende Tabelle, welche diese tabellarisch beschreibt. Am Startaktor erkennt man zu welchem System das Use-Case gehört.

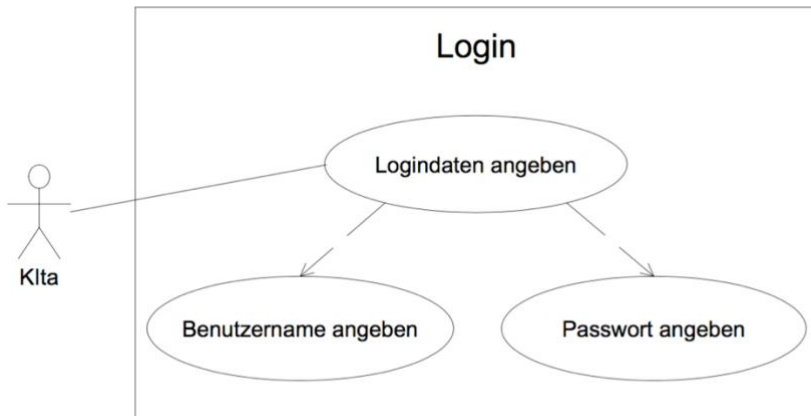


Abbildung 10: Use Case Kita Login

Name des Use Case	Login
System	Kita
Kurzbeschreibung	Kitas können sich einloggen
Autor	Joshua Perolini
Auslöser	Eine Kita möchte sich in das Kita Essenbestellungstool einloggen
Ergebnis	Kita hat sich eingeloggt
Vorbedingungen	Kita hat ein Konto bei NeoTropical
Nachbedingungen	<ul style="list-style-type: none"> <li>• Benutzername ist richtig</li> <li>• Passwort ist richtig</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Benutzername und Passwort angeben</li> <li>• Benutzername und Passwort überprüfen und einloggen</li> </ul>
Alternativer Ablauf	-

Tabelle 4: Use Case Kita Login

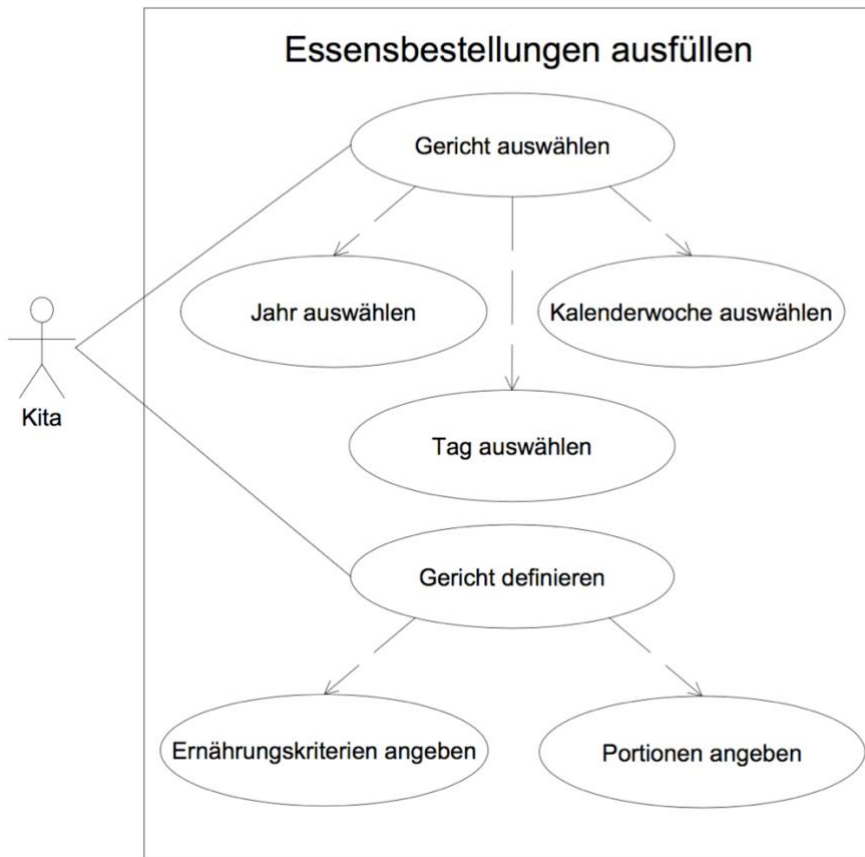


Abbildung 11: Use Case Kita Essensbestellungen ausfüllen

Name des Use Case	Essensbestellungen ausfüllen
System	Kita
Kurzbeschreibung	Eine Kalenderwoche auswählen, für die ein Essensplan erstellt worden ist und zu jedem Tag die Gerichte definieren
Autor	Joshua Perolini
Auslöser	Eine Kita hat sich eingeloggt und möchte eine Essensbestellung erstellen
Ergebnis	Eine Kita hat für eine oder zwei Wochen eine Essensbestellung eingereicht
Vorbedingungen	Kita hat sich eingeloggt (Benutzername und Passwort war richtig)
Nachbedingungen	<ul style="list-style-type: none"> <li>• Eine Kalenderwoche wurde ausgewählt</li> <li>• Zu 5 oder 10 Tagen wurde eine Essensbestellung abgespeichert</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Jahr auswählen</li> <li>• Kalenderwoche auswählen</li> <li>• Tag auswählen</li> <li>• Essensbestellungen erstellen zu allen Tagen der Kalenderwoche</li> <li>• Essensbestellungen der Kalenderwoche abspeichern</li> </ul>
Alternativer Ablauf	<ul style="list-style-type: none"> <li>• Jahr auswählen</li> <li>• Kalenderwoche auswählen</li> <li>• Tag auswählen</li> <li>• Essensbestellungen erstellen zu allen Tagen der Kalenderwoche und der darauffolgenden Kalenderwoche</li> <li>• Essensbestellungen der Kalenderwochen abspeichern</li> </ul>

Tabelle 5: Use Case Kita Essensbestellungen ausfüllen

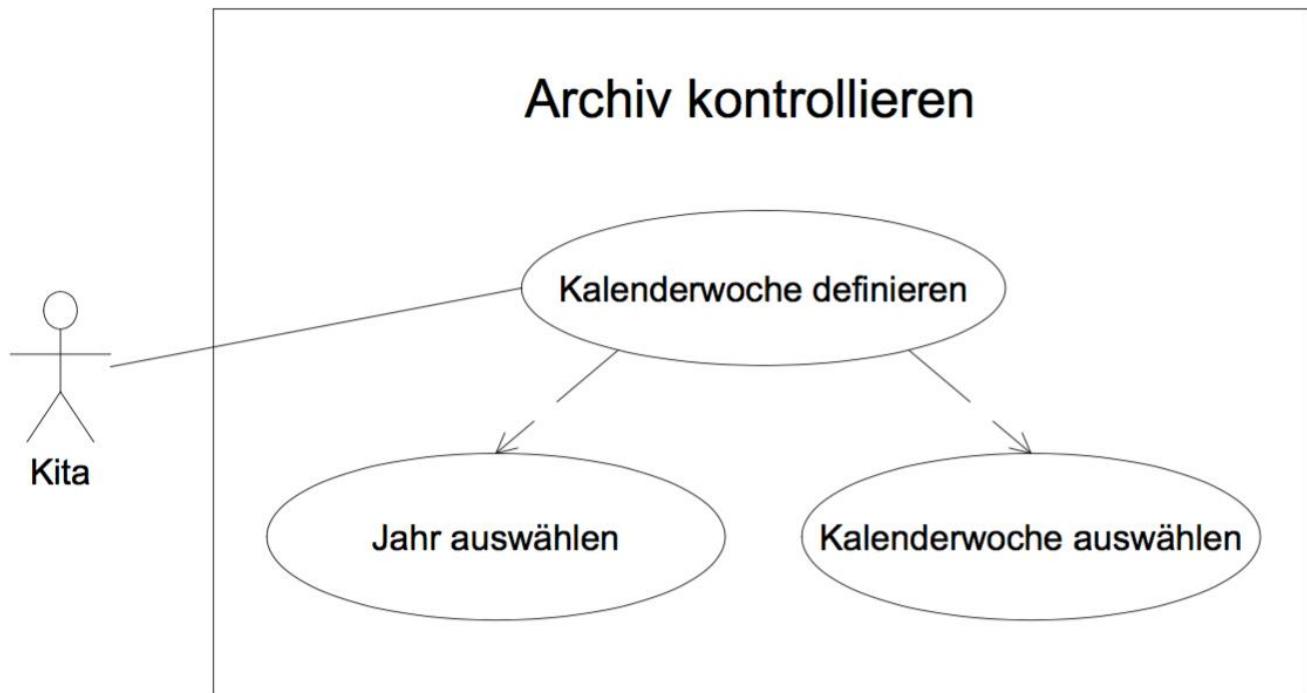


Abbildung 12: Use Case Kita Archiv kontrollieren

Name des Use Case	Archiv kontrollieren
System	Kita
Kurzbeschreibung	Schon erstellte Essensbestellungen ansehen
Autor	Joshua Perolini
Auslöser	Eine Kita hat sich eingeloggt und möchte die schon erstellten Essensbestellungen ansehen
Ergebnis	Essensbestellung anzeigen
Vorbedingungen	<ul style="list-style-type: none"> <li>• Kita hat sich eingeloggt (Benutzername und Passwort war richtig)</li> <li>• Essensbestellungen wurden schon erstellt</li> </ul>
Nachbedingungen	<ul style="list-style-type: none"> <li>• Eine Kalenderwoche wurde ausgewählt</li> <li>• Essensbestellung anzeigen</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Jahr auswählen</li> <li>• Kalenderwoche auswählen</li> <li>• Tag auswählen</li> <li>• Essensbestellung anzeigen</li> </ul>
Alternativer Ablauf	-

Tabelle 6: Use Case Kita Archiv kontrollieren

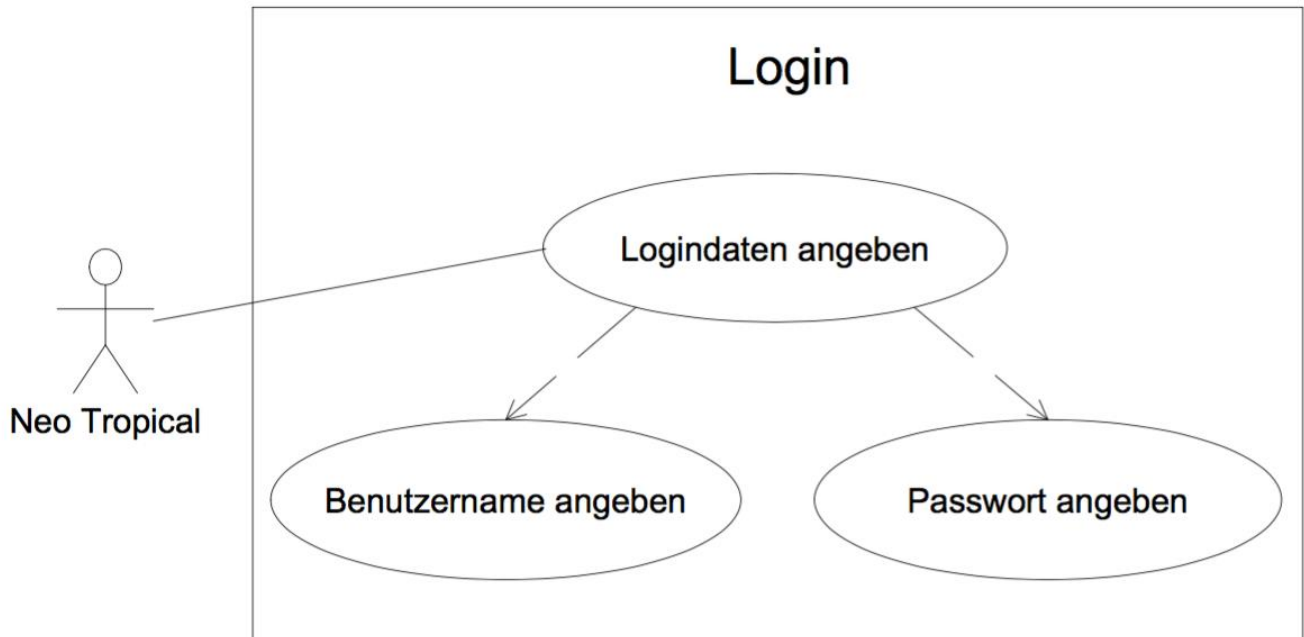


Abbildung 13: Use Case NeoTropical Login

Name des Use Case	Login
System	NeoTropical
Kurzbeschreibung	Ein NeoTropical Mitarbeiter kann sich einloggen
Autor	Joshua Perolini
Auslöser	Ein NeoTropical Mitarbeiter möchte sich in das Kita Essenbestellungstool einloggen
Ergebnis	Ein NeoTropical Mitarbeiter hat sich eingeloggt
Vorbedingungen	-
Nachbedingungen	<ul style="list-style-type: none"> <li>• Benutzername ist richtig</li> <li>• Passwort ist richtig</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Benutzername und Passwort angeben</li> <li>• Benutzername und Passwort überprüfen und einloggen</li> </ul>
Alternativer Ablauf	-

Tabelle 7: Use Case NeoTropical Login

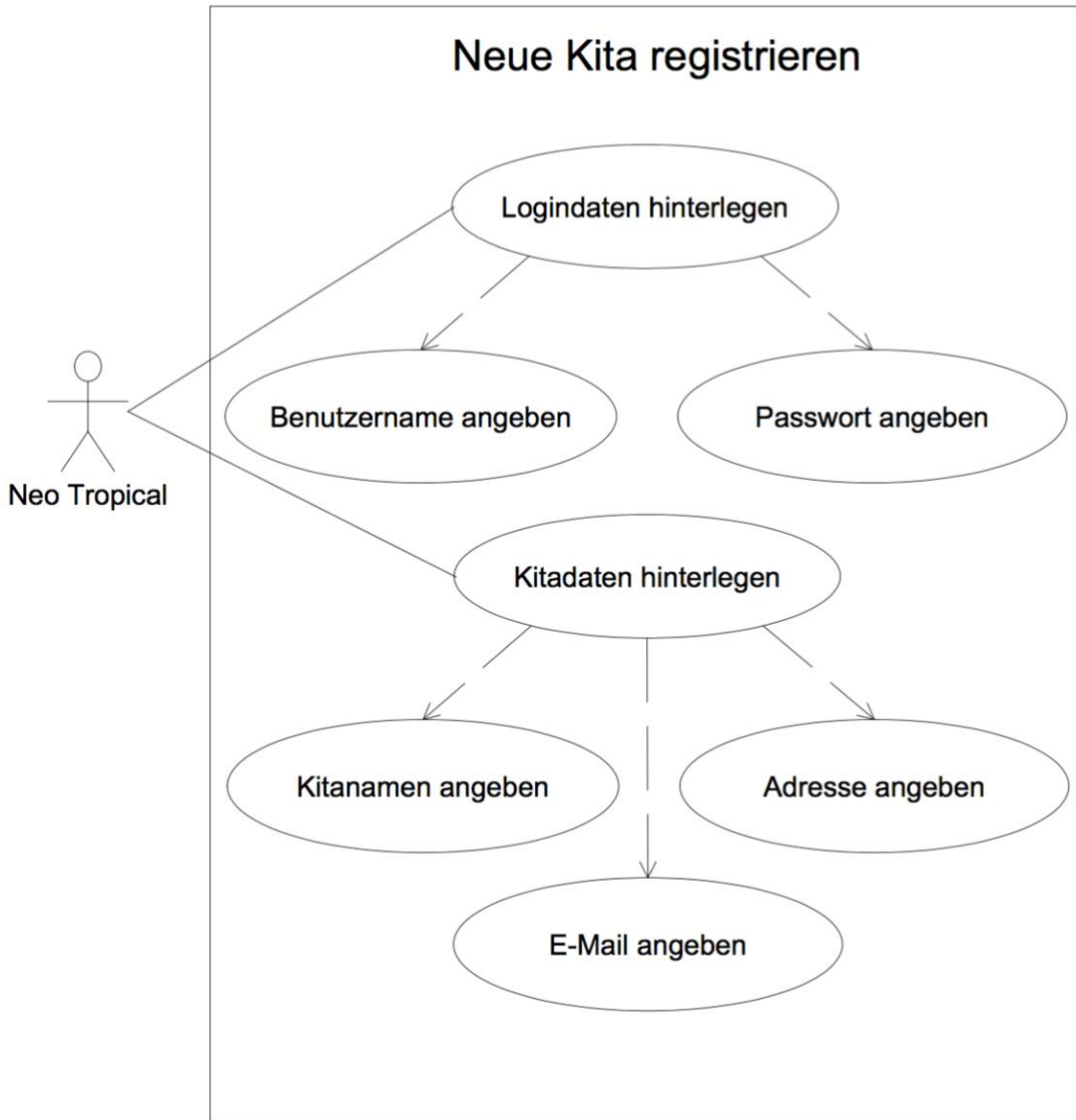


Abbildung 14: Use Case NeoTropical Neue Kita registrieren

Name des Use Case	Neue Kita registrieren
System	NeoTropical
Kurzbeschreibung	Ein NeoTropical Mitarbeiter registriert eine neue Kita
Autor	Joshua Perolini
Auslöser	Eine Kita, die von NeoTropical beliefert wird, möchte das Essensbestellungstool benutzen
Ergebnis	Eine neue Kita wurde registriert
Vorbedingungen	Ein NeoTropical Mitarbeiter hat sich eingeloggt
Nachbedingungen	<ul style="list-style-type: none"> <li>• Logindaten in der Datenbank gespeichert</li> <li>• Kitadaten in der Datenbank gespeichert</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Kitadaten angeben</li> <li>• Logindaten angeben</li> <li>• Kita- und Logindaten überprüfen</li> <li>• Kita- und Logindaten abspeichern</li> </ul>
Alternativer Ablauf	-

Tabelle 8: Use Case NeoTropical Neue Kita registrieren

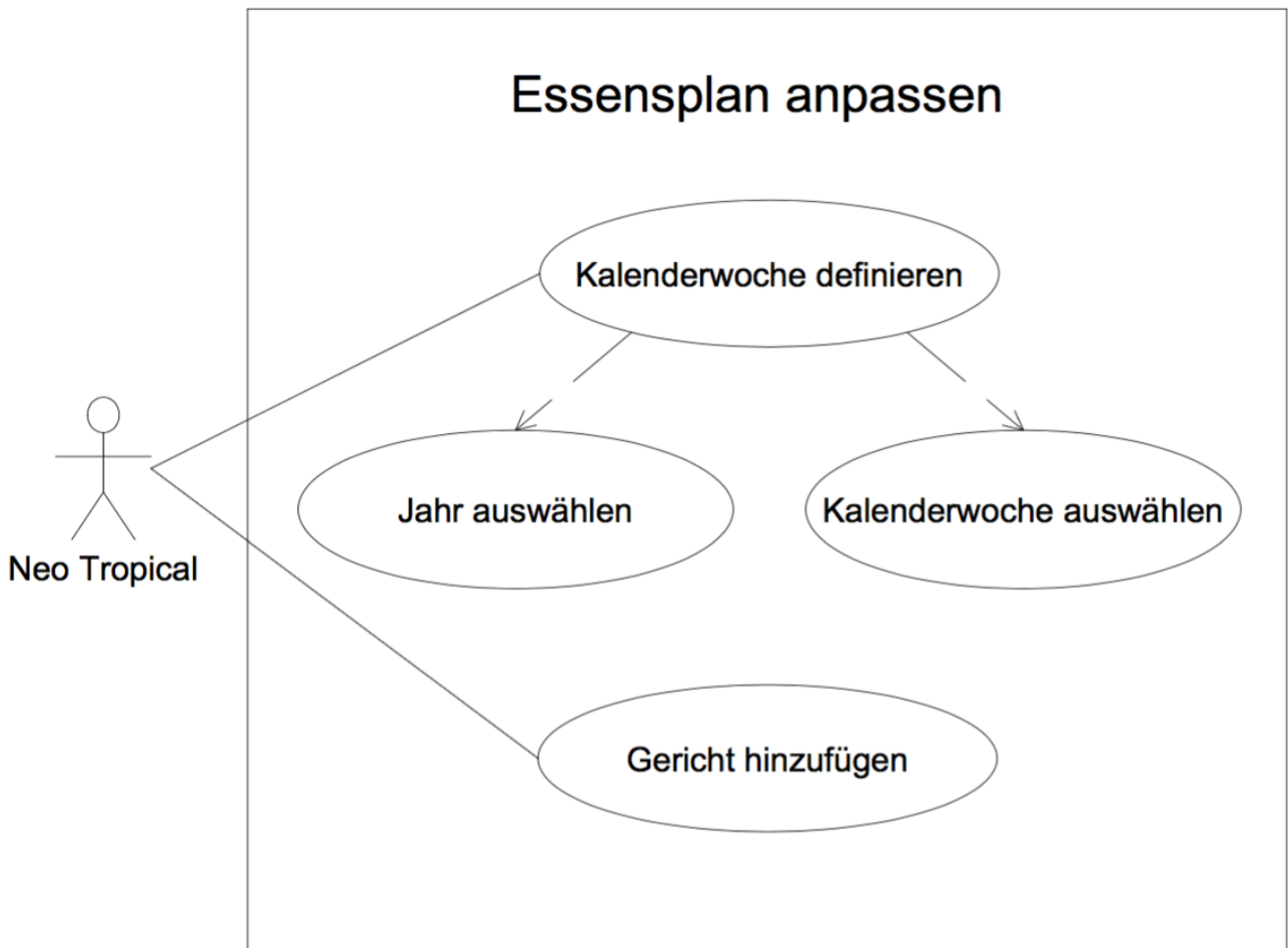


Abbildung 15: Use Case NeoTropical Essensplan anpassen

Name des Use Case	Essensplan anpassen
System	NeoTropical
Kurzbeschreibung	Ein NeoTropical Mitarbeiter passt den Essensplan an
Autor	Joshua Perolini
Auslöser	Ein NeoTropical Mitarbeiter möchte den aktuellen Essensplan anpassen
Ergebnis	Für eine Woche wurde ein Essensplan erstellt
Vorbedingungen	Ein NeoTropical Mitarbeiter hat sich eingeloggt
Nachbedingungen	<ul style="list-style-type: none"> <li>• Jahr und Kalenderwoche wurden ausgewählt</li> <li>• Ein Gericht wurde zu jedem Wochentag hinzugefügt</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Jahr auswählen</li> <li>• Kalenderwoche auswählen</li> <li>• Zu jedem Wochentag ein Gericht hinzufügen</li> <li>• Essensplan speichern</li> </ul>
Alternativer Ablauf	-

Tabelle 9: Use Case NeoTropical Essensplan anpassen

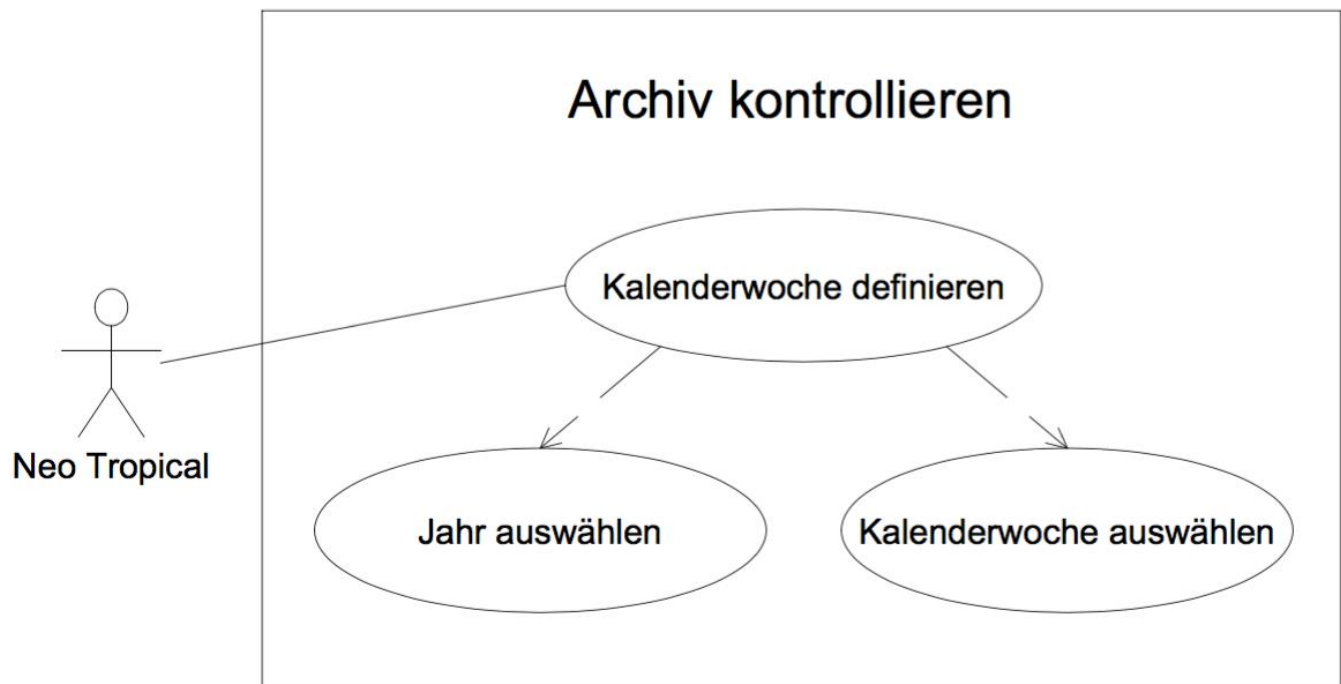


Abbildung 16: Use Case NeoTropical Archiv kontrollieren

Name des Use Case	Archiv kontrollieren
System	NeoTropical
Kurzbeschreibung	Die alten Essenspläne ansehen
Autor	Joshua Perolini
Auslöser	Ein NeoTropical Mitarbeiter möchte die alten Essenspläne ansehen
Ergebnis	Anzeigen eines Gerichts
Vorbedingungen	<ul style="list-style-type: none"> <li>• Ein NeoTropical Mitarbeiter hat sich eingeloggt</li> <li>• Es wurden schon Essenspläne erstellt</li> </ul>
Nachbedingungen	<ul style="list-style-type: none"> <li>• Jahr und Kalenderwoche wurden ausgewählt</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Jahr auswählen</li> <li>• Kalenderwoche auswählen</li> </ul>
Alternativer Ablauf	-

Tabelle 10: Use Case NeoTropical Archiv kontrollieren

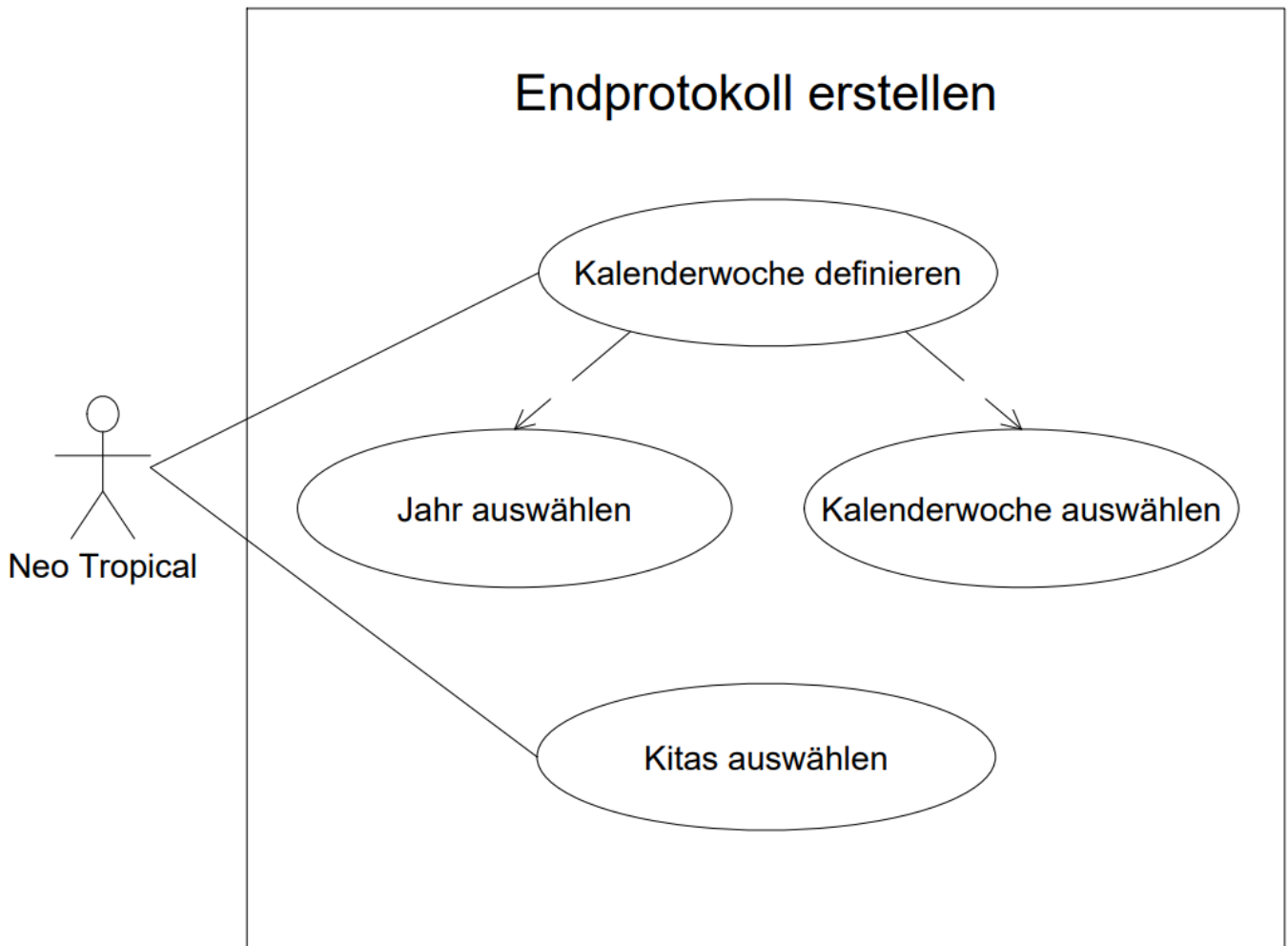


Abbildung 17: Use Case NeoTropical Endprotokoll erstellen

Name des Use Case	Endprotokoll erstellen
System	NeoTropical
Kurzbeschreibung	Endprotokoll erstellen
Autor	Joshua Perolini
Auslöser	Ein NeoTropical Mitarbeiter möchte für die nächste Woche ein Endprotokoll erstellen
Ergebnis	Endprotokoll erstellt
Vorbedingungen	<ul style="list-style-type: none"> <li>• Ein NeoTropical Mitarbeiter hat sich eingeloggt</li> <li>• Es wurden schon Essenspläne erstellt</li> <li>• Kits haben ihre Bestellung schon ausgefüllt und abgeschickt</li> </ul>
Nachbedingungen	<ul style="list-style-type: none"> <li>• Jahr und Kalenderwoche wurden ausgewählt</li> <li>• Kits wurden ausgewählt</li> </ul>
Typischer Ablauf	<ul style="list-style-type: none"> <li>• Jahr auswählen</li> <li>• Kalenderwoche auswählen</li> <li>• Kits auswählen, für die ein Endprotokoll erstellt werden soll</li> <li>• Endprotokoll erstellen</li> </ul>
Alternativer Ablauf	-

Tabelle 11: Use Case NeoTropical Endprotokoll erstellen

## 2.4 Sequenzdiagramme

Sequenzdiagramme modellieren die Interaktionen zwischen Akteuren und Objekten in einem System sowie der Interaktion zwischen den Objekten selbst. Das dient dazu, die innere Struktur des Systems und eine zeitliche Logik aufzuzeigen. Sequenzdiagramme zeigen die Abfolge der Interaktionen, die während eines bestimmten Anwendungsfalls beziehungsweise der Instanz eines Anwendungsfalls stattfinden.

Das erste Sequenzdiagramm, das ich erstellt habe, zeigt den Use-Case Essensbestellungen ausfüllen in dem System Kita. Darin werden die einzelnen Objekte definiert, die die Kita benötigt, wenn sie eine Essensbestellungen ausfüllt. Weiter zeigt es, wo es eine Schleife brauchen wird.

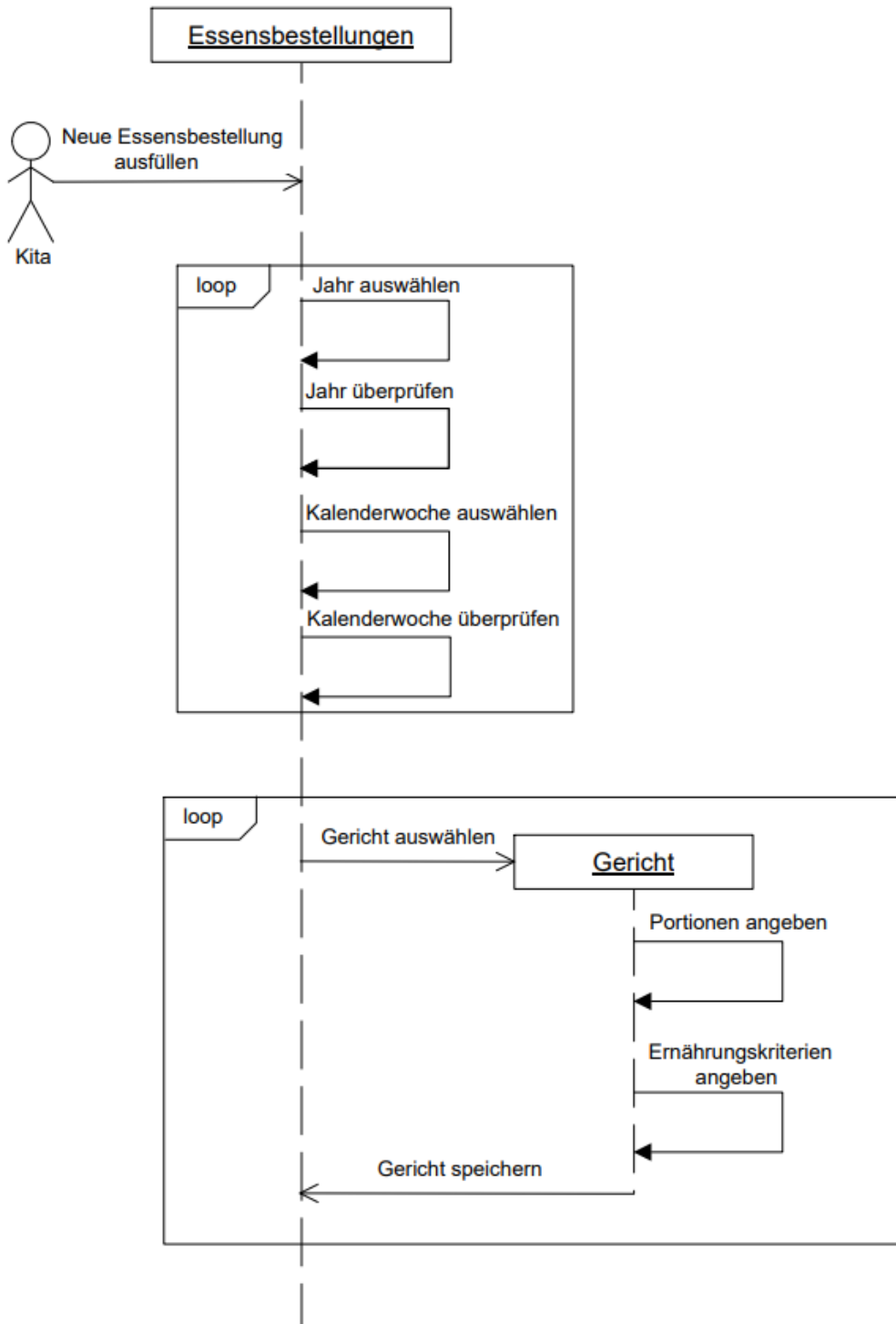


Abbildung 18: Sequenzdiagramm Kita Essensbestellungen ausfüllen

Als nächstes erstellte ich ein Sequenzdiagramm für den Use Case Archiv kontrollieren, das auch im System Kita liegt. Darin werden die einzelnen Schritte, wie man zur Ansicht der Details eines schon bestellten Gerichts kommt, graphisch dargestellt.

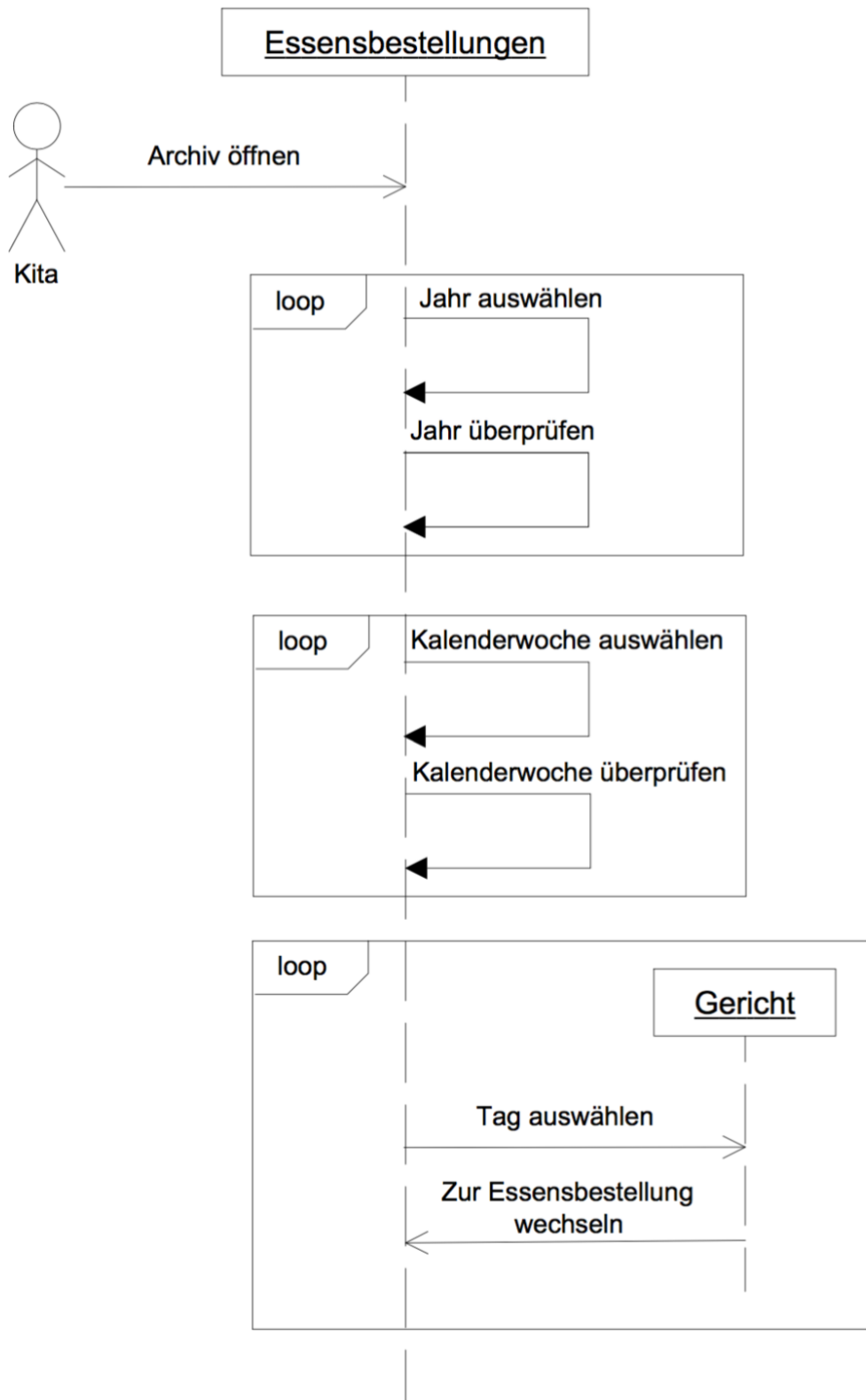


Abbildung 19: Sequenzdiagramm Kita Archiv kontrollieren

Danach erstellte ich das erste Sequenzdiagramm für das System NeoTropical, für den Use Case Essensplan anpassen. NeoTropical kann jetzt den Essensplan mit neuen Gerichten erweitern und diese einer Kalenderwoche zuteilen.

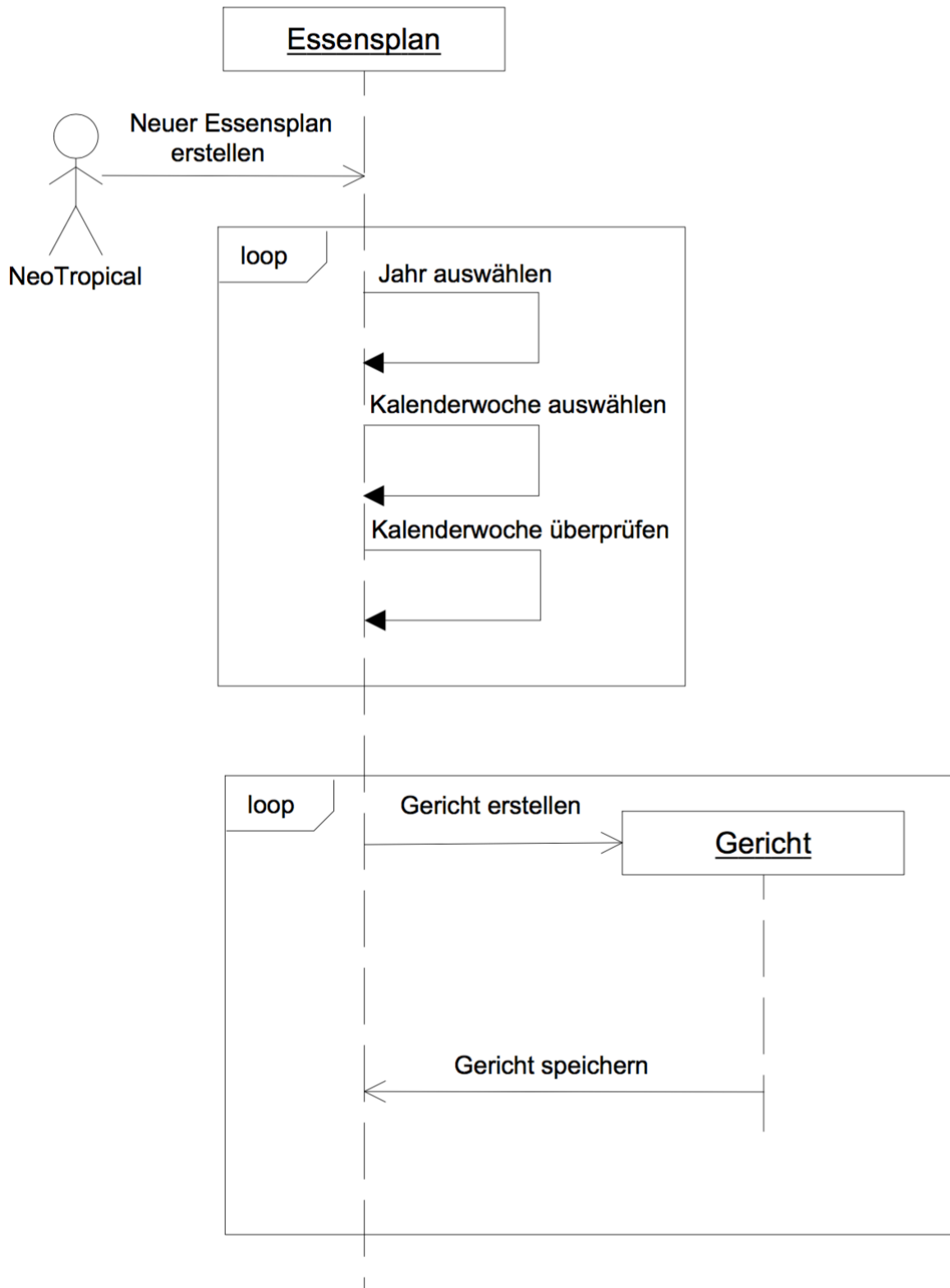


Abbildung 20: Sequenzdiagramm NeoTropical Essensplan anpassen

Als nächstes erstellte ich ein Sequenzdiagramm für den Use Case Neue Kita registrieren. Der Use Case Neue Kita registrieren liegt im System NeoTropical. NeoTropical kann in diesem Abschnitt eine neue Kita registrieren.

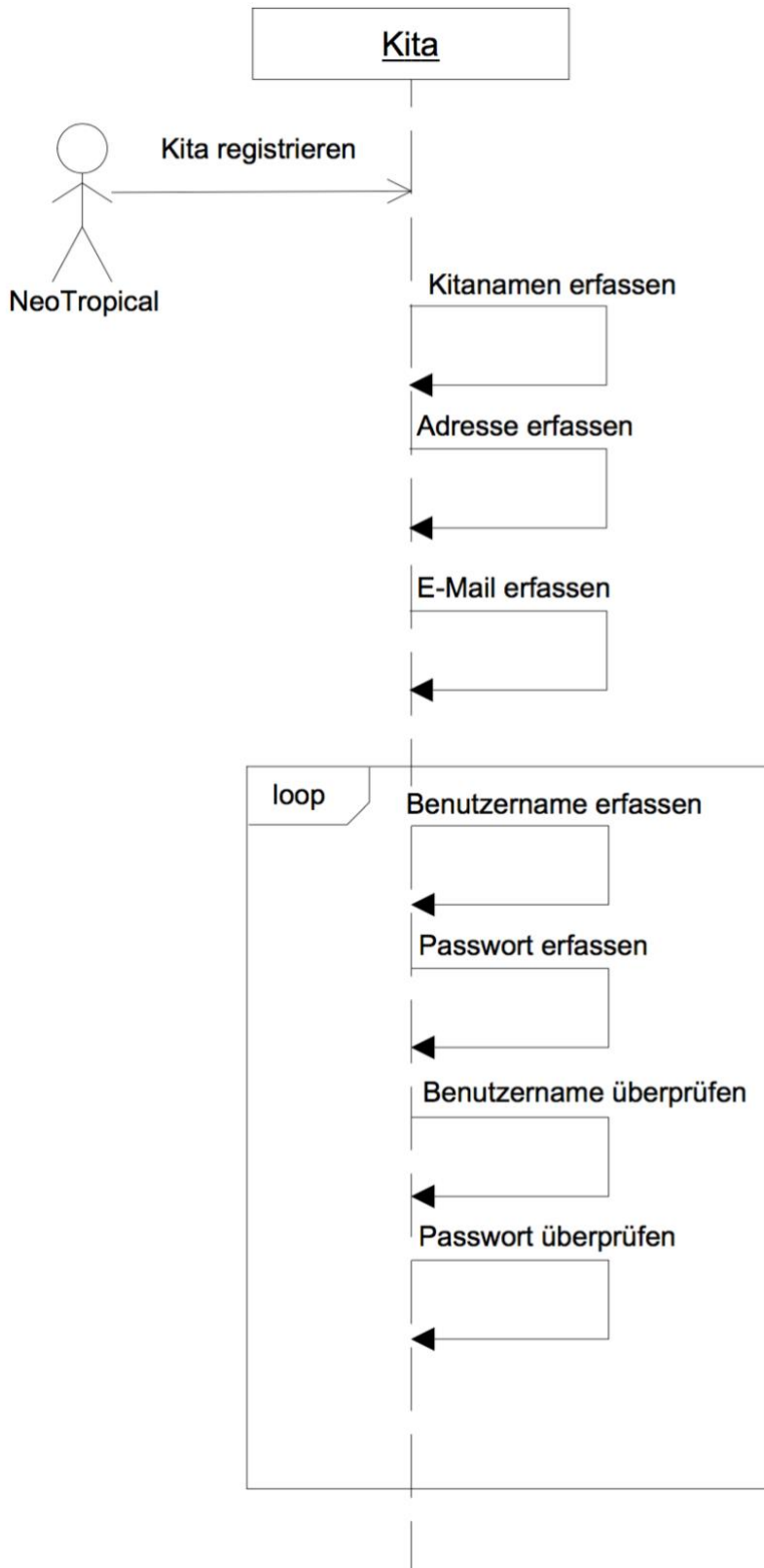


Abbildung 21: Sequenzdiagramm NeoTropical Neue Kita registrieren

Der zweitletzte Use Case, der noch kein Sequenzdiagramm hat, ist für den Use Case Archiv kontrollieren. Darin kann man schon erstellte Essenspläne ansehen. Dabei wird graphisch verdeutlicht, wie die Objekte bei der Gerichtauswahl miteinander kommunizieren.

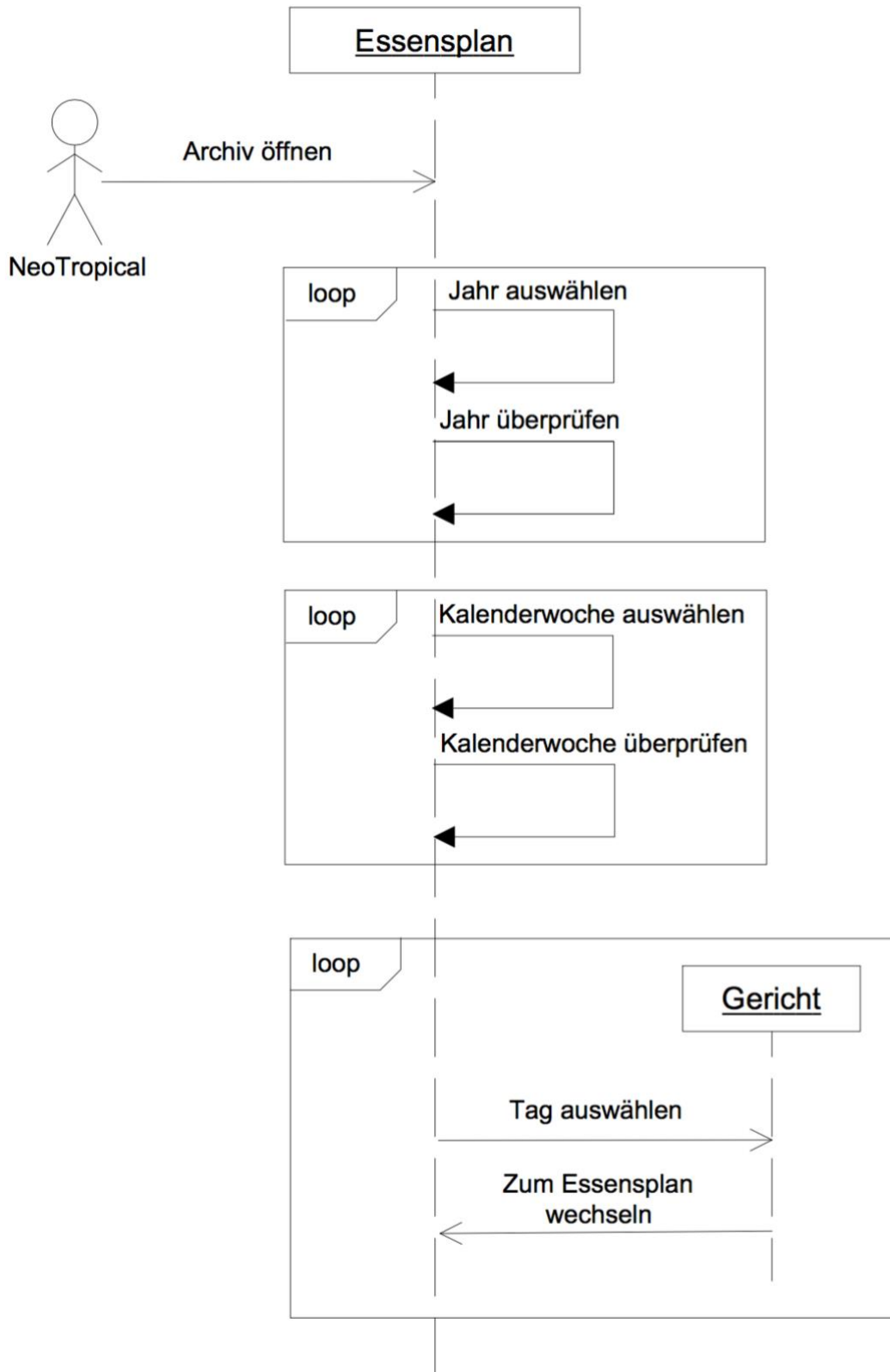


Abbildung 22: Sequenzdiagramm NeoTropical Archiv kontrollieren

Der letzte Use Case, der noch kein Sequenzdiagramm hat, ist für den Use Case Endprotokoll erstellen. NeoTropical kann am Ende einer Woche ein Endprotokoll erstellen. Bei der Erstellung kann NeoTropical die Kalenderwoche auswählen und bestimmen, für welche Kitas das Endprotokoll erstellt wird.

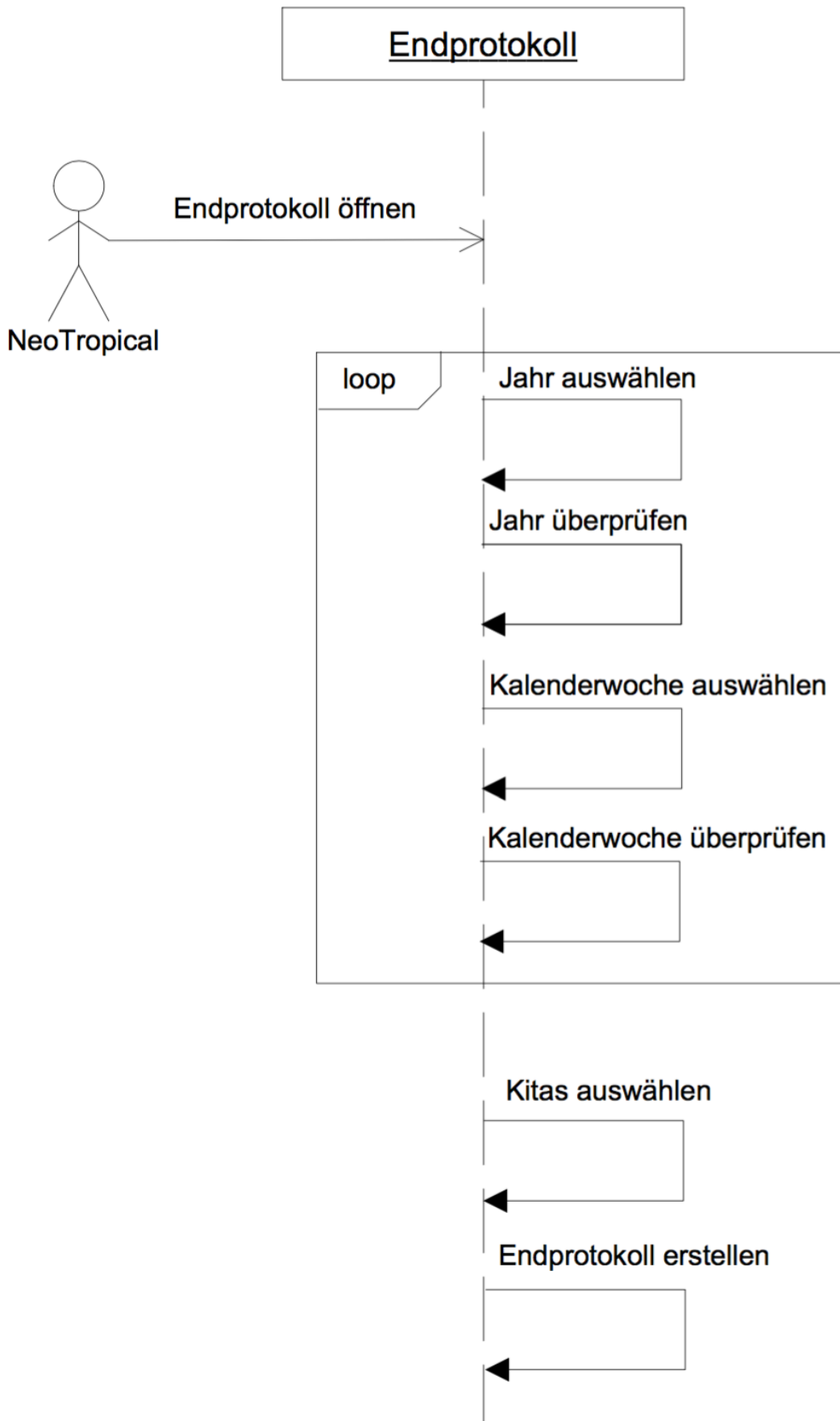


Abbildung 23: NeoTropical Endprotokoll erstellen

## 2.5 Zustandsdiagramme

Zustandsdiagramme zeigen Systemzustände und Ergebnisse, die Übergänge von einem Zustand zum nächsten auslösen. Zustandsdiagramme sind nicht für den Datenfluss innerhalb des Systems da, sie können aber zusätzliche Informationen, über die in jedem Zustand auszuführenden Berechnungen enthalten.

Weiter erstellte ich eine Tabelle, in der alle aufgetretenen Zustände mit der zugehörigen Beschreibung angezeigt werden.

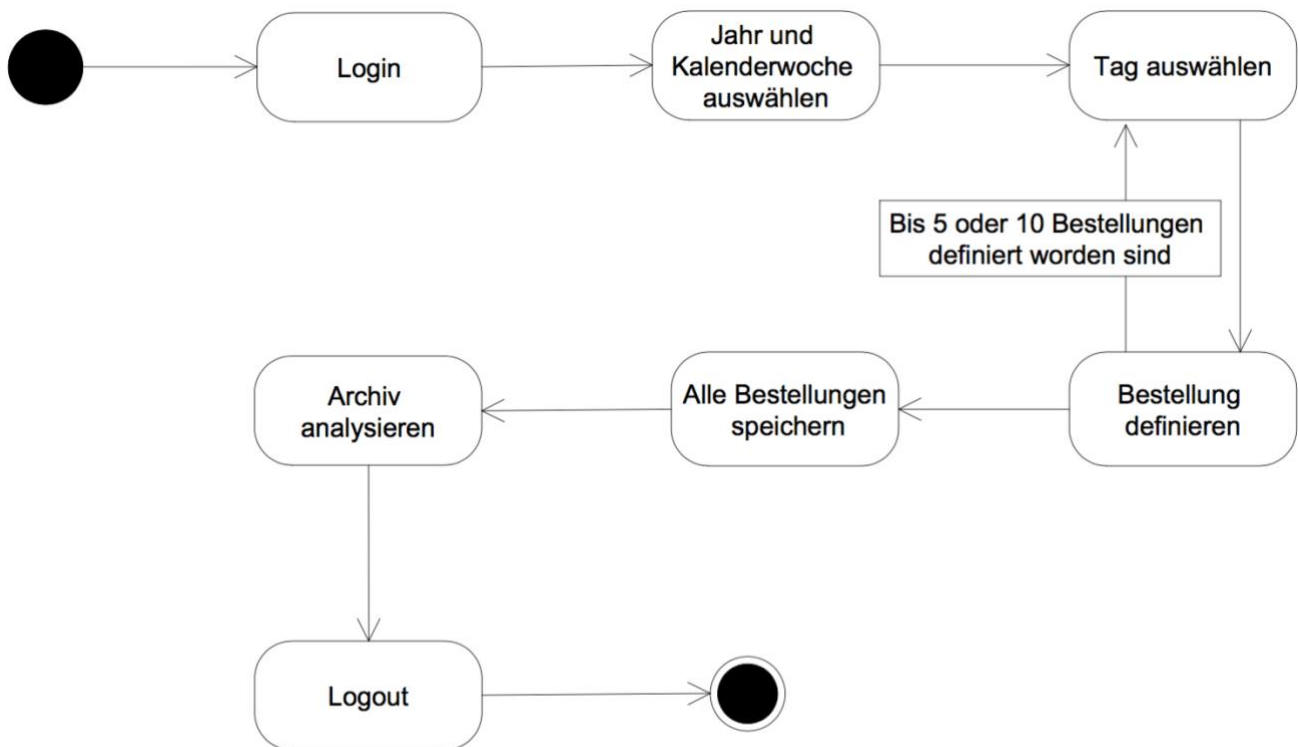


Abbildung 24: Kita Zustandsdiagramm

Zustand	Beschreibung
Login	Das System wartet auf eine erfolgreiche Eingabe des Benutzernamens und Passworts.
Jahr und Kalenderwoche auswählen	Ein Jahr auswählen und in diesem eine Kalenderwoche bestimmen, die einen Essensplan hat.
Tag auswählen	Einen Wochentag wählen, für den eine Bestellung aufgegeben werden möchte.
Bestellung definieren	Für die Erwachsenen und die Kinder die jeweiligen Ernährungskriterien auswählen.
Alle Bestellungen speichern	Falls 5 oder 10 Bestellungen aufgegeben sind, können die Bestellungen gespeichert werden.
Archiv analysieren	Ein Jahr auswählen und in diesem eine Kalenderwoche bestimmen, für die die Bestellungen angesehen werden möchten.
Logout	Das System löscht alle Daten, die nicht gespeichert wurden.

Tabelle 12: Kita Zustandsdiagrammanalyse

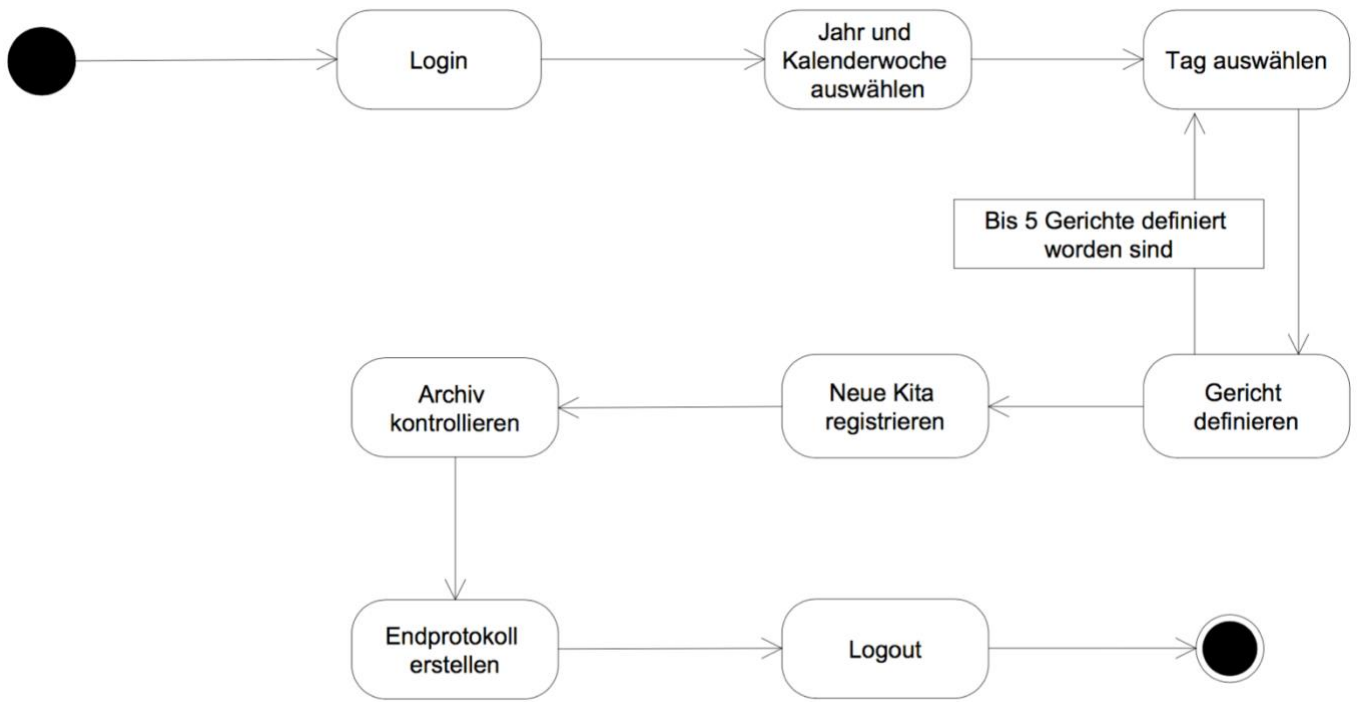


Abbildung 25: NeoTropical Zustandsdiagramm

Zustand	Beschreibung
Login	Das System wartet auf eine erfolgreiche Eingabe des Benutzernamens und Passworts.
Jahr und Kalenderwoche auswählen	Ein Jahr auswählen und in diesem eine Kalenderwoche definieren.
Tag auswählen	Ein Wochentag bestimmen, für den ein Gericht festgelegt werden möchte.
Gericht definieren	Zum ausgewählten Wochentag ein Gericht erstellen.
Neue Kita registrieren	Eine neue Kita kann registriert werden, indem die Logindaten und Kitadaten angegeben werden.
Archiv kontrollieren	Ein Jahr auswählen und in diesem eine Kalenderwoche bestimmen, für die die Bestellungen angesehen werden.
Endprotokoll erstellen	Ein Jahr auswählen und in diesem eine Kalenderwoche bestimmen, dann die Kitas auswählen, für die ein Endprotokoll erstellt wird.
Logout	Das System löscht alle Daten, die nicht gespeichert wurden.

Tabelle 13: NeoTropical Zustandsdiagrammanalyse

## 2.6 Klassendiagramm

Mit Hilfe der zuvor erstellten Sequenzdiagramme wird ein Klassendiagramm erstellt. Klassendiagramme sind Designndiagramme und wesentlich für die Umsetzung. Sie dienen dazu, eine optimale Anordnung der erkannten Klassen respektive der definierten Methoden und Eigenschaften zu ermöglichen.



Abbildung 26: Kita Klassendiagramm

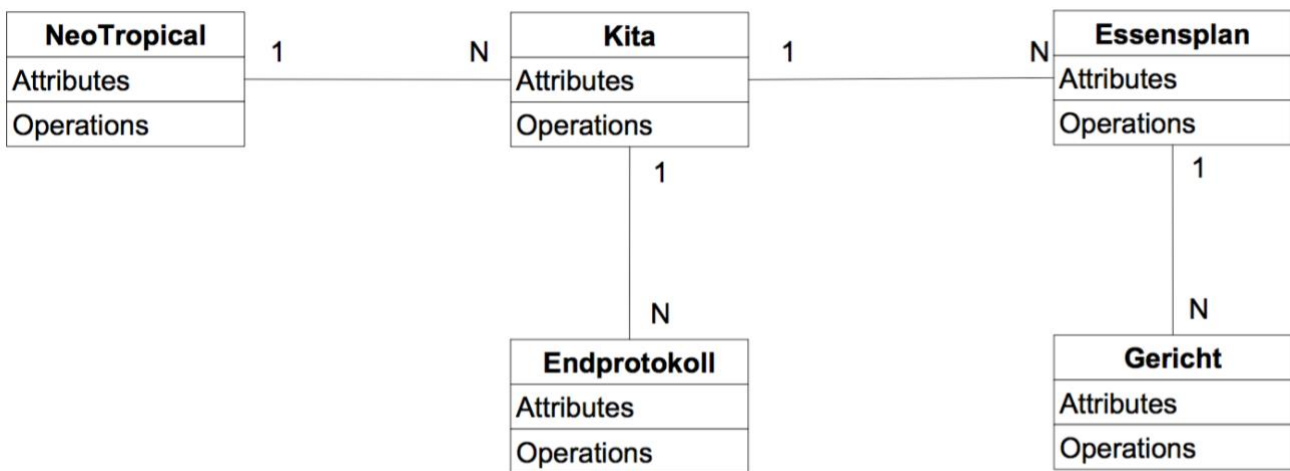


Abbildung 27: NeoTropical Klassendiagramm

## 2.7 Modellierung der Datenbank

### 2.7.1 Leistungsdiagramm

Für die Modellierung der Datenbank wird als erstes ein Leistungsdiagramm erstellt. Das Leistungsdiagramm enthält alle Prozesse des Essenbestellungstools. Ein Prozess erzeugt oder verbraucht eine Leistung.

Ich benötige für das Essenbestellungstool zwei Leistungsdiagramme, da das Essenbestellungstool zwei unterschiedliche Prozesse hat.

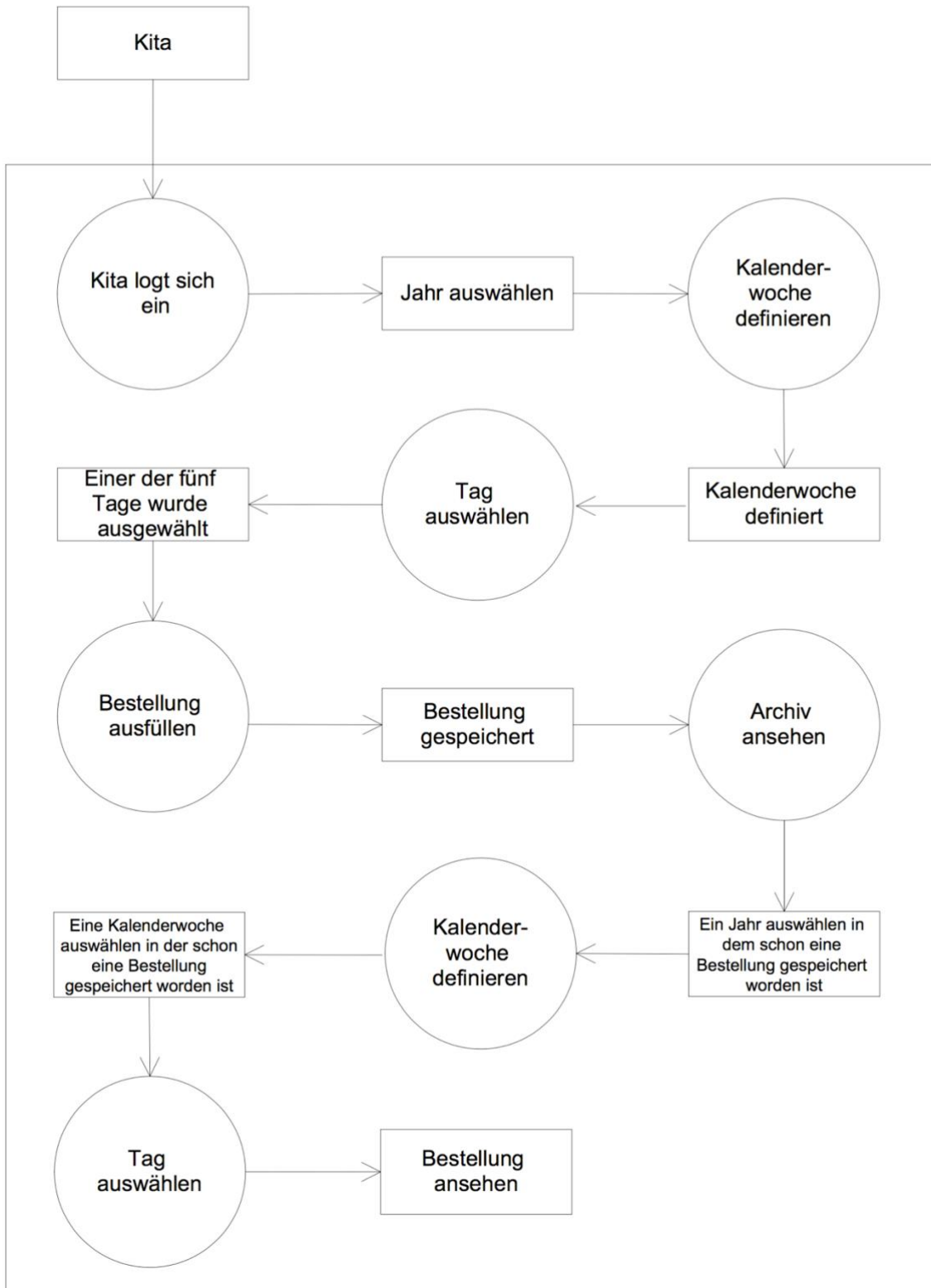


Abbildung 28: Leistungsdiagramm Kita

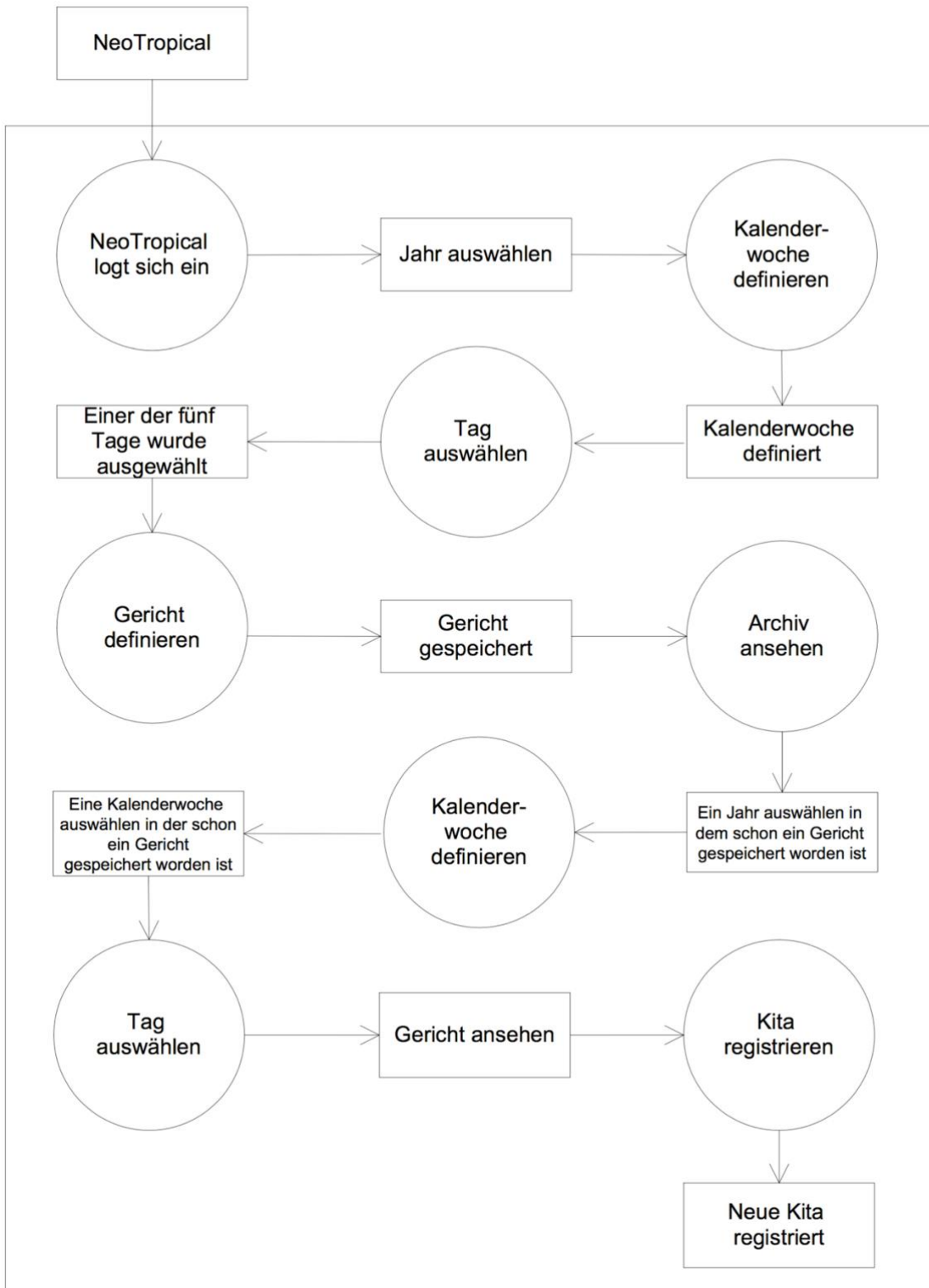


Abbildung 29: Leistungsdiagramm NeoTropical

### 2.7.2 Entitäts-Matrix

Mich interessierten nun die Leistungen aus den beiden Leistungsdiagrammen, welche ich identifiziert hatte. Diese zerlegte ich in die einzelnen Informationselemente (Merkmale) wie in der Entität-Matrix aufgezeigt. So konnte ich die Entitäten jeder Leistung erfassen.

Mir ist während dem Erstellen der Entitäts-Matrix aufgefallen, dass es viele Leistungen gibt, die in beiden Leistungsdiagrammen vorhanden sind. Deshalb entschied ich mich, hier eine einzelne Entitäts Matrix zu erstellen und schreibe eine doppelte Leistung nur einmal auf.

Leistung	Merkmale	Entität?
Kita	Kitadaten Logindaten	Kita
Jahr auswählen	Jahr	Jahr
Kalenderwoche definiert	Kalenderwoche	Kalenderwoche
Einer der fünf Wochentage wurde ausgewählt	Wochentag	Tag
Bestellung gespeichert	Mengenangabe Ernährungskriterien Erwachsene / Kind	Bestellung
Ein Jahr auswählen, in der schon eine Bestellung gespeichert worden ist	Jahr	Jahr
Eine Kalenderwoche auswählen, in der schon eine Bestellung gespeichert worden ist	Kalenderwoche	Kalenderwoche
Bestellung ansehen	Mengenangabe Ernährungskriterien Erwachsene / Kind	Bestellung
NeoTropical	Benutzername Passwort	NeoTropical
Gericht gespeichert	Gericht	Tag
Ein Jahr auswählen in der schon ein Gericht gespeichert worden ist	Jahr	Jahr
Eine Kalenderwoche auswählen, in der schon ein Gericht gespeichert worden ist	Kalenderwoche	Kalenderwoche

Tabelle 14: Entitäts-Matrix

### 2.7.3 Entity-Relationship-Modell

Aus der Entität-Matrix konnte ich die einzelnen Entitäten entnehmen. Mit den Entitäten baue ich ein Entity-Relationship-Modell, abgekürzt mit ERM. Das ERM dient als Grundlage eines Datenbankentwurfs. Bevor mittels SQL damit begonnen wird, Tabellen und Beziehungen anzulegen, wird mittels ERM geplant, wie die Datenbankstruktur aufgebaut wird und funktionieren sollte.

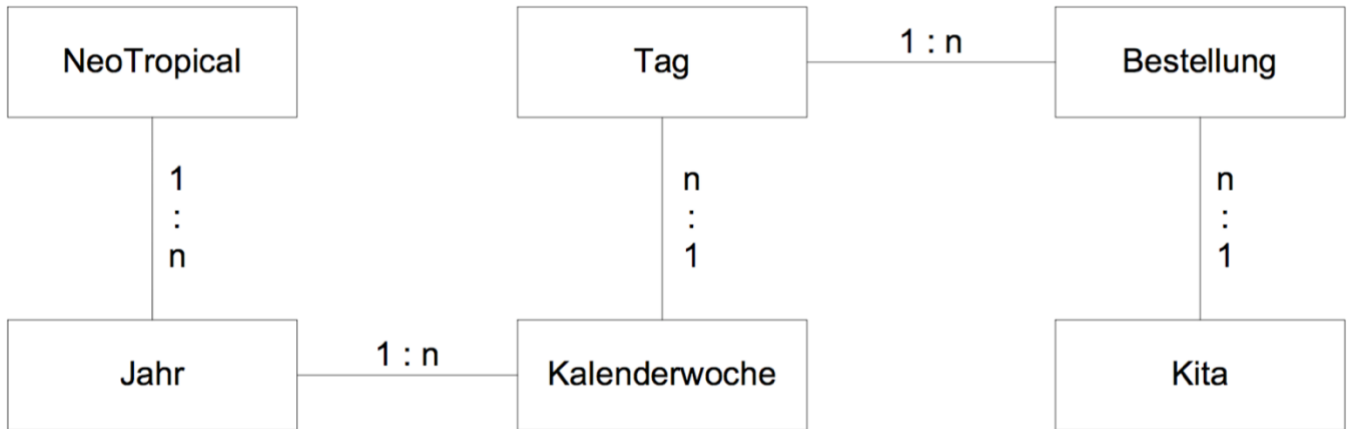


Abbildung 30: Entity-Relationship-Modell

## 2.7.4 Entity-Relationship-Diagramm

Ein Entity-Relationship-Diagramm abgekürzt mit ERD ist ein Diagramm, das zeigt, welche Beziehungen zwischen den Entitäten, die beim ERM definiert wurden, innerhalb eines Systems bestehen. ER-Diagramme werden eingesetzt, um eine relationale Datenbank zu entwerfen oder zu debuggen.

Ich baute nach dem Erstellen des Entity-Relationship-Modells direkt die Datenbank. Aus der Datenbank entnehme ich direkt das Entity-Relationship-Modell, dass ich im Datenbankdiagramm schon erstellt habe. Im Datenbankdiagramm erstelle ich alle Verknüpfungen von den Tabellen her, das heisst ein Primary-Key mit einem Foreign-Key verbinden. Da die Zeit fürs Erstellen der Diplomarbeit sowieso sehr knapp ist, habe ich ohne Archiv gerechnet. Die anderen Ziele muss ich zuerst erreichen. Das Archiv wäre bei der realen Umsetzung wichtig aber für den Prototyp ist es keine Pflicht, sondern nur "nice to have".

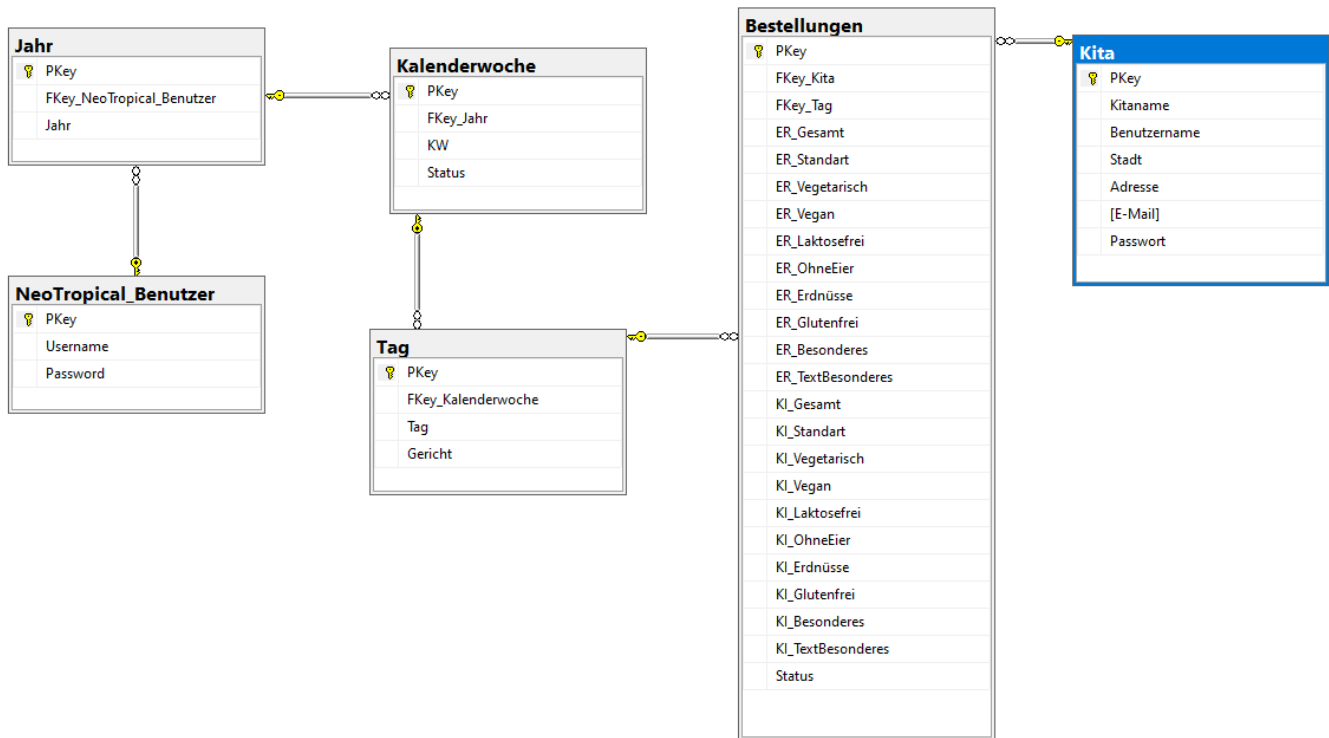


Abbildung 31: Entity-Relationship-Diagramm

## 2.8 Testkonzept

In dem man die positiven Szenarien und die negativen Szenarien aufschreibt, kann man aus den Use Cases die Anforderungsanalyse erstellen. Wie wir festgestellt haben, hat es immer mehr negative als positive Szenarien.

Aus den negativen Szenarien konnte ich die Testfälle erstellen. Es ist sicherer aus negativen Szenarien Testfälle zu erstellen, da man bei negativen Szenarien gezielt einen Fehler testet. Wenn man aus positiven Szenarien die Testfälle erstellt, könnte es passieren, dass das System keine Fehler erkennt und zwanzig Mal ohne Probleme läuft. Aber beim darauf folgenden Durchlauf könnte plötzlich ein Fehler auftauchen, weil ein bestimmtes Ereignis erst beim einundzwanzigsten Durchlauf eintritt.

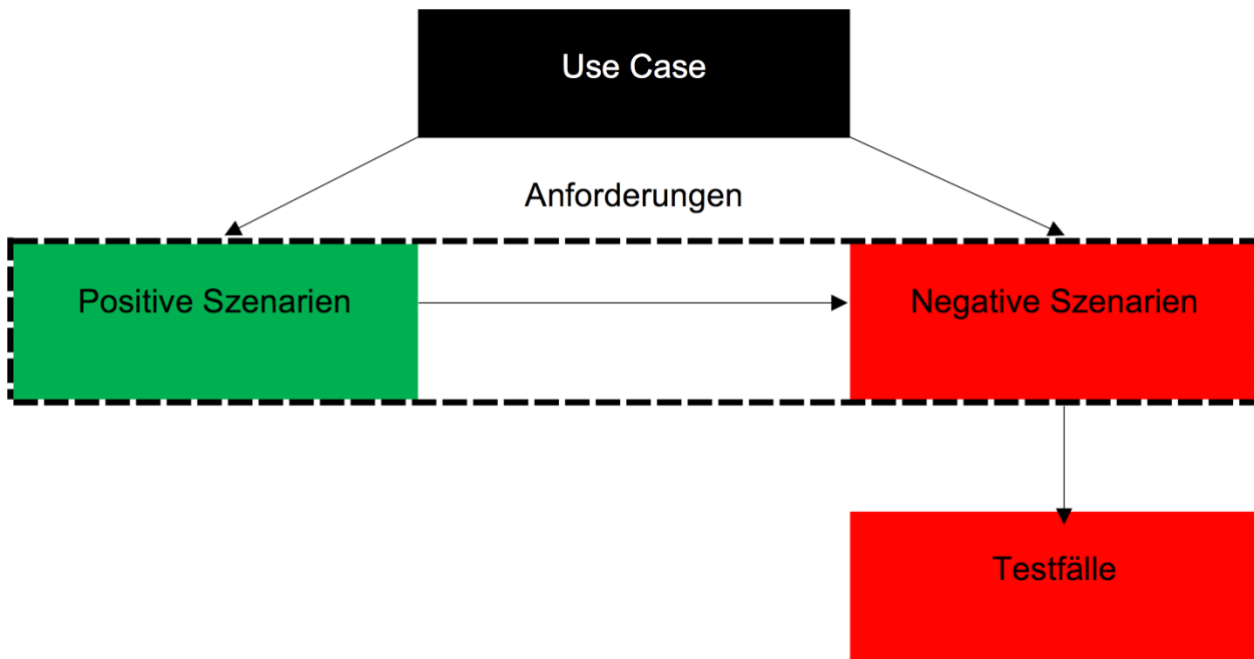


Abbildung 32: Testkonzept

## 2.8.1 Anforderungsanalyse

Das Ziel einer Anforderungsanalyse ist, die Anforderungen an das zu entwickelnde System zu ermitteln und zu strukturieren. Ich arbeitete mit Hilfe der Use Cases, zu jedem passenden Include bei den Use Case überlegte ich mir mindestens ein negatives Szenario.

Use Case	Szenario			Risiko			Herkunft		Bemerkung
	Include	Positiv	Negativ	«ETW(1-5)»	«Effekt(1-5)»	RPN	Technisch	Organisat.	
Login	Logindaten angeben	Richtiger Benutzername wurde vom System erkannt	Richtiger Benutzername wurde vom System nicht erkannt	1	5	5	X		
			Falscher Benutzername wurde vom System erkannt	1	5	5	X		
		Richtiges Passwort wurde vom System erkannt	Richtiges Passwort wurde vom System nicht erkannt	1	5	5	X		
			Falsches Passwort wurde vom System erkannt	1	5	5	X		
Essensbestellungen ausfüllen	Gericht auswählen	Eine Kalenderwoche konnte ausgewählt werden in der schon ein Essensplan erstellt worden ist.	Eine Kalenderwoche konnte ausgewählt werden in der kein Essensplan erstellt worden ist.	3	4	12	X		
			Das gewünschte Jahr kann nicht ausgewählt werden.	1	5	5	X		
			Die gewünschte Kalenderwoche kann nicht ausgewählt werden.	1	5	5	X		
		In der ausgewählten Kalenderwoche können alle fünf Wochentage ausgewählt werden und wird das dazugehörige Gericht angezeigt.	In der ausgewählten Kalenderwoche werden keine Gerichte angezeigt.	2	4	8	X		
			In der ausgewählten Kalenderwoche können die Wochentage nicht ausgewählt werden.	1	5	5	X		
		Falls zu allen fünf Wochentagen ein Gericht erstellt worden ist und in der darauffolgenden Kalenderwoche schon ein Essensplan erstellt worden ist, wird der Essensplan der darauffolgenden Kalenderwoche angezeigt und können die nächsten fünf Wochentage ausgewählt werden.	Nicht zu allen fünf Wochentagen wurde ein Gericht erstellt und der Essensplan der nächsten Woche wird trotzdem schon angezeigt	3	3	9	X		
			Zu allen fünf Wochentagen wurde ein Gericht erstellt und die nächste	2	4	8	X		

			Kalenderwoche wird nicht angezeigt, obwohl es einen Essensplan für die Kalenderwoche gibt.						
			In der darauffolgenden Kalenderwoche werden keine Gerichte angezeigt.	2	4	8	X		
			In der darauffolgenden Kalenderwoche können die Wochentage nicht ausgewählt werden	2	5	10	X		
	Gericht definieren	Zum ausgewählten Tag können Ernährungskriterien und Mengenangabe für Erwachsene und Kinder definiert werden	Zum ausgewählten Tag können keine Ernährungskriterien ausgewählt werden.	2	5	10	X		
			Zum ausgewählten Tag können keine Mengenangaben angegeben werden.	3	5	15	X		
			Zum ausgewählten Tag können nur Gerichte für Erwachsene oder Kinder definiert werden.	1	5	5	X		
Archiv kontrollieren	Jahr auswählen	Ein Jahr, in welchem schon ein Essensplan erstellt worden ist, kann ausgewählt werden.	Das gewünschte Jahr kann nicht ausgewählt werden.	2	4	8	X		
	Kalenderwoche auswählen	Eine Kalenderwoche, in der schon ein Essensplan erstellt worden ist, kann ausgewählt werden.	Die gewünschte Kalenderwoche kann nicht ausgewählt werden.	1	5	5	X		
			Eine Kalenderwoche konnte ausgewählt werden in der kein Essensplan erstellt worden ist.	1	5	5	X		
		In der ausgewählten Kalenderwoche können alle fünf Wochentage ausgewählt werden und die jeweilige Essensbestellung wird angezeigt.	In der ausgewählten Kalenderwoche werden keine Gerichte angezeigt.	2	4	8	X		
			In der ausgewählten Kalenderwoche können die Wochentage nicht ausgewählt werden.	1	5	5	X		
			Der Wochentag konnte ausgewählt werden aber die Bestellung wird nicht angezeigt.	3	5	15	X		

Tabelle 15: Anforderungsanalyse Kita

Use Case	Szenario			Risiko			Herkunft		Bemerkung
	Include	Positiv	Negativ	«ETW(1-5)»	«Effekt(1-5)»	RPN	Technisch	Organisat.	
Login	Logindaten angeben	Richtiger Benutzername wurde vom System erkannt	Richtiger Benutzername wurde vom System nicht erkannt	1	5	5	X		
			Falscher Benutzername wurde vom System erkannt	1	5	5	X		
		Richtiges Passwort wurde vom System erkannt	Richtiges Passwort wurde vom System nicht erkannt	1	5	5	X		
			Falsches Passwort wurde vom System erkannt.	1	5	5	X		
Neue Kita registrieren	Logindaten hinterlegen	Der Benutzername und das Passwort wurden erfolgreich hinterlegt.	Neuer Account kann erstellt werden ohne Login daten.	1	5	5	X		
			Ein ungültiges Passwort kann erstellt werden.	2	5	10	X		
	Kitadaten hinterlegen	Ein neuer Account kann erstellt werden mit dem Kitanamen, Kitaadresse und der Kita E-Mailadresse.	Neuer Account kann erstellt werden ohne Kitadaten.	1	5	5	X		
Essensplan anpassen	Kalenderwoche definieren	Zuerst muss ein Jahr ausgewählt werden und danach eine Kalenderwoche, die noch keinen Essensplan hat, in die neuen Gerichte gespeichert werden.	Das gewünschte Jahr kann nicht ausgewählt werden.	1	5	5	X		
			Die gewünschte Kalenderwoche kann nicht ausgewählt werden.	1	5	5	X		
			Eine Kalenderwoche in der schon Gerichte definiert worden sind, kann ausgewählt werden.	3	4	12	X		
		Gericht hinzufügen	Zu einem ausgewählten Wochentag kann ein Gericht hinzugefügt werden.	Zu einem ausgewählten Wochentag kann kein Gericht hinzugefügt werden.	1	5	5	X	
Archiv kontrollieren	Jahr auswählen	Ein Jahr, in welchem schon ein Essensplan erstellt worden ist, kann ausgewählt werden.	Das gewünschte Jahr kann nicht ausgewählt werden.	1	5	5	X		
	Kalenderwoche auswählen	Eine Kalenderwoche, in der schon ein Essensplan erstellt worden ist, kann ausgewählt werden und die Gerichte werden angezeigt.	Die gewünschte Kalenderwoche kann nicht ausgewählt werden.	1	5	5	X		
			Eine Kalenderwoche konnte ausgewählt werden in der kein	3	4	12	X		

			Essensplan erstellt worden ist.						
			In der ausgewählten Kalenderwoche werden keine Gerichte angezeigt.	2	5	10	X		
Endprotokoll erstellen	Kalenderwoche definieren	Ein Jahr, in welchem schon eine Essensbestellung erstellt worden ist, kann ausgewählt werden.	Das gewünschte Jahr kann nicht ausgewählt werden.	1	5	5	X		
		Eine Kalenderwoche, in der schon eine Essensbestellung erstellt worden ist, kann ausgewählt werden.	Die gewünschte Kalenderwoche kann nicht ausgewählt werden.	1	5	5	X		
			Eine Kalenderwoche konnte ausgewählt werden in der keine Essensbestellungen erstellt worden sind.	2	5	10	X		
	Kitas auswählen	Nachdem die Kalenderwoche ausgewählt worden ist, können Kitas ausgewählt werden für die ein oder kein Endprotokoll erstellt werden soll.	Nachdem die Kalenderwoche ausgewählt worden ist, können keine Kitas ausgewählt werden.	1	5	5	X		
			Nachdem die Kalenderwoche ausgewählt worden ist, werden keine Kitas angezeigt.	2	5	10	X		

Tabelle 16: Anforderungsanalyse NeoTropical

## 2.9 GUI-Konzept

Für den Prototyp möchte NeoTropical schon ein einheitliches attraktives GUI, welches auch modern aussieht. Aus diesem Grund habe ich mich entschieden ein Konzept für das GUI zu erstellen. Wichtig für ein modernes GUI ist, dass der User nicht mit zu vielen Eingabe Modulen überfordert ist. Der User muss den roten Faden des Tools sehen, damit er selbständig damit arbeiten kann. Zuerst musste ich entscheiden, mit welchen Farben ich arbeiten möchte. Ich beschloss, das Tool mit den Farben von NeoTropical zu gestalten und eine ähnliche Struktur wie auf der Webseite von NeoTropical aufzubauen.

Schon zu Beginn der Konzeptphase habe ich dank dem Kontextdiagramm herausgefunden, dass ich zwei Systeme programmieren muss. Das eine System ist für NeoTropical, darin werden die Gerichte erstellt und Kitas registriert. Das andere System ist für die Kitas, darin werden die Essensbestellungen ausgefüllt.

Nach der Erstellung von allen GUI Konzepten, traf ich mich mit Ueli Enggist. Während dieser Sitzung besprachen wir alle GUI Konzepte, so dass der Prototyp ein attraktives GUI hat und die Firma NeoTropical zufrieden ist. Wir besprachen auch den Ablauf der beiden Systeme, den ich geplant hatte und Ueli Enggist gab seine Inputs.

So konnte ich für beide Systeme bei ein paar View's dasselbe Konzept verwenden, wie zum Beispiel beim Login.

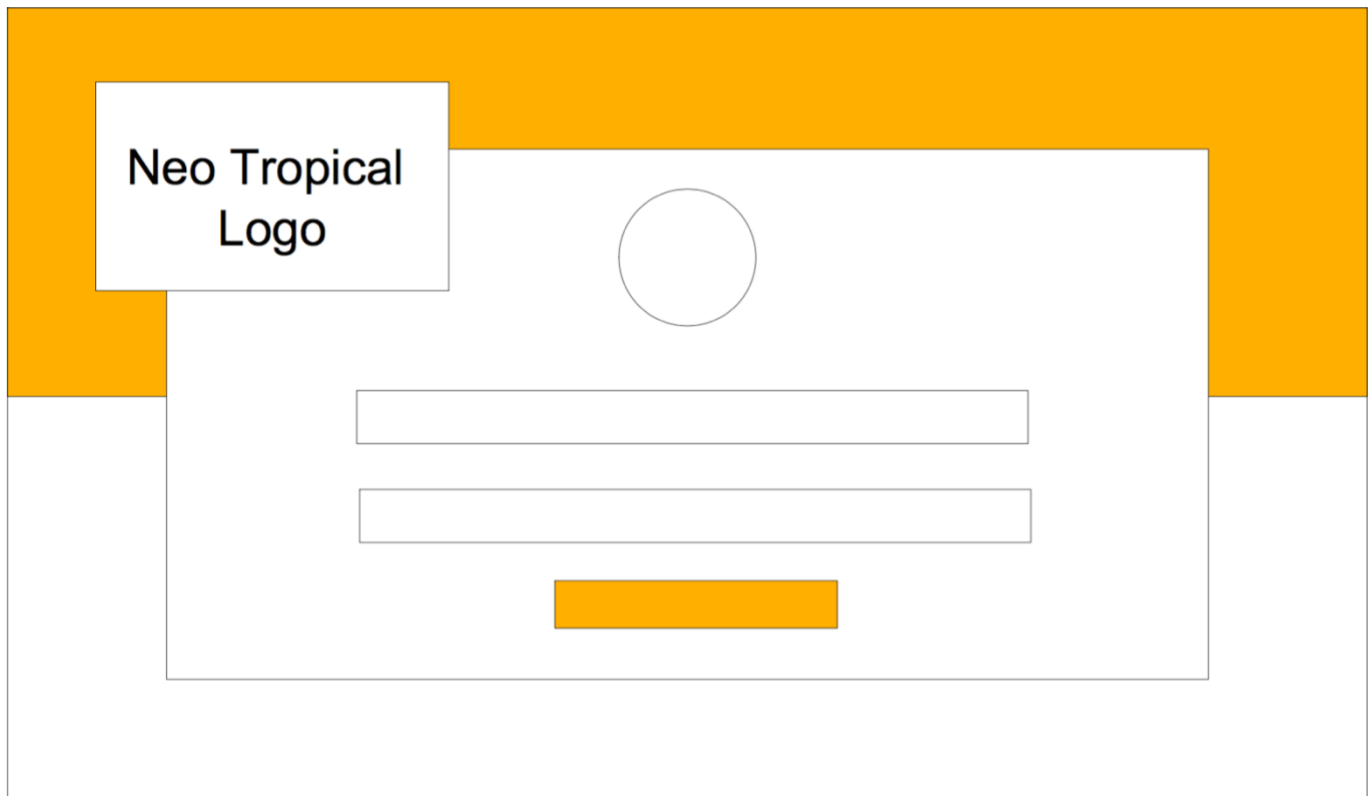


Abbildung 33: GUI-Login

Ueli Enggist und ich waren derselben Meinung, dass es am sinnvollsten ist, wenn man direkt nach dem Login auf die Seite kommt, in der das meiste erledigt wird. Deshalb wollte ich für diese Systeme keine Hauptseite entwickeln. Beim Kitasystem ist das die Seite Essensbestellung, in der die Kitas ihre Essensbestellungen für eine bestimmte Kalenderwoche aufgeben können. Bei dem zweiten System für NeoTropical ist das die Seite Essensplan. Auf der Seite Essensplan kann ein Angestellter von NeoTropical einen Essensplan mit allen Gerichten für eine Woche erstellen.

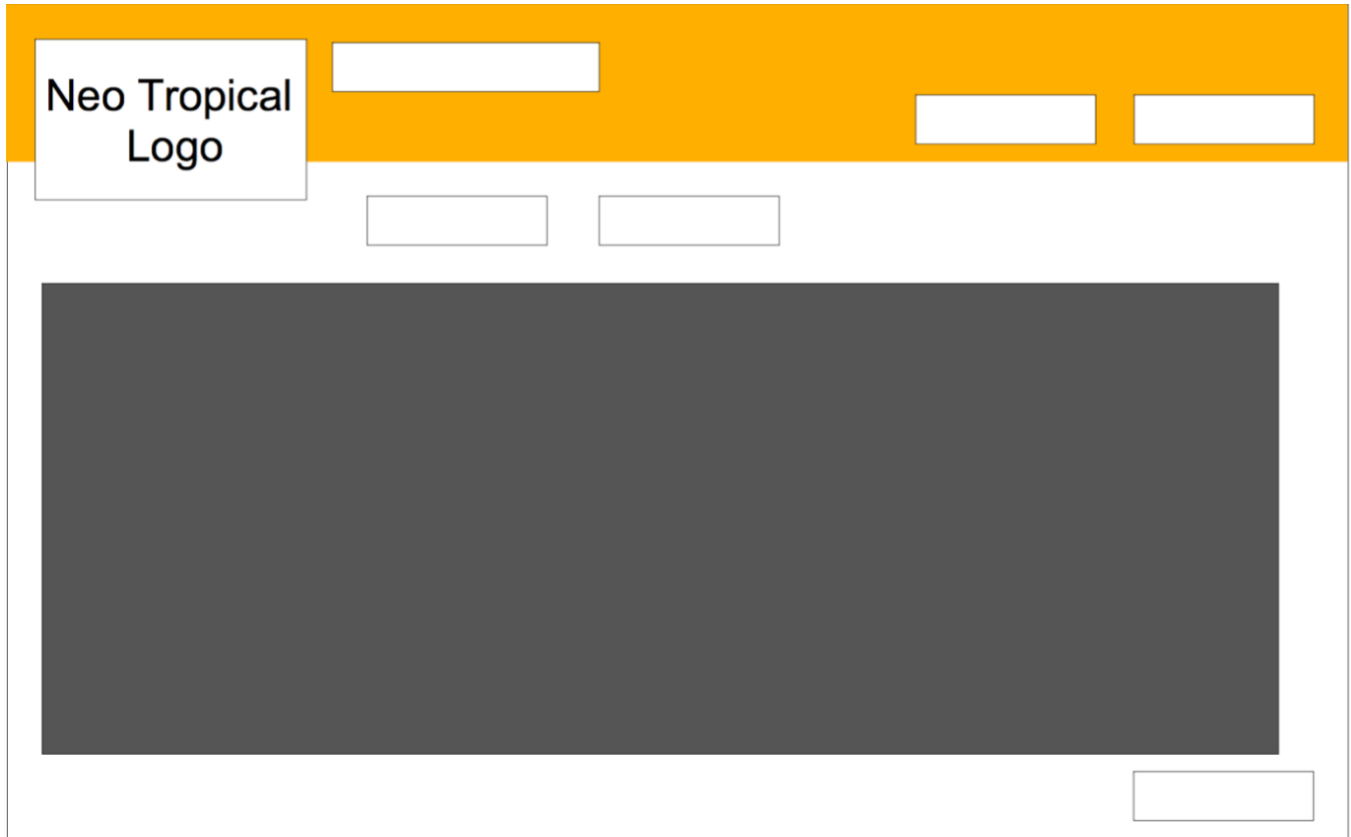


Abbildung 34: GUI-Essensbestellung

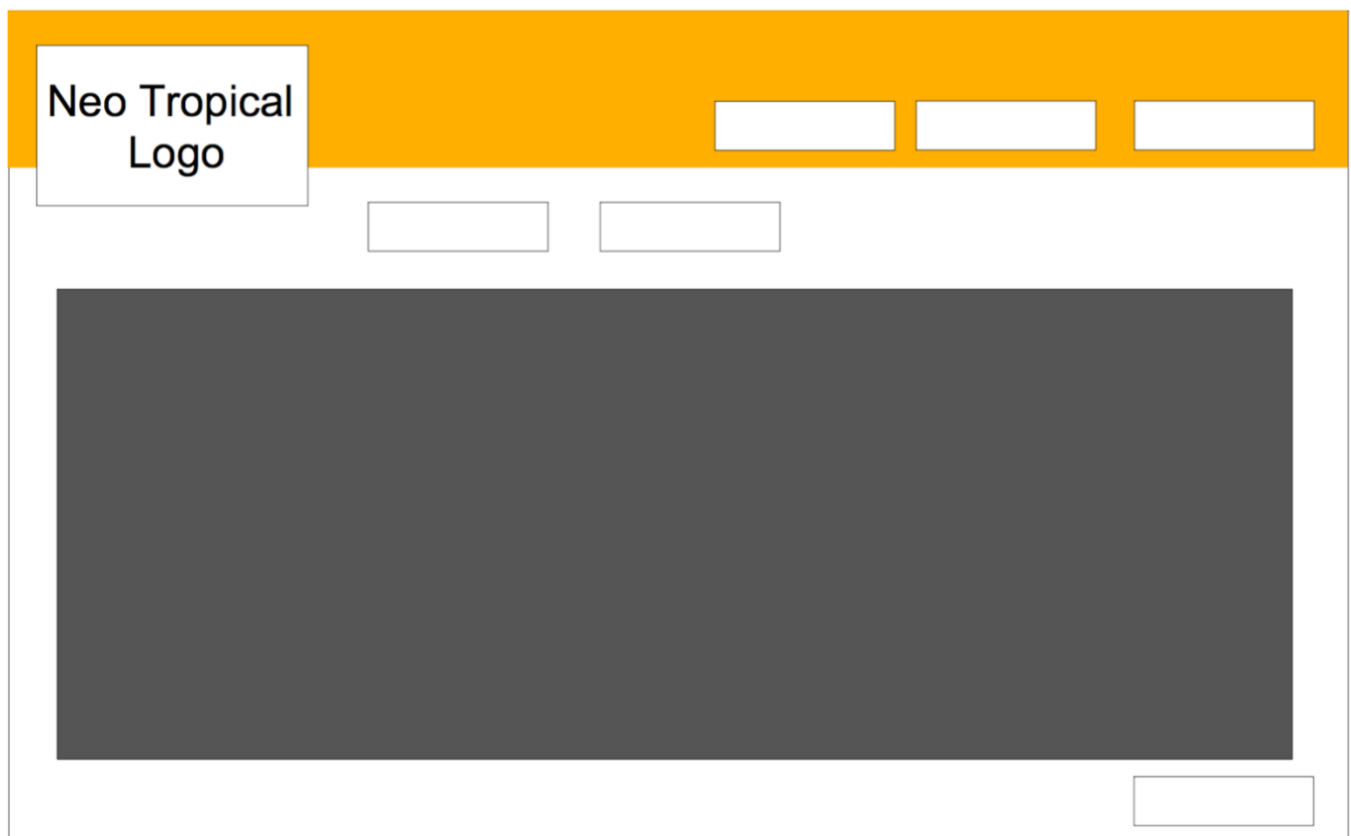


Abbildung 35: GUI-Essensplan

Für das Kitasystem benötigte ich noch ein weiteres GUI, für die spezifische Essensbestellung für jeden Tag. Meine Überlegung war, wenn ich zusätzlich alle Eingabe Module für die Ernährungskriterien für Kinder und Erwachsene auf der Essensbestellungs Seite aufführen würde, würde die Seite zu überladen. Dann würde sie nicht mehr geordnet und strukturiert wirken. Deshalb entschied ich mich für eine weitere Seite. Darin wird das aktuelle Gericht für den ausgewählten Tag angezeigt und man kann die Ernährungskriterien für Kinder und Erwachsene definieren.

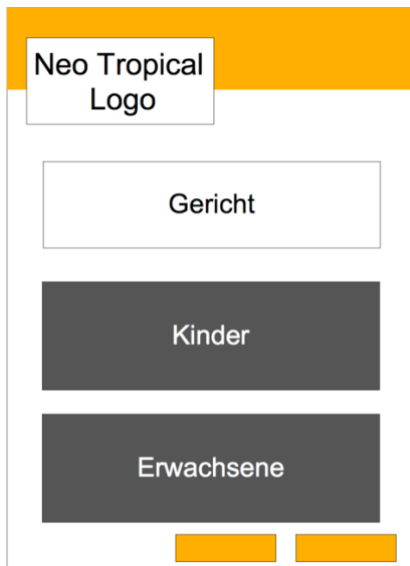


Abbildung 36: GUI-Gericht

Die zweitletzte Seite, für die ich noch ein GUI Konzept benötigte, ist für die Seite, in der neue Kitas registriert werden. Hierzu fragte ich Ueli Enggist, welche Kitadaten, neben den Login Daten, für das System notwendig sind. Wir entschieden, dass die Kitaadresse, Kita E-Mailadresse und der Kitanamen wichtig sind.

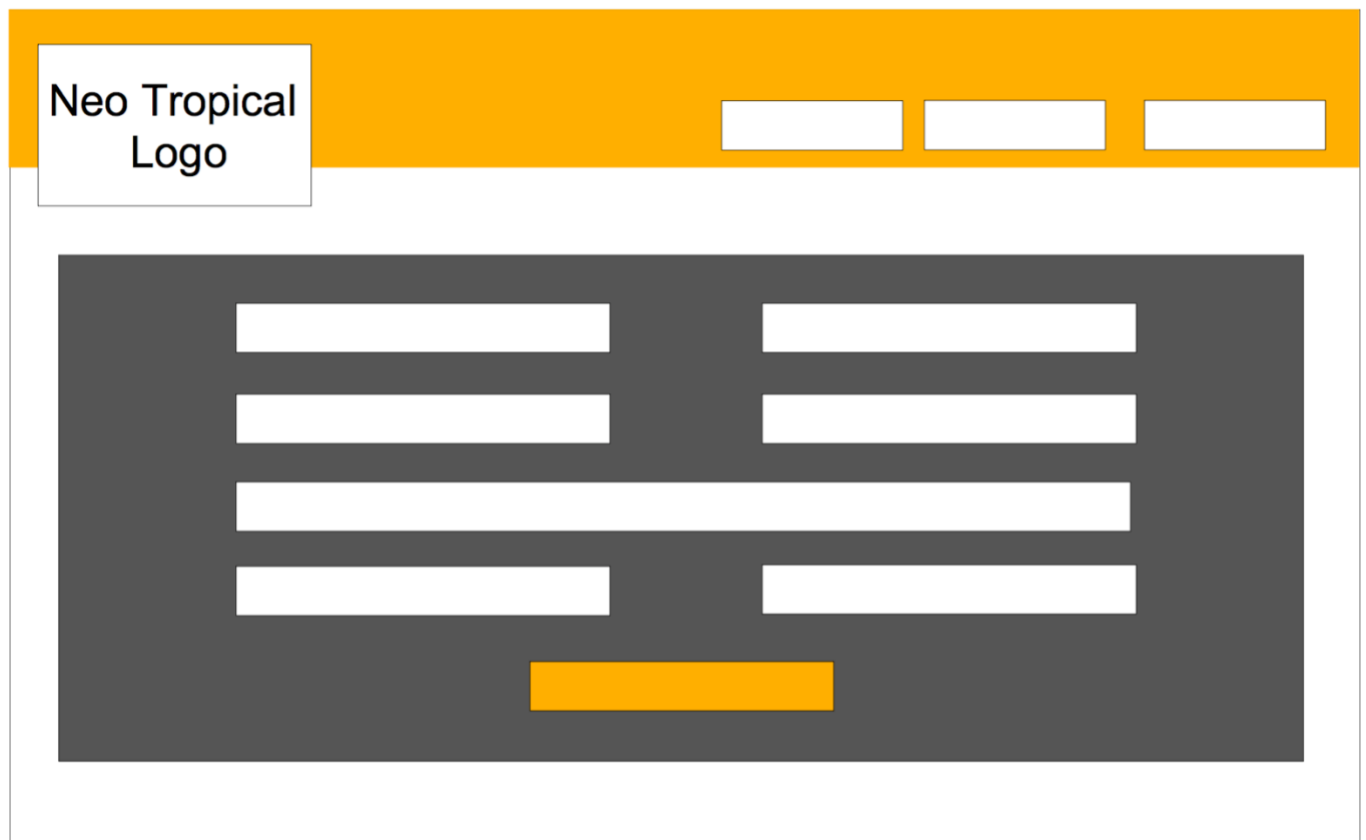


Abbildung 37: GUI-Registrieren

Das letzte GUI Konzept ist für das Archivieren. Ich weiss noch nicht, ob ich das GUI Konzept für das Archivieren überhaupt benötige. Das Archivieren ist ein mögliches Ziel und hat deshalb nicht oberste Priorität. Ich wollte der Vollständigkeit halber dennoch ein Konzept für das GUI erstellen, falls ich schon während der Diplomarbeit zum Archivieren kommen würde. So habe ich ein Konzept für beide Systeme erstellt, da der Aufbau vom GUI in beiden Systemen ähnlich ist. Der einzige Unterschied ist, dass NeoTropical zusätzlich noch Kitas auswählen kann, wenn Sie eine Kalenderwoche ausgewählt haben. Eine Kita kann nur eine Kalenderwoche auswählen.



Abbildung 38: GUI-Archiv

## 2.10 Endprotokoll Konzept

Ueli Enggist hat mir schon zu Beginn gesagt, dass ein ordentliches, gut strukturiertes Endprotokoll sehr wichtig für ihn sei, da bis jetzt fast jede Kita ihr eigenes Endprotokoll habe und es so sehr schwierig sei, das jeweilige Endprotokoll zu entschlüsseln. Bis jetzt ist es nicht zu 100% ersichtlich wie viele Portionen es insgesamt gibt und wie viele davon ein Ernährungskriterium haben. So erlebte die Firma NeoTopical schon viele Probleme, die ich mit einem klar strukturierten und einheitlichem Endprotokoll beheben möchte.

Damit ich so ein klar strukturiertes und einheitliches Endprotokoll erstellen kann, entschied ich mich ein Konzept für das Endprotokoll zu verfassen. Im Konzept will ich ermitteln, welche Daten für das Endprotokoll notwendig sind und wie ich Sie optisch schön und sinnvoll darstellen kann.

Nachdem ich das Konzept für das Endprotokoll erstellt hatte, traf ich mich wieder mit Ueli Enggist damit ich ihm meinen Vorschlag präsentieren konnte. Mir ist es wichtig in engem Kontakt mit dem Kunden zu stehen, damit das Endprodukt auch seinen Vorstellungen entspricht. Falls Ueli Enggist auch noch konstruktive Kritik zum Endprotokoll hat, besprechen wir zusammen den Kritikpunkt und finden eine Lösung, die für beide stimmig ist.

### Menü Bestellung

Kita: \_\_\_\_\_ KW: \_\_\_\_\_ Monat: \_\_\_\_\_ Jahr: \_\_\_\_\_

Gericht					
Datum					
Gericht					

Kinder					
Ges					
Veg					
Vegan					
Glu					
Lak					
OhEi					
OhErd					
Bes					

Erwachsene					
Ges					
Veg					
Vegan					
Glu					
Lak					
OhEi					
OhErd					
Bes					

Ges = Gesamt                      Veg = Vegetarisch                      Glu = Glutenfrei  
 Lak = Laktosefrei                      OhEi = Ohne Eier                      OhErd = Erdnüsse  
 Bes = Besonderes

Abbildung 39: Endprotokoll

## 3 Realisierung

### 3.1 Programmierumgebung

Als Programmierumgebung wählte ich Microsoft Visual Studio und als Programmiersprache C-Sharp (C#). C# ist eine objektorientierte Allzweck-Programmiersprache und kann dadurch das gesamte System modellieren.

Ich entschied mich für das GUI-Toolkit Windows Forms des Microsoft .NET Frameworks. Es ermöglicht die Erstellung grafischer Benutzeroberflächen (GUI) für Windows. Mir wurde immer wieder mitgeteilt, dass das Windows GUI-Toolkit Forms veraltet ist, nicht mehr oft verwendet wird und stattdessen Windows Presentation Foundation (WPF). Ich entschied mich trotzdem für Windows Forms, da wir in den letzten Semestern in der Schule auch ein paar GUI gestalten mussten und uns Windows Forms erklärt wurde. Ich arbeitete bis jetzt nur mit dem Windows GUI-Toolkit Forms.

### 3.2 Softwareaufbau

Bei der Definition aller Ziele hatte ich festgelegt, welches Muss- und welches Kann-Ziele sind. Durch diese Gliederung konnte ich mich zuerst auf alle Muss-Ziele fokussieren und wenn danach die Zeit noch reicht, kann ich mit den Kann-Zielen starten. Für mich ist es wichtig alle Muss-Ziele abgeschlossen zu haben.

#### 3.2.1 Kita

Das erste System, das ich realisieren wollte, war das Kita-System. Ich entschied mich deshalb dafür, weil es den Hauptprozess, des gesamten Essenbestellungstools, beinhaltet. Da es ein Prototyp wird, ist es am wichtigsten, den Hauptprozess eines Systems darzustellen.

Der Hauptprozess des Essenbestellungstools und des Kita-Systems ist das Essenbestellen. Die Ernährungskriterien müssen zu jedem Gericht und zu jedem Tag individuell auswählbar sein. Wichtig bei dem Essenbestellen ist auch, dass man nicht zweimal zur selben Kalenderwoche eine Essensbestellung ausfüllen kann. Jedoch muss unbedingt jede Kita zu einer bestimmten Kalenderwoche ihre Bestellungen ausfüllen können.

##### 3.2.1.1 Login

Ich erstellte zuerst das GUI des Logins. Da ich schon ein GUI-Konzept erstellt hatte, wusste wie die einzelnen Tools anzuordnen sind. Danach strukturierte ich mein Programm nach Model-View-Controller und erstellte die einzelnen Instanzen damit die einzelnen Klassen miteinander kommunizieren können. Als erstes programmierte ich die Datenauslesung und das Datenschreiben der Datenbank. Der Datentransfer auf die Datenbank findet in der Model Klasse statt, somit erstellte ich als erstes die Methoden auslesen, schreiben und löschen (weitere Infos für die Datenbankimplementierung und Datenbankankbindung im Kapitel 3.3).

In der View für das Login werden die beiden TextBoxen für das Passwort und den Benutzernamen über einen Button ausgelesen. Zusätzlich ist es möglich, das Passwort mit Hilfe einer CheckBox anzuzeigen. Wird der Zustand der CheckBox geändert, wird auch der PasswordChar der TextBox vom Passwort geändert. Damit der angegebene Benutzername zum angegebenen Passwort passt, wird er in der Klasse Controller überprüft. Ich überprüfe den Benutzernamen und das eingegebene Passwort, indem ich einen SQL String erstelle, mit welchem mit Hilfe von Transact-SQL nur User ausgelesen werden, die zum Benutzernamen und zum Passwort passen. Wenn der Count vom DataRow mehr als eins ist, stimmen die E-Mail-Adresse und das Passwort nicht überein. Danach funktionierte das Login.

```
//Überprüfen ist die E-Mail adresse und das Passwort richtig
if (my_dt.Rows.Count == 1)
{
    RegistrationSuccessfully = "Successfully";
    Kitaname = my_dt.Rows[0]["Kitaname"].ToString();
}
else
{
    MessageBox.Show("Either your E-Mail or Password is incorrect", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

## 3.2.1.2 Essenbestellen

Nachdem ich das Login programmiert hatte, erstellte ich die Form für die Essensbestellung. Zu diesem GUI hatte ich zuvor ein Konzept erstellt, somit hatte ich in kürzester Zeit das GUI fertig. Der nächste Schritt war, wieder eine Instanz zum Controller aufzubauen. Sobald ich die Grundlagen hatte, wie zum Beispiel einen Button, um wieder zum Login-Form zurück zu gelangen, konnte ich mit der Programmierung der Essenbestellung beginnen.

Zuerst erstellte ich eine Methode, mit der ich das ausgewählte Jahr und die Kalenderwoche auslesen konnte. Die beiden Daten schickte ich zum Controller. Dieser überprüfte, ob beide Felder ausgefüllt sind und ob zur ausgewählten Kalenderwoche schon ein Essensplan erstellt worden ist. Ich entschied, fürs Überprüfen, ob schon ein Essensplan erstellt worden ist, eine eigene Methode zu verwenden. So konnte ich das Programm besser strukturieren. Falls zur ausgewählten Kalenderwoche ein Essensplan existiert, wird dieser ausgelesen und in der View dargestellt.

```
Jahr = Convert.ToInt32(comboBox_Jahr.SelectedItem);
KW = Convert.ToInt32(comboBox_KW.SelectedItem);

//Gerichte aus Datenbank holen und zu jedem Button und Label zuordnen
DataTable Gerichte = my_controller.pruefe_JahrKW(Jahr, KW, Kitaname);
if (Controller.Controller.RegistrationSuccessfully == "Successfully")
{
    Controller.Controller.RegistrationSuccessfully = "";

    panel_Monatsplan.Visible = true;
    buttonSpeichernMonatsplan.Visible = true;

    label_Tag1.Text = Gerichte.Rows[0]["Tag"].ToString();
    label_Tag2.Text = Gerichte.Rows[1]["Tag"].ToString();
    label_Tag3.Text = Gerichte.Rows[2]["Tag"].ToString();
    label_Tag4.Text = Gerichte.Rows[3]["Tag"].ToString();
    label_Tag5.Text = Gerichte.Rows[4]["Tag"].ToString();

    button_Tag1.Text = Gerichte.Rows[0]["Gericht"].ToString();
    button_Tag2.Text = Gerichte.Rows[1]["Gericht"].ToString();
    button_Tag3.Text = Gerichte.Rows[2]["Gericht"].ToString();
    button_Tag4.Text = Gerichte.Rows[3]["Gericht"].ToString();
    button_Tag5.Text = Gerichte.Rows[4]["Gericht"].ToString();
}
```

Abbildung 41: Essensplan anzeigen

Ich erstellte eine weitere Methode, die beim Laden der Essenbestellungs-Form ausgeführt wird. Die Methode begrüßt die Kita und schreibt ins Label «Herzlich willkommen 'Kitanamen'». Diese Methode startet noch eine zweite Methode im Controller. Die zweite Methode löscht alle Bestellungen, die nicht gespeichert worden sind.

Der nächste Stepp im Programm ist, ein Gericht für diesen Tag auszuwählen. Da es zehn Methoden geworden sind und der Inhalt dieser fast derselbe ist, entschied ich eine elfte Methode mit demselben Inhalt aber mit Variablen, die sich immer ändern können, zu erstellen. So kann ich bei den anderen zehn Methoden den Inhalt löschen und muss bei diesen zehn Methoden nur die elfte Methode öffnen. Der Inhalt der elften Methode ist, eine weitere Form zu öffnen und zu dieser Form Variablen zu senden. Die Form, die geöffnet werden sollte, ist die in der alle Ernährungskriterien dargestellt werden sollen. Für diese elf Methoden baute ich eine Region und konnte mein Programm so noch besser gestalten.

### 3.2.1.3 Gericht definieren

Als erstes erstellte ich wieder das GUI. Dies und die Instanz auf den Controller war schnell erledigt. Sobald ich das GUI gebaut hatte und die Instanz eingerichtet worden war, erstellte ich eine Methode, die aufgerufen wird, wenn das Form lädt. In dieser Methode schreibe ich das Gericht in die TextBox und das Datum zum Label. Die beiden Variablen habe ich vom Essenbestellen Form mitgenommen, wie ein DataTable mit der Bestellung, falls schon eine ausgefüllt worden ist. Falls schon eine Essensbestellung für diesen Tag ausgefüllt worden war, wird die Essensbestellung genau so dargestellt wie gespeichert, mit allen Ernährungskriterien und deren Anzahl. Aber wenn noch keine Essensbestellung für diesen Tag erstellt worden ist, wird eine 0 bei den Ernährungskriterien dargestellt.

Die Gesamtzahl aller Ernährungskriterien wird automatisch zusammengezählt, so kann es weniger zu Missverständnissen kommen. Es gibt insgesamt zwei Gesamtzahlen, eine für die Kinder und eine für die Erwachsenen. Für jede Gesamtzahl wird eine Region erstellt. In der jeweiligen Region werden alle TextBoxen zusammengezählt und die Gesamtzahl dargestellt.

Eine weitere wichtige Methode wird gebraucht, wenn die Kita sich verklickt hat. Dann möchte die Kita vielleicht keine Essensbestellung ausfüllen. Dafür gibt es einen Button, der Abbrechen heisst. Wenn man den Button drückt, schliesst man diese Form ohne zu speichern und kehrt zurück zum Essenbestellen Form.

Die letzte Methode, die notwendig ist, wird für das Speichern der Essensbestellung benötigt. Die Methode überprüft zuerst, ob das Ernährungskriterium Besonderes ausgewählt worden war. Falls dies der Fall ist, muss auch eine Beschreibung erstellt worden sein. Falls keine Beschreibung ausgefüllt worden ist, wird eine Benachrichtigung erstellt, in welcher der Kita mitgeteilt wird, dass sie entweder eine Anzahl an besonderen Gerichten hinterlegen muss oder eine Beschreibung für die besonderen Gerichte. Sobald die Überprüfung in Ordnung ist, können die Variablen dem Controller gesendet werden.

```
//Überprüfen wurde das Ernährungskriterium Besonderes ausgewählt und beschrieben
if (((comboBox_BesoEr.SelectedItem.ToString() == "0" && textBox_BesoEr.Text == "") || (comboBox_BesoEr.SelectedItem.ToString() != "0" && textBox_BesoEr.Text != "")) && ((comboBox_BesoKi.SelectedItem.ToString() == "0" && tex
{
    //Essensbestellung speichern
    my_controller.BestellungSpeichern(Kitaname, label_Datum.Text, comboBox_StaEr.SelectedItem.ToString(), comboBox_VegeEr.SelectedItem.ToString(), comboBox_VegaEr.SelectedItem.ToString(), comboBox_LaktEr.SelectedItem.ToStr:
    if (Controller.Controller.RegistrationSuccessfully == "Successfully")
    {
        Controller.Controller.RegistrationSuccessfully = "";
        this.Close();
    }
}
else
{
    if (comboBox_BesoKi.SelectedItem.ToString() == "0" && textBox_BesoKi.Text != "" || comboBox_BesoEr.SelectedItem.ToString() == "0" && textBox_BesoEr.Text != "")
    {
        MessageBox.Show("Bitte wählen Sie wie viel besonders sind.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (comboBox_BesoKi.SelectedItem.ToString() != "0" && textBox_BesoKi.Text == "" || comboBox_BesoEr.SelectedItem.ToString() != "0" && textBox_BesoEr.Text == "")
    {
        MessageBox.Show("Bitte beschreiben Sie was besonders ist.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}
```

Abbildung 42: Überprüfen des Ernährungskriteriums Besonderes

Im Controller wird überprüft, ob es die erste Bestellung dieser Kita zu genau diesem Tag ist. Falls es die erste Bestellung ist, wird diese in der Datenbank gespeichert. Aber falls schon eine Bestellung gespeichert worden ist, wird der Eintrag in der Datenbank updatet. Nachdem die Essensbestellung erfolgreich in die Datenbank geschrieben worden ist, schliesst sich die Form.

### 3.2.1.4 Essenbestellen abschliessen

Ich konnte nun Essensbestellungen für einem Tag definieren, aber ich musste noch die Kalenderwoche abspeichern. Deswegen musste ich, nach dem ich den Vorgang Gericht definieren erfolgreich abgeschlossen hatte, die Essenbestell Form noch erweitern. Die Erweiterung beinhaltet zwei neue Methoden. Mit einer Methode kann man die Essensbestellungen, die zuvor definiert worden sind, abspeichern und mit der zweiten Methode wird überprüft, ob in der darauffolgenden Kalenderwoche schon ein Essensplan erstellt worden ist.

Die Methode, mit der man Essensbestellungen abspeichern kann, stellte mich vor keine Probleme. Beim Erstellen vom Essenbestellen-GUI hatte ich einen Button fürs Abspeichern entwickelt. Dadurch benötigte ich für den Button nur noch eine Click Methode. Die Click Methode in der View vom Essenbestellen Form startet eine neue Methode im Controller. Die Methode, die im Controller aufgerufen wird, überprüft, ob zu jedem Tag ein Gericht definiert worden ist oder ob ausgewählt wurde, dass die Kita am gewählten Tag keine Gerichte definieren möchte. Falls zu jedem Wochentag ein Gericht definiert oder die CheckBox für keine Gerichte aktiviert worden ist, wird der Status von jedem Gericht auf Bestellt gesetzt.

Die zweite Methode, die für das Überprüfen der darauffolgenden Kalenderwoche zuständig ist, stellte mich vor grössere Herausforderungen. Ich hatte zuerst keine Idee, wie ich die Methode starten könnte, da ich hier keinen Button erstellen wollte, mit der ich die nächste Kalenderwoche anzeigen kann. Ich fand in der Form eine Methode, welche MouseEnter heisst. Diese tritt ein, wenn die Maus über den sichtbaren Bereich des Steuerelements bewegt wird. So musste ich zuerst sicherstellen, dass die darauffolgende Kalenderwoche erst angezeigt wird, wenn zu jedem Wochentag in der aktuellen Kalenderwoche ein Gericht definiert worden ist. Da ich mit der Softwarearchitektur MVC arbeite, überprüfte ich das nicht direkt in der Methode, sondern erstellte für die Überprüfung eine Methode im Controller, den ich über die Methode MouseEnter starte. Falls die Überprüfung erfolgreich war, konnte ich die Gerichte in der Datenbank auslesen. Da ich schon eine Methode im Controller erstellte hatte, der die Gerichte auslesen kann, konnte ich diese Methode verwenden, um die erste Kalenderwoche anzuzeigen.

Die Methode hatte nur ein Problem. Sobald ich zu allen fünf Wochentagen eine Essensbestellung aufgegeben hatte und mit der Maus auf das Form kam, startete die Methode neu. Wenn die Methode neu startete, überprüfte sie immer wieder ob für die nächste Kalenderwoche schon ein Essensplan erstellt worden ist. Falls kein Essensplan erstellt worden ist, kommt eine Fehlermeldung und wenn einer erstellt worden ist, werden die Wochentage neu angezeigt. Aber der Vorgang sollte nur einmalig durchlaufen werden. So kam ich auf die Idee, eine Variable zu erstellen, die sich ändert, wenn der Vorgang einmalig durchlaufen worden ist. Vor jedem weiteren Durchlaufen wird zuerst überprüft, ob die Variable schon geändert worden ist. Falls die Variable geändert worden ist, schliesst sich die Methode wieder.

```
//Überprüfen wurde schon ein Druchlauf durchgeführt
if (Durchlauf == 0)
{
    my_controller.Überprüfen_Bestellungen();

    //Überprüfen wurden schon fünf Essensbestellungen ausgeführt
    if (Controller.Controller.RegistrationSuccessfully == "Successfully")
    {
        Controller.Controller.RegistrationSuccessfully = "";

        Durchlauf = 1;

        DataTable Gerichte = my_controller.pruefe_JahrKW(Jahr, KW + 1, Kitaname);

        //Überprüfen wurde für die nächste KW schon ein Essensplan erstellt
        if (Controller.Controller.RegistrationSuccessfully == "Successfully")
        {
            Controller.Controller.RegistrationSuccessfully = "";
        }
    }
}
```

Abbildung 43: Durchläufe überprüfen

### 3.2.1.5 Archiv

Da ich mit allen Muss Zielen vom Kita System fertig geworden war, konnte ich noch ein Archiv erstellen. Das Archivieren ist eigentlich nur ein Kann Ziel, dennoch wollte ich so viele Ziele wie möglich abschliessen und entwickelte deshalb noch ein Archiv für die Kitas.

Da ich auch für das Archivieren schon ein GUI-Konzept erstellt hatte, konnte ich dies schnell umsetzen. Wie bei den anderen Forms benötigte ich wieder eine Instanz zum Controller und zwei Methoden. Mit der einen kann ich mich aus dem Tool ausloggen und mit der anderen das Form zum Bestellen Form wechseln.

Auf der View musste ich nicht viel neues programmieren, da ich das meiste von der View vom Bestellen übernehmen konnte. Aber im Controller benötigte ich eine neue Methode, die die Essensbestellungen nur anzeigt, wenn die eingeloggte Kita zur ausgewählten Kalenderwoche eine Essensbestellung abgesendet hatte.

Die grösste Herausforderung war, die einzelnen Bestellungen anzuzeigen. Ich hatte schon eine View entwickelt, da ich dieselbe wie beim Essenbestellen verwenden wollte. Die einzelnen Angaben zu den Essenbestellungen darf ich aber nicht in eine Combobox schreiben, da man sonst im Archiv die Essensbestellung ändern kann. Deshalb musste ich alle ComboBoxen durch eine TextBox ersetzen, in der die Kita den Wert nicht ändern kann. Weiter benötigte ich auch noch einen Button mit welchem ich zurück zum Archiv gelangen kann. Dafür erstellte ich einen neuen Button, der das Gericht Form wieder schliesst.

## 3.2.2 NeoTropical

Nachdem ich das Kita System programmiert und erstellt hatte, konnte ich mit dem NeoTropical System anfangen. Da ich ein einheitliches, modernes GUI erstellen möchte, konnte ich viel vom Kita System übernehmen.

### 3.2.2.1 Login

Zuerst musste wieder das Login GUI erstellt werden. Da ich im Kita System schon ein Login GUI erstellt hatte, konnte ich das ganze Form kopieren. Danach mussten, wie beim Kita System, alle Instanzen zu jeder Klasse erstellt werden. Die Methode, mit der ich das Passwort anzeigen kann und die Methode, mit der ich den Benutzernamen und das Passwort auslesen kann, konnte ich auch vom Kita System kopieren. Die Methode im Controller, um die Benutzernamen und Passwörter zu überprüfen, konnte ich auch vom Kita System kopieren. Ich musste in dieser Methode nur zwei Sachen ändern: Da die Benutzer von NeoTropical in einer anderen Tabelle der Datenbank abgespeichert werden, musste ich den SQL String ändern. Die zweite Änderung war die Form, welche geöffnet werden soll, sobald das Login erfolgreich ist.

```
//Passwort anzeigen und wieder bedecken
1-Verweis
private void Passwortanzeigen_checkBox_CheckedChanged(object sender, EventArgs e)
{
    if (Passwortanzeigen_checkBox.Checked)
    {
        textpassword.PasswordChar = '\0';
    }
    else
    {
        textpassword.PasswordChar = '•';
    }
}
```

Abbildung 44: Passwort anzeigen

Abbildung 45: Login-Form

## 3.2.2.2 Wochenplan

Nach dem Erstellen des Logins war der nächste Schritt, eine Form für den Wochenplan zu erstellen. Das entsprechende Konzept hatte ich schon im GUI-Konzept erstellt. Die Grundlage des Konzepts war dieselbe, wie für das Essenbestellen Form. Dadurch konnte ich ein paar Steuerungselement vom Essenbestellen Form kopieren und einfügen. So konnte ich Zeit sparen und mussten nur noch ein paar Erweiterungen vornehmen. Der nächste Schritt war wieder eine Instanz zum Controller aufzubauen. Sobald ich die Grundlagen hatte, wie einen Button, mit dem ich wieder zum Login-Form gelange, einen Button, mit dem ich zum Registrieren-Form gelange oder einen Button zum Ausloggen, konnte ich mit der Programmierung des Monatsplanes beginnen.

Ich entschied - wie beim Essenbestellen Form - das man zuerst nach Jahr und Kalenderwoche suchen muss. Sobald dies geschieht, kann man auch überprüfen, ob in dieser Kalenderwoche schon ein Essensplan erstellt worden war. Für die Überprüfung wurde im Controller eine Methode erstellt, welche in der Datenbank ausliest, ob für diese Kalenderwoche schon ein Essensplan erstellt worden war. Falls dies der Fall ist, taucht eine Fehlermeldung auf und man muss eine neue Kalenderwoche aussuchen. Sobald man eine Kalenderwoche ausgewählt hat, in der kein Essensplan erstellt worden ist, wird die Kalenderwoche angezeigt. Mein Ziel war, dass zu jedem Tag der Kalenderwoche das exakte Datum dargestellt wird. Ich entdeckte im Internet, wie ich für jedes Jahr mit Hilfe der Kalenderwoche das exakte Datum von jedem Wochentag herausfinden kann. Ich konnte mit der Struktur DateTime einen Tag hinterlegen und mit .AddDays Tage der DateTime Struktur hinzufügen. Somit konnte ich jeder Kalenderwoche die exakten Daten zuordnen.

```
//Datum vom Montag von der Kalenderwoche ermitteln
2 Verweise
public static DateTime KW(string Jahr, int KW)
{
    //1. Tag im Jahr x
    DateTime myDT = new DateTime(Convert.ToInt32(Jahr), 1, 4);

    //Beginn auf Montag setzten
    myDT = myDT.AddDays(-(int)((myDT.DayOfWeek != DayOfWeek.Sunday) ? myDT.DayOfWeek - 1 : DayOfWeek.Saturday));

    //Wochen dazu addieren
    return myDT.AddDays(--KW * 7);
}
```

Abbildung 46: Datum ermitteln

Zuletzt benötigte ich noch eine Methode, mit der ich den erstellten Wochenplan abspeichern kann. Dafür entwickelte ich im Wochenplan View eine Button Click Methode. Diese Methode öffnet eine Methode im Controller, welche den Wochenplan überprüft. Die Methode überprüft, ob zu jedem Tag ein Gericht erstellt worden ist und speichert die einzelnen Gerichte beim entsprechenden Tag ab. Falls nicht zu jedem Tag ein Gericht definiert worden ist, taucht eine Fehlermeldung auf und fordert auf, dass zu jedem Tag ein Gericht definiert werden muss.

### 3.2.2.3 Registrieren

Im NeoTropical Tool gibt es aktuell nur drei Aktionen, die man durchführen kann, da das Essenbestellungstool nur ein Prototyp ist. Neben dem Essensplan erstellen, gibt es noch die Möglichkeit, einer Kita einen Useraccount zuzuteilen. Dafür muss zuerst eine Kita mit unterschiedlichen Variablen, die sich von Kita zu Kita ändert, registriert werden. Das Konzept vom GUI hatte ich schon erarbeitet. Während des Erstellens des GUI-Konzepts hatte ich abgeklärt, welche Kitadaten ich benötigen würde. Nach dem Erstellen vom GUI musste ich wieder eine Instanz zum Controller aufbauen. Sobald ich die Grundlagen hatte, wie einen Button, mit dem ich wieder zum Login-Form gelange, einen Button, mit dem ich zum Registrieren-Form gelange oder einen Button zum Ausloggen, konnte ich mit der Programmierung der Registrierung beginnen.

Zuerst erstellte ich eine Methode, mit der ich alle Textfelder - von den Logindaten und Kitadaten - auslesen kann. Die ausgelesenen Textfelder schicke ich in den Controller zu einer neuen Methode, die alle Textfelder kontrolliert. Zuerst wird kontrolliert, ob in jede TextBox etwas geschrieben worden war. Danach wird das erstellte Passwort untersucht. Das erstellte Passwort darf nicht kürzer als sechs Buchstaben oder Zahlen sein. Ein weiterer Punkt, der beim Passwort überprüft wird, ist, dass es nur aus Zahlen und Gross- oder Kleinbuchstaben bestehen darf. Zuletzt wird noch überprüft, ob die beiden Passwörter der beiden TextBoxen Passwort und Passwort bestätigen, identisch sind. Sobald alle Kriterien erfüllt worden sind, werden die Daten der Kita in die Datenbank geschrieben.

Ich wollte das Registrieren optisch noch schöner gestalten. Deshalb entschied ich, dass die TextBoxen immer einen Mustertext enthalten. Dieser Mustertext ändert sich von TextBox zu TextBox, da er eine Vorlage sein sollte, was NeoTropical hineinschreibt. Der Mustertext sollte beim Anwählen der TextBox verschwinden. Aber wenn der User entscheidet, zu einem neuen Steuerungselement zu gehen und die angewählte TextBox noch keinen Text enthält, wird der Mustertext wieder angezeigt. Ich benötigte zum Realisieren vom Mustertext zwei Methoden in der TextBox. Die erste Methode heisst Enter. Diese wird ausgeführt, wenn das gewählte Steuerungselement zum aktiven Steuerelement des Formulars wird. Die zweite Methode heisst Leave. Diese Methode wird ausgeführt, wenn das gewählte Steuerungselement nicht mehr das aktive Steuerelement des Formulars ist. Ich benötigte für jede TextBox, die einen Mustertext hat, beide Methoden. Dadurch wurde der Code sehr gross. Aber weil der Inhalt von jeder Enter Methode und von jeder Leave Methode fast identisch war, entschied ich, zwei weitere Methoden zu entwickeln. In diesen beiden Methoden steht der Inhalt der Leave und Enter Methoden. Mit Variablen kann ich den Inhalt der Methoden auf jede TextBox anpassen.

```
public void textSampleEnter(TextBox textbox, String textsample1, String textsample2)
{
    if (textbox.Text.Equals(textsample1))
    {
        textbox.ForeColor = System.Drawing.Color.Black;
        textbox.Text = textsample2;
    }
}

1-Verweis
private void textKita_Enter(object sender, EventArgs e)
{
    textSampleEnter(textKita, @"Muster-Kita", "");
}
```

Abbildung 47: Enter-Methode

```
public void textSampleLeave(TextBox textbox, String textsample1, String textsample2)
{
    if (textbox.Text.Equals(textsample1))
    {
        textbox.ForeColor = System.Drawing.Color.Gray;
        textbox.Text = textsample2;
    }
}

1-Verweis
private void textKita_Leave(object sender, EventArgs e)
{
    textSampleLeave(textKita, "", @"Muster-Kita");
}
```

Abbildung 48: Leave-Methode

## 3.2.2.4 Archiv

Da ich im Kita System schon ein Archiv entwickelt hatte, war das Grundgerüst im NeoTropical System schnell fertig. Da ich aber im NeoTropical System nicht nur die schon erstellten Essenspläne anzeigen möchte, sondern zusätzlich einen Essensplan zur ausgewählten Kalenderwoche, stellte das mich vor grössere Herausforderungen als zuerst gedacht.

Ich wollte, nachdem eine Kalenderwoche ausgewählt worden ist, alle Kitas anzeigen, die schon zu der ausgewählten Kalenderwoche einen Wochenplan erstellt hatten. Somit sieht NeoTropical sofort welche Kitas einen Wochenplan erstellt haben und welche noch nicht. Danach kann NeoTropical bestimmen, zu welchen Kitas ein Endprotokoll erstellt wird oder alle Kitas auswählen. Somit können einzelne Kitas für ein Endprotokoll ausgeschlossen werden. Falls man zu einer Kita, welche erst später einen Wochenplan erstellt hat, noch ein Endprotokoll erstellen möchte, muss man nicht alle Endprotokolle neu erstellen, sondern kann nur eine Kita auswählen.

Damit ich alle Kitas anzeigen konnte, die zu einer Kalenderwoche einen Wochenplan erfasst hatten, musste ich eine Liste mit allen Kitas erstellen. Die Kitas schrieb ich danach im Archiv View in eine CheckedListBox. NeoTropical kann durch eine weitere CheckBox alle Items in der CheckedListBox auswählen oder nicht mehr auswählen, dadurch wird das Tool benutzerfreundlicher.

```
for (int i = 0; i < my_dt.Rows.Count; i++)
{
    SQL_string = "select Kitaname from [Kita] Where PKey = '" + my_dt.Rows[i]["FKey_Kita"] + "'";

    //Holen der Daten
    my_ds = my_model.auslesen(SQL_string);

    //Nutzen des DataSets
    //Umwandeln in DataTable
    DataTable mydt = my_ds.Tables["mein_mds"];

    Kita.Add(mydt.Rows[0]["Kitaname"].ToString());
}
```

Abbildung 49: Kita-List erstellen

```
if (Kita.Count != 0)
{
    panel_Kitas.Visible = true;
    buttonSpeichernEndprotokoll.Visible = true;

    foreach (var _Kita in Kita)
    {
        checkedListBox1.Items.Add(_Kita, false);
    }
}
```

Abbildung 50: Kita-List in CheckedListBox einfügen

Der schwierigste Teil, für den ich auch am meisten Zeit gebraucht hatte, musste ich jetzt noch entwickeln. Der NeoTropical User kann, nachdem er die gewünschten Kitas ausgewählt hat, ein Endprotokoll für die ausgewählte Kalenderwoche für diese Kitas erstellen. Das erstellte Endprotokoll wird mit Excel dargestellt. Damit ich aus C# ein Excel Dokument machen kann, habe ich mir aus dem Unterricht von Christian Herren den Code für das Erstellen und Schreiben in ein Excel Dokument kopiert. Bis zu diesem Zeitpunkt hatte ich noch nie ein Excel Dokument in C# erstellt. Deshalb entwickelte ich zuerst ein separates Visual Studio Projekt. In diesem probierte ich, wie ein schon erstelltes Excel Dokument zu öffnen ist und wie ich in das Dokument Werte einfügen kann. Ich recherchierte viel im Internet aber fand keine Lösung. Deshalb fragte ich Christian Herren in meinem Review, wie ich das Problem beheben könnte. Eine weitere Schwierigkeit war, dass mir ein Verweis im Visual Studio Projekt gefehlt hat. Deshalb erkannte C# die Namespaces von Excel nicht. Ich schilderte meine Situation und darauf schickte Christian Herren mir die fehlende Assembly «Microsoft.Office.Interop.Excel» zu. Danach musste ich im Projekt nur noch den fehlenden Verweis aktivieren. Jetzt konnte ich ein Excel Dokument erstellen.

Sobald mir dies möglich war, konnte ich an der Diplomarbeit weiterarbeiten. Zuerst musste ich im Archiv View alle Kitas auslesen, die ausgewählt worden sind. Danach hatte ich alle notwendigen Parameter zum

Erstellen des Endprotokolls. Ich baute im Controller eine weitere Methode, mit der ich sicherstellen konnte, dass zu jeder Kita ein Endprotokoll erstellt wird. Zudem muss die Methode noch alle Ernährungskriterien zu jedem Tag auslesen. Sobald ich zu jedem Tag und für die Gerichte ein DataTable hatte, schrieb ich die notwendigen Parameter in die Model Klasse.

```
//SQL-String definieren
string SQL_string = "select * from [Tag] Where [FKey_Kalenderwoche] = '" + PKey_KW + "'";

//Holen der Daten
my_ds = my_model.auslesen(SQL_string);

//Nutzen des DataSets
//Umwandeln in DataTable
DataTable my_dt = my_ds.Tables["mein_mds"];

for (int i = 0; i < kita.Count; i++)
{
    //SQL-String definieren
    SQL_string = "select * from [Kita] Where [Kitaname] = '" + kita[i] + "'";

    //Holen der Daten
    my_ds = my_model.auslesen(SQL_string);

    DataTable dt_Kita = my_ds.Tables["mein_mds"];

    int PKey_Kita = Convert.ToInt32(dt_Kita.Rows[0]["PKey"]);

    //Tag 1
    int PKey_Tag = PKey_tag(my_dt.Rows[0]["Tag"].ToString());

    SQL_string = "select * from [Bestellungen] Where FKey_Kita = '" + PKey_Kita + "' and FKey_Tag = '" + PKey_Tag + "'";

    //Holen der Daten
    my_ds = my_model.auslesen(SQL_string);

    //Nutzen des DataSets
    //Umwandeln in DataTable
    DataTable dt_Tag1 = my_ds.Tables["mein_mds"];
}
```

Abbildung 51: DataTable erstellen für Endprotokoll

### 3.3 Datenbankimplementierung und Datenbankanbindung

Die Datenbankimplementierung und Datenbankanbindung wurde für das System Kita und für das System NeoTropical genau gleich gestaltet und implementiert. Im NeoTropical System gibt es noch eine zusätzliche Methode fürs Erstellen des Excel Dokuments.

#### 3.3.1 ADO.NET Framework

Für den Datentransport zwischen dem C-Sharp und Microsoft SQL-Server verwendete ich das ADO.NET Framework. Dies ist das älteste Framework. Es ist deshalb nicht minder robust, allerdings muss jede Operation auf der Datenbank als eigenes SQL Statement konzipiert und die Daten dem entsprechend ausgelesen, aufbereitet und geschrieben werden.

ADO.NET stellt konsistenten Zugriff auf Datenquellen wie SQL-Server und XML bereit. Verbraucheranwendungen mit Datenfreigabe können mit ADO.NET eine Verbindung mit diesen Datenquellen herstellen und die enthaltenen Daten abrufen, verarbeiten und aktualisieren.

ADO.NET trennt den Datenzugriff von der Datenbearbeitung in einzelne Komponenten, die separat oder zusammen verwendet werden können. ADO.NET schließt .NET Framework-Datenanbieter zum Verbinden mit einer Datenbank, zum Ausführen von Befehlen und zum Abrufen von Ergebnissen ein. Diese Ergebnisse werden entweder direkt verarbeitet oder in einem ADO.NET DataSet Objekt platziert, um sie dem Benutzer, kombiniert mit Daten aus mehreren Quellen, bei Bedarf verfügbar zu machen oder um sie an eine andere Ebene zu übergeben. Das DataSet Objekt kann auch unabhängig von einem .NET Framework Datenanbieter verwendet werden, um Daten zu verwalten, die lokal für die Anwendung sind.

#### 3.3.2 Programm realisieren

Da ich für die Softwarestruktur MVC verwende, existiert eine ganze Klasse, die den Datentransport übernimmt. Die Klasse, die diese Aufgaben übernimmt, heisst Model. In der Modelklasse befinden sich drei Methoden im Kita System und im NeoTropical System vier.

Mit der ersten Methode kann ich Daten aus der Datenbank auslesen und das erstellte DataSet mit allen Daten zurücksenden.

```
//Auslesen
15 Verweise
public DataSet auslesen(string SQL_query)
{
    //connection objekt
    SqlConnection my_conn = new SqlConnection(conn_str);

    //Dataset definieren
    DataSet my_mds = new DataSet();

    //Versuchen auf die DB zuzugreifen
    try
    {
        //Erfolgreich
        //Zugriff physisch
        my_conn.Open();

        //dataadaptor füllen
        SqlDataAdapter my_da1 = new SqlDataAdapter(SQL_query, my_conn);
        //dataset füllen
        my_da1.Fill(my_mds, "mein_mds");
    }
    catch (Exception ex)
    {
        //Nicht erfolgreich
        MessageBox.Show(ex.Message);
    }
    finally
    {
        //Schliessen
        my_conn.Close();
    }

    return my_mds;
}
```

Abbildung 52: DB-auslesen

Die zweite Methode in der Klasse Model ist für das Datenschieben zuständig. Hier werden neue Daten in die Datenbank geschrieben und bei erfolgreichem Schreiben wird ein String mit dem Wert fertig zurückgeschickt.

```
//Schreiben
30 Verweise
public string schreiben(string SQL_query)
{
    SqlConnection my_conn = new SqlConnection(conn_str);

    //Commandobjekt bilden
    SqlCommand my_cmd = new SqlCommand(SQL_query, my_conn);

    try
    {
        //Connection öffnen
        my_conn.Open();

        //Befehl abschicken
        my_cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        //Fehlermeldung
        MessageBox.Show(ex.Message);
    }
    finally
    {
        //DB-Connection schliessen
        my_conn.Close();
    }

    return "fertig";
}
```

Abbildung 53: DB-schreiben

Die letzte Methode in der Klasse Model im System Kita ist für das Daten löschen verantwortlich. In dieser Methode werden Daten aus der Datenbank gelöscht und wird auch ein String mit dem Wert fertig zurückgeschickt.

```
//Löschen
8 Verweise
public string löschen(string SQL_query)
{
    //Connection aufbauen
    SqlConnection my_conn = new SqlConnection(conn_str);

    //Commandobjekt bilden
    SqlCommand my_cmd = new SqlCommand(SQL_query, my_conn);

    try
    {
        //Connection öffnen
        my_conn.Open();

        //Befehl abschicken
        my_cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        //Fehlermeldung
        MessageBox.Show(ex.Message);
    }
    finally
    {
        //DB-Connection schliessen
        my_conn.Close();
    }

    return "fertig";
}
```

Abbildung 54: DB-löschen

Im NeoTropical System brauchte ich noch eine viere Methode. Diese arbeitet nicht mit der Datenbank zusammen, sondern hat die Aufgabe, in das Endprotokoll zu schreiben. Erstens überprüfe ich in dieser Methode, ob für diese Kalenderwoche schon eine Excel Tabelle erstellt worden ist. Dafür erstellte ich einen bool, der mit Hilfe vom vorher erstellen Path einen true oder false Wert erhält. Falls der Path noch nicht existiert, erstelle ich Verweise auf verwendete Excel Strukturen. Ich wollte meine Werte nicht in eine neue Excel Tabelle schreiben, sondern in eine Excel Tabelle, die ich schon erstellt hatte. Damit ich so eine Vorlage verwenden kann, muss ich beim Öffnen des Workbooks den Path hinterlegen, in welchen die Vorlage abgespeichert worden ist. Sobald ich das Workbook gestartet und das richtige Arbeitsblatt ausgewählt habe, kann ich meine Werte aus den DataTable in das Excel Arbeitsblatt schreiben. Danach muss ich den neuen Path erstellen, in dem das neue Endprotokoll abgespeichert werden kann. Am Schluss muss ich noch den Path definieren in dem das Excel Dokument abgespeichert wird.

```
string Path = "C:\\Temp\\NeoTropical\\Endprotokoll\\" + Kita + "\\ " + Jahr + "\\ " + KW;

//Überprüfen wurde schon ein Endprotokoll erstellt
bool exist = System.IO.Directory.Exists(Path);
if (exist)
{
    MessageBox.Show("Für die Kita " + Kita + " wurde schon ein Endprotokoll ertsellit, für die Kalenderwoche " + KW + " im Jahr " + Jahr, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
{
    //Verweise auf die verwendete EXCEL-Struktur
    Microsoft.Office.Interop.Excel.Application oXL;
    Microsoft.Office.Interop.Excel._Workbook oWB;
    Microsoft.Office.Interop.Excel._Worksheet oSheet;

    object misvalue = System.Reflection.Missing.Value;
    //Start Excel and get Application object.
    oXL = new Microsoft.Office.Interop.Excel.Application();
    oXL.Visible = true;

    //Get a new workbook.
    oWB = (Microsoft.Office.Interop.Excel._Workbook)(oXL.Workbooks.Add("C:\\Temp\\NeoTropical\\Endprotokoll\\Endprotokoll.xlsx"));
    oSheet = (Microsoft.Office.Interop.Excel._Worksheet)oWB.ActiveSheet;

    Zellen befüllen

    oXL.Visible = false;
    oXL.UserControl = false;

    //Path erstellen
    exist = System.IO.Directory.Exists(Path);
    if (!exist)
    {
        System.IO.Directory.CreateDirectory(Path);
    }

    //abspeichern
    oWB.SaveAs(Path + "\\Endprotokoll_" + Kita + "_" + Jahr + "_" + KW + ".xlsx", Microsoft.Office.Interop.Excel.XlFileFormat.xlWorkbookDefault, Type.Missing, Type.Missing,
        false, false, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChange,
        Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

    oWB.Close();
}
```

Abbildung 55: Endprotokoll erstellen

```
#region Zellen befüllen
// Excel Kopf mit Werten füllen
oSheet.get_Range("B5").Value2 = "Kita: " + Kita;
oSheet.get_Range("I5").Value2 = "KW: " + KW;
oSheet.get_Range("L5").Value2 = "Jahr: " + Jahr;

// Excel Zellen mit dem Datum füllen
oSheet.get_Range("D11").Value2 = dt_Gericht.Rows[0]["Tag"].ToString();
oSheet.get_Range("F11").Value2 = dt_Gericht.Rows[1]["Tag"].ToString();
oSheet.get_Range("H11").Value2 = dt_Gericht.Rows[2]["Tag"].ToString();
oSheet.get_Range("J11").Value2 = dt_Gericht.Rows[3]["Tag"].ToString();
oSheet.get_Range("L11").Value2 = dt_Gericht.Rows[4]["Tag"].ToString();

```

Abbildung 56: Endprotokoll Zellen befüllen

### 3.4 Testprotokoll

Nachdem ich mit dem Programmieren fertig geworden war, habe ich mit Testfällen geprüft, ob die wichtigsten Eigenschaften des Programms fehlerfrei funktionieren. Die Testfälle entnahm ich der zuvor erstellten Anforderungsanalyse. Ich wählte alle Risiken mit einem Risiko von acht oder höher für die Testfälle aus.

Use Case	Include	Szenarien		Analyse		Realisation	
		Positiv	Negativ	Testfall	Soll	Ist	Wichtigkeit
Essensbestellungen ausfüllen	Gericht auswählen	Eine Kalenderwoche konnte ausgewählt werden, in der schon ein Essensplan erstellt worden ist.	Eine Kalenderwoche konnte ausgewählt werden in der kein Essensplan erstellt worden ist.	Es werden zehn Kalenderwochen ausgewählt in der kein Essensplan erstellt worden ist.	Alle zehn Kalenderwochen konnten nicht ausgewählt werden.	Alle zehn Kalenderwochen konnten nicht ausgewählt werden.	
		In der ausgewählten Kalenderwoche können alle fünf Wochentage ausgewählt werden und wird das dazugehörige Gericht angezeigt.	In der ausgewählten Kalenderwoche werden keine Gerichte angezeigt.	Es werden drei Kalenderwochen ausgewählt, in denen schon ein Essensplan erstellt worden ist.	Zu jedem Wochentag der drei Kalenderwochen werden die Gerichte angezeigt.	Zu jedem Wochentag der drei Kalenderwochen werden die Gerichte angezeigt.	
		Falls zu allen fünf Wochentagen ein Gericht erstellt worden ist und in der darauffolgenden Kalenderwoche schon ein Essensplan erstellt worden ist, wird der Essensplan von der darauffolgenden Kalenderwoche angezeigt und können die nächsten fünf Wochentage ausgewählt werden.	Nicht zu allen fünf Wochentagen ist ein Gericht erstellt worden und trotzdem wird der Essensplan der nächsten Woche schon angezeigt.	Es werden zu drei Kalenderwochen Bestellungen aufgegeben und in den darauffolgenden drei Kalenderwochen wurde schon ein Essensplan erstellt.	Die darauffolgende Kalenderwoche wird erst angezeigt, wenn zu jedem Wochentag einer Kalenderwoche eine Bestellung aufgegeben worden ist.	Die darauffolgende Kalenderwoche wird erst angezeigt, wenn zu jedem Wochentag einer Kalenderwoche eine Bestellung aufgegeben worden ist.	
			Zu allen fünf Wochentagen wurde ein Gericht erstellt und die nächste Kalenderwoche wird nicht angezeigt, obwohl es einen Essensplan für die Kalenderwoche gibt.	Es werden zu drei Kalenderwochen Bestellungen aufgegeben und in den darauffolgenden drei Kalenderwochen wurde schon ein Essensplan erstellt.	Die darauffolgende Kalenderwoche wird erst angezeigt, wenn zu jedem Wochentag einer Kalenderwoche eine Bestellung aufgegeben worden ist.	Die darauffolgende Kalenderwoche wird erst angezeigt, wenn zu jedem Wochentag einer Kalenderwoche eine Bestellung aufgegeben worden ist.	
			In der darauffolgenden Kalenderwoche werden keine Gerichte angezeigt.	Es werden zu drei Kalenderwochen Bestellungen aufgegeben und in den darauffolgenden	In den drei darauffolgenden Kalenderwochen werden alle Gerichte angezeigt.	In den drei darauffolgenden Kalenderwochen werden alle Gerichte angezeigt.	

				en drei Kalenderwochen wurde schon ein Essensplan erstellt.			
			In der darauffolgenden Kalenderwochen können die Wochentage nicht ausgewählt werden	Es werden zu drei Kalenderwochen Bestellungen aufgegeben und in den darauffolgenden drei Kalenderwochen wurde schon ein Essensplan erstellt.	In den drei darauffolgenden Kalenderwochen können die Wochentage angezeigt und ausgewählt werden.	In den drei darauffolgenden Kalenderwochen können die Wochentage angezeigt und ausgewählt werden.	
	Gericht definieren	Zum ausgewählten Tag können Ernährungskriterien und Mengenangabe für Erwachsene und Kinder definiert werden	Zum ausgewählten Tag können keine Ernährungskriterien ausgewählt werden.	Es werden zu fünfzehn Tagen Gerichte definiert.	Zu zehn Tage kann man Ernährungskriterien definieren und zu fünf Tagen kann man einen Tag speichern ohne Ernährungskriterien.	Zu zehn Tage kann man Ernährungskriterien definieren und zu fünf Tage kann man einen Tag speichern ohne Ernährungskriterien.	
			Zum ausgewählten Tag kann keine Mengenangabe angegeben werden.	Es werden zu fünfzehn Tage Gerichte definiert.	Zu zehn Tage können unterschiedliche Mengenangaben definiert werden.	Zu zehn Tage können unterschiedliche Mengenangaben definiert werden.	
Archiv kontrollieren	Jahr auswählen	Ein Jahr, in dem schon ein Essensplan erstellt worden ist, kann ausgewählt werden.	Das gewünschte Jahr kann nicht ausgewählt werden.	Es wird in drei unterschiedlichen Jahren ein Essensplan erstellt.	Drei Jahre in denen schon ein Essensplan erstellt worden ist, können ausgewählt werden.	Drei Jahre in denen schon ein Essensplan erstellt worden ist, können ausgewählt werden.	
	Kalenderwoche auswählen	In der ausgewählten Kalenderwoche können alle fünf Wochentage ausgewählt werden und die jeweilige Essensbestellung wird angezeigt.	In der ausgewählten Kalenderwoche werden keine Gerichte angezeigt.	Es werden in unterschiedlichen Jahren drei Kalenderwochen ausgewählt, in denen eine Essensbestellung abgeschickt worden ist.	In allen drei Kalenderwochen werden alle Gerichte angezeigt.	In allen drei Kalenderwochen werden alle Gerichte angezeigt.	
			Der Wochentag konnte ausgewählt werden aber die Bestellung wird nicht angezeigt.	Es werden in unterschiedlichen Jahren drei Kalenderwochen ausgewählt, in denen eine Essensbestellung abgeschickt worden ist.	Fünfzehn Wochentage können ausgewählt werden und an den Wochentagen kann die Bestellung angezeigt werden.	Fünfzehn Wochentage können ausgewählt werden und an den Wochentagen kann die Bestellung angezeigt werden.	

Tabelle 17: Testprotokoll Kita

Use Case	Include	Szenarien		Analyse		Realisation	
		Positiv	Negativ	Testfall	Soll	Ist	Wichtigkeit
Neue Kita registrieren	Logindaten hinterlegen	Der Benutzername und das Passwort wurden erfolgreich hinterlegt.	Ein ungültiges Passwort kann erstellt werden.	Es wird versucht, fünf ungültige Passwörter zu erstellen.	Alle fünf ungültigen Passwörter können nicht gespeichert werden.	Alle fünf ungültigen Passwörter können nicht gespeichert werden.	
Essensplan anpassen	Kalenderwoche definieren	Zuerst muss ein Jahr ausgewählt werden und danach eine Kalenderwoche, die noch keinen Essensplan hat, in die die neuen Gerichte gespeichert wird.	Eine Kalenderwoche in der schon Gerichte definiert worden sind, kann ausgewählt werden.	Es wird bei drei Kalenderwochen, in denen schon ein Essensplan erstellt worden ist, getestet, ob man sie auswählen kann.	Alle drei Kalenderwochen konnten nicht ausgewählt werden.	Alle drei Kalenderwochen konnten nicht ausgewählt werden.	
Archiv kontrollieren	Kalenderwoche auswählen	Eine Kalenderwoche, in der schon ein Essensplan erstellt worden ist, kann ausgewählt werden und die Gerichte anzeigen.	Eine Kalenderwoche konnte ausgewählt werden in der kein Essensplan erstellt worden ist.	Es werden fünf Kalenderwochen ausgewählt, in der kein Essensplan gespeichert worden ist.	Es konnte keine der fünf Kalenderwochen ausgewählt werden.	Es konnte keine der fünf Kalenderwochen ausgewählt werden.	
			In der ausgewählten Kalenderwoche werden keine Gerichte angezeigt.	Drei Kalenderwochen, in denen ein Essensplan abgespeichert worden ist, werden ausgewählt.	Zu allen drei Kalenderwochen werden alle Gerichte angezeigt.	Zu allen drei Kalenderwochen werden alle Gerichte angezeigt.	
Endprotokoll erstellen	Kalenderwoche definieren	Eine Kalenderwoche, in der schon eine Essensbestellung erstellt worden ist, kann ausgewählt werden.	Eine Kalenderwoche konnte ausgewählt werden in der keine Essensbestellungen erstellt worden ist.	Es werden drei Kalenderwochen ausgewählt, in denen keine Essensbestellung ausgefüllt worden ist.	Für drei Kalenderwochen, in denen keine Essensbestellung erstellt worden ist, kann kein Endprotokoll erstellt werden.	Für drei Kalenderwochen, in denen keine Essensbestellung erstellt worden ist, kann kein Endprotokoll erstellt werden.	
	Kitas auswählen	Nachdem die Kalenderwoche ausgewählt worden ist, können Kitas ausgewählt werden, für die kein Endprotokoll erstellt werden soll.	Nachdem die Kalenderwoche ausgewählt worden ist, werden keine Kitas angezeigt.	Es werden drei Kalenderwochen ausgewählt, in denen eine Essensbestellung ausgefüllt worden ist.	Zu jeder der drei Kalenderwochen werden die Kitas angezeigt, die für diese Kalenderwoche eine Essensbestellung ausgefüllt haben.	Zu jeder der drei Kalenderwochen werden die Kitas angezeigt, die für diese Kalenderwoche eine Essensbestellung ausgefüllt haben.	
		Sobald alle Kitas ausgewählt worden sind, wird für jede Kita ein Endprotokoll erstellt.	Sobald alle Kitas ausgewählt worden sind, wird kein Endprotokoll erstellt.	Es werden drei Kalenderwochen ausgewählt und in jeder zusätzlich eine, zwei oder drei Kitas.	Für alle Kitas zu jeder Kalenderwoche wurde ein Endprotokoll erstellt.	Für alle Kitas zu jeder Kalenderwoche wurde ein Endprotokoll erstellt.	

Tabelle 18: Testprotokoll NeoTropical

## 4 Schlussbericht

### 4.1 Vergleich Ist/Soll

Die Software wurde anhand der von NeoTropical und Joshua Perolini definierten Anforderungen konfiguriert. Der Prototyp ist funktionsbereit. Alle Anforderungen wurden erfolgreich entwickelt und zusätzlich wurden alle Ziele Muss und Kann erfüllt.

Der Prototyp ist getestet und bereit für die eigentliche Entwicklung. Es wurden immer wieder Fehler entdeckt, die vollständig behoben worden sind. Es wurden auch Verbesserungen entdeckt, welche zu einem späteren Zeitpunkt der Entwicklung der Endlösung möglich werden.

Als Abnahme wurde der Prototyp Ueli Enggist präsentiert. Dabei hätte dieser Einwände anbringen können, die noch zu beheben gewesen wären. Bei der Endabnahme wurde kein offener Punkt angesprochen und Ueli Enggist war sehr zufrieden.

### 4.2 Persönliches Fazit

Durch die Diplomarbeit konnte ich mein Wissen über Visual Studio, C# Windows Form und Microsoft SQL-Server verinnerlichen. Auch das strukturierte Programmieren konnte ich genauer analysieren und im Projekt übernehmen. Immer wieder bin ich zu neuen Aufgaben vorgestossen, die ich zuerst im Internet und anhand der Dokumente analysieren musste, bevor ich sie umsetzen konnte. Während der Arbeit bin ich auf mehrere Hindernisse gestossen, doch mit Hilfe des Internets und dank Christian Herren konnte ich die Arbeit erfolgreich abschliessen.

Da ich wusste, dass ich für die Realisierung mehr Zeit investieren muss, probierte ich immer ein bisschen vor der geplanten Zeit zu sein. Das hat aber nicht immer funktioniert. Ich war sehr froh, dass wir jeweils am Mittwoch keine Schule mehr hatten. So hatte ich jeweils frei und konnte an diesen Tagen intensiv an der Diplomarbeit arbeiten. Da wir in der Klasse sehr oft das Erarbeiten von UML-Diagrammen geübt hatten, hatte ich beim Konzept fast keine Probleme. Das einzige Konzept, mit welchem ich Mühe hatte, war für das Endprotokoll. Ich hatte den Essensplan mit dem NeoTropical aktuell arbeitet, bekommen. Jedoch ist NeoTropical mit diesem nicht zufrieden, da sie beim alten Essensplan keine schöne Übersicht haben, wer welche Ernährungskriterien ausgewählt hat und wie viele Gerichte es insgesamt waren. In der Initialisierung Phase lief alles tiptop.

Ich habe die Aufgaben gut aufgeteilt und konnte zielstrebig arbeiten. Es war trotzdem eine Herausforderung, das Projekt pünktlich abzugeben, da ich bis jetzt nicht oft eine Visual Studio Microsoft Forms Anwendung entwickelt hatte. Zudem arbeite ich nicht in der Informatik und hatte keine Berufserfahrungen. Zu Beginn musste ich, um verstehen zu können, was ich genau entwickeln sollte, viel Zeit in Recherche stecken.

Für mich war diese Diplomarbeit eine grosse Chance, um sehr viel Neues zu lernen und um den ersten Prototyp für eine Anwendung zu erstellen, die im echten Leben verwendet wird. Vor der Diplomarbeit hatte ich noch nie eine Excel Tabelle aus einem Projekt erstellt, durch die Diplomarbeit konnte ich auch dieses Wissen aneignen.

Bei dieser Arbeit habe ich viele spannende Erfahrungen und Kenntnisse im Programmieren gesammelt. Ich habe auch erkannt, dass die Planung (Diagramme, Testfälle, usw.), wenn sie gut durchdacht ist, beim Realisieren hilft, damit es weniger Probleme gibt.

Bestimmt benötigte ich viel mehr Zeit, um zu einem erfolgreichen Abschluss zu gelangen, als geplant war. Jedoch schätze ich sehr, wieviel ich während der gesamten Recherche, Umsetzung und Dokumentierung lernen konnte.

## 5 Quellenverzeichnis

Hermes 5.1: <https://www.hermes.admin.ch/de/projektmanagement/verstehen/ubersicht-hermes/methodenubersicht.html>

MVVM: <https://akrick.wordpress.com/2013/05/18/mvvm/>


MVC: <https://bachasoftware.com/benefits-of-mvc-model-to-web-application-development/>

Funktionale, nicht funktionale Anforderungen:

[https://www.anforderungsmanagement.ch/in\\_depth\\_vertiefung/funktionale\\_nicht\\_funktionale\\_anforderungen/](https://www.anforderungsmanagement.ch/in_depth_vertiefung/funktionale_nicht_funktionale_anforderungen/)

## 6 Eigenständigkeitserklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbstständig geschrieben und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Name, Datum	Unterschrift
Joshua Perolini, 30.10.2021	

## 7 Anhang

- Projektplan
- Themeneingabe
- Essensplan
- Vierwochenmenüplan

Planung Diplomarbeit			Kalenderwoche 37					Kalenderwoche 38					Kalenderwoche 39					Kalenderwoche 40					Kalenderwoche 41					Kalenderwoche 42					Kalenderwoche 43					Kalenderwoche 44																											
			M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
<b>Arbeiten</b>	<b>Soll (in d)</b>	<b>Ist (in d)</b>																																																															
Initialisierung	27	24																																																															
Konzept	50	57																																																															
Realisierung	108	121																																																															
Einführung	6	4																																																															
	<b>Soll (in h)</b>	<b>Ist (in h)</b>																																																															
<b>Initialisierung</b>	<b>27</b>	<b>24</b>																																																															
Stakeholderanalyse	2	1	2	1																																																													
Zeitplanung	3	3	1	2	3																																																												
Qualitätsmanagement	2	2	2	2																																																													
Konfigurationsmanagement	2	2	2																																																														
Situationsanalyse	1	1	1																																																														
Lösungskonzept	2	2	2																																																														
Dokumentation	15	13	3	6	6	3	8	2																																																									
<b>Konzept</b>	<b>50</b>	<b>57</b>																																																															
Kontextdiagramm	2	2	2																																																														
Use Case	2	2	2																																																														
Sequenzdiagramm	4	5	1	4																																																													
Zustandsdiagramm	2	3	2	3																																																													
Klassendiagramm	1	1	1																																																														
Leistungsflussdiagramm	2	2	2																																																														
Entitäts-Matrix	3	3	3																																																														
ERM	3	3	3																																																														
ERD	2	1	1	2																																																													
Planung Endprotokoll	2	4	3	1																																																													
Planung GUI	3	4	2	2																																																													
Testkonzept	4	4	1	3																																																													
Dokumentation	20	23	1	4	3	3	3	3	3																																																								
<b>Realisierung</b>	<b>108</b>	<b>121</b>																																																															
DB erstellen	5	5	3	2																																																													
GUI	8	9	1	3																																																													
C# Umsetzen	55	62	3	5	8	3	1	8	2	5	8	4	2	7	4	2																																																	
Fehler behebung	10	8	2	3																																																													
Dokumentation	30	37	8	8	4	4	4	4	4	8	2																																																						
<b>Einführung</b>	<b>6</b>	<b>4</b>																																																															
Testfälle durchführen	4	2	2																																																														
Dokumentieren	2	2	2																																																														
<b>Saldo Soll-Zeiten</b>	<b>191</b>																																																																
<b>Saldo Ist-Zeiten</b>		<b>206</b>																																																															
Soll-Zeit																																																																	
Ist-Zeit																																																																	
<b>Meilensteine:</b>																																																																	
Meilenstein 1:	Voranalyse fertig																																																																
Meilenstein 2:	Konzept fertig																																																																
Meilenstein 3:	Review																																																																
Meilenstein 4:	Realisieren fertig																																																																
Meilenstein 5:	Abgabe Diplomarbeit																																																																

Themeneingabe zur Diplomarbeit

Kita Essenbestellungstool

B-TIN-18-T-a

Standort: Bern

Eingereicht von

Perolini Joshua

3422 Kirchberg, Reinhardweg 4

joshua.perolini@edu.teko.ch

076 382 69 50

Firmenbetreuer

Enggist Ueli

Neo Tropical Catering & Events

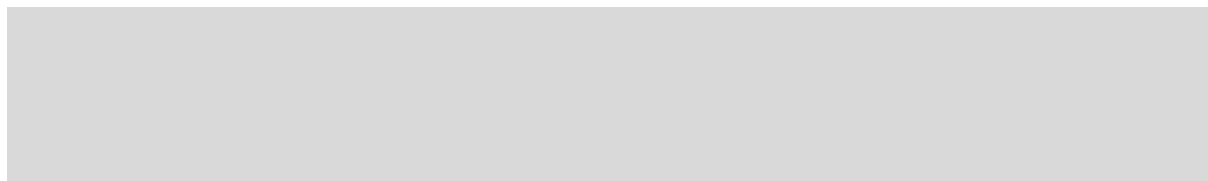
Belpbergstrasse 34b, 3110 Münsingen BE

Eingereicht am: 26.06.2021

Bewilligt:  Ja /  Nein

Datum: \_\_\_\_\_ / Unterschrift:

Begründung:



## Management Summary

### Themenbeschreibung

Ein enger Freund von mir, der Teilhaber von Neo Tropical Catering & Events ist, fragte mich, ob ich ein Tool programmieren kann. Die Firma verwendet aktuell eine Software, mit der sie nicht zufrieden ist. Das von mir zu entwickelnde Tool soll diese Software ersetzen. Es wird ein Bestellsystem für die Kitas sein. In dem Bestellsystem werden verschiedene Angaben getätigt, damit die Firma Neo Tropical einfacher und schneller das Essen zubereiten kann.

Zu Beginn war ich skeptisch, doch mit der Zeit entdeckte ich, dass die Idee perfekt für meine Diplomarbeit ist, da ich selbstständig ein reales Problem lösen kann. Da das Bestellsystem zu aufwendig ist für die Diplomarbeit, wird nur ein Prototyp vom Bestellsystem erstellt.

### Kunde

Ich entwickle ein Tool für die Firma Neo Tropical Catering & Events. Diese Firma befindet sich in Münsingen im Kanton Bern und wurde 2010 ins Leben gerufen. Sie kochen täglich frische und gesunde Gerichte für Kindertagesstätten und Tagesschulen.

### Erfolgskriterien

Der Prototyp muss mindestens folgende Informationen verwalten können:

- ◆ Monatlich oder wöchentlich Essensbestellungen
- ◆ Gut ersichtliche Mengenangaben bei den Menü Bestellungen
- ◆ Monatlich änderbarer Essensplan
  - Nur Fleischgericht ersichtlich
- ◆ Filter für besondere Ernährungskriterien
  - Vegetarisch
  - Vegan
  - Glutenfrei
  - Laktosefrei
  - Ohne Eier
  - Erdnüsse
  - Besonderes
- ◆ Login
  - Kita Login
  - Neo Tropical Login

## Inhaltsverzeichnis

1	Zweck des Dokuments .....	4
2	Ausgangslage .....	4
3	Aufgabenstellung .....	4
4	IST-Zustand .....	4
5	Abgrenzung.....	4
	5.1 Generelle Abgrenzung der Arbeit .....	4
	5.2 Geleistete Vorarbeiten .....	4
6	Rahmenbedingungen .....	5
7	Ziele und Anforderungen .....	5
	7.1 Ziele.....	5
	7.2 Funktionale Anforderungen .....	6
	7.3 Nicht-funktionale Anforderungen .....	7
8	Projektmanagement.....	8
	8.1 Vorgehensmethodik .....	8
	8.2 Grobe Zeitplanung .....	8
9	Nutzen.....	8
10	Mittelbedarf .....	8
11	Erfolgskriterien .....	9
	Anhang .....	10

## 1 Zweck des Dokuments

Durch die Diplomarbeit erhalte ich die Möglichkeit, eine selbst gewählte Arbeit auszuführen und zu zeigen, dass ich in der Lage bin ein Thema, eine Fragestellung oder ein Problem selbstständig anzugehen und entsprechende Lösungen zu erarbeiten. Die Diplomarbeit muss als Informatikprojekt durchgeführt werden und das zu erstellende Programm muss einen beurteilbaren Nutzen bringen.

## 2 Ausgangslage

Zu entwickeln ist eine Applikation, welche in vereinfachter Weise erlaubt, die Essensplanung der Kitas zu erleichtern. Dabei sollen sowohl die Kitadaten als auch die effektiven Angaben über die Bestellung hinterlegt werden können. Die Applikation soll einerseits den Köchen einen besseren Überblick über die Essenbestellungen erlauben, andererseits auch die Grundlage für ein effizientes Buchhaltungssystem liefern. Die Kitaangestellten sollen in der Lage sein, Bestellungen in einem Archiv direkt in der Applikation zu sehen.

## 3 Aufgabenstellung

Der Essensplan wird immer vor Beginn des nächsten Monats manuell geändert. Im Essensplan werden nur die Fleischgerichte aufgelistet. Die Essenbestellungen müssen entweder wöchentlich oder monatlich von den Kitas geschickt werden, sonst ist keine Bestellung möglich. Anfang der Woche wird ein Dokument für jede einzelne Kita erstellt. In diesem Dokument befinden sich die Mengenangaben zu den einzelnen Tagen mit gut ersichtlichen Ernährungskriterien. Die Firma Neo Tropical liefert nur von Montag bis Freitag aus.

## 4 IST-Zustand

Die Firma Neo Tropical schickt ihren Menüplan jeder Kita, welche Kunde ist. Die Kitas legen ein Blatt in die Essensboxen, welche die Firma Neo Tropical jeden Tag zu den Kitas bringt. Darauf steht der Menü Beststellungsplan der nächsten Woche oder des nächsten Monats.

## 5 Abgrenzung

### 5.1 Generelle Abgrenzung der Arbeit

Von der zu erstellenden Applikation wird nur ein Prototyp erstellt. In dem Prototyp wird keine Historie erstellt werden. Der Prototyp kann eine Stand Alone Applikation sein und es müssen keine Sicherheitsaspekte beachtet werden.

### 5.2 Geleistete Vorarbeiten

Erstes Zusammentreffen mit Ueli Enggist in der Catering Firma. Bei dem Zusammentreffen wurde mir physisch gezeigt, wie sie in der Firma arbeiten. Es wurde mir erklärt, wie die Firma aktuell die Menüs plant. Mir wurde auch der Menüplan, den die Kita's bekommen, gezeigt sowie der Menü Beststellungsplan.

## 6 Rahmenbedingungen

Für die Diplomarbeit bestehen folgende Rahmenbedingungen, die von der TEKO bestimmt worden sind:

Rahmenbedingungen Administrativ der Projektarbeit

- Starttermin: Freitag, 17. September 2021
- Abgabetermin: Montag, 1. November 2021
- Dauer der Arbeit: ca. 6 Wochen
- Präsentationstermin: Freitag/Samstag, 12./13. November 2021

Rahmenbedingungen des Projektes

- Die Diplomarbeit ist eine Einzelarbeit und wird als Informatikprojekt durchgeführt.
- Die Diplomarbeit wird mit einem Textverarbeitungsprogramm im Format A4 verfasst, einseitig beschrieben, fortlaufend nummeriert, gebunden oder in einem Ordner.
- Ca. 150 – 250 Stunden Arbeitsaufwand
- Die Arbeit umfasst eine schriftliche Dokumentation.

## 7 Ziele und Anforderungen

### 7.1 Ziele

<b>Nr.</b>	<b>Ziele</b>	<b>Wichtigkeit</b>
1	Die Kitas können ihre Essensbestellung ausfüllen.	Muss
2	Durchführen einer Stakeholder Analyse.	Muss
3	Erstellen von einem Qualitätsmanagement.	Muss
4	Erstellen von einem Konfigurationsmanagement.	Muss
5	Planung von einem Endprotokoll, dass die Firma Neo Tropical bekommt von den Kittas.	Muss
6	Die Firma Neo Tropical kann jeden Monat den Essensplan anpassen.	Muss
7	Die Kitas können bei den Bestellungen pro Portion Ernährungskriterien auswählen.	Muss
8	Die Firma Neo Tropical kann selbständig neue Kitas registrieren.	Muss
9	Die Dokumentation ist in elektronischer Form abzugeben.	Muss
10	Am Ende der Woche wird ein Endprotokoll in Excel erstellt mit allen Kitas der nächsten Woche.	Kann
11	Alle Bestellungen der jeweiligen Kita sind in einem Archiv abgelegt.	Kann

## 7.2 Funktionale Anforderungen

<b>Nr.</b>	<b>Anforderungen</b>	<b>Wichtigkeit</b>
1	Kitas können sich einloggen.	Muss
2	Neo Tropical können sich einloggen.	Muss
2	Kitas können Essensbestellungen einsenden.	Muss
3	Anpassbarer Essensplan.	Muss
4	Kitas können sich nicht selbständig registrieren, nur Neo Tropical kann eine neue Kita registrieren.	Muss
5	Die Kitas können bei den Bestellungen pro Portion Ernährungskriterien auswählen: <ul style="list-style-type: none"> <li>➤ Vegetarisch</li> <li>➤ Vegan</li> <li>➤ Glutenfrei</li> <li>➤ Laktosefrei</li> <li>➤ Ohne Eier</li> <li>➤ Erdnüsse</li> <li>➤ Besonderes</li> </ul>	Muss
6	Ein Endprotokoll wird in Excel erstellt.	Kann
7	Bestellungen werden in einem Archiv gespeichert.	Kann
8	Kitas müssen ihre Essensbestellungen eine Woche im Voraus abgeschickt haben.	Kann

### 7.3 Nicht-funktionale Anforderungen

<b>Nr.</b>	<b>Anforderungen</b>	<b>Wichtigkeit</b>
1	Ein elektronisches Exemplar in einer PDF-Datei an das TEKO-Sekretariat via Upload im Extranet.	Muss
2	In der Dokumentation befindet sich die gesamte Projektplanung mit dem entsprechenden Controlling.	Muss
3	In der Dokumentation befindet sich ein Qualitätsmanagement.	Muss
4	In der Dokumentation befindet sich Konfigurationsmanagement in einer sinnvollen Granularität.	Muss
5	Die Analyse und das Design werden mittels UML vorgenommen und in der Dokumentation dokumentiert.	Muss
6	Die Diplomarbeit wird in einem Textverarbeitungsprogramm im Format A4, einseitig beschrieben, fortlaufend nummeriert.	Muss
7	Ca. 150 – 250 Stunden Arbeitsaufwand.	Muss
8	Die Arbeit erfasst eine schriftliche Dokumentation.	Muss
9	<p>Gliederung der Diplomarbeit</p> <ul style="list-style-type: none"> <li>➤ Deckblatt</li> <li>➤ Inhaltsverzeichnis</li> <li>➤ Management Summary</li> <li>➤ Kurzer beruflicher Lebenslauf</li> <li>➤ Aufgabenstellung / Pflichtenheft / Zieldefinition</li> <li>➤ Terminplan</li> <li>➤ Lösung der Aufgabe</li> <li>➤ Vollständige Quellenangaben und Literaturverzeichnis</li> <li>➤ Lessons learned</li> <li>➤ Persönliches Schlusswort</li> </ul>	Muss

## 8 Projektmanagement

### 8.1 Vorgehensmethodik

Ich werde bei der Diplomarbeit auf ein **5 Phasenmodell** setzen. Die erste Phase ist die Initialisierungsphase, die vor dem Diplomarbeitsantrag schon am Laufen sein muss. In der Diplomarbeit werden drei Phasen vom 5 Phasenmodell verwendet. Zu Beginn der Diplomarbeit kommt die Voranalyse danach die Konzeptphase und am Schluss die Realisierung und das Testen. Die fünfte Phase ist die Einführungsphase, welche erst nach der Diplomarbeit beginnt.

### 8.2 Grobe Zeitplanung

Start der Diplomarbeit:	17. September 2021
Voranalyse fertig Meilenstein:	28. September 2021
Konzept fertig Meilenstein:	10. Oktober 2021
Realisieren fertig Meilenstein:	28. Oktober 2021
Abgabe der Diplomarbeit:	1. November 2021
Präsentationen:	12./13. November 2021

## 9 Nutzen

Durch ein effizientes Bestellungstool benötigt die Firma Neo Tropical weniger Zeit für Arbeiten, welche das Tool übernimmt, wie zum Beispiel, um alle Essensbestellungen abzuarbeiten, da die Essensbestellungen für die nächste Woche in einem Excel-Dokument dargestellt werden können. Oder der Essensplan muss nicht jeden Monat separat zu jeder Kita gesendet werden.

Alle eingegangenen Essensbestellungen werden in einem Archiv gespeichert und können wieder angeschaut werden. Durch ein Archiv kann die Firma einfacher dokumentieren, wenn Sie was gekocht haben und wie viel.

Mit einem Bestellungstool ist Neo Tropical immer abgesichert, falls eine Kita reklamiert, dass Neo Tropical zu wenig Bestellungen oder das Falsche geliefert hat.

## 10 Mittelbedarf

- HP 250 G6 Intel Core i5-7200U, 8GB, 256GB
- Visual Studio 2019
- C#
- Microsoft SQL Server Management Studio 18
- ConceptDraw 7 Professional
- Microsoft Office 365

## 11 Erfolgskriterien

Der Prototyp muss mindestens folgende Informationen verwalten können:

- ◆ Monatlich oder wöchentlich Essensbestellungen
- ◆ Gut ersichtliche Mengenangaben bei den Menü Bestellungen
- ◆ Monatlich änderbarer Essensplan
  - Nur Fleischgericht ersichtlich
- ◆ Filter für besondere Ernährungskriterien
  - Vegetarisch
  - Vegan
  - Glutenfrei
  - Laktosefrei
  - Ohne Eier
  - Erdnüsse
  - Besonderes
- ◆ Login
  - Kita Login
  - Neo Tropical Login

## Anhang

- Essensplan
- Vierwochenmenüplan

**NEO TROPICAL**  
**CATERING & EVENTS**

**MENÜ BESTELLUNG**

KITA: **Vela**

MONAT: **Mai**

JAHR: **2021**

DATUM	ERWACHSENE	KINDER	VEGI	LAKTOSEFREI	GLUTENFREI	ALERGIE
03 Montag	1 N / 3 V	11				
04 Dienstag	3 V	11	1			
05 Mittwoch	1 N / 2 V	11				
06 Donnerstag	1 N / 2 V	11	1			
07 Freitag	1 N / 3 V	11				

DATUM	ERWACHSENE	KINDER	VEGI	LAKTOSEFREI	GLUTENFREI	ALERGIE
10 Montag	1 N / 3 V	11				
11 Dienstag	3 V	11	1			
12 Mittwoch	1 N / 2 V	11				
<del>13 Donnerstag</del>	<del>1 N / 2 V</del>	<del>11</del>	<del>1</del>			
<del>14 Freitag</del>	<del>1 N / 3 V</del>	<del>11</del>				

DATUM	ERWACHSENE	KINDER	VEGI	LAKTOSEFREI	GLUTENFREI	ALERGIE
17 Montag	1 N / 3 V	11				
18 Dienstag	3 V	11	1			
19 Mittwoch	1 N / 2 V	11				
20 Donnerstag	1 N / 2 V	11	1			
21 Freitag	1 N / 3 V	11				


DATUM	ERWACHSENE	KINDER	VEGI	LAKTOSEFREI	GLUTENFREI	ALERGIE
<del>24 Montag</del>	<del>1 N / 3 V</del>	<del>11</del>				
25 Dienstag	3 V	11	1			
26 Mittwoch	1 N / 2 V	11				
27 Donnerstag	1 N / 2 V	11	1			
28 Freitag	1 N / 3 V	11				

# NEO TROPICAL

## CATERING & EVENTS

### Kita Sputnik

### Vierwochenmenüplan 26. April bis 21. Mai 2021

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
<b>KW 17</b> <b>26-30.4</b> 	Curry Bratkügel (Poulet) Serviert mit Ebly Brokkoli-Blumenkohl Knackiger Grüner Salat	Vollkorn Spaghetti Serviert mit Tomatensauce Geriebener Käse Knackiger Grüner Salat Gemüse Dip	Kartoffelgratin mit Hackfleisch (Rindfleisch) Serviert mit Salatplatte Knackiger Grüner Salat	Gemüsekuchen Serviert mit Knackiger Gemischter Salat	Äplermagnonen Serviert mit Apfelmus Knackiger Grüner Salat Gemüse Dip
<b>KW 18</b> <b>03-07.5</b>	Kichererbsen-Gemüse ✓ Eintopf Serviert mit Nudel ✓ Knackiger Gemischter Salat	Riz Casimir mit Poulet ✓ und Früchte Serviert mit Reis ✓ Knackiger Grüner Salat Randen Salat	Fusilli ✓ Serviert mit Pesto ✓ Geriebener Käse ✓ Knackiger Grüner Salat Gemüseplatte	Hackfleischbällchen mit ✓ Tomatensauce (Rindfleisch) Kartoffelstock Grüner Bohnen ✓ Knackiger Grüner Salat	Gnocchi Gemüse Pfanne Serviert mit cinque Pi Sauce Geriebener Käse Knackiger Grüner Salat
<b>KW 19</b> <b>10-14.5</b>	Hamburger (Rindfleisch) Serviert mit Bratkartoffel Karotten-Erbsen Knackiger Grüner Salat	Spinat-Ricotta Cannelloni Serviert mit Knackiger Gemischter Salat	Pouletschnitzel Serviert mit Parmesan-Risotto Zucchini Knackiger Grüner Salat	Vollkorn Pasta Serviert mit Brokkoli- Rahmsauce Geriebener Käse Knackiger Grüner Salat Gemüse Dip	Tofu Piccata Serviert mit Gemüsereis Rahmspinat Knackiger Grüner Salat
<b>KW 20</b> <b>17-21.5</b>	Penne Serviert mit Käsesauce Geriebener Käse Knackiger Grüner Salat Gemüse Dip	Hackfleisch mit Gemüse (Rindfleisch) Serviert mit Bulgur Knackiger Grüner Salat Maissalat	Mini Pizza Rösti Serviert mit Salatplatte Knackiger Grüner Salat	Gemüseschnitzel Serviert mit Bohnen-Reis Knackiger Grüner Salat Hüttenkäse	Lasagne Bolognese (Rindfleisch) Serviert mit Geriebener Käse Knackiger Grüner Salat Gemüse Dip

**Guten Appetit**

**Fischdeklaration** nur MSC-Fang

**Fleischdeklaration** Rinds und Poulet aus der Schweiz.

Leomar de Oliveira - Belpbergstrasse 34b - CH- 3110 Münsingen - Tel. +41 (0) 79 267 75 63 [neotropical@bluewin.ch](mailto:neotropical@bluewin.ch)