

# Dokumentation Shopping Mall

Ein Werk von Eveline Vizante und Mathias Jungkind

Version: 2.01  
Language: deutsch  
Release: 06.02.2024

1	Einleitung.....	3
2	Allgemeines .....	3
2.1	Ziel und Zweck des Dokuments.....	3
2.2	Ausgangssituation .....	3
2.3	Projektbezug.....	3
2.5	Teams und Schnittstellen .....	3
3	Konzept.....	3
3.1	Ziele der Shopping Mall.....	3
3.2	Ziele und Nutzen des Verwaltungssystems.....	3
3.3	Zielgruppen .....	3
4	Funktionale Anforderungen .....	4
4.1	Verwaltung von Mitarbeitern .....	4
4.2	Verwaltung von Geschäften .....	4
4.3	Benutzeroberfläche (GUI) .....	4
4.4	Datenbank .....	4
5	Nichtfunktionale Anforderungen .....	6
5.1	Performance .....	6
5.2	Sicherheit.....	6
5.3	Technische Anforderungen .....	6
6	Rahmenbedingungen .....	6
6.1	Zeitplan.....	6
6.2	Problemanalyse.....	6
6.3	Qualität.....	7
7	Einblick in das Klassensystem.....	7
7.1	Genereller Aufbau .....	7
7.2	Funktionsweisen und Methoden der Klassen .....	7
8	Diagrammansichten .....	9
8.1	Klassendiagramm .....	9
8.2	Struktogramm .....	10
8.3	Gantt-Diagramm.....	10
8.4	Use-Case-Diagramm .....	11
8.5	Sequenz-Diagramm .....	12
8.6	Relationen Modell .....	13

# 1 Einleitung

Das vorliegende Pflichtenheft definiert die funktionalen und nicht-funktionalen Anforderungen an das Verwaltungssystem einer Shopping Mall. Es dient als Grundlage für die Entwicklung und Implementierung des Systems.

## 2 Allgemeines

### 2.1 Ziel und Zweck des Dokuments

Dieses Pflichtenheft beschreibt die Anforderungen an das Verwaltungssystem einer Shopping Mall zur effektiven Verwaltung von Geschäften und Mitarbeitern.

### 2.2 Ausgangssituation

Die Shopping Mall hat viele Geschäfte und Mitarbeiter, und ihre effiziente Verwaltung ist entscheidend für ihren Erfolg.

### 2.3 Projektbezug

Das Verwaltungssystem ist ein eigenständiges Projekt der Shopping Mall.

### 2.5 Teams und Schnittstellen

## 3 Konzept

### 3.1 Ziele der Shopping Mall

Das Hauptziel der Shopping Mall ist es, so viel Geld wie möglich zu verdienen und eine gute Bewertung durch hochwertige Geschäfte und Mitarbeiter zu erhalten.

### 3.2 Ziele und Nutzen des Verwaltungssystems

Das Verwaltungssystem soll die Shopping Mall dabei unterstützen, die Produktivität der Mitarbeiter zu steigern, unrentable Geschäfte zu identifizieren und zu entfernen sowie die Kapitalrendite zu maximieren.

### 3.3 Zielgruppen

Die Zielgruppe des Verwaltungssystems umfasst das Management und die Verwaltungsmitarbeiter der Shopping Mall.

# 4 Funktionale Anforderungen

## 4.1 Verwaltung von Mitarbeitern

Erfassung von Mitarbeiterdaten (Name, Kontaktdaten, Position, Gehalt)  
Überwachung der Arbeitsleistung und Produktivität der Mitarbeiter anhand von Metriken wie Arbeitsstunden, Umsatzbeitrag und Rating.

## 4.2 Verwaltung von Geschäften

Erfassung von Geschäftsdaten (Name, Standort, Umsatz, Mietkosten)  
Analyse der Rentabilität von Geschäften anhand von Metriken wie Umsatz und Rating. Möglichkeit zur Identifikation unrentabler Geschäfte und Durchführung entsprechende

## 4.3 Benutzeroberfläche (GUI)

Benutzerfreundliche Oberfläche zur Darstellung von Daten und Interaktive Funktionen zur Verwaltung von Mitarbeitern und Geschäften.

## 4.4 Datenbank

Aufbau einer Datenbank zur Speicherung aller relevanten Informationen über Geschäfte, Mitarbeiter und die Shopping Mall Sicherstellung der Datenintegrität und -sicherheit

Die Datenbank besteht aus vier Tabellen:

- Mall
- Shop
- Restaurant
- Worker

Die Datenspalten zu den dazugehörigen Tabellen lauten wie folgt:

Mall:

- Id
  - Die eindeutige Identifikationsnummer der Mall
- Name
  - Bezeichnung des Standortes
- Capacity
  - Anzahl der möglichen Gäste in der Mall
- Address
  - Die Adresse der Mall
- Capital

- Gesamter Geldwert des Standortes
- RentAmount
  - Monatlich erhaltener Geldwert von den einzelnen Shops
- Status
  - Öffnungsstatus der Mall
- VisitorCount
  - Momentane Besucherzahl des Standortes
- AverageVisitorRevenue
  - Durchschnittlicher Gewinn pro Besucher der Mall

#### Shop:

- Id
  - Eindeutige Identifikationsnummer für das Geschäft
- Name
  - Name des Geschäfts
- Size
  - Geschäftsgröße, angegeben in m<sup>3</sup>
- Category
  - Angabe des Shoptyps für Identifizierung des zu kreirenden Objekts
- Status
  - Gibt den Öffnungsstatus des Programms an
- Capital
  - Angabe des Geldwertes, des dazugehörigen Shops
- Rating
  - Sterne bewertung des Shops (von 1 bis 5)
- AvgVisitorCount
  - Durchschnitts Besucherzahl des Shops
- AvgVisitorRevenue
  - Durchschnittlicher Geldwert der von einem Besucher erreicht wird

#### Restaurant:

- Id
  - Eindeutige Identifikationsnummer für das Restaurant
- HealthInspection
  - Anzeige, ob eine aktuelle Gesundheitsinspektion vorhanden ist
- Stars
  - Gibt die dazugehörigen Sterne des Restaurants an
- ShopId
  - Foreign Key zu Shop, um allgemeine Daten des Restaurants mit diesen Daten zu verbinden

#### Worker:

- Id
  - Eindeutige Identifikationsnummer für die Arbeitskräfte

- Name
  - Name der Arbeitskraft
- Age
  - Alter der Arbeitskraft
- SocialSecurityNumber
  - Staatliche Identifikationsnummer der Arbeitskraft
- WorkField
  - Arbeitsbereich des Arbeiters
- WeekHours
  - Wochenarbeitszeit des Arbeiters
- Position
  - Hierarchische Position des Arbeiters
- Rating
  - Die Arbeitseinschätzung des Arbeiters (von 1 bis 5)

Eine Visualisierung des Datenbankmodells ist in 8.6 zu finden.

## 5 Nichtfunktionale Anforderungen

### 5.1 Performance

Das System soll eine schnelle und zuverlässige Leistung bieten, auch bei einer großen Anzahl von Datensätzen.

### 5.2 Sicherheit

Das System muss sicherstellen, dass alle sensiblen Daten geschützt und vor unbefugtem Zugriff gesichert sind, z.B. durch die Verwendung von Attributen für die Zugriffssteuerung.

### 5.3 Technische Anforderungen

Das System wird in C# entwickelt und verwendet eine Datenbank, die mit Visual Studio erstellt wird.

## 6 Rahmenbedingungen

### 6.1 Zeitplan

Festlegung der Projektphasen und Meilensteine Zeitrahmen für die Entwicklung, Implementierung und Testphase

### 6.2 Problemanalyse

Identifikation potenzieller Risiken wie technische Schwierigkeiten bei der Integration von Datenbanken.

## 6.3 Qualität

Anforderungen an die Qualitätssicherung und -kontrolle während des gesamten Projekts.

# 7 Einblick in das Klassensystem

## 7.1 Genereller Aufbau

Das Klassensystem startet mit einer übergeordneten Klasse des Namens "Mall", darunter erbt die Klasse "Shop". Zum Schluss der Vererbungskette sind die Klassen "Restaurant", "Clothstore", "Drugstore" und "Techstore" aufzufinden. Separat von den anderen Klassen, existiert noch "Worker". Zusätzlich besitzen wir noch 5 Forms-Klassen "DetailScreen", "ShopOverview", "WorkersPage", "Add", "AddWorkers" und "Form1". Für eine visuelle Repräsentation der Klassen, wird auf Punkt 8.1 verwiesen.

## 7.2 Funktionsweisen und Methoden der Klassen

Shop:

"Shop" ist, zusammen mit "Worker", das Fundament unserer Arbeit, hier werden die essenziellen Informationen abstrakt definiert, welche danach von den Unterklassen "Restaurant", "Clothstore", "Drugstore" und "Techstore" genauer ausgearbeitet werden.

"Shop" deklariert die im Bereich 4.4 genannten Attribute mit Get-Methoden und Set-Methoden. Weitere Methoden werden nicht benötigt.

Restaurant, Clothstore, Drugstore und Techstore:

Dies sind die von uns zu erstellende Objekte, mit dessen Daten am Ende gearbeitet wird

Sie übernehmen die vorgegebenen Attribute von Shop und definieren dazu noch ihre eigenen Konstruktoren.

Worker:

Eine Klasse die zum Erstellen von Objekten, der Arbeiter der Mall gedacht ist. Als Attribute findet man die essenziellen Daten der Arbeiterschaft und diese Liste

der Attribute ist in Bereich 4.4 zu sehen. Außerhalb davon existiert als Methode nur der Konstruktor als notwendige Funktion der Klasse.

DetailScreen:

Zeigt die Daten des ausgewählten Geschäfts oder Mitarbeiters an.

Auf dieser Seite existiert ein Delete-Button, welcher für das Entfernen des dazugehörigen Datensatzes zuständig ist und dies durch den Methodenaufruf "delete\_button\_Click()" erfolgt. Wenn ein Datensatz nicht gelöscht werden möchte, so betätigt man stattdessen den Back-Button, welcher einen auf die vorherige Seite, durch die Methode "backButton\_Click\_1()", befördert.

StoreOverview:

Dies ist der Bildschirm, wo für die Anzeige der Geschäfte zuständig ist. Die Geschäfte selbst bekommt die Seite aus einer Liste namens "list" vom Typ "List<Shop>" und erstellt daraus die benötigten Panels.

Die hier verwendeten Methoden lauten "EditButton\_Click()", "createPanels(List<Shop> list)", "openPage(Shop shop)" und "addButton\_Click()".

"EditButton\_Click()" wird verwendet, um einen Editierungsmodus zu starten, indem Daten abgeändert werden können.

"createPanels(List<Shop> list)" wird genutzt um die vorhin genannten Panels für die Anzeige der Geschäfte zu kreieren.

"openPage(Shop shop)" versteckt das momentane Fenster und erstellt + öffnet einen "DetailScreen" zu dem dazugehörigen Datensatz.

"addButton\_Click()" bringt einen zum Hinzufügen eines neuen Geschäft-Datensatzes.

WorkersPage:

Als Gegenstück zum "StoreOverview" haben wir die "WorkersPage". Diese hat eine identische Funktion wie die "StoreOverview" und implementiert dieselben Methoden, nur dass die Methoden auf der Liste "listWorker" vom Datentyp "List<Worker>" aufbauen, somit alle Vorkommen von Shop auf Worker übertragen werden können.

Add:

Ist die Seite für die Erstellung neuer Geschäftsdatensätze.



Hier gibt es vielerlei Eingabefelder und zum Schluss noch ein Hinzufüge-Knopf, der die angegebenen Daten in die Datenbank hinzufügt.

Als einzige Methode dieser Seite gibt es “addButton\_Click()”, welcher die Daten aus der TextInputs liest und einen SQL-Command absenden, welcher eine neue Zeile in der Datenbank, mit den angegebenen Daten, erstellt.

AddWorker:

Wie bei der WorkersPage auch, ist dies das Gegenstück zu Add und erfüllt dadurch eine identische Rolle. Somit werden alle angegebenen Daten auf gleiche Art und Weise, diesmal in Worker, eingetragen, wie auch in Add.

Form1(unglücklicher Name ☹):

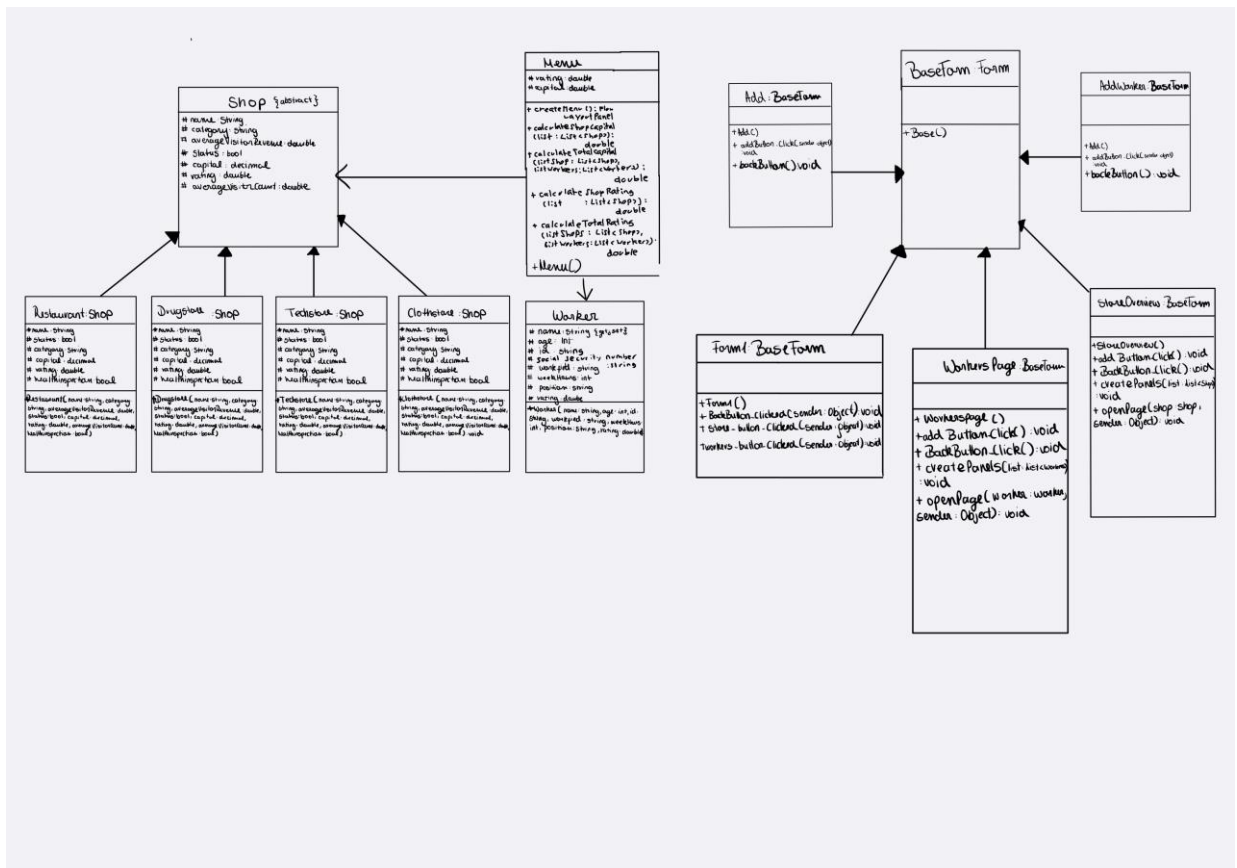
Fungiert als erster Bildschirm in unserer App und kann damit auch als “Home-Bildschirm” bezeichnet werden. Generell ist diese Seite ausschließlich für die Entscheidung der einzusehenden Daten wichtig. Also ob man über Geschäfte oder über Mitarbeiter einsehen möchte.

So hat die Seite auch bloß zwei Methoden, welche jeweils bloß für den Übergang in die dazugehörige Seite zuständig sind.

Die Namen der Methoden lauten: “store\_button\_Click()” und “workers\_button\_Click()”

## 8 Diagrammansichten

### 8.1 Klassendiagramm

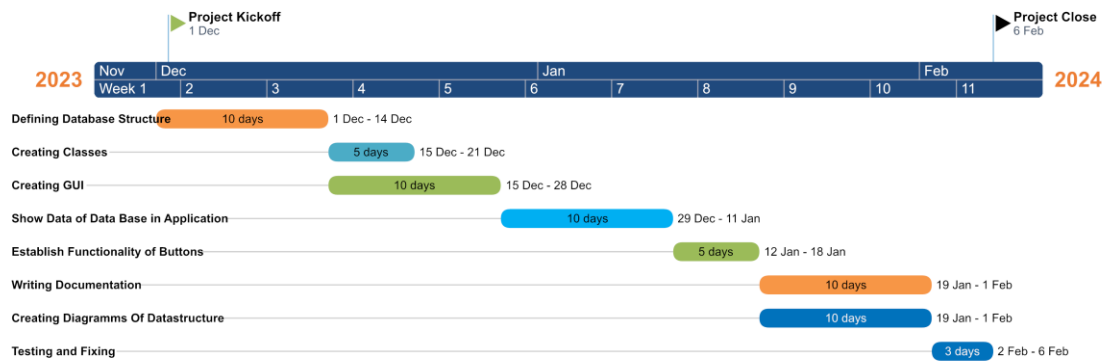


## 8.2 Struktogramm

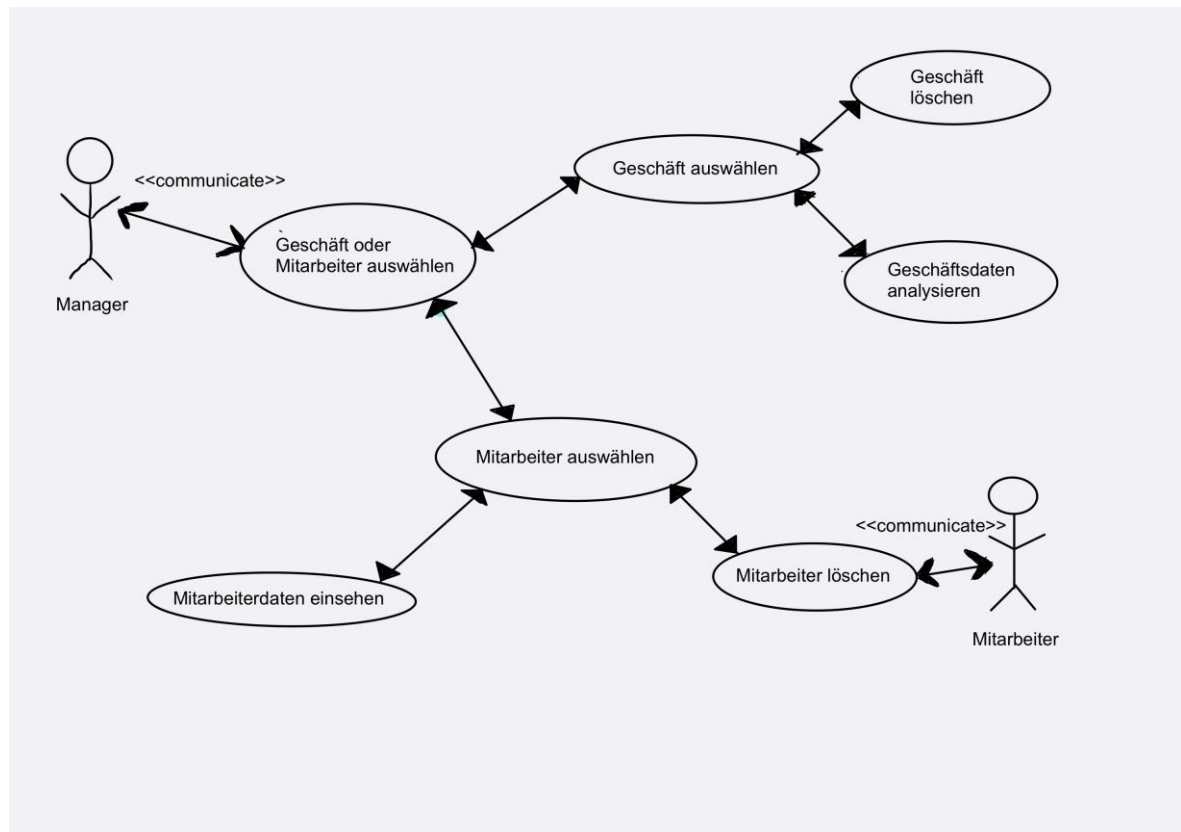
Geschäft in der Übersicht ausgewählt?	
Wahr	Falsch
Anzeigen der Geschäftsübersicht	Anzeigen der Mitarbeiterübersicht
E: Anklicken des gewünschten Geschäfts	E: Anklicken des gewünschten Mitarbeiters
Anzeigen der Geschäftsdaten	Anzeigen der Mitarbeiterdaten
Unzufrieden mit der Leistung?	
Wahr	Falsch
E: Löschtaaste betätigen	E: Zum vorherigen Bildschirm zurückkehren
Datensatz wird gelöscht	
Zurückgehen in vorherigen Bildschirm	

### 8.3 Gantt-Diagramm

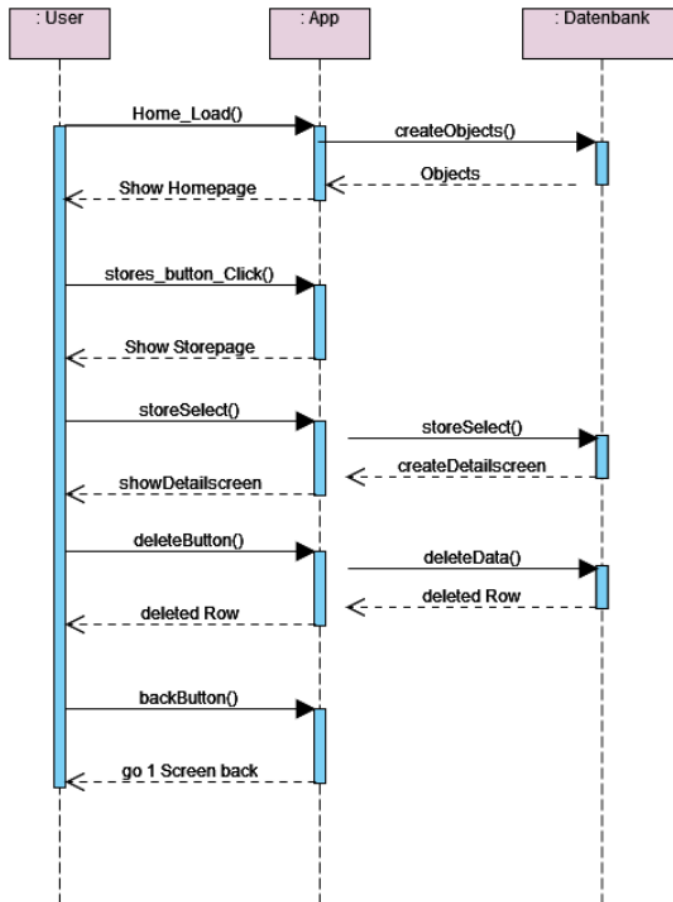
# Gantt-Diagramm Mall



## 8.4 Use-Case-Diagramm



## 8.5 Sequenz-Diagramm



## 8.6 Relationen Modell

