



# DOKUMENTATION ZUR SAVEUP-APP

Modul 335



30. JUNI 2024

IPSO BILDUNG AG  
Nesim Abdelaziz

## Inhaltsverzeichnis

<b>1</b>	<b>Informieren</b>	<b>2</b>
1.1	Ausgangslage .....	2
1.2	Zielsetzung .....	2
1.3	Randbedingungen .....	2
1.4	Ressourcen .....	3
1.5	1.5 Zielgruppe .....	3
<b>2</b>	<b>Planen</b>	<b>3</b>
2.1	Arbeitspakete.....	3
2.1.1	Projektinitialisierung .....	3
2.1.2	Design.....	3
2.1.3	Entwicklung .....	3
2.1.4	Testing .....	3
2.1.5	Dokumentation .....	4
2.1.6	Bereitstellung.....	4
2.2	Zeitplan (Gantt-Diagramm) .....	4
2.2.1	Phase 1: Projektinitialisierung (1 Woche) .....	5
2.2.2	Phase 2: Design (2 Wochen).....	5
2.2.3	Phase 3: Entwicklung (4 Wochen) .....	5
2.2.4	Phase 4: Testing (2 Wochen).....	5
2.2.5	Phase 5: Dokumentation (1 Woche) .....	5
<b>3</b>	<b>Entscheiden</b>	<b>6</b>
3.1	Technologien und Werkzeuge .....	6
3.1.1	NET MAUI.....	6
3.1.2	C# .....	6
3.1.3	Visual Studio.....	6
3.1.4	Design-Tools.....	6
3.2	Entwurfsmuster .....	8
3.2.1	Model-View-ViewModel (MVVM) .....	8
<b>4</b>	<b>Realisieren</b>	<b>8</b>
4.1	Entwicklung der App .....	8
4.1.1	Implementierung der MVVM-Struktur .....	8
4.1.2	Erstellung der Views.....	8
4.1.3	Datenpersistenz .....	8
4.2	Testing .....	8
4.2.1	Unit Tests .....	8
4.2.2	Integrationstests .....	9
4.2.3	UI Tests .....	9
4.3	Dokumentation .....	9
4.3.1	Projektdokumentation .....	9
<b>5</b>	<b>Kontrollieren</b>	<b>9</b>
5.1	Qualitätskontrolle .....	9
5.1.1	Code Reviews .....	9
5.1.2	Testberichte.....	9
5.2	Abnahme.....	9
5.2.1	Abnahmetests.....	9
5.2.2	Feedback.....	9

<b>6</b>	<b>Auswerten</b>	<b>10</b>
6.1	Projektauswertung .....	10
6.1.1	Erfolgsfaktoren .....	10
6.1.2	Herausforderungen .....	11
6.2	Lessons Learned .....	11
6.2.1	Positive Erfahrungen .....	11
6.2.2	Verbesserungspotential .....	12
<b>7</b>	<b>Verkaufspräsentation</b>	<b>12</b>
7.1	Präsentationsstruktur .....	12
7.1.1	Einführung.....	12
7.1.2	Live Demo .....	12
7.1.3	Fazit.....	12
7.2	Präsentationsmittel.....	12
7.2.1	PowerPoint .....	12
7.2.2	Live Demo .....	12
<b>8</b>	<b>Schlussfolgerung</b>	<b>13</b>
8.1	Zusammenfassung .....	13
8.2	Ausblick.....	13

## **1 Informieren**

### **1.1 Ausgangslage**

Die SaveUp-App wird im Rahmen des Modul 335 entwickelt. Ziel der App ist es, Benutzern zu ermöglichen, kleine tägliche Einsparungen zu erfassen und den angesparten Betrag zu überwachen. Diese Einsparungen können beispielsweise für größere private Investitionen wie Urlaub genutzt werden. Die App wird in .NET MAUI realisiert und soll eine benutzerfreundliche Oberfläche bieten.

### **1.2 Zielsetzung**

**Funktionalität:** Die App soll es ermöglichen, Produkte mit einer kurzen Beschreibung und dem Preis zu erfassen und in einer Liste anzuzeigen.

**Benutzerfreundlichkeit:** Die App soll eine einfache und intuitive Bedienung bieten.

**Technologie:** Die App wird unter Verwendung von .NET MAUI und dem MVVM-Entwurfsmuster entwickelt.

**Dokumentation und Testing:** Umfassende Dokumentation und gründliche Tests werden durchgeführt, um die Qualität der App sicherzustellen.

### **1.3 Randbedingungen**

**Plattform:** Die App wird auf Android und iOS laufen.

**Technologie:** .NET MAUI und C# werden als Programmiersprachen verwendet.

**Entwicklungsumgebung:** Visual Studio wird als IDE verwendet.

**Design:** Die App soll ansprechende und konsistente Styles sowie ein eigenes App-Icon haben.

## 1.4 Ressourcen

**Dokumentation:** .NET MAUI Online-Dokumentation

**Design-Tools:** Mockup-Tools wie Balsamiq

**Icons:** Tools wie Android Asset Studio

## 1.5 1.5 Zielgruppe

Die Zielgruppe der App sind Personen, die regelmäßig kleine Beträge einsparen möchten, um für größere Investitionen zu sparen. Diese Benutzer sind an einer einfachen und übersichtlichen Möglichkeit interessiert, ihre Einsparungen zu verfolgen.

## 2 Planen

### 2.1 Arbeitspakete

Die Arbeitspakete werden gemäß dem Gantt-Diagramm geplant und umfassen folgende Schritte:

#### 2.1.1 Projektinitialisierung

- **Projektstruktur aufsetzen**
- **Tools und Frameworks einrichten**

#### 2.1.2 Design

- **Erstellung von Mockups**
- **Definition der UI-Elemente und Navigation**

#### 2.1.3 Entwicklung

- **Implementierung der MVVM-Struktur**
- **Erstellung der Views (Content Pages)**
- **Logik zur Produkterfassung und -anzeige**
- **Datenpersistenz**

#### 2.1.4 Testing

- **Unit Tests für die Geschäftslogik**
- **UI Tests zur Überprüfung der Benutzeroberfläche**

### 2.1.5 Dokumentation

- **Erstellung der Projektdokumentation gemäß IPERKA**
- **Vorbereitung der Verkaufspräsentation**

### 2.1.6 Bereitstellung

- **Veröffentlichung der App im Google Play Store**

## 2.2 Zeitplan (Gantt-Diagramm)

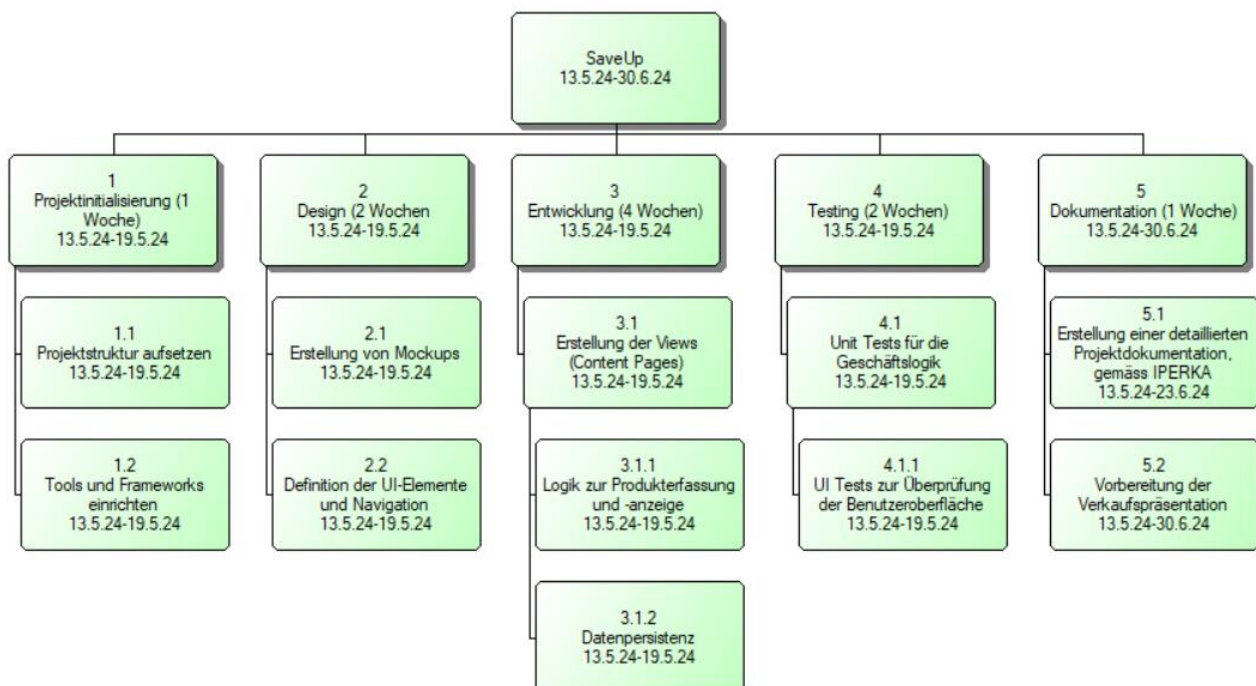


Abbildung 1 PSP vom WBS-Tool

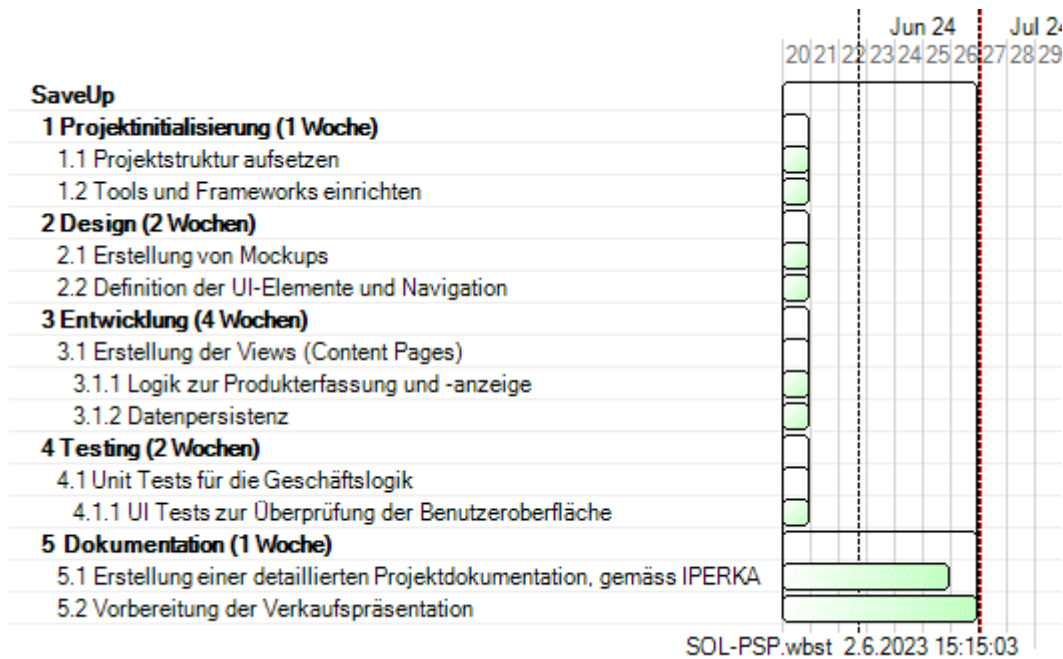


Abbildung 2 GANTT Ansicht vom WBS-Tool

### 2.2.1 Phase 1: Projektinitialisierung (1 Woche)

- Projektstruktur aufsetzen
- Tools und Frameworks einrichten

### 2.2.2 Phase 2: Design (2 Wochen)

- Erstellung von Mockups
- Definition der UI-Elemente und Navigation

### 2.2.3 Phase 3: Entwicklung (4 Wochen)

- Implementierung der MVVM-Struktur
- Erstellung der Views (Content Pages)
- Logik zur Produkterfassung und -anzeige
- Datenpersistenz

### 2.2.4 Phase 4: Testing (2 Wochen)

- Unit Tests für die Geschäftslogik
- UI Tests zur Überprüfung der Benutzeroberfläche

### 2.2.5 Phase 5: Dokumentation (1 Woche)

- Erstellung der Projektdokumentation gemäss IPERKA
- Vorbereitung der Verkaufspräsentation

## **3 Entscheiden**

### **3.1 Technologien und Werkzeuge**

#### **3.1.1 NET MAUI**

.NET MAUI wird als Framework für die Entwicklung der mobilen App verwendet. Es bietet eine plattformübergreifende Lösung und ermöglicht die Erstellung nativer Benutzeroberflächen für Android und iOS.

#### **3.1.2 C#**

C# wird als Programmiersprache verwendet. Es bietet eine starke Typisierung und ist gut in die .NET-Umgebung integriert.

#### **3.1.3 Visual Studio**

Visual Studio dient als integrierte Entwicklungsumgebung (IDE). Es bietet umfassende Tools für die Entwicklung, das Debugging und das Testen von .NET MAUI-Anwendungen.

#### **3.1.4 Design-Tools**

Tools wie Balsamiq werden verwendet, um Mockups und UI-Designs zu erstellen. Diese helfen dabei, die Benutzeroberfläche der App visuell zu planen und zu gestalten.



Abbildung 3 Artikel erfassen (SaveUp-App Mockup)

Abbildung 4 Startseite SaveUp (Mockup)



Abbildung 5 Startseite mit erfasstem Artikel

Abbildung 6 Liste anzeigen (Löschbutton implementiert einzel/alles)



## 3.2 Entwurfsmuster

### 3.2.1 Model-View-ViewModel (MVVM)

Das MVVM-Muster wird verwendet, um eine klare Trennung von Logik und Darstellung zu gewährleisten. Es erleichtert die Wartbarkeit und Testbarkeit der Anwendung.

## 4 Realisieren

### 4.1 Entwicklung der App

#### 4.1.1 Implementierung der MVVM-Struktur

- **Model:** Datenmodelle, die die zu speichernden Produkte repräsentieren.
- **View:** Benutzeroberflächen, die die Interaktion mit dem Benutzer ermöglichen.
- **ViewModel:** Vermittler zwischen Model und View, der die Logik der Anwendung enthält.

#### 4.1.2 Erstellung der Views

- **MainPage:** Startseite der App mit Navigation zu den anderen Seiten.
- **AddProductPage:** Seite zur Erfassung neuer Produkte.
- **ProductListPage:** Seite zur Anzeige der gespeicherten Produkte.

#### 4.1.3 Datenpersistenz

- **Speicherung der Daten:** Implementierung der Funktionalität zur Speicherung der Produkte in einer lokalen Datei oder Datenbank.

### 4.2 Testing

#### 4.2.1 Unit Tests

- **Test der Geschäftslogik:** Überprüfung der korrekten Berechnung und Speicherung der gesparten Beträge.

#### 4.2.2 Integrationstests

- **Sicherstellung der Interaktion:** Überprüfung der korrekten Zusammenarbeit zwischen den Views und dem ViewModel.

#### 4.2.3 UI Tests

- **Benutzerfreundlichkeit:** Test der Benutzeroberfläche auf verschiedenen Geräten und Bildschirmgrößen.

### 4.3 Dokumentation

#### 4.3.1 Projektdokumentation

- **Erstellung der Dokumentation:** Dokumentation aller Entwicklungsphasen gemäß IPERKA.
- **Vorbereitung der Verkaufspräsentation:** Erstellung einer Präsentation zur Vorstellung der App.

## 5 Kontrollieren

### 5.1 Qualitätskontrolle

#### 5.1.1 Code Reviews

- **Überprüfung des Codes:** Regelmäßige Überprüfung des Codes durch Teammitglieder zur Sicherstellung der Codequalität.

#### 5.1.2 Testberichte

- **Erstellung von Testberichten:** Dokumentation der Testergebnisse und identifizierten Fehler.

### 5.2 Abnahme

#### 5.2.1 Abnahmetests

- **Durchführung von Abnahmetests:** Überprüfung, ob alle Anforderungen erfüllt sind und die App korrekt funktioniert.

#### 5.2.2 Feedback

- **Einholung von Feedback:** Sammeln von Feedback von Testbenutzern zur weiteren Verbesserung der App.

## 6 Auswerten

### 6.1 Projektauswertung

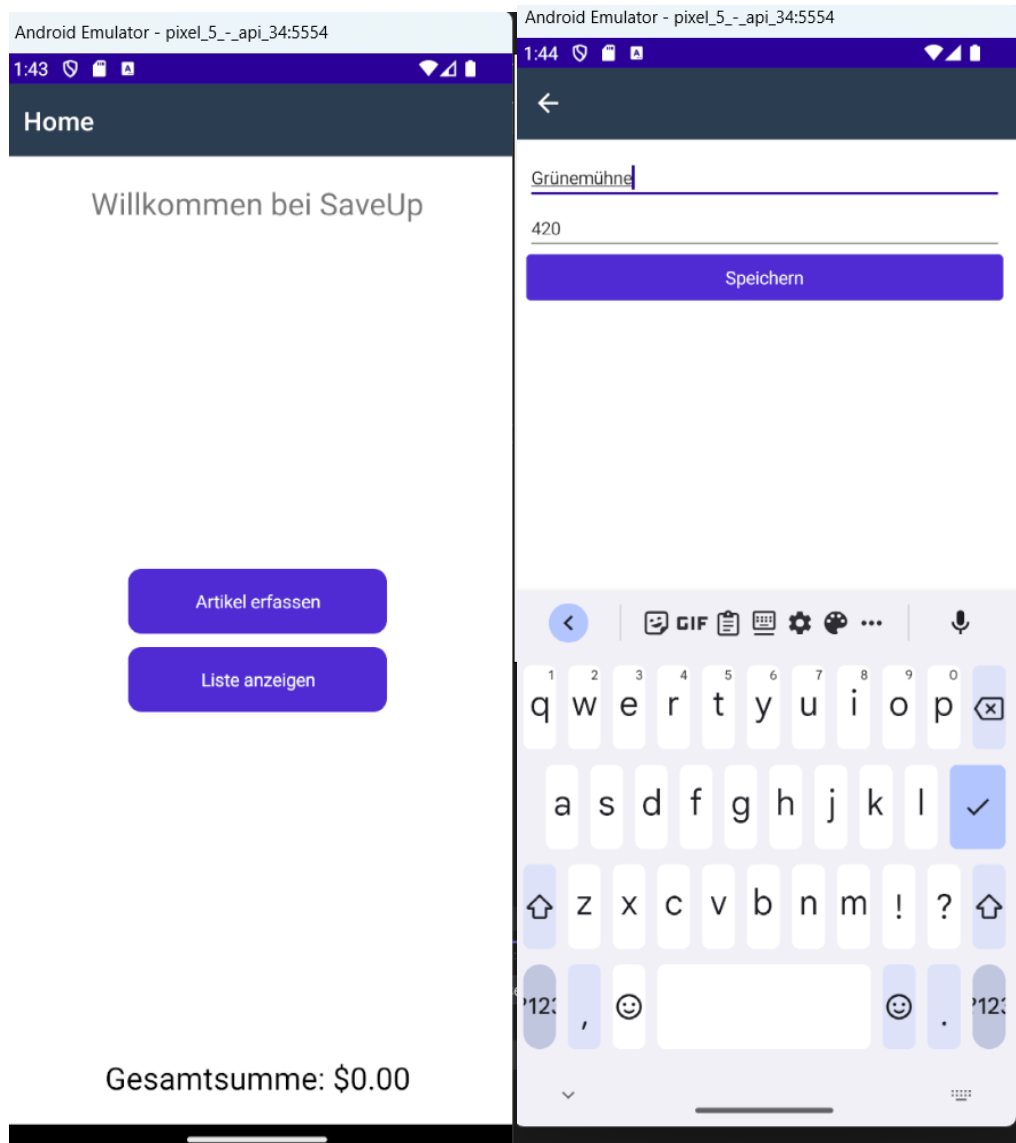


Abbildung 7 Artikel erfassen Seite der SaveUp-App

Abbildung 8 Startseite der SaveUp-App

#### 6.1.1 Erfolgsfaktoren

**Erfüllung der Anforderungen:** Überprüfung, ob alle funktionalen und nicht-funktionalen Anforderungen erfüllt wurden.

**Benutzerfreundlichkeit:** Bewertung der Benutzerfreundlichkeit der App durch Testbenutzer.

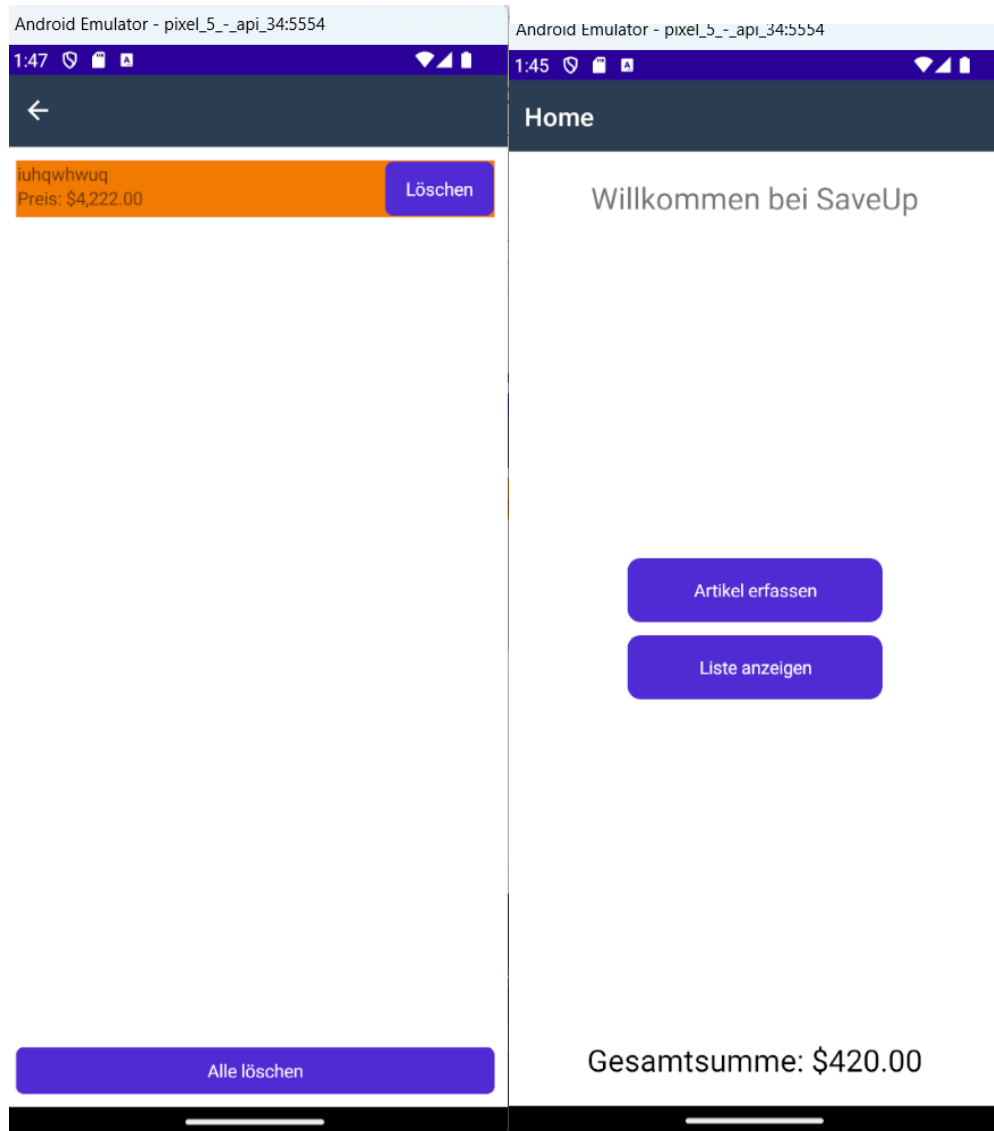


Abbildung 9 Liste anzeigen der SaveUp-App

Abbildung 10 Startseite mit dem erfassten Artikel der SaveUp-App

### 6.1.2 Herausforderungen

**Technische Herausforderungen:** Analyse der technischen Herausforderungen und deren Bewältigung während der Entwicklung.

**Zeitmanagement:** Bewertung der Einhaltung des Zeitplans und Identifizierung von Verzögerungen.

## 6.2 Lessons Learned

### 6.2.1 Positive Erfahrungen

**Erfolgreiche Implementierung:** Positive Aspekte der Implementierung und der Zusammenarbeit im Team.

**Effektive Nutzung von Tools:** Bewertung der verwendeten Tools und Technologien.

## 6.2.2 Verbesserungspotential

**Optimierung des Entwicklungsprozesses:** Identifizierung von Bereichen, die in zukünftigen Projekten verbessert werden können.

**Zusätzliche Funktionalitäten:** Vorschläge für zukünftige Erweiterungen und Verbesserungen der App.

# 7 Verkaufspräsentation

## 7.1 Präsentationsstruktur

### 7.1.1 Einführung

- **Vorstellung des Projekts und der Ziele:** Kurze Einführung in das Projekt und die Hauptziele der SaveUp-App.

### 7.1.2 Live Demo

- **Präsentation der App-Funktionen:** Demonstration der Hauptfunktionen der App.
- **Benutzeroberfläche und Navigation:** Vorstellung der Benutzeroberfläche und der Navigation innerhalb der App.

### 7.1.3 Fazit

- **Zusammenfassung der Erfahrungen:** Reflektion über die Entwicklungsphase und die erzielten Ergebnisse.
- **Ausblick auf zukünftige Erweiterungen:** Vorstellung möglicher zukünftiger Erweiterungen und Verbesserungen der App.

## 7.2 Präsentationsmittel

### 7.2.1 PowerPoint

- **Erstellung einer PowerPoint-Präsentation:** Visuelle Unterstützung für die Verkaufspräsentation.

### 7.2.2 Live Demo

- **Vorführung der App:** Live-Demonstration der App auf einem mobilen Gerät oder Emulator.

## 8 Schlussfolgerung

### 8.1 Zusammenfassung

Die SaveUp-App ermöglicht es Benutzern, ihre täglichen Einsparungen zu erfassen und den gesamten angesparten Betrag zu überwachen. Durch die Verwendung von .NET MAUI und dem MVVM-Muster konnte eine plattformübergreifende und wartbare Lösung entwickelt werden. Die umfassende Dokumentation und das gründliche Testing tragen dazu bei, die Qualität der App sicherzustellen.

### 8.2 Ausblick

Zukünftige Erweiterungen könnten zusätzliche Funktionen wie die Visualisierung der Einsparungen, Erinnerungen zur regelmäßigen Erfassung und die Integration von sozialen Funktionen umfassen. Diese könnten die Benutzerfreundlichkeit und den Mehrwert der App weiter erhöhen.

Abbildung 1 PSP vom WBS-Tool .....	4
Abbildung 2 GANTT Ansicht vom WBS-Tool .....	5
Abbildung 3 Artikel erfassen (SaveUp-App Mockup).....	7
Abbildung 4 Startseite SaveUp (Mockup) .....	7
Abbildung 5 Startseite mit erfasstem Artikel .....	7
Abbildung 6 Liste anzeigen (Löschbutton implementiert einzel/alles) .....	7
Abbildung 7 Artikel erfassen Seite der SaveUp-App .....	10
Abbildung 8 Startseite der SaveUp-App .....	10
Abbildung 9 Liste anzeigen der SaveUp-App .....	11
Abbildung 10 Startseite mit dem erfassten Artikel der SaveUp-App.....	11

<https://github.com/Nesimo05/SaveUp.git>