IPA Dokumentation

Für internen Gebrauch

autor(en) : Bosshard Fabrice

version : 1.01.0

status : Final

quelle : Atos

dokumentendatum : 18 Oktober 2018­

­

­

anzahl der seiten : 1

Änderungshistorie

| Version | Datum | Beschreibung | Autor(en) |
| --- | --- | --- | --- |
| 0.1 | 30.10.2018 | Initiale Dokumentationsstruktur erstellt. | 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**

1 Aufgabenstellung 4

1.1 Projekt 4

1.2 Ausgangslage 4

1.3 Detaillierte Aufgabenstellung 4

1.4 Management Summary 4

1.5 Applikationsstruktur 5

1.6 Technologie-Stack 5

1.7 Firmenstandards 5

1.8 Anforderungen an die Applikation 5

1.9 Vorkenntnisse 6

1.10 Vorarbeiten 6

2 Projektorganisation 7

2.1 Beteiligte Personen 7

2.2 Projektmanagementmethode 7

2.3 Backup-Konzept 7

3 Zeitplanung 8

3.1 Aufwandschätzung 8

3.2 Meilensteine 9

3.3 Massnahmen bei Verzug 9

3.4 Gantt-Diagramm 10

4 Arbeitsprotokoll 11

4.1 Mittwoch, 31. Oktober 2018 11

4.2 Donnerstag, 01. November 2018 12

4.3 Freitag, 02. November 2018 13

4.4 Mittwoch, 07. November 2018 14

4.5 Freitag, 09. November 2018 15

5 Kurzfassung 16

6 Informieren 17

7 Planen 18

8 Entscheiden 19

9 Realisierung 20

10 Kontrollieren 21

11 Auswerten 22

12 Glossar 23

13 Verzeichnisse 24

13.1 Quellenverzeichnis 24

13.2 Abbildungsverzeichnis 24

13.3 Tabellenverzeichnis 24

14 Aufgabenstellung 25

15 Programmcode 34

# 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 1.8 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

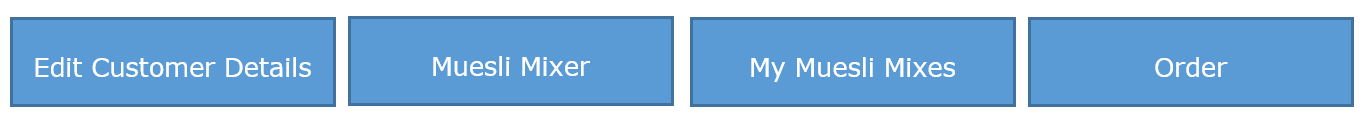
Das folgende Diagramm zeigt die verschiedenen Teile der Applikation. Jeder Knoten repräsentiert ein Fenster, welche jeweils im Anhang (Aufgabenstellung ICT-Skills) detailliert beschrieben werden.

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 | |  | |
| Systemübersicht definieren | 0.75 | |  | |
| Planen | Use Cases erstellen | 2 | |  | |
| Testkonzept erstellen | 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 | | | 30.75 h  (30 Stunden 45 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 |
| Projektorganisation definieren | 0.75 | 0.5 | - |
| Aufwandschätzung und Meilensteine definieren | 1 | 1 | - |
| Ist- / Sollanalyse beschreiben | 1.5 | 1.75 | - |
| Systemübersicht definieren | 0.75 | 1 | - |
| Use Cases angefangen | 1 | 1 | Nur kleine Überlegungen und Vorbereitung für wirkliche Use Cases am nächsten Tag |
| Total | 5.25 Stunden | 5.5  Stunden | - |

Tabelle 6: Arbeitsprotokoll 31.10.2018

### Erfolge

### Misserfolge

### Code Analyse

### Massnahmen

### Fazit

Mit remo gespräch der Aufgabenstellung => änderung von sqlite zu mssql

## Donnerstag, 01. November 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 2 (01.11.2018) | | | |
| Phase: | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 7: Arbeitsprotokoll 01.11.2018

### Erfolge

### Misserfolge

### Code Analyse

### Massnahmen

### Fazit

## Freitag, 02. November 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 3 (02.11.2018) | | | |
| Phase: | | | |
| 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- / ModulTests
  + 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

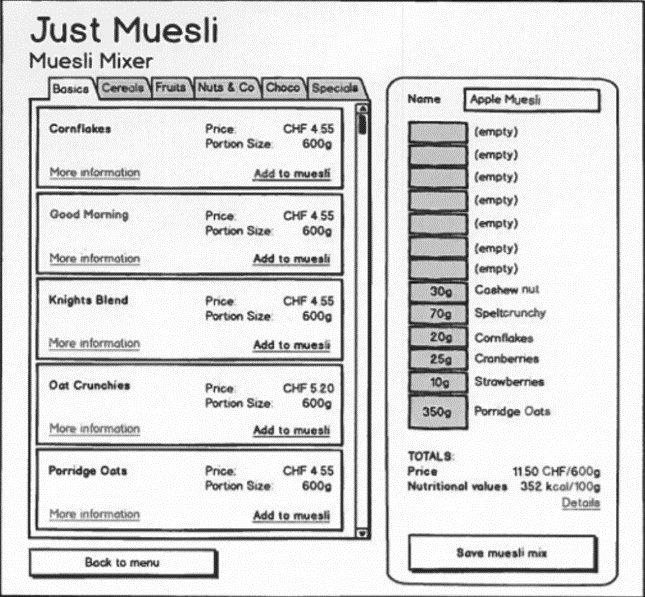
**Muesli Mixer**

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)).

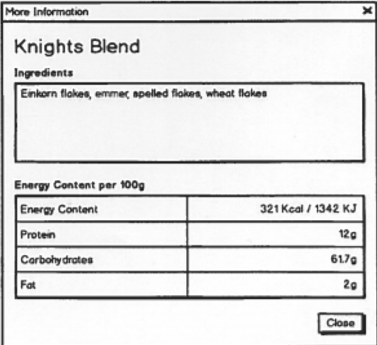
***Components list (Linke Seite von Muesli Mixer)***

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:

|  |  |  |
| --- | --- | --- |
| 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)***

**My muesli mixes**

**Order**

**Order Confirmation**

### Backend

## 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 6: 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)

## Anwendungsfälle

# 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)

# 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 |  |

Tabelle 13: 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

## Abbildungsverzeichnis

[Abbildung 1: Applikationstruktur 5](file:///C:\Arbeitsordner\_ImportantStuff\IPA\Probe\IPA_Projektdokumentation_FabriceBosshard.docx#_Toc528756373)

[Abbildung 2: Just Muesli "Menu" 25](#_Toc528756374)

[Abbildung 3: Just Muesli "Customer Details" 26](#_Toc528756375)

[Abbildung 4: Just Muesli "Menu Part 2" 27](#_Toc528756376)

[Abbildung 5: Just Muesli "Muesli Mixer" 28](#_Toc528756377)

[Abbildung 6: Just Muesli "Component list" 29](#_Toc528756378)

[Abbildung 7: Just Muesli "Current Muesli" 30](#_Toc528756379)

[Abbildung 8: Just Muesli "My Muesli Mixes" 31](#_Toc528756380)

[Abbildung 9: Just Muesli "Order" 32](#_Toc528756381)

[Abbildung 10: Just Muesli "Order Confirmation" 33](#_Toc528756382)

## Tabellenverzeichnis

[Tabelle 1: Änderungshistorie 1](#_Toc528756383)

[Tabelle 2: Meilensteine 9](#_Toc528756384)

[Tabelle 3: Gantt-Diagramm 10](#_Toc528756385)

[Tabelle 4: Arbeitsprotokoll 31.10.2018 11](#_Toc528756386)

[Tabelle 5: Arbeitsprotokoll 01.11.2018 12](#_Toc528756387)

[Tabelle 6: Arbeitsprotokoll 02.11.2018 13](#_Toc528756388)

[Tabelle 7: Arbeitsprotokoll 07.11.2018 14](#_Toc528756389)

[Tabelle 8: Arbeitsprotokoll 09.11.2018 15](#_Toc528756390)

[Tabelle 9: Glossar 23](#_Toc528756391)

# Aufgabenstellung

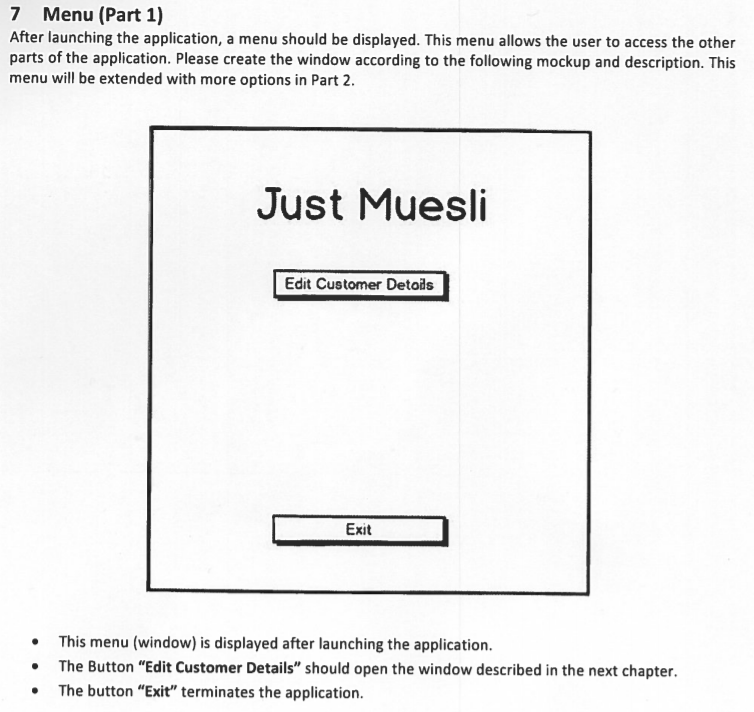


Abbildung 7: Just Muesli "Menu"

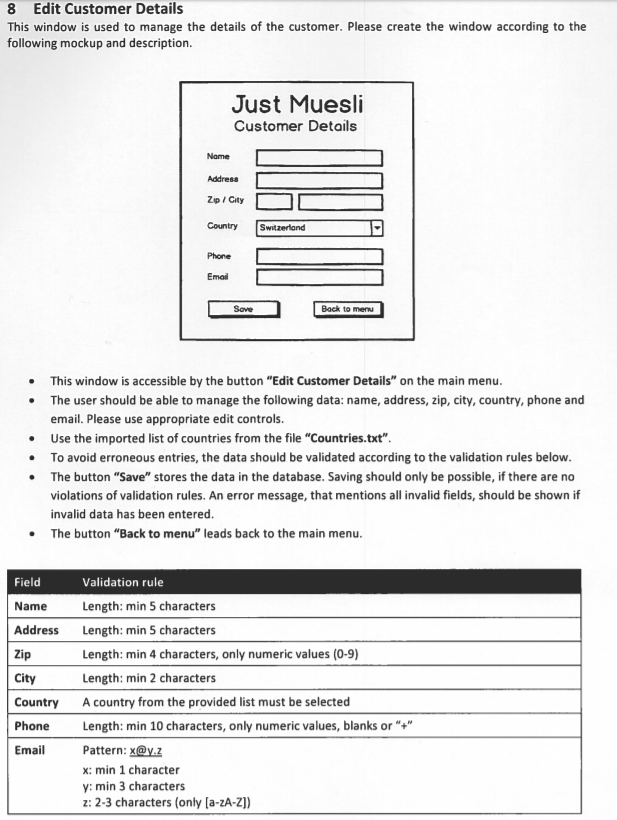


Abbildung 8: Just Muesli "Customer Details"

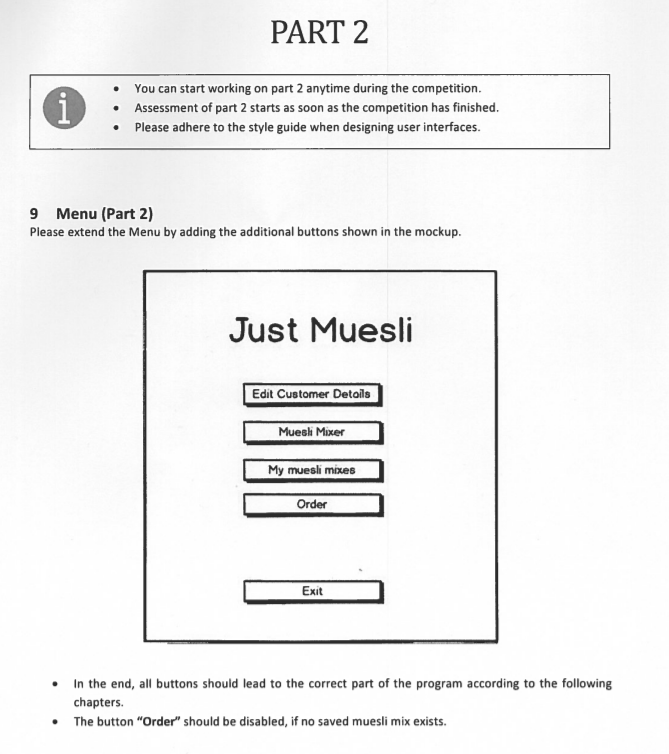


Abbildung 9: Just Muesli "Menu Part 2"

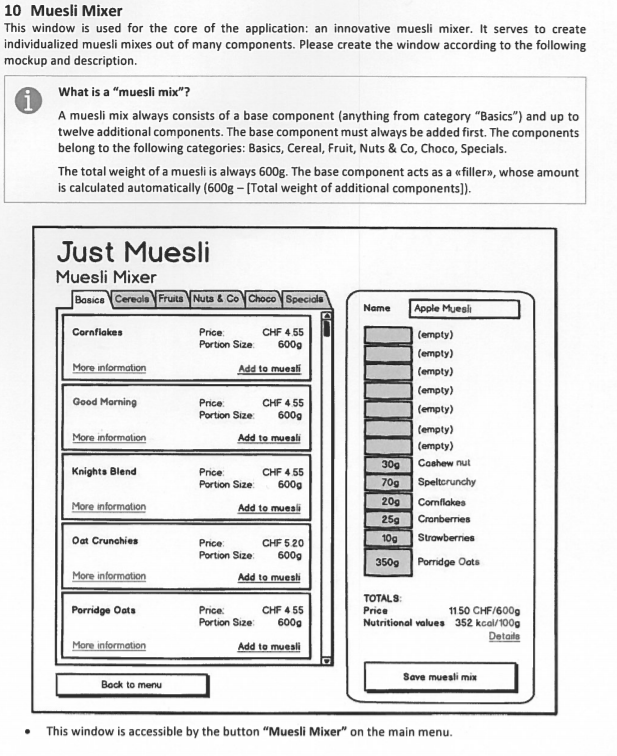


Abbildung 10: Just Muesli "Muesli Mixer"

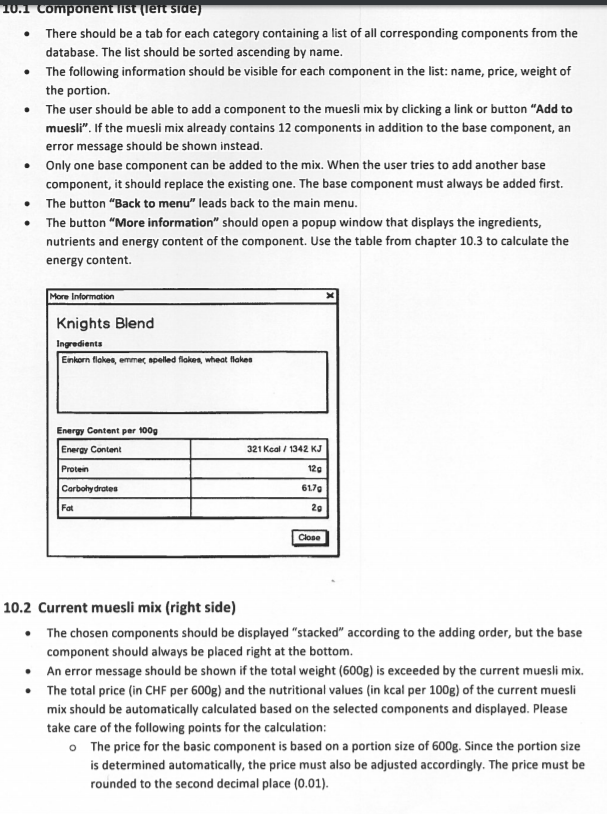


Abbildung 11: Just Muesli "Component list"

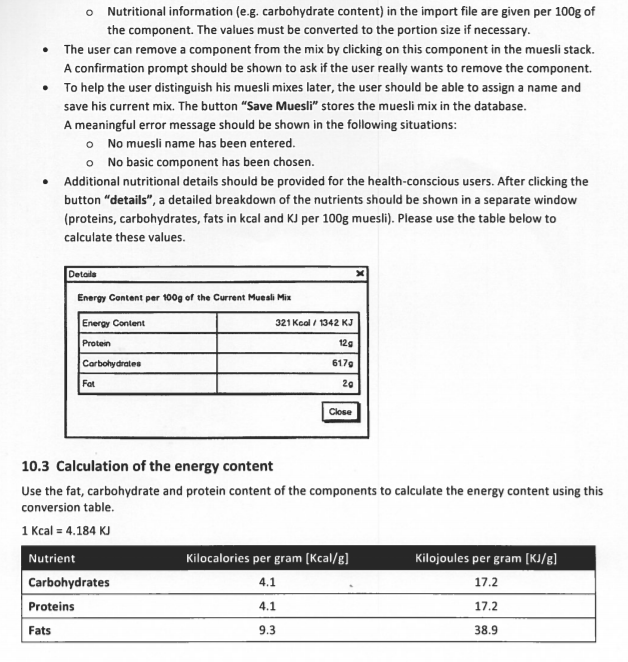


Abbildung 12: Just Muesli "Current Muesli"

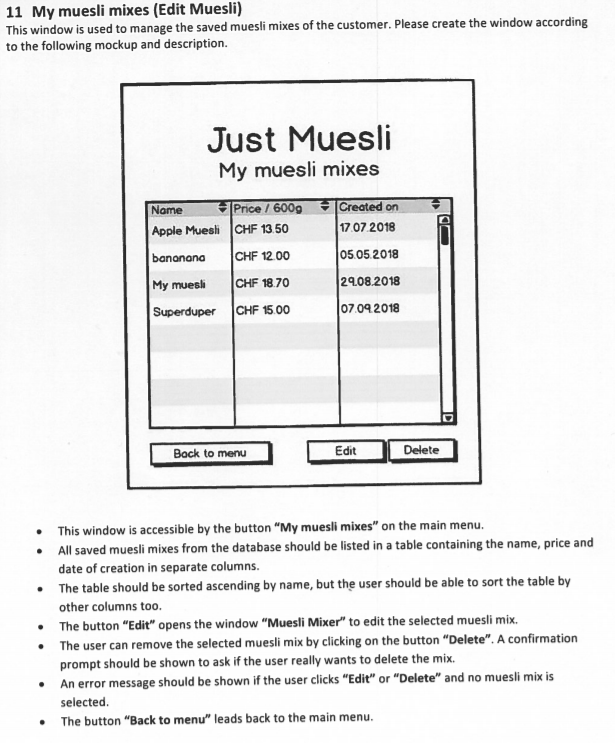


Abbildung 13: Just Muesli "My Muesli Mixes"

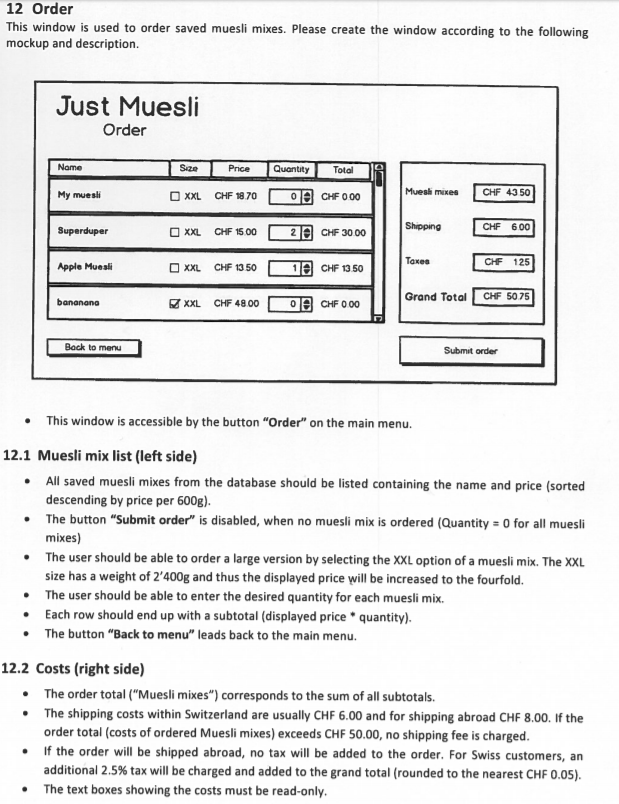


Abbildung 14: Just Muesli "Order"

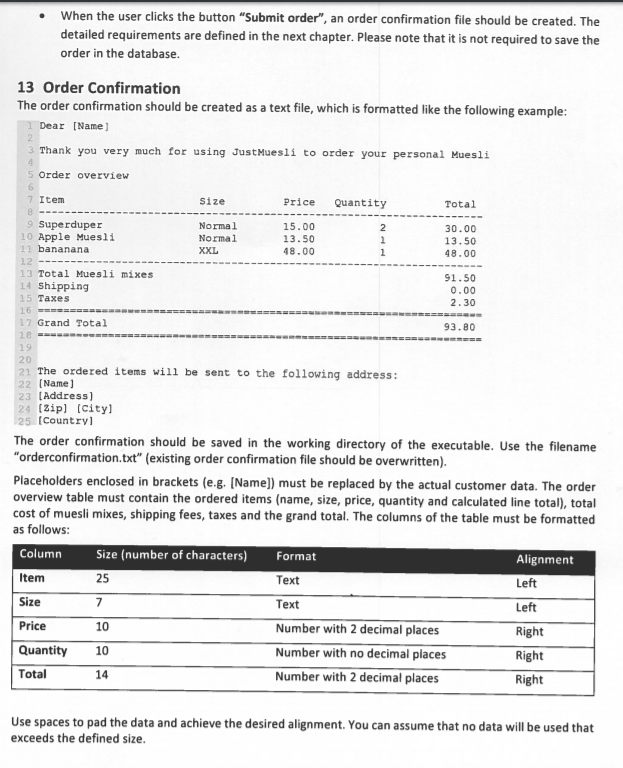


Abbildung 15: Just Muesli "Order Confirmation"

# Programmcode