IPA Bericht

Für internen Gebrauch

autor(en) : Bosshard Fabrice CCT-FFF-XXXX

version : 1.0

status : Draft

quelle : Atos

dokumentendatum : 18 Oktober 2018­

­

­

anzahl der seiten : 46

Änderungshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Beschreibung | Autor(en) |
| 0.1 | 30.10.2018 | Initiale Dokumentationsstruktur erstellt. | Fabrice Bosshard |
| 0.2 | 31.10.2018 | Kapitel «Informieren» abgeschlossen | Fabrice Bosshard |
| 0.3 | 01.11.2018 | Kapitel «Planen» abgeschlossen | Fabrice Bosshard |

Tabelle 1: Änderungshistorie

© Copyright 2018, Atos AG Alle Rechte vorbehalten. Reproduktion von Teilen oder dem Gesamten ist ohne vorherige schriftliche Genehmigung des Urhebers untersagt. Für Fragen oder Anmerkungen zu diesem Dokument wenden Sie sich bitte an Atos, 078 620 00 35.

­­­Inhalt

**Teil 1: Umfeld und Ablauf**

Aufgabenstellung 4

Projekt 4

Ausgangslage 4

Detaillierte Aufgabenstellung 4

Management Summary 4

Applikationsstruktur 5

Technologie-Stack 5

Firmenstandards 5

Anforderungen an die Applikation 5

Vorkenntnisse 6

Vorarbeiten 6

Projektorganisation 7

Beteiligte Personen 7

Projektmanagementmethode 7

Backup-Konzept 7

Zeitplanung 8

Aufwandschätzung 8

Meilensteine 9

Massnahmen bei Verzug 9

Gantt-Diagramm 11

Arbeitsprotokoll 12

Mittwoch, 31. Oktober 2018 12

Donnerstag, 01. November 2018 14

Freitag, 02. November 2018 15

Mittwoch, 07. November 2018 16

Freitag, 09. November 2018 17

Kurzfassung 18

Informieren 19

Ist-Analyse 19

Soll-Analyse 19

Systemübersicht 24

Planen 25

Use Cases 25

Testkonzept 27

Software-Architektur 30

Datenbank-Architektur 30

Logging-Konzept 30

Entscheiden 31

Tools und Frameworks 31

Datenbank-Implementation 31

Realisierung 32

Kontrollieren 33

Auswerten 34

Glossar 35

Verzeichnisse 36

Quellenverzeichnis 36

Abbildungsverzeichnis 36

Tabellenverzeichnis 36

Aufgabenstellung 37

Programmcode 46

# Aufgabenstellung

**Aufgabenstellung Modul 223 nach PkOrg**. Die Aufgabenstellung wird vom Auftraggeber festgelegt und ist anbei dokumentiert.

## ****Projekt****

Projekttitel: Just Muesli

Im Auftrag von: Remo Steinmann Modul 223

Auftragnehmer: Fabrice-Ronny Bosshard

Starttermin: 31. Oktober 2018

Abgabetermin: 09. November 2018

Geplanter Projektaufwand: 5 Arbeitstage à 6 Stunden (30h)

Erstellung einer Multi-User-Applikation, mit Frontend, Backend und Anbindung an eine relationale Datenbank.

## Ausgangslage

Die Aufgabenstellung existiert bereits aus den Informatik-Schweizermeisterschaften vom Jahr 2018 (ICT Skills 2018). Sie wurde für dieses Modul angepasst, um den Anforderungen an die Applikation zu entsprechen (siehe 0 Anforderungen an die Applikation).

Das Projekt wird von Grund auf aufgebaut und implementiert. Der Auftragnehmer muss neben den Standard .NET-Kenntnissen auch solide Kenntnisse in Entity Framework und WPF (Windows Presentation Foundation) haben.

Die Struktur der Dokumentation soll bereits als Vorlage existieren, damit bei Projektbeginn direkt mit dem Schreiben angefangen werden kann.

## Detaillierte Aufgabenstellung

Im Anhang finden sie die detaillierte Aufgabenstellung der ICT-Skills Schweiz. Sie beschreibt die funktionalen Anforderungen an die Applikation und gibt durch Wire Frames vor, wie das Layout der Benutzeroberfläche aussehen muss. Es existieren neben Benutzerfreundlichkeit keine weiteren Kriterien für das Styling der Applikation.

## Management Summary

Kurze Repräsentation der Aufgabenstellung von ICT-Skills Schweiz.

„Just Muesli“ ist eine neu gegründete Firma welche sich als Ziel gesetzt hat, allen Kunden ein perfektes, konfigurierbares Muesli bieten zu können. Die Kunden können sich selber, aus einer Auswahl von Zutaten, Muesli online erstellen. So wird ermöglicht, dass man unzählige einzigartigen Muesli für jeden Benutzer erstellen kann.

## Applikationsstruktur

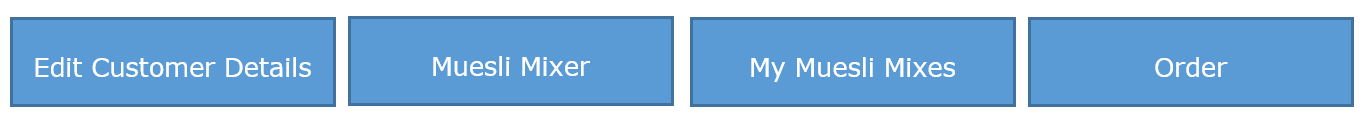
Das folgende Diagramm zeigt die verschiedenen Teile der Applikation. Jeder Knoten repräsentiert ein Fenster, welche jeweils im Anhang (Aufgabenstellung ICT-Skills) detailliert beschrieben werden.

Abbildung 1: Applikationstruktur

Menu

## ****Technologie-Stack****

IDE: Visual Studio 2017 (mit ReSharper)

Database: SQLite mit Entity Framework

Front- / Backend: C# 7.0 mit Windows Presentation Foundation (WPF)

Testing: NUnit

Versionsverwaltung: Git mit TortoiseGit

Programmarchitektur wird nach MVVM (Model View ViewModel) umgesetzt.

## Firmenstandards

Für Qualitätssicherheit und Vulnerabilitätsüberprüfung wird SonarQube verwendet.

Als Dokumentationsstandard gibt es bei der Atos ein spezielles Template für Reports. Dieses Template gibt Layout, Schriftarten, Farben, Kopf- / Fußzeilen und Absätze vor.

## Anforderungen an die Applikation

Folgende Anforderungen müssen von der Applikation erfüllt werden:

* Multi-User-Applikation
* Relationale Datenbank
* Objektorientierte Programmierung
* Mehrere Clients müssen gleichzeitig auf den gleichen Datenbestand zugreifen
* Transaktionssicherheit muss gewährleistet werden
* Zentrale Datenbank

### ****Zusätzliche Kriterien****

Während der Arbeit müssen weiterhin folgende Kriterien beachtet werden:

* 225 Versionsverwaltung mit Verwaltungs-SW
* 235 Entwurf mit UML
* 166 Codingstyle – lesbarer Code
* 250 Schichtentrennung
* 130 Vollständiges ERM bzw. Datenmodell
* 125 Gliederung des Programms
* 164 Codierung: Fehlerbehandlung

## Vorkenntnisse

In diesem Abschnitt werden die bereits erlangten Kenntnisse aufgelistet.

Projektkenntnisse:

* C# (2 Jahre)
* WPF (2 Jahre)
* Visual Studio (2 Jahre)
* Angular *TypeScript* (1 Jahr)
* Git (2 Jahre)
* CI (1 Jahr)
* DI (1 Jahr)

## Vorarbeiten

In der IPA muss eine vorgegebene Arbeit umgesetzt und dokumentiert werden. Oftmals sprengen alle Tätigkeiten in dieser Arbeit den Zeitrahmen und deshalb werden gewisse Arbeiten bereits zuvor erledigt. In diesem Kapitel werden alle Vorbereitungen für die Probe-IPA deklariert.

* Initiale Dokumentationsstruktur aufbauen nach IPERKA.
  + Struktur
  + Layout
  + Verzeichnisse
* Wissen für Entity Framework aufgebaut.
* Transaktionssicherheit implementiert in Lernprojekt

# Projektorganisation

Dieses Kapitel gibt einen Überblick über den Aufbau der Organisation, informiert über die beteiligten Personen und zeigt die Vorgehensweise im Allgemeinen.

## Beteiligte Personen

|  |  |  |  |
| --- | --- | --- | --- |
| **Person** | **Rolle** | **Verantwortung** | **Kontakt** |
| Torben Dziuk | Auftraggeber & Verantwortliche Fachkraft | Aufgabe definieren & Unterstützung Kandidat | torben.dziuk@atos.net |
| Remo Steinmann | Hauptexperte | Rahmenbedingungen setzen & Bewertung der IPA | remo.steinmann@siemens.com |
| Fabrice Bosshard | Kandidat | Umsetzung der Arbeit | fabrice.bosshard.external@atos.net  fabrice.bosshard@siemens.com |

Tabelle 2: Beteilgte Personen

## Projektmanagementmethode

Im Betrieb arbeiten wir mit der Projektmanagementmethode Scrum, welche sich sehr gut für die agile Prozessentwicklung eignet. Bei Scrum werden die verschiedenen Arbeitsschritte in Zyklen (Sprints) abgehandelt, was das Anpassen von Vorgaben zwischen den Zyklen und die transparente Kommunikation innerhalb des Entwicklerteams ermöglicht.

Scrum ist jedoch für eine IPA nicht gut geeignet, da der Mehraufwand für die Formalitäten (Daily, Retro etc.) zu hoch wäre.

Ich habe mich deshalb für die Projektmanagementmethode IPERKA (**I**nformieren, **P**lanen, **E**ntscheiden, **R**ealisieren, **K**ontrollieren, **A**uswerten) entschieden, da ich damit schon viel Erfahrungen in der Schule sammeln konnte. Diese klassische Projektmanagementmethode eignet sich für die IPA perfekt, da der Ablauf respektive die Struktur mit den Anforderungen der Arbeit übereinstimmt.

## Backup-Konzept

Bei der Umsetzung der Arbeit ist es essentiell, dass man stets auf eine valide Version der Arbeit zurückgreifen kann. Wenn man zum Beispiel einen groben Fehler macht und die ganze Applikation nicht mehr zum Laufen kriegt, ist es wichtig, dass man nicht von Neuem beginnen muss, sondern auf den Stand vor dem Fehler zurückgreifen kann.

Die Dokumentation und der Quellcode werden in ein Git-Repository geladen und sind somit zentral in einer Cloud gespeichert. Das Repository muss stets aktualisiert werden, damit möglichst immer der neuste Stand verfügbar ist. Um dies zu ermöglichen, sollte für jede neue Funktionalität ein Commit erstellt werden, welcher den neuen Stand darstellt (Daumenregel: 2-4 Mal am Tag).

Als weitere Absicherung wird ein USB-Stick genutzt, auf welchem mindestens ein Mal am Tag ein komplettes Backup erstellt wird. Im Ernstfall würde man als erstes versuchen auf das Git-Repository zurückzugreifen, doch wenn dies ebenfalls fehlschlagen sollte, gibt es noch eine Sicherung auf dem Stick.

# Zeitplanung

In diesem Kapitel werden die verschiedenen Aufwände geschätzt und aufgelistet. Die Zeitplanung dient als Leitfaden für den Projektablauf und hilft bei der Selbstkontrolle.

## Aufwandschätzung

Bei der Aufwandschätzung, oder auch Kostenrechnung, werden die gröberen Tätigkeiten zeitlich geschätzt und übereinandergestellt. Somit sieht man einen ungefähren Totalaufwand für die Arbeit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Tätigkeit | Soll-Zeit  (In Stunden) | | Ist-Zeit  (In Stunden) | |
| Informieren | Projektorganisation definieren | 0.75 | | 0.5 | |
| Aufwandschätzung und Meilensteine definieren | 1 | | 1 | |
| Ist- / Sollanalyse beschreiben | 1.5 | | 2 | |
| Systemübersicht definieren | 0.75 | | 0.75 | |
| Planen | Use Cases erstellen | 1.5 | | 2 | |
| Testkonzept erstellen | 1 | | 1 | |
| Software-Architektur beschreiben | 1.5 | |  | |
| Datenbank-Architektur beschreiben | 1 | |  | |
| Logging-Konzept | 1 | |  | |
| Entscheiden | Tools und Framework entscheiden und dokumentieren | 0.75 | |  | |
| Datenbank-Implementation dokumentieren | 0.75 | |  | |
| Realisieren | Umgebung aufsetzen | 0.5 | |  | |
| Verknüpfung zu SonarQube | 0.5 | |  | |
| Logger aufsetzen | 1 | |  | |
| Datenbankstruktur erstellen | 2 | |  | |
| Applikationsstruktur erstellen | 3 | |  | |
| Benutzer-Authentifikation implementieren | 2 | |  | |
| Business-Logik mit View und DatabaseClient verknüpfen | 4 | |  | |
| Kontrollieren | Test durchführen | 1 | |  | |
| Testprotokoll & Fazit erstellen | 1.25 | |  | |
| Auswerten | Relfexion & Fazit schreiben | 1.5 | |  | |
| Erweiterbarkeit dokumentieren | 0.5 | |  | |
| Kurzfassung schreiben | 1.5 | |  | |
| Total | | | 30.25 h  (30 Stunden 15 Minuten) | |  | |

Tabelle 3: Aufwandschätzung

## Meilensteine

Meilensteine sind wichtige Teilpunkte im Projektverlauf. Sie werden als Prüfpunkte verwendet und wirken sich positiv auf die Qualitätssicherung aus. Man kann somit die erledigten Ergebnisse mit den erwarteten Vorgaben abgleichen.

|  |  |  |
| --- | --- | --- |
| Meilenstein | Erfüllungskriterien | Datum |
| Projektbeginn | Kick-off der Probe IPA | 31.10.2018 08:30 Uhr |
| Informieren | Auftragsanalyse durchgeführt und dokumentiert | 31.11.2018 17:00 Uhr |
| Planen | Anwendungsfälle, Architektur, Testfälle planen, Logging-konzept geplant | 01.11.2018 17:00 Uhr |
| Entscheidung | Tools / Framework & Datenbankentscheidungen definiert | 02.11.2018 11:00 Uhr |
| Realisieren | Implementation der Arbeit & Dokumentation realisiert | 07.11.2018 17:00 Uhr |
| Kontrollieren | Die zuvor definierten Kontrollkriterien überprüft und Fazit daraus gezogen | 09.11.2018 11:00 Uhr |
| Auswerten | Reflexion der Arbeit und des Arbeitsverhalten erstellt & Kurzfassung geschrieben | 09.11.2018 16:00 Uhr |
| Projektabschluss | Projektarbeit abgeschlossen | 09.11.2018 17:00 Uhr |

Tabelle 4: Meilensteine

## Massnahmen bei Verzug

Sobald ein Meilenstein oder eine Tätigkeit nicht zur definierten Zeit erledigt werden konnte, muss eine Massnahme ergriffen werden, damit weitere wichtige Tätigkeiten nicht in Gefahr geraten. Dazu würden die kleineren Teil-Tätigkeiten, welche nicht grossen Einfluss auf den Fertigstellungsgrad haben, ignoriert und als optional für nach der Fertigstellung markiert werden.

Es werden jedoch nur Teil-Tätigkeiten aus der Realisierungs-Phase weggenommen, damit der allgemeine Projektablauf nicht manipuliert wird. Das heisst, wenn eine Massnahme ergriffen werden müsste, würde als Erstes die letzte Ansicht der Benutzeroberfläche ignoriert werden (In dem Beispiel => «Order Confirmation» würde als optional gesetzt werden (Siehe Anhang Aufgabenstellung)).

## Gantt-Diagramm



Tabelle 5: Gantt-Diagramm

# Arbeitsprotokoll

Das Arbeitsprotokoll dient zur Repräsentation des Tagesablaufs und der Fortschrittsbelegung. An jedem Arbeitstag wird das Arbeitsprotokoll erweitert und gepflegt. Es wird für jeden Tag mindestens eine Stunde für das Protokoll eingerechnet, damit die Qualität des Produkts sichergestellt werden kann. Es zeigt die geplanten Aufgaben pro Tag und stellt dar, ob es von der Tagesplanung Abweichungen gegeben hat. Zudem enthält das Protokoll jeweils eine Reflexion für jeden Tag, bei der man auf Erfolge, Misserfolge und den daraus zu schliessenden Massnahmen achtet.  
Zur Unterstützung der Reflexion wird mit Continious Integration (CI) gearbeitet. Die CI nimmt den eingecheckten Code und prüft ihn auf folgende Kriterien: Code Coverage, Check-Style, Vulnerabilities.

## Mittwoch, 31. Oktober 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (31.10.2018) | | | | |
| Phase: Informieren | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
| Aufgabenstellung mit Hauptexperte festlegen | 0.25 | 0.25 | Änderung der Aufgabenstellung wird im Arbeitsjournal und nicht in der konkreten Aufgabenstellung festgehalten |
| Gantt-Diagramm erstellen | 0.75 | 0.75 | - |
| Projektorganisation definieren | 0.75 | 0.5 | - |
| Aufwandschätzung und Meilensteine definieren | 1 | 1 | - |
| Ist- / Sollanalyse beschreiben | 1.5 | 2 | - |
| Systemübersicht definieren | 0.75 | 1 | - |
| Use Cases beginnen | 0.5 | 0 | Zeit war zu knapp / Am nächsten nachholen. |
| Arbeitsjournal schreiben | 1 | 0 | Zeit war zu knapp / Am nächsten Morgen geschrieben. |
| Total | 6.5 Stunden | 5.5 Stunden | Wir hatten an diesem Tag weniger Zeit |

Tabelle 6: Arbeitsprotokoll 31.10.2018

Heute war der erste Tag an dem wir an der IPA arbeiten konnten. Zu Beginn des Tages hatten wir noch einen Kurs über das designen von Flip-Charts. Aus diesem Grund haben wir bereits ein bisschen später mit der Arbeit angefangen. Diese fehlende Zeit wird am nächsten Tag mit Mehrarbeit kompensiert. Ich hatte mit dem Hauptexperten anschliessend eine kurze Besprechung in der wir über den Zeitplan und die Aufgabenstellung gesprochen haben. Wie ich einen Tag vor Projektbeginn gemerkt habe, kann ich nicht ganz mit der Aufgabenstellung fortfahren, welche wir definiert hatten. Ich wollte eigentlich die Datenbank mit SQLite erstellen, jedoch ist die Anbindung an das Entity Framework ein zu grosser Mehraufwand und deshalb habe ich mich für MSSQL umentschieden. Diese Entscheidung wurde mit dem Hauptexperten abgesprochen und akzeptiert.

### Erfolge

Trotz der fehlenden Zeit für den heutigen Tag, konnte ich sehr viele Arbeiten im Teil «Informieren» erledigen. Ich habe mich vor allem intensiv mit der Ist-/Sollanalyse beschäftigt und diese ausführlich dokumentiert. Es hat zwar etwas länger als geplant gedauert, dafür habe ich eine hohe Qualität sichergestellt.

Obwohl der ganze Raum relativ laut war konnte ich mich sehr gut auf meine Arbeit konzentrieren und war stets voll dabei. Ich denke ich habe trotz verschiedenen Zwischenfälle, für den heutigen Tag, das volle Potenzial mögliche herausgeholt.

### Misserfolge

Der Einstieg in die Arbeit war etwas holprig, da wir sowieso schon weniger Zeit als geplant erhielten, und ich etwas nervös wegen der Arbeit war. Deshalb war ich bei den ersten Tätigkeiten, wie Zeitplan oder Projektorganisation, ziemlich unsicher und wusste nicht, ob das richtig ist was ich mache. Auch mein Zeitplan konnte ich nicht so ausführlich machen wie ich es eigentlich wollte, da der Zeitplan (Gantt-Diagramm) eigentlich ein grobes Planungstool ist und wir nicht genügend Zeit hatten dafür hatten. Aus diesem Grund habe ich schnell das Gefühl bekommen, dass ich zu sehr vom Zeitplan abweiche. Mit diesem Stress konnte ich nicht sehr gut arbeiten und hatte auch mühe mich zu konzentrieren am Anfang. Mit der Zeit wurde mir bewusst, dass es nicht bringt mich selber zu stressen, da meine Arbeitsleistung dann nur abnimmt.

### Massnahmen

Der Tag ist nicht so ganz gelaufen, wie ich es mir erhofft habe. Ich kam in Stress und das hat mich schlussendlich nur blockiert. Ich muss für die restlichen Tage unbedingt versuchen einfach ruhig zu bleiben. Ich darf mich selber nicht stressen. Auch wenn ich mit dem Zeitplan hinterher hange, muss ich versuchen «cool» zu bleiben, um die Qualität und vor allem auch die Arbeitsmotivation nicht in den Keller zu hauen.

Ich habe die fehlende Zeit des Tages auch nicht wirklich in meinen Zeitplan mitaufgenommen und deshalb kam ich dann sehr rasch in Verzug. Ich muss versuchen eine genauere Tagesplanung zu machen, denn ich bin eigentlich ein sehr schneller Arbeiter, jedoch hatte ich mir gestern einfach zu viel vorgenommen.

### Fazit

Trotz den erlebten Missstände finde ich der erste Tag war ein Erfolg. Es ist eine Probe-IPA (zum Glück) und da dürfen gewisse Sachen schieflaufen und für mich war es halt ein Erfolg, weil ich daraus eine wichtige Erkenntnis erlangt habe. Ich habe gelernt, dass ich einfach unter Stress nicht gut arbeite und das ist etwas, dass ich in den letzten Jahren nie wirklich gehabt habe. Ich war eigentlich immer sehr ruhig und davon überzeugt, dass ich gut abschliessen werde in einem Test oder ähnlichem. Nun war es heute mal anders und ich fand es spannend zu sehen wie ich darauf reagiere. Diese Kenntnis werde ich sicherlich für die restlichen tage nutzen. Es ist für mich auch wichtig, dass ich eigentlich möglichst viele Fehler oder Fehlüberlegungen während der Probe-IPA mache, denn nur so kann ich mich auf die richtige IPA vorbereiten. Ich sehe die heutigen Fehler nicht als schlimm an, sondern konzentriere mich darauf, dass sie nicht mehr passieren und dass ich daraus lernen kann.

## Donnerstag, 01. November 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 2 (01.11.2018) | | | |
| Phase: Planen | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
| Use Cases erstellen | 1.5 | 2 |  |
| Testkonzept erstellen | 1 | 1.5 |  |
| Software-Architektur beschreiben | 1.5 | 1 |  |
| Datenbank-Architektur beschreiben | 1 | 1 |  |
| Logging-Konzept | 1 | 0.5 |  |
| Arbeitsjournal führen | 1 | 1 |  |
| Total | 7 Stunden | 7 Stunden |  |

Tabelle 7: Arbeitsprotokoll 01.11.2018

Heute war definitiv besser als Gestern. Ich konnte den ganzen Tag durcharbeiten und habe so ziemlich alles abgeschlossen, was ich mir heute vorgenommen habe. Man muss dazu auch sagen, dass ich eigentlich noch ein Gespräch mit dem Hauptexperten gehabt hätte und ich deshalb wie 45 Minuten mehr arbeiten konnte. Dieses Gespräch wird morgen nachgeholt, weshalb ich befürchte, dass es morgen etwas enger wird mit dem Zeitplan.

### Erfolge

Das Beste was ich aus dem Tag berichten kann ist, dass ich meine Massnahme des letzten Tages umgesetzt habe. Ich war heute viel bequemer unterwegs und habe direkt gemerkt, dass die Information viel flüssiger aus mir herauskommen. Gestern hatte ich Mühe, da ich durch den Stress blockiert war. Ich habe als Hilfe praktisch den ganzen Tag Musik gehört. Die Musik konnte mich sehr gut beruhigen und ich konnte mich wesentlich besser konzentrieren. Natürlich ist es auch ein Erfolg, dass ich meine Tagesplanung vollständig umsetzen konnte, jedoch muss man eben auch dazu beachten, dass ich mein Expertengespräch nicht wie geplant hatte und deshalb mehr Zeit hatte.

### Misserfolge

Das Planen ist etwas, was mir immer wieder Mühe macht. Ich brauche zum Teil etwas länger, bis ich bei solchen Themen mir eine Vorstellung geschaffen habe und deshalb musste ich dann einfach viel intensiver dokumentieren. Momentan spüre ich sehr stark meine beanspruchten Finger und hoffe das bessert sich morgen. Was mir ausserdem überhaupt nicht passt, ist der Arbeitsort. Ich kann nicht einen ganzen Tag auf einem «schäbigen» Stuhl sitzen, weil ich sehr schnell Rückenschmerzen bekomme.

### Massnahmen

Auf jeden Fall muss ich mich heute etwas ausruhen. Ich sollte ein bisschen früher ins Bett, da ich gegen Ende des Tages ziemlich Mühe hatte mich noch zu Leistungen zu erzwingen. Ich trinke eigentlich keinen Kaffee mehr, doch ich denke für diese Arbeit könnte es was nutzen.

## Fazit

Ich bin sehr zufrieden von den Leistungen, welche ich erbracht habe. Ich bin auf jeden Fall für das Expertengespräch morgen vorbereitet. Die ersten beiden Schritte von IPERKA sind nun endlich abgeschlossen und ich kann mit dem Implementieren beginnen.

## Freitag, 02. November 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (31.10.2018) | | | | |
| Phase: Informieren | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 8: Arbeitsprotokoll 02.11.2018

### Erfolge

### Misserfolge

### Code Analyse

### Massnahmen

### Fazit

## Mittwoch, 07. November 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 4 (07.11.2018) | | | |
| Phase: | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 9: Arbeitsprotokoll 07.11.2018

### Erfolge

### Misserfolge

### Code Analyse

### Massnahmen

### Fazit

## Freitag, 09. November 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 5 (09.11.2018) | | | |
| Phase: | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 10: Arbeitsprotokoll 09.11.2018

### Erfolge

### Misserfolge

### Code Analyse

### Massnahmen

### Fazit

# Kurzfassung

In der Kurzfassung wird kurz und bündig das Kapitel «Teil 2: Projekt» zusammengefasst. Es zeigt die wichtigsten Stichpunkte aus diesem Teil und gibt dem Leser einen raschen Überblick der Arbeit.

# Informieren

Das Informieren ist die erste Stufe in der gewählten Projektmethode «IPERKA». In diesem Schritt geht es vor allem um die Auftragsanalyse. Hier werden benötigte Information herausgesucht und zusammengestellt. Dabei werden Kriterien wie Umgebung, Bedingungen, Vorkenntnisse, Ressourcen und Ziel ganz genau hinterfragt. Man versucht in diesem Schritt den Auftrag ganz genau zu untersuchen, damit man bereits offene Fragen abdecken und mögliches Fehlerpotenzial aufweisen kann, um einen groben Überblick zu erhalten. Dieser Schritt dient als Grundlage für das nächste Kapitel «Planung».

## Ist-Analyse

Im Modul 223, welches von Remo Steinmann geleitet wird, muss eine IPA als Vorbereitung erstellt werden. Dazu wurden die Teilnehmer aufgeboten eine Aufgabenstellung zu definieren. Ich habe bei der Atos, explizit bei Torben Dziuk, nachgefragt, ob er eine Aufgabe für mich hätte. Nach gemeinsamen Überlegungen haben wir uns für die Aufgabe aus den ICT-Skills-Schweizermeisterschaften aus diesem Jahr entschieden.

Diese Aufgabe wurde während den Schweizermeisterschaften in einem Tag gelöst, jedoch wurde dabei nur auf die allgemeine Funktionalität geachtet und nicht wirklich auf die Qualitätsmerkmale einer Applikation. Folgende Aspekte wurden bei den Schweizermeisterschaften nicht ausführlich bis gar nicht behandelt:

* Testing
  + Unit- / Modul Tests
  + RegressionTests
* Transaktionssicherheit
* Code Quality (z.B. mit SonarQube)
* Clean Code Prinzipien

## Soll-Analyse

Die Soll-Analyse ist eine genauere Analyse der Anforderungen und zeigt den Zustand, welcher als Ziel definiert wurde. In diesem Kapitel werde ich zur Veranschaulichung die Screenshots, aus der Aufgabenstellung von ICT-Skills, als Wire-Frames für meine Benutzeroberfläche nutzen.

### Frontend

Abbildung 2: Soll-Analyse Menu

Soll-Analyse der Frontend-Implementation.

**Menu**

Das Menu ist die Startseite der Applikation und dient sozusagen als Navigation für den Benutzer. Über das Menu kann man die unten beschriebenen Funktionalitäten auswählen und deren Benutzeroberfläche öffnen.

* Edit Customer Details
* Muesli Mixer
* My Muesli Mixes
* Order (Deaktiviert, wenn keine Muesli Mixes erfasst wurden)
* Exit (Applikationsaustritt)

**Edit Customer Details**

Abbildung 3: Soll-Analyse Edit Customer Details

Diese Benutzeransicht wird für das editieren eines Kunden benutzt und kann vom Menu aus geöffnet werden. Folgende Daten müssen für den Benutzer editierbar sein: Name, Adresse, Postleitzahl, Stadt, Land, Telefonnummer und email. Dafür müssen angemessene UI-Controls implementiert werden.

Beim Klicken auf den Save-Knopf müssen die vorhandenen Einträge in der Datenbank abgespeichert werden => Bei Missachtung der Validation Regel muss eine Fehlermeldung erstellt werden, in welcher der Benutzer über die fehlerhaften Felder informiert wird.

Beim Klicken auf den «Back to menu»-Knopf wird der Benutzer zum Menu zurückgeleitet.

Damit fehlerhafte Daten in der Datenbank verhindert werden, müssen folgende Validierungen implementiert werden.

|  |  |
| --- | --- |
| Feld | Validationsregel |
| Name | Länge => mindestens 5 Zeichen |
| Adresse | Länge => mindestens 5 Zeichen |
| Postleitzahl | Länge => mindestens 4 Zeichen, nur numerische Werte |
| Stadt | Länge => mindestens 2 Zeichen |
| Land | Es muss ein Land von einer vordefinierten Liste selektiert werden. |
| Telefonnummer | Länge => mindestens 10 Zeichen, nur numerische Werte, Leerzeichen oder «+» |
| Email | Muster: [x@y.z](mailto:x@y.z)  X => mindestens 1 Zeichen  Y => mindestens 3 Zeichen  Z => 2-3 Zeichen (nur [a-zA-z]) |

Tabelle 11: Validationsregeln Customer Details

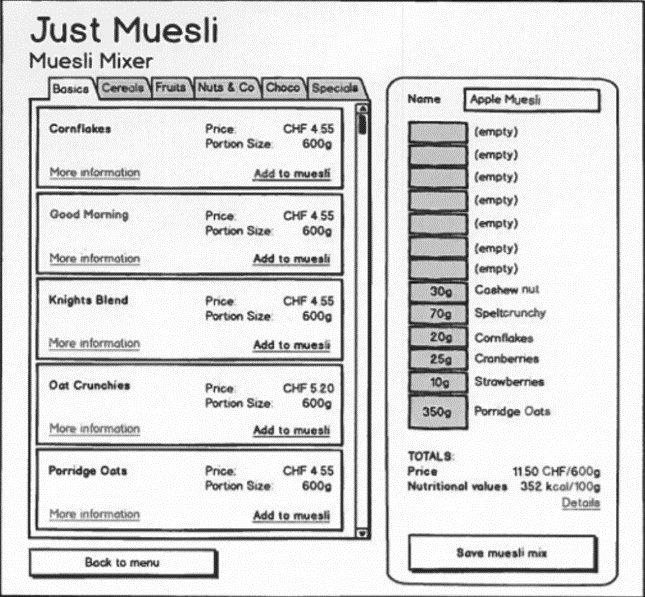
**Muesli Mixer**

Abbildung 4: Soll-Analyse Muesli Mixer

Diese Ansicht ist vom Menu aufrufbar und ist sozusagen der Kern der Applikation. Man kann auf dieser Seite sein individualisiertes Muesli erstellen.

Ein Muesli-Mix muss immer aus einer Basis-Zutat (Kategorie Basics) bestehen und kann bis zu 12 zusätzliche Zutaten beinhalten. Als erstes muss immer die Basis-Zutat ausgewählt werden und erst dann kann man aus einer der folgenden Kategorien auswählen: Cerealien, Früchte, Nüsse & Co, Schokolade und Spezialitäten.

Das Gesamtgewicht von einem Muesli ist immer 600g. Die Basis-Zutat wird als «Füller» genutzt und wird automatisch berechnet (600g – (Totalgewicht der zusätzlichen Zutaten)).

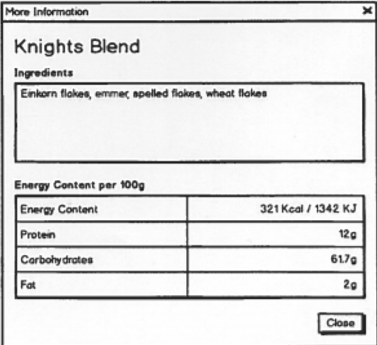
*Components list (Linke Seite von Muesli Mixer)*

Abbildung 5: Soll-Analyse Components List

Auf der linken Seite vom Muesli Mixer kann man die verschiedenen Zutaten auswählen. Es muss für jede Kategorie ein Tab erstellt werden, welcher eine Liste seiner Zutaten enthält und anzeigt. Diese Liste sollte nach dem Namen sortiert sein. In der Liste muss für jede Zutat folgendes ersichtlich sein: Name, Preis, Gewicht einer Portion.

Der Benutzer muss diese Zutaten per Klick auf «Add to Muesli» in sein Muesli hinzufügen können. Wenn bereits 12 Zutaten hinzugefügt wurden, muss eine Fehlermeldung angezeigt werden. Als Erstes muss immer eine Basis-Zutat ausgewählt werden und es kann nur eine einzige ausgewählt sein. Wenn der User eine zusätzliche Basis-Zutat hinzufügen möchte, wird die Alte ersetzt.

Beim Klick auf den «Back to menu»-Knopf wird der Benutzer zum Menu zurückgeleitet.

Beim Klick auf den «More Information»-Knopf muss ein Popup erscheinen, welches die Zutaten, Nährstoffe und den Energiegehalt einer Komponente anzeigt. Für die Berechnung des Energiegehaltes wird folgende Kalkulation benutzt:

(1 Kcal = 4.184 KJ)

|  |  |  |
| --- | --- | --- |
| Nährstoff | Kilocalorien pro gram (Kcal/g) | Kilojoules per gram (KJ/g) |
| Kohlenhydrate | 4.1 | 17.2 |
| Proteine | 4.1 | 17.2 |
| Fette | 9.3 | 38.9 |

Tabelle 12: Nährstoff-Kalkulation

*Current muesli mix list (Rechte Seite von Muesli Mixer)*

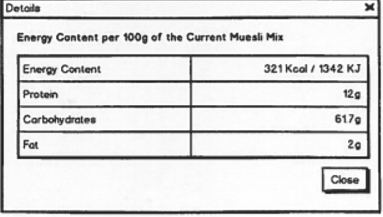
Es muss auf der rechten Seite von Muesli Mixer ein Stapel mit den gewählten Zutaten existieren. Die Basis-Zutat muss immer zuunterst sein. Es soll eine Fehlermeldung erscheinen, wenn das gesamt Muesli schwerer als 600g ist. Der Gesamtpreis des Muesli (In CHF pro 600g) und dessen Nährstoffwerte (In kcal pro 100g) müssen kalkuliert und angezeigt werden. Für diese Berechnung und Anzeige muss folgendes beachtet werden:

Abbildung 6: Soll-Analyse Current Muesli Mix list

* Der Preis muss auf die zweite Dezimalstelle gerundet sein (0.01).
* Der Benutzer kann eine Zutat entfernen, wenn er auf sie klickt. Da sollte eine Bestätigung für den Benutzer erforderlich sein (Popup - yes/no).
* Es soll für den Benutzer möglich sein, einen Namen für sein eigenes Muesli zu definieren.
* Über den «Save Muesli»-Button kann das Muesli in der Datenbank gespeichert werden.
  + Eine Fehlermeldung muss angezeigt werden, wenn folgendes nicht erfüllt wurde
    - Muesli hat einen Namen.
    - Eine Basis-Zutat ist selektiert
* Für die gesundheitsbewussten Benutzer soll es für jede Zutat eine eigene Detail-Seite geben auf der die Nährstoffgehalte angezeigt werden.

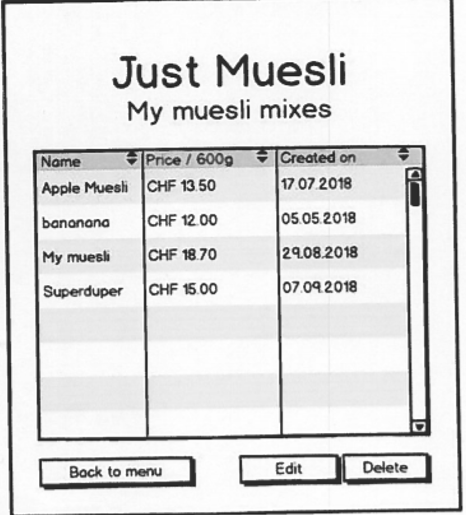
**My Muesli Mixes**

Abbildung 7: Soll-Analyse My Muesli Mixes

Auf dieser Ansicht kann der Benutzer seine eigenen Muesli-Kreationen ansehen. Diese Ansicht kann man vom Menu aus aufrufen. Alle Muesli sollen mit Name, Preis und Erstelldatum angezeigt werden. Die Tabelle soll nach dem Namen sortiert sein, doch es soll für den Benutzer möglich sein auf anderen Spalten zu sortieren. Der Edit-Knopf öffnet den Muesli Mixer mit dem ausgewählten Muesli als Inhalt. Der Benutzer kann das Muesli löschen, wenn er auf den Delete-Knopf drückt. Dabei wird eine Bestätigungsmeldung angezeigt, wo der Benutzer noch bestätigen muss, dass er das Muesli wirklich löschen möchte. Wenn Edit oder Delete geklickt wird ohne, dass ein Muesli selektiert wird, soll eine Fehlermeldung angezeigt werden. Der «Back to menu»-Knopf soll den Benutzer zurück aufs Menu leiten.

**Order**

Diese Ansicht wird genutzt, um bestehende Muesli zu bestellen. Sie ist aufrufbar über das Menu.

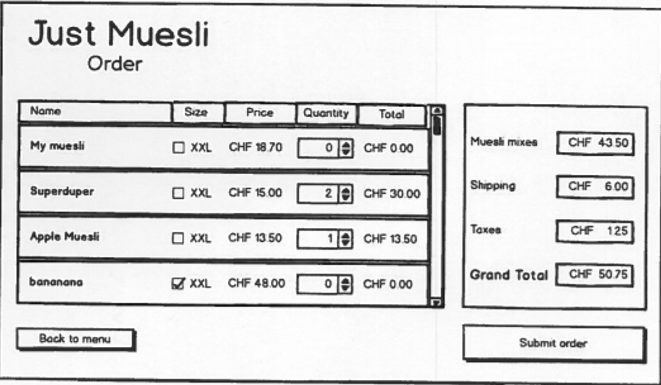


Abbildung 8: Soll-Analyse Order

M*uesli Mix list (linke Seite von My Muesli Mixes)*

Alle gespeicherten Muesli sollen mit dem Namen und Preis angezeigt (Nach Preis abwärts sortiert). Sobald ein Müesli eine Anzahl von Bestellungen (Quantity) zugewiesen bekommt, soll der Order-Knopf verfügbar sein. Der Benutzer soll die Möglichkeit haben eine XXL Version seines Muesli zu bestellen (2’400g / Preis 4x). Jeder Eintrag soll am Schluss den Totalpreis für ein Muesli anzeigen (Angezeigter Preis \* Anzahl Bestellungen). Der «Back to Menu»-Knopf leitet den Benutzer auf das Menu zurück.

*Costs (rechte Seite von My Muesli Mixes)*

Der angezeigt Bestellpreis soll aus den verschiedenen Zwischenpreisen entstehen. Die Lieferkosten innerhalb der Schweiz sind immer 6.- CHF und ausserhalb immer 8.- CHF. Wenn der Bestellpreis höher als 50.- CHF ist, sollen keine Lieferkosten angezeigt werden.  
Wenn die Bestellung ausserhalb der Schweiz liegt, sollen keine Steuern berechnet werden. Für inländische soll eine Steuer von 2.5 % (gerundet auf nächste 0.05 Stelle). Die Anzeige der Kosten muss Read-Only sein. Wenn der Benutzer auf «Submit order» klickt, soll eine Bestellungsbestätigung angezeigt werden (siehe unten).

**Order Confirmation**

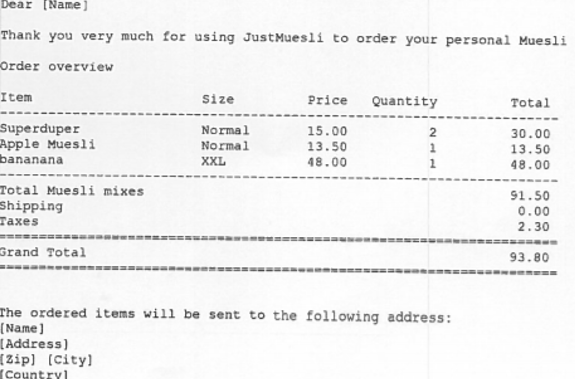
Die Bestellungsbestätigung soll als eine Textdatei generiert werden und soll folgendermassen formatiert sein

Abbildung 9: Soll-Analyse Order Confirmation

Diese Datei soll im Arbeitsverzeichnis der Applikation abgelegt werden und der Filename soll «orderconfirmation.txt» sein (Existierende Datei soll überschrieben werden). Die Platzhalter in den Klammern (z.B. [Name]) muss mit den Kundendaten ersetzt werden. Die Tabelle muss die bestellten Müslis enthalten (Name, Preis, Grösse, Anzahl und eigenes Total), Kosten der Müslis, Lieferkosten, Steuern und den Totalpreis der Bestellung. Die Spalten der Tabelle müssen folgendermassen formatier werden.

|  |  |  |  |
| --- | --- | --- | --- |
| Spalte | Grösse (Zeichenanzahl) | Format | Ausrichtung |
| Item | 25 | Text | Links |
| Size | 7 | Text | Links |
| Price | 10 | Nummer mit 2 Kommastellen | Rechts |
| Quantity | 10 | Nummer ohne Kommastellen | Rechts |
| Total | 14 | Nummer mit 2 Kommastellen | Rechts |

Tabelle 13: Bestellungsbestätigung Formatierung

Wir können dabei annehmen, dass kein Wert die definierte Grösse überschreitet.

### Backend

Alle Frontend-Schnittstellen interagieren mit der Backend-Implementation. Das Backend ist dafür zuständig die erforderlichen Daten in der Datenbank zu persistieren.

**Datenbank**

Mit dem Entity Framework wird die Datenbank erstellt, erweitert, und manipuliert. Im Hintergrund steckt eine MSSQL-Datenbank. Über die Benutzerschnittstelle wird das backend angesteuert und Entity Framework kommuniziert die gewünschten Änderungen mit der Datenbank.

## Systemübersicht

In diesem Projekt gibt es eigentlich nur zwei Systeme welche miteinander kommunizieren. Es gibt die Client-Applikation welche jeweils auf die MSSQL-Datenbank zugreift. Dazwischen sitzt das Entity Framework, welches als ORM (object-relational Mapper) fungiert und die Client-Applikation bei Datenbankabfragen unterstützt.

*Entity Framework*

**.NET Framework**

**(Client Application)**

**WPF**

**MSSQL**

**C#**

**SQL-Datenbank**

Abbildung 10: Systemübersicht

Wie man in der obigen Abbildung sehen kann, wird innerhalb des .NET-Framework mit C# und WPF gearbeitet. WPF wird für das Frontend, respektive für die Benutzeroberfläche, genutzt und C# für die Business-Logik und Daten-Aufbereitung der Benutzeroberfläche.

# Planen

Im zweiten Schritt der IPERKA-Methode werden nun die möglichen Lösungsvarianten und das Vorgehen ausgearbeitet. Es werden einzelne Arbeitsschritte geplant, um einen gesamten Arbeitsablauf zu erstellen. Im Arbeitsablauf werden Hilfsmittel und Werkzeuge ermittelt, sowie der Zeitbedarf für die Arbeitsschritte geschätzt. Zudem werden Qualitäts-Kriterien für den Schritt «Kontrollieren» gesammelt. (VSSM, 2014)

## Use Cases

In diesem Kapitel beschreibe ich die verschiedenen Anwendungsfälle anhand von Use Cases. Bei Use Cases ist es nicht nur wichtig, dass man darstellt, welche Funktionalität von wem aufgerufen werden kann, sondern auch was nach dieser Aktion passiert und ob es Vorbedingungen gibt.

In meiner Applikation gibt es eigentlich nur eine Rolle – Den Kunden. Dieser interagiert mit der ganzen Applikation. Er hat keine Einschränkungen und ist vollkommen dazu berechtigt jede Funktionalität aufzurufen.

Abbildung 11: Use Case Diagramm Kunde

|  |  |
| --- | --- |
| Use Case 1 – Kundendaten anpassen | |
| Beschreibung | Als Kunde möchte ich meine Daten anpassen können |
| Konsequenz | Man wird auf die Ansicht «Edit Customer Details» weitergeleitet und kann seine Benutzer Daten anpassen und speichern. |
| Ablauf | 1. Auf dem Menu sein. 2. Knopf «Edit Customer Details» klicken. 3. Seite «Edit Customer Details» wird angezeigt. 4. Daten anpassen. 5. Auf den Save-Knopf drücken. |

Tabelle 14: Use Case Kundendaten anpassen

|  |  |
| --- | --- |
| Use Case 2 – Muesli Mixer aufrufen | |
| Beschreibung | Als Kunde möchte ich ein individualisiertes Muesli erstellen können, wobei ich aus einer Vielzahl von Zutaten auswählen kann. |
| Konsequenz | Der Muesli Mixer wird geöffnet und man hat die Möglichkeit ein neues Muesli zu konfigurieren und abzuspeichern. Nebenbei sieht man immer die verschiedenen Nährstoff-/ und Energiegehalte der einzelnen oder des gesamten Mueslis. |
| Ablauf | 1. Auf dem Menu sein. 2. Knopf «Muesli Mixer» klicken. 3. Seite «Muesli Mixer» wird angezeigt. 4. Zutaten auswählen und hinzufügen. 5. Energie-/ und Nährstoffgehalte anzeigen lassen. 6. Muesli speichern. |

Tabelle 15: Use Case Muesli Mixer aufrufen

|  |  |
| --- | --- |
| Use Case 3 – My Muesli Mixes editieren | |
| Beschreibung | Als Kunde möchte ich die Möglichkeit haben meine bereits erstellten Müeslis zu bearbeiten. |
| Vorbedingung | Es müssen bereits Muesli erfasst worden sein. |
| Konsequenz | Das entsprechend ausgewählte Muesli wird als Inhalt im Muesli Mixer geöffnet. |
| Ablauf | 1. Auf dem Menu sein. 2. Knopf «My Muesli Mixes» klicken. 3. Seite «My Muesli Mixes» wird angezeigt. 4. Eigenes Muesli auswählen. 5. Auf den Edit-Knopf klicken. 6. Seite «Muesli Mixer» wird mit gewähltem Inhalt geöffnet. 7. Muesli bearbeiten. 8. Mit dem Save-Knopf das Muesli speichern. |

Tabelle 16: Use Case My Muesli Mixes editieren

|  |  |
| --- | --- |
| Use Case 4 – Bestellung aufgeben | |
| Beschreibung | Als Kunde möchte ich meine bereits erfassten Mueslis bestellen können. Ich möchte auswählen zwischen XXL und normal Portion und ich möchte eine Anzahl Bestellungen eingeben können. |
| Vorbedingung | Es müssen bereits Muesli erfasst worden sein. |
| Konsequenz | Die definierte Bestellung wird erfasst und eine Bestellungsbestätigung wird erstellt. |
| Ablauf | 1. Auf dem Menu sein. 2. Knopf «Order» klicken. 3. Seite «Order» wird angezeigt. 4. Anzahl Bestellungen & Grösse der Muesli definieren. 5. Auf «Submit order» klicken. |

Tabelle 17: Use Case: Bestellung aufgeben

|  |  |
| --- | --- |
| Use Case 5 – Bestellungsbestätigung einsehen | |
| Beschreibung | Als Kunde möchte ich, nachdem ich die Bestellung aufgegeben habe, meine Bestellung nochmals einsehen. |
| Vorbedingung | Es muss eine Bestellung aufgegeben worden sein. |
| Konsequenz | Bestellbestätigung wird geöffnet und angezeigt. |
| Ablauf | 1. Im Arbeitsverzeichnis der Applikation nach dem File «orderconfirmation.txt» und öffnen. 2. Bestellbestätigung einsehen. |

Tabelle 18: Use Case Bestellungsbestätigung einsehen

## Testkonzept

Das Testkonzept wird in der Planung erstellt und wird danach in der Phase «Kontrollieren» wiedergenutzt. Im Testkonzept werden alle gewünschten Funktionalitäten festgehalten, damit später sichergestellt werden, dass die Vorgaben eingehalten wurden. Um dieses Prozedere zu ermöglichen werden Testfälle erstellt.

### Unit- / Modultests

Als erste Qualitätssicherung werden verschiedene Unit- / oder auch Modultests erfasst. Diese sind automatisiert und decken entweder einzelne Funktionen oder ganze Funktionsketten (Module) ab. Da die Zeit zu knapp ist, um für Alles Test zu schreiben, wird vor allem auf Kernfunktionalität geachtet. Mit dem Tool SonarQube kann auf die Code Coverage geachtet werden. Ich versuche mit den Unit-/ Modultest eine Code Coverage von 60 % zu erreichen. Sobald die Phase «Realisieren» beginnt, wird in jedem Arbeitsjournal ein kurzer Beitrag zu den Ergebnissen von SonarQube erstellt.

Die automatisierten Tests werden vor jedem Push auf das Git-Repository ausgeführt, um sicherzustellen, dass keine Funktionalität, welche kaputtgegangen ist, auf das «Backup» kommt.

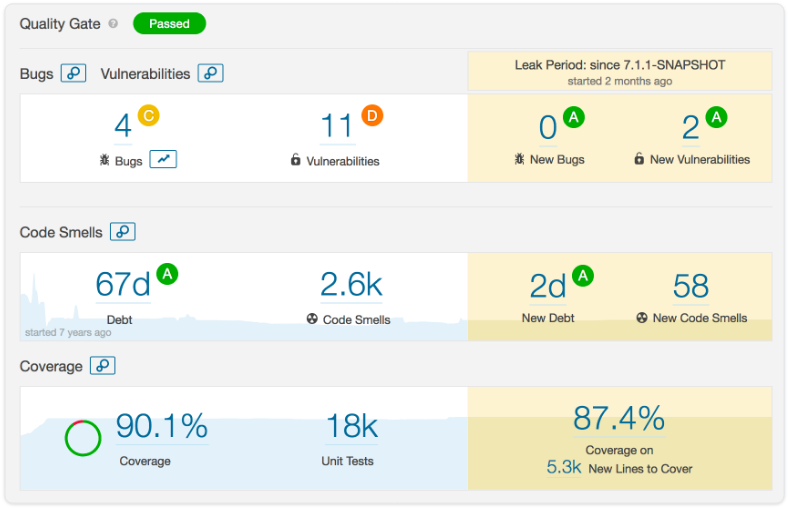


Abbildung 12: Code Coverage Beispiel SonarQube

### Regression-Tests

Im zweiten Schritt für die Qualitätssicherung werden manuelle Regressionstest. «Unter einem Regressionstest versteht man in der Softwaretechnik die Wiederholung von Testfällen, um sicherzustellen, dass Modifikationen in bereits getesteten Teilen der Software keine neuen Fehler („Regressionen“) verursachen.» (Wikipedia, Regressionstest, 2015)

### Testfälle

Anbei werden verschiedene manuelle Testfälle erstellt, welche mit den Anforderungen in den Use Cases korrelieren. Die Testfälle werden nach den Use Cases benannt (Nummer + Bezeichnung).

|  |  |  |  |
| --- | --- | --- | --- |
| (Use Case 1) Nr. | Vorbedingung | Testschritte | Erwartetes Ergebnis |
| 1.1 | Seite «Edit Customer Details» ist geöffnet. | 1. Vollständiger Kunde erfassen mit validen Datensätzen 2. Save-Knopf drücken | Neuer Kunde wird in der Datenbank gespeichert. |
| 1.2 | Siehe oben | 1. Vollständiger Kunde mit Name «hans» erfassen. 2. Save-Knopf drücken | Fehlermeldung erscheint. Feld wird als falsch markiert. |
| 1.3 | Siehe oben | 1. Vollständiger Kunde mit Adresse «hier» erfassen. 2. Save-Knopf drücken | Fehlermeldung erscheint. Feld wird als falsch markiert. |
| 1.4 | Siehe oben | 1. Vollständiger Kunde mit Postleitzahl «3e» erfassen. 2. Save-Knopf drücken | Fehlermeldung erscheint. Feld wird als falsch markiert. |
| 1.5 | Siehe oben | 1. Vollständiger Kunde mit Stadt «b» erfassen. 2. Save-Knopf drücken | Fehlermeldung erscheint. Feld wird als falsch markiert. |
| 1.6 | Siehe oben | 1. Vollständiger Kunde mit Handnummer «sdfsa34234» erfassen. 2. Save-Knopf drücken | Fehlermeldung erscheint. Feld wird als falsch markiert. |
| 1.7 | Siehe oben | 1. Vollständiger Kunde mit Email «a.a.a@d» erfassen. 2. Save-Knopf drücken | Fehlermeldung erscheint. Feld wird als falsch markiert. |

Tabelle 19: Testfälle 1

|  |  |  |  |
| --- | --- | --- | --- |
| (Use Case 2) Nr. | Vorbedingung | Testschritte | Erwartetes Ergebnis |
| 2.1 | Seite «Muesli Mixer» ist geöffnet. | 1. Muesli erfassen    1. Basiszutat    2. Zusätzliche Zutaten    3. Name 2. Auf «Save Muesli mix» klicken | Muesli wird in Datenbank gespeichert. |
| 2.2 | Siehe oben | 1. Muesli ohne Basiszutat erfassen 2. Auf «Save Muesli mix» klicken | Fehlermeldung wird angezeigt |
| 2.3 | Siehe oben | 1. Muesli mit mehr als 12 Zutaten erfassen. | Fehlermeldung wird angezeigt |
| 2.4 | Siehe oben | 1. Muesli mit mehr als 600g Gesamtgewicht erfassen. | Fehlermeldung wird angezeigt |
| 2.5 | Siehe oben | 1. Mit Muesli Mixer interagieren. | Preis & Nährstoff-/Energiegehalt wird neu berechnet und angezeigt. |
| 2.6 | Siehe oben | 1. Auf «Details» klicken. | Detail Seite wird angezeigt und zeigt Nährstoff-/Energiegehalt für das Ganze Muesli |
| 2.7 | Siehe oben | 1. Auf «More Information» klicken. | «More Information»-Seite wird angezeigt und zeigt Nährstoff-/Energiegehalt & Unterzutaten für eine zutat |
| 2.8 | Zutat muss in Muesli Mixer hinzugefügt sein | 1. Klick auf Zutat in Zutatenliste | Zutat wird aus Liste entfernt |

Tabelle 20: Testfälle 2

|  |  |  |  |
| --- | --- | --- | --- |
| (Use Case 3) Nr. | Vorbedingung | Testschritte | Erwartetes Ergebnis |
| 3.1 | Seite «My muesli mixes» muss geöffnet sein.  Ein Muesli muss bereits erstellt worden sein. | 1. Muesli auswählen 2. Auf Edit klicken | Muesli Mixer mit gewähltem Muesli als Inhalt wird geöffnet. |
| 3.2 | Siehe oben | 1. Muesli Auswählen 2. Auf Delete klicken | Muesli wird entfernt |

Tabelle 21: Testfälle 3

|  |  |  |  |
| --- | --- | --- | --- |
| (Use Case 4) Nr. | Vorbedingung | Testschritte | Erwartetes Ergebnis |
| 4.1 | Seite «Order» muss geöffnet sein.  Ein Muesli muss bereits erstellt worden sein. | 1. Muesli auswählen 2. Anzahl Bestellung auf 2 Setzen | Subtotal und Total soll angezeigt werden. Lieferkosten und Steuern sollen angezeigt werden. |
| 4.2 | Siehe oben | 1. Muesli als XXL auswählen | Preis muss um 4-faches steigen |
| 4.3 | Siehe oben | 1. Keine Anzahl für Bestellung eingeben | «Submit order»-Knopf soll deaktiviert sein |
| 4.4 | Siehe oben | 1. Muesli Anzahl auf 2 setzen 2. «Submit order» klicken | Bestellungsbestätigung wird erstellt und im Applikationsverzeichnis hinterlegt. |

Tabelle 22:Testfälle 4

|  |  |  |  |
| --- | --- | --- | --- |
| (Use Case 5) Nr. | Vorbedingung | Testschritte | Erwartetes Ergebnis |
| 5.1 | Muesli muss, wie unter 4.4 beschrieben, ausgewählt und bestellt werden | 1. Ins Applikationsverzeichnis gehen. 2. Datei «orderconfirmation.txt» öffnen | Bestellbestätigung wird angezeigt in richtiger Formatierung. |

Tabelle 23: Testfälle 5

## Software-Architektur

In diesem Kapitel werden alle Themen bezüglich der allgemeinen Software-Architektur behandelt. Es werden die geplanten Komponenten definiert, ohne sie zu konkretisieren. Ich werde die funktionalen Verantwortungen der einzelnen Komponenten genauer beschreiben, jedoch ihre konkrete Implementation offenlassen, damit ich in der Phase «Realisieren» noch Anpassungsmöglichkeiten habe.

|  |  |  |
| --- | --- | --- |
| Komponente | Beschreibung | Abhängigkeiten |
| DatabaseService | Verknüpft die Models mit der Datenbank. Auf diesem Service werden alle Datenbankabfragen aufgerufen. | Entity Framework |
| LoggingService | Dieser Service wird von praktisch überall in der Applikation genutzt. Er empfängt die Logging- Einträge und leitet sie ans Log4Net-Framework weiter. | Log4Net |
| Models | Korrespondieren mit Datenbanktabellen. Behälter für die Daten | Keine Spezialitäten |
| ViewModels | Verbindung zwischen Models und View. Datenaufbereitung für Benutzerschnittstelle | Keine Spezialitäten |
| Views | Konkrete XAML Implementation mit Anknüpfung an ViewModels | Keine Spezialitäten |
| DI-Service | Wrapper für IOC-Container. Zuständig für Dependency Injection. | Keine Spezialitäten |

### Klassendiagramm

## Datenbank-Architektur

Unter diesem Kapitel wird die Struktur der Datenbank festgelegt. Die Struktur wurde im ersten Schritt auf einem Notizblatt skizziert und wurde nach mehreren kleinen Verbesserungen in Draw.io als erstes ERD (Ohne Datentypen) abgebildet. Die Version von Draw.io wird für die Planung genutzt. Es wird jedoch nach der Datenbank-Implementation ein ERD generiert und ebenfalls unter diesem Kapitel beigefügt.

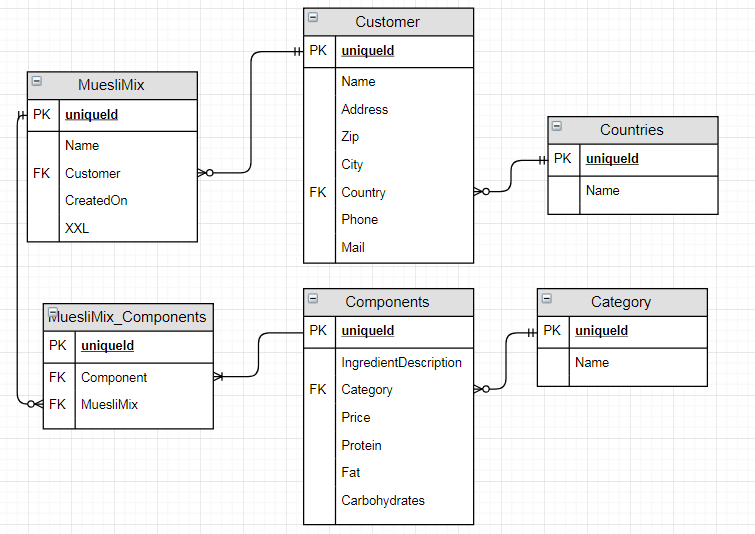


Abbildung 13: Erstes ERD mit Draw.io

### ERD

## Logging-Konzept

Für das Verständnis des Entwicklers und des Benutzers werden Fehlermeldungen oder sonstige Ereignisse in einer Log-Datei festgehalten.

Damit der Anwender dieser Log-Datei nicht ewig in unnötigen Informationen suchen muss, gibt es zwei verschiedene Log-Dateien.

|  |  |  |
| --- | --- | --- |
| Datei | Konfiguration | Inhalt |
| Error\_Log.txt | Dieses File wird im Applikationsverzeichnis abgelegt und wird für jeden Tag neu erstellt. Es kann maximal bis zu 100 MB gross werden, bevor eine weitere Datei erstellt wird. | Alle Ereignisse mit Log-Level höher als «Warn»(Error & Fatal). |
| Log.txt | Dieses File wird im Applikationsverzeichnis abgelegt und wird nur neu erzeugt, wenn die maximal Grösse von 100 MB erreicht wurde. | Beinhaltet alle Ereignisse, welche einen Log-Level tiefer als «Error» haben (Debug, Info, Warn). |

**Log-Eintrag**

Ein Log-Eintrag soll folgendermassen aufgebaut werden:

[Datum\_Zeit] [Log-Level] [Methode] [Nachricht]

# Entscheiden

Damit man mit dem dritten Schritt von IPERKA beginnen kann, müssen zuerst die Anforderungen aus dem zweiten Schritt «Planen» erfüllt sein. Nach der Planung müssen die verschiedenen Lösungsvarianten gegenübergestellt werden und eine davon ausgewählt werden. Hierbei werden verschiedene Faktoren miteinander abgeglichen und ausgewertet. (VSSM, 2014)

### Architekturkonzept

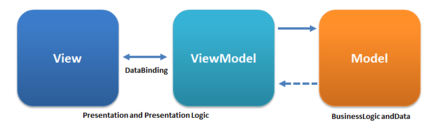
Ich arbeite seit 2 Jahren mit C# und entwickle seit je her mit dem Entwurfsmuster «MVVM» (Model View ViewModel). Anders als bei MVC (Model View Controller) trennt man die Darstellung von der Logik der Benutzerschnittstelle. Dieses Konzept zielt, unter anderem, auf WPF (Windows Presentation Foundation), da ein Datenbindungsmechanismus erforderlich ist. Wegen meiner Erfahrung in diesem Bereich und der Kompatibilität zu WPF, arbeite ich nach MVVM in meiner Applikation. (Wikipedia, MVVM, 2018)

Abbildung 14: MVVM

## Tools und Frameworks

Da ich bereits in der Aufgabenstellung meinen Technologie-Stack vorgestellt habe, gibt es nicht mehr viel was ich wirklich entscheiden kann. Anbei werde ich trotzdem noch kurz die Entscheidungen zu den verwendeten Tools und Frameworks aufzeigen.

|  |  |  |
| --- | --- | --- |
| Tool / Framework | Alternative | Begründung |
| Entity Framework | Datenbank mit vordefinierten SQLCommands erstellen und warten. | Das Entity Framework ist sozusagen ein Wrapper für alle SqlCommand, SqlReader, SqlWriter Klassen etc.  EF vereinfacht die Arbeit und spart extrem viel Zeit. |
| Log4Net | NLog | Ich arbeite seit 2 Jahren mit Log4Net. Ich kenne mich viel besser aus damit und habe Praxiserfahrung, was ich bei NLog nicht behaupten kann. |
| MSSQL | Sqlite | Ich wollte eigentlich zuerst mit Sqlite arbeiten, da aber die Sqllite-Anbindung an das EF nicht wirklich optimal ist, habe ich mich für die altbewährte MSSQL-Datenbank entschieden |

Tabelle 24: Tool & Framework Entscheide

## Datenbank-Implementation

Bei Entity Framework gibt es verschiedenen Möglichkeiten, wie man die Datenbank, respektive die Datenbankverbindung, realisiert. Die zwei meist gebrauchten Methoden sind Code-First und Database-First. Wie eigentlich schon in den beiden Namen ersichtlich, wird bei der ersten Methode aus dem Code eine Datenbank generiert und bei der zweiten wird zuerst die Datenbank erstellt und danach der Code.

Bei Code-First ist der Vorteil, dass man mehr Kontrolle über die Struktur der Datenbank hat und bei Database-First ist das Problem, dass man zum Teil Schwierigkeiten hat, die Datenbank-Struktur anzupassen und den Code neu daraus zu generieren. Database-First wird oft verwendet, wenn man mit einem Legacy-System arbeitet, da man dort meistens gewisse Sicherheitsschritte durchführen muss, was bei Code-First die Migrationen unterbinden würde. Da ich nicht mit einem Legacy-System arbeite und mir die Code-First Variante besser liegt, arbeite ich mit der Code-First-Methode.

# Realisierung

Die Realisierung und somit der vierte Schritt der IPERKA-Methode, ist die zeitintensivste Phase in den meisten Projekten. Hier werden die zuvor erfüllten Schritte von IPERKA (Planen & Entscheiden) aus der Theorie in die Praxis umgewandelt. Dabei ist es wichtig den Arbeitsablauf einzuhalten und nicht ohne zwingende Gründe zu ändern. In diesem Schritt ist es äusserst wichtig genügend Zeit für eine gute Reflexion einzuplanen, sodass man schnell auf Misserfolge reagieren kann und die Massnahmen daraus planen und umsetzen kann. (VSSM, 2014)

# Kontrollieren

Der wichtigste Schritt in der IPERKA-Methode ist das Kontrollieren. Man kann seine Arbeit nicht abgeben, ohne dass man alles nochmals kontrolliert hat. Hierbei werden vor allem die Vorgaben, welche sich aus dem Schritt «Planen» ergeben haben, mit dem tatsächlichen Ergebnis verglichen und die Resultate festgehalten. Neben der Kontrolle des eigentlichen Produktes, ist es auch wichtig das Arbeitsverhalten zu überprüfen und allenfalls Abweichungen aus den Plänen aufzuzeigen. (VSSM, 2014)

# Auswerten

Im letzten Schritt von IPERKA reflektiert man nochmals, vom Informieren bis zum Kontrollieren die gesamte Bearbeitung des Auftrags. Beim Auswerten geht es um eine Selbsteinschätzung der erledigten Tätigkeiten. Man überlegt sich darüber hinaus was man aus der Arbeit gelernt hat, was man verbessern kann und wo man das Gelernte in der Zukunft anwenden kann. (VSSM, 2014)

# Glossar

Im Glossar werden die wichtigsten Fremdwörter und Technische Ausdrücke erklärt.

| Begriff | Erklärung |
| --- | --- |
| **Vulnerability** |  |
| **Continious Integration (CI)** |  |
| **Model View ViewModel (MVVM)** |  |
| **Daily** |  |
| ORM |  |
| Wire-Frames |  |
| UI-Controls |  |
| Inversion of Control (IOC) |  |

Tabelle 25: Glossar

# Verzeichnisse

Anbei werden alle verwendeten Ressourcen aufgelistet.

## Quellenverzeichnis

S.A, S. (2008). Leading Product for CI. Abgerufen am 30. Oktober 2018 von https://www.sonarqube.org/

VSSM, V. s. (2014). Die 6-Schritte-Methode. Abgerufen am 24. Oktober 2018 von https://www.vssm.ch/sites/default/files/vssm/BB/docs/AusbildungEFZ/IPERKA\_1\_4\_D2017.pdf

Wikipedia. (20. November 2015). Regressionstest. Abgerufen am 01. November 2018 von https://de.wikipedia.org/wiki/Regressionstest

Wikipedia. (16. September 2018). MVVM. Abgerufen am 01. November 2018 von https://de.wikipedia.org/wiki/Model\_View\_ViewModel

## Abbildungsverzeichnis

[Abbildung 1: Applikationstruktur 5](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846116)

[Abbildung 2: Soll-Analyse Menu 19](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846117)

[Abbildung 3: Soll-Analyse Edit Customer Details 20](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846118)

[Abbildung 4: Soll-Analyse Muesli Mixer 20](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846119)

[Abbildung 5: Soll-Analyse Components List 21](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846120)

[Abbildung 6: Soll-Analyse Current Muesli Mix list 21](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846121)

[Abbildung 7: Soll-Analyse My Muesli Mixes 22](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846122)

[Abbildung 8: Soll-Analyse Order 22](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846123)

[Abbildung 9: Soll-Analyse Order Confirmation 23](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846124)

[Abbildung 10: Systemübersicht 24](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846125)

[Abbildung 11: Use Case Diagramm Kunde 25](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846126)

[Abbildung 12: Code Coverage Beispiel SonarQube 27](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846127)

[Abbildung 13: MVVM 31](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\MyMuesli\Documents\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528846128)

[Abbildung 14: Just Muesli "Menu" 37](#_Toc528846129)

[Abbildung 15: Just Muesli "Customer Details" 38](#_Toc528846130)

[Abbildung 16: Just Muesli "Menu Part 2" 39](#_Toc528846131)

[Abbildung 17: Just Muesli "Muesli Mixer" 40](#_Toc528846132)

[Abbildung 18: Just Muesli "Component list" 41](#_Toc528846133)

[Abbildung 19: Just Muesli "Current Muesli" 42](#_Toc528846134)

[Abbildung 20: Just Muesli "My Muesli Mixes" 43](#_Toc528846135)

[Abbildung 21: Just Muesli "Order" 44](#_Toc528846136)

[Abbildung 22: Just Muesli "Order Confirmation" 45](#_Toc528846137)

## Tabellenverzeichnis

[Tabelle 1: Änderungshistorie 1](#_Toc528846138)

[Tabelle 2: Beteilgte Personen 7](#_Toc528846139)

[Tabelle 3: Aufwandschätzung 9](#_Toc528846140)

[Tabelle 4: Meilensteine 9](#_Toc528846141)

[Tabelle 5: Gantt-Diagramm 11](#_Toc528846142)

[Tabelle 6: Arbeitsprotokoll 31.10.2018 12](#_Toc528846143)

[Tabelle 7: Arbeitsprotokoll 01.11.2018 14](#_Toc528846144)

[Tabelle 8: Arbeitsprotokoll 02.11.2018 15](#_Toc528846145)

[Tabelle 9: Arbeitsprotokoll 07.11.2018 16](#_Toc528846146)

[Tabelle 10: Arbeitsprotokoll 09.11.2018 17](#_Toc528846147)

[Tabelle 11: Validationsregeln Customer Details 20](#_Toc528846148)

[Tabelle 12: Nährstoff-Kalkulation 21](#_Toc528846149)

[Tabelle 13: Bestellungsbestätigung Formatierung 23](#_Toc528846150)

[Tabelle 14: Use Case Kundendaten anpassen 25](#_Toc528846151)

[Tabelle 15: Use Case Muesli Mixer aufrufen 26](#_Toc528846152)

[Tabelle 16: Use Case My Muesli Mixes editieren 26](#_Toc528846153)

[Tabelle 17: Use Case: Bestellung aufgeben 26](#_Toc528846154)

[Tabelle 18: Use Case Bestellungsbestätigung einsehen 26](#_Toc528846155)

[Tabelle 19: Glossar 35](#_Toc528846156)

# Aufgabenstellung

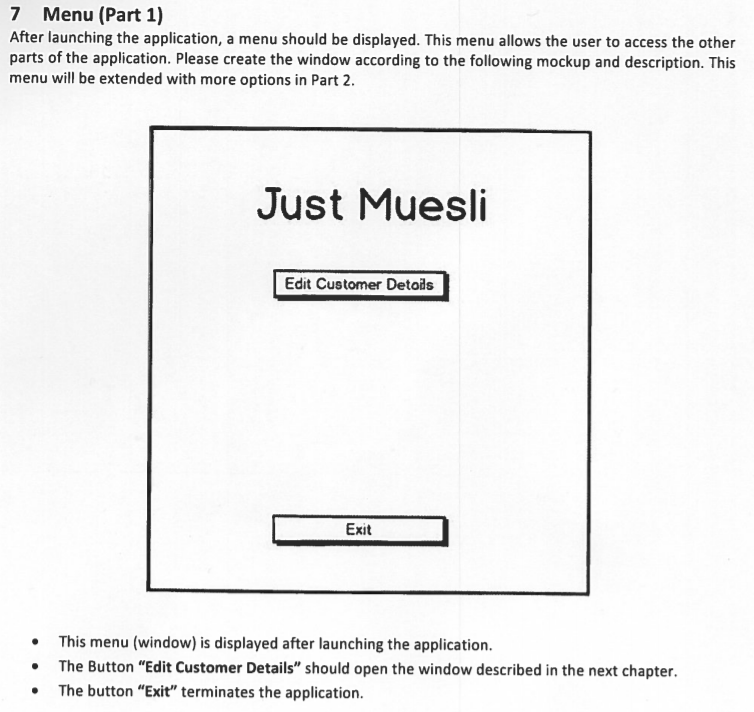


Abbildung 15: Just Muesli "Menu"

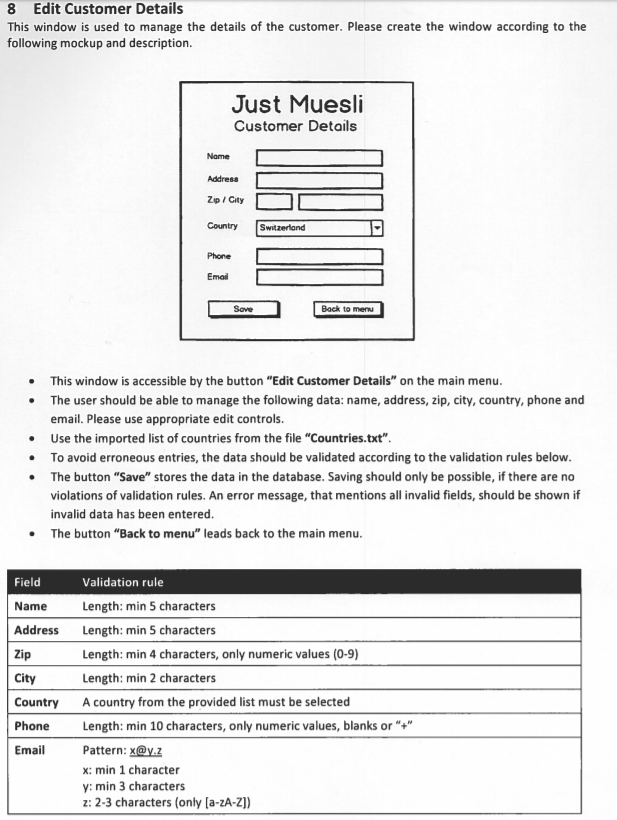


Abbildung 16: Just Muesli "Customer Details"

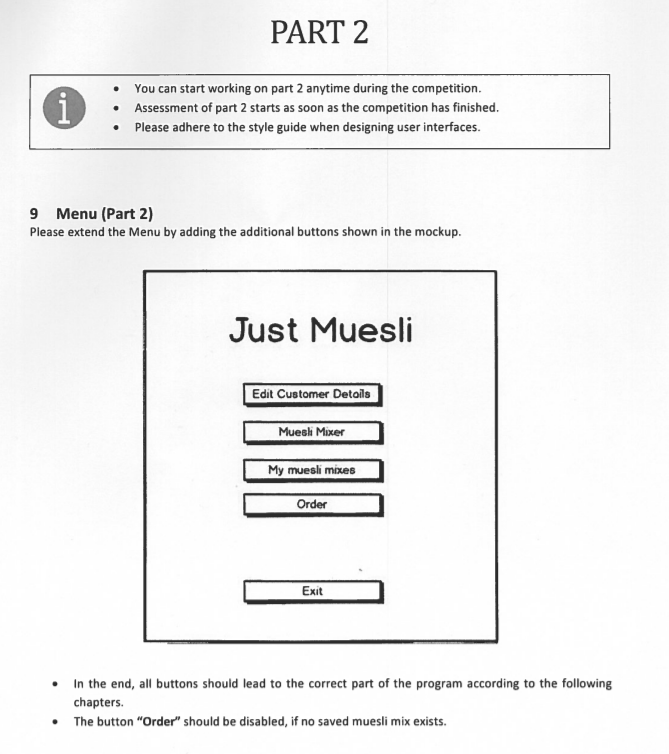


Abbildung 17: Just Muesli "Menu Part 2"

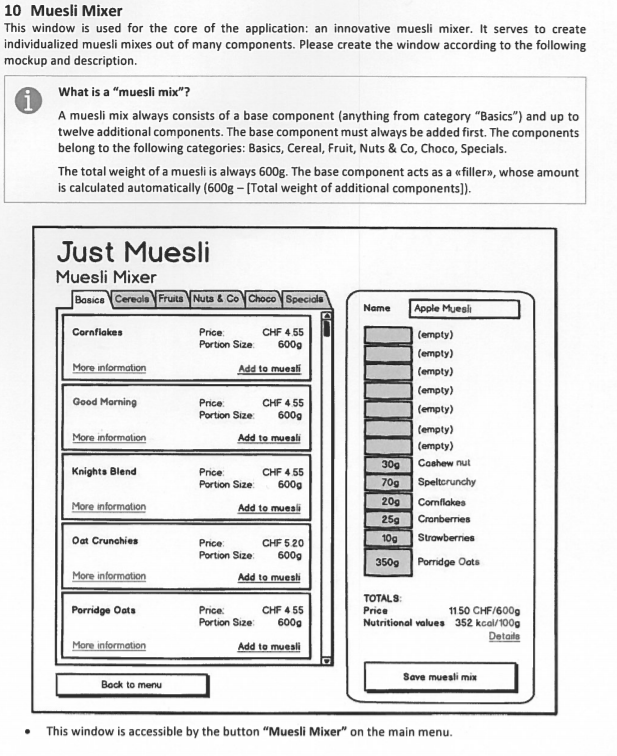


Abbildung 18: Just Muesli "Muesli Mixer"

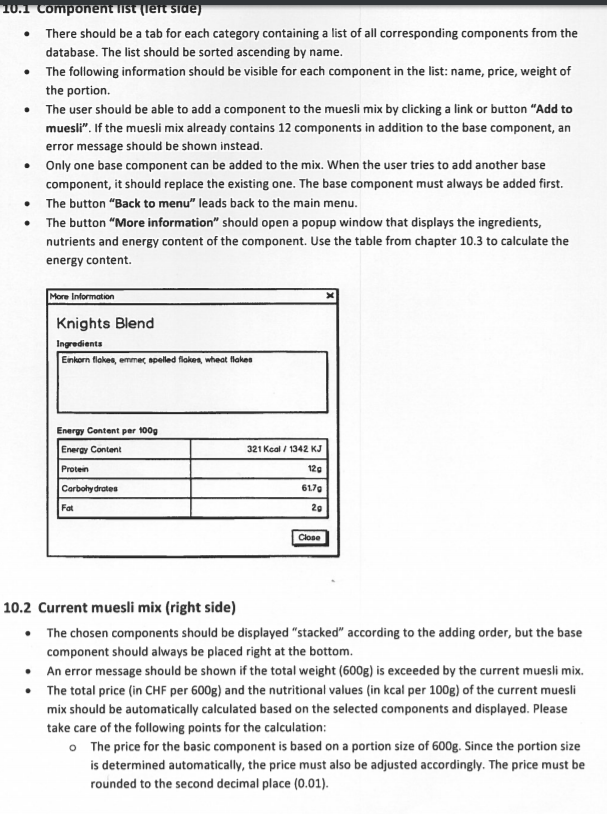


Abbildung 19: Just Muesli "Component list"

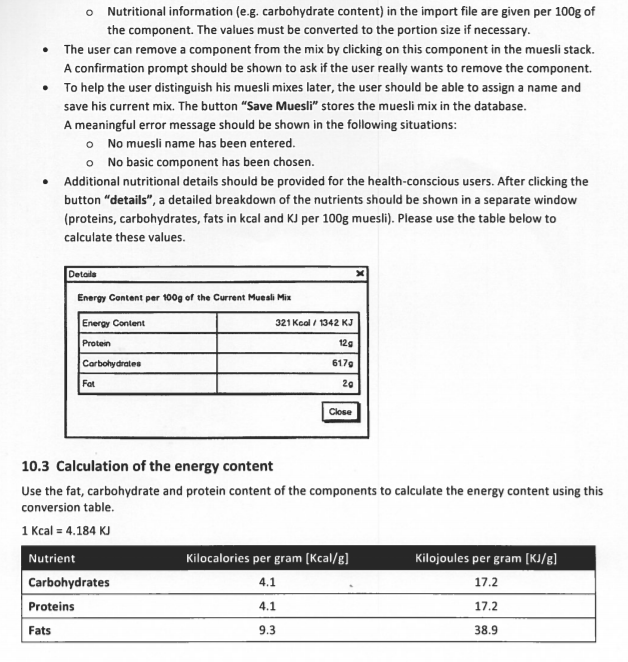


Abbildung 20: Just Muesli "Current Muesli"

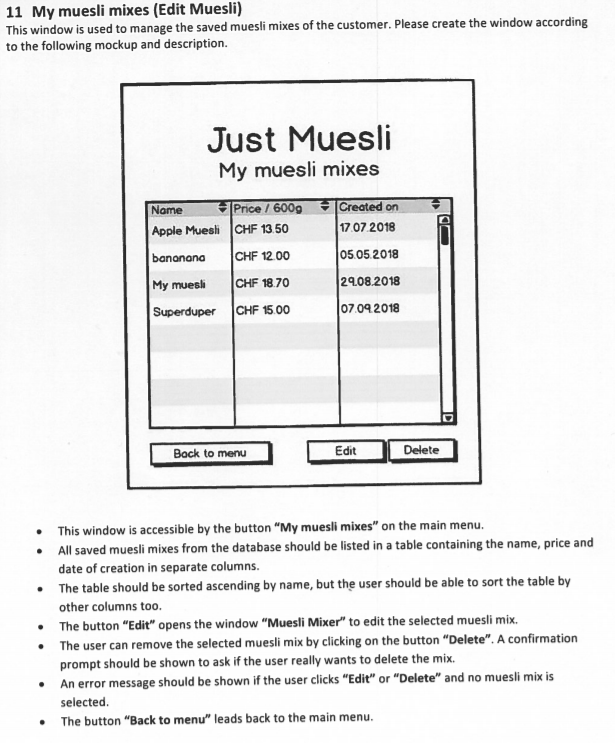


Abbildung 21: Just Muesli "My Muesli Mixes"

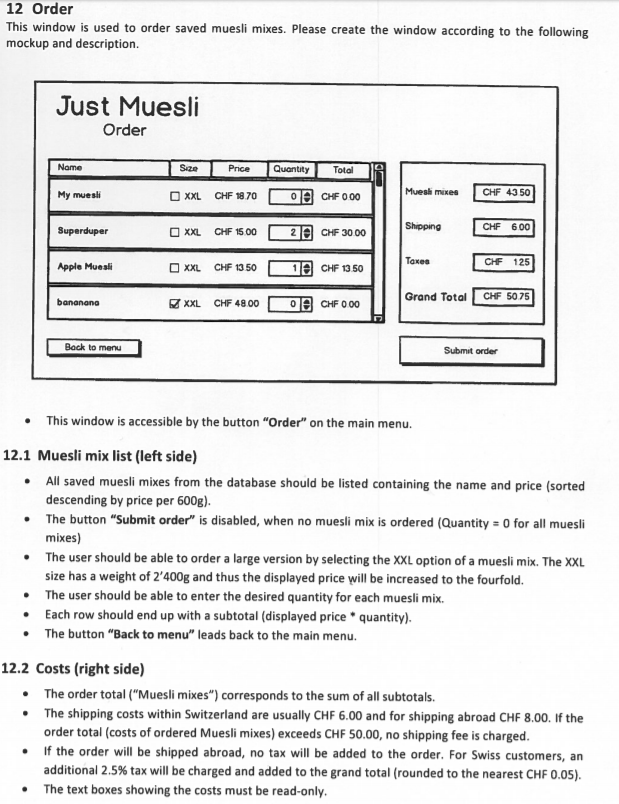


Abbildung 22: Just Muesli "Order"

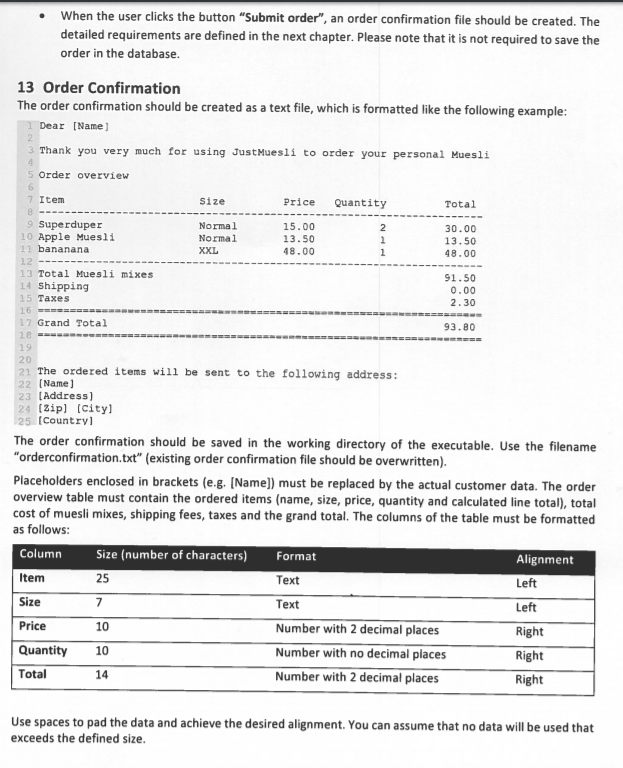


Abbildung 23: Just Muesli "Order Confirmation"

# Programmcode