Tatyana Merlo – Natalie Stalder – Nadja Stadelmann

Starbucks Manager

M120 – Benutzerschnittstelle entwickeln

# Einleitung

## Systemanforderungen

## Vorwissen

## Dokument

### Grund

### Aufbau

* Analyse
  + Beschrieb und erste Aufsplittung in einzelne Teilbereiche
* Vorgehen
  + Abhängigkeiten, Aufgabenübersicht
* Design
  + Detaillierte Analyse DB-Design
  + Detaillierte Analyse Programmstruktur
* Implementation
* Testing
  + Testcases basieren auf UseCases

Inhalt

[Einleitung 1](#_Toc494206374)

[Systemanforderungen 1](#_Toc494206375)

[Vorwissen 1](#_Toc494206376)

[Dokument 1](#_Toc494206377)

[Grund 1](#_Toc494206378)

[Aufbau 1](#_Toc494206379)

[Vision 3](#_Toc494206380)

[Analyse 3](#_Toc494206381)

[Anforderungen 3](#_Toc494206382)

[Muss-Kriterien 3](#_Toc494206383)

[Kann-Kriterien 3](#_Toc494206384)

[Use Cases 3](#_Toc494206385)

[Vorgehensplan 4](#_Toc494206386)

[Design 5](#_Toc494206387)

[DB 5](#_Toc494206388)

[Tabellen 5](#_Toc494206389)

[User 5](#_Toc494206390)

[Location 5](#_Toc494206391)

[Designanpassungen 5](#_Toc494206392)

[Programm 5](#_Toc494206393)

[Model 5](#_Toc494206394)

[View 5](#_Toc494206395)

[Controller 5](#_Toc494206396)

[Designanpassungen 5](#_Toc494206397)

[Implementation 6](#_Toc494206398)

[Implementierung 6](#_Toc494206399)

[Testing 7](#_Toc494206400)

# Vision

Im Modul 326 haben wir ein Starbucks-Menümanager mit Java erstellt. Bisher konnte der jedoch nur textbasiert verwendet werden (mittels Konsole).

Unser Ziel ist nun, den Mitarbeitern das Mutieren des Menüs mit einem GUI zu erleichtern. Das GUI soll alle bisherigen Userinteraktionen abdecken, sprich:

* Menü anzeigen
* Produkt hinzufügen
* Produkt bearbeiten
* Produkt löschen
* Programm beenden

Weiter möchten wir eine Login- sowie Logoutfunktion einbauen, um die Sicherheit der Daten zu gewährleisten.

# Analyse

## Anforderungen

### Muss-Kriterien



### Kann-Kriterien

## Use Cases

# Design

## Filestruktur

Der bisherige Starbucksmanager hat keine Datenbank, welche er verwaltet. Die Daten werden in einem einfachen File abgespeichert. Dennoch ist die Struktur des Files natürlich extrem wichtig für die Datenverarbeitung, weshalb sie hier kurz deklariert ist.

Jede Zeile steht für einen Menü-Eintrag, der in einzelne Komponenten unterteilt ist. Je nachdem, welche Attribute der Eintrag mitführt, wird er in eine andere Kategorie unterteilt.

**Kategorie Recordstruktur**

Beverage Name | Preis | null | true (hot) / false (cold)  
beverage¦2.3¦null¦true

Coffee Name | Preis |Zutaten  
coffee¦1.4¦ingr

Extra Name | Preis  
extra¦2.0

Food Name | Preis |Zutaten | Vegi / Vegan / Glutenfrei / usw.  
food¦1.5¦Ingr¦Vegi

## Programm

### Model

Wie bereits erwähnt werden die Daten in einem einzigen File gehandhabt. Das File und sein Handling besteht bereits.

### View

Das GUI ist komplett neu umzusetzen. Bisher waren nur Kommandozeileneingaben zur Steuerung des Programmes da. Dies soll neu über ein GUI geregelt werden, um den Mitarbeitern einen leichteren Einstieg zu ermöglichen.

So stellen wir uns das Ergebnis des GUIs vor:

### Controller

Die Controller bestehen teilweise, da bereits Usereingaben verarbeitet wurden. Dennoch müssen sie an das GUI angepasst werden.

# Implementation

## Implementierung

# Testing