



---

# SAVEUP

---

Von Davud Ponjevic & Sebastian Thurnbauer



IPSO BILDUNG AG  
[Firmenadresse]

## **1 Technical Summary**

### **1.1 Ausgangslage**

Um finanzielle Mittel für eine grössere private Investition zu sparen, wurde beschlossen, auf kleinere alltägliche Ausgaben zu verzichten. Zu diesen Ausgaben zählen etwa der Kauf von Kaffee oder Süssigkeiten. Diese gesparten Beträge sollen in einer speziell entwickelten App dokumentiert werden. Ziel ist es, jederzeit eine Übersicht über die eingesparten Beträge zu haben, um die Fortschritte in Richtung des Sparziels transparent nachzuvollziehen.

### **1.2 Ziel**

Das Projektziel ist die Entwicklung einer benutzerfreundlichen .NET MAUI-App mit dem Namen SaveUp. Die App soll es den Nutzern ermöglichen:

1. Erfassung von Verzichtprodukten: Es sollen Produkte, auf die bewusst verzichtet wurde, mit einer Kurzbeschreibung und dem entsprechenden Preis eingetragen werden können.
2. Gesamtübersicht: Alle erfassten Produkte werden in einer Liste angezeigt, wobei der angesparte Gesamtbetrag summiert dargestellt wird.
3. Benutzerfreundliche Bedienung: Die App zeichnet sich durch eine einfache, benutzerfreundliche und ansprechende Gestaltung aus. Sie beinhaltet mindestens drei Content Pages und folgt dem MVVM-Entwurfsmuster.
4. Hochwertige Umsetzung: Neben der Funktionalität wird grosser Wert auf die Codestruktur, XAML-Styles für das GUI-Design sowie auf ein individuelles App-Icon gelegt.
5. Dokumentation und Testing: Eine umfassende Dokumentation und Testdurchführung begleiten das gesamte Projekt.

## Inhalt

<b>1</b>	<b>Technical Summary</b>	<b>1</b>
1.1	Ausgangslage .....	1
1.2	Ziel .....	1
<b>2</b>	<b>Initialisierung</b>	<b>3</b>
2.1	Projektbesprechung .....	3
2.2	Zieldefinition .....	3
<b>3</b>	<b>Planung</b>	<b>4</b>
3.1	Erstellung von Mockups .....	4
3.2	Erstellung eines Gantt-Diagramms .....	4
3.3	Recherche .....	4
3.4	Zeitplan und Aufgabenverteilung .....	4
3.4.1	Zeitplan .....	4
3.4.2	Aufgabenverteilung .....	4
3.5	Grafisch Dargestellt .....	5
3.5.1	Mockup .....	5
3.5.2	Gant Diagramm .....	6
3.5.3	Zeitplan und Aufgabenverteilung .....	7
3.5.4	Aufgabenverteilung .....	8
<b>4</b>	<b>Entscheidung</b>	<b>9</b>
4.1.1	Auswahl der Technologien .....	9
4.1.2	Gründe für die Auswahl .....	9
4.1.3	Ergebnis der Entscheidungsphase .....	9
<b>5</b>	<b>Realisieren</b>	<b>10</b>
5.1	Backend-Entwicklung .....	10
5.2	Frontend-Entwicklung .....	10
5.2.1	Funktionale Logik .....	10
5.2.2	Persistenz .....	10
5.2.3	Funktionserweiterungen .....	10
5.3	Tests .....	11
5.4	Ergebnis der Realisierungsphase .....	11
<b>6</b>	<b>Kontrollieren</b>	<b>12</b>
6.1	Funktionale Tests .....	12
6.2	Fehlerbehebung .....	12
6.3	Feedback und Iterationen .....	12
6.4	Ergebnis der Kontrollphase .....	12
<b>7</b>	<b>Auswertung</b>	<b>13</b>
7.1	Fazit .....	13

## 2 Initialisierung

In diesem Schritt haben Sebastian Thurnbauer und Davud Ponjevic das gesamte Projekt analysiert und daraufhin die Planung begonnen. Es war für uns von grosser Bedeutung, diesen Schritt gründlich zu durchführen, um von Anfang an die Grundlage für ein erfolgreiches Projekt zu schaffen. Im Rahmen der Initialisierung wurden folgende Schritte unternommen:

### 2.1 Projektbesprechung

Zunächst haben wir eine umfassende Besprechung durchgeführt, um die Anforderungen und den Umfang des Projekts zu definieren. Dabei haben wir uns damit auseinandergesetzt, was genau umgesetzt werden muss und welche Funktionen die Applikation enthalten soll. Diese Besprechung diente dazu, ein gemeinsames Verständnis über die Ziele und Herausforderungen des Projekts zu schaffen.

### 2.2 Zieldefinition

Im Anschluss an die Besprechung haben wir die konkreten Ziele des Projekts definiert. Die Zielsetzung wurde in zwei Bereiche unterteilt:

#### 1. **Primäre Ziele:**

- Entwicklung einer benutzerfreundlichen .NET MAUI-App mit den zentralen Funktionen zur Erfassung und Darstellung von Verzichtsprodukten.
- Implementierung einer Übersicht, die den gesamten angesparten Betrag zeigt.
- Einhaltung des MVVM-Entwurfsmusters zur Strukturierung der App.

#### 2. **Sekundäre Ziele** (optional):

- Persistente Datenspeicherung über lokale Dateien oder ein Backend.
- Erweiterte Funktionalitäten wie das Löschen von Einträgen und grafische Darstellungen.

### 3 Planung

Die Planungsphase ist entscheidend, um den Projekterfolg sicherzustellen, indem alle Schritte, Ressourcen und Verantwortlichkeiten klar definiert werden. In dieser Phase wurden folgende Schritte durchgeführt:

#### 3.1 Erstellung von Mockups

Um eine klare Vorstellung vom Aufbau und Design der Anwendung zu erhalten, wurden Mockups und Wireframes erstellt. Diese visualisierten die Struktur der App, die Navigation zwischen den Seiten und die Darstellung der Hauptfunktionen. Ziel war es, ein benutzerfreundliches und ansprechendes Layout zu entwerfen. Diese Aufgabe wurde von **Davud Ponjevic** übernommen und nahm **2 Stunden** in Anspruch. Das Mockup sehen wir am Ende dieses Kapitels.

#### 3.2 Erstellung eines Gantt-Diagramms

Zur besseren Organisation und Übersichtlichkeit der einzelnen Aufgaben wurde ein detailliertes Gantt-Diagramm entwickelt. Es zeigt die Abfolge aller Projektphasen, dies half dabei, Meilensteine zu identifizieren und den Fortschritt regelmässig zu kontrollieren. Auch diese Aufgabe wurde von **Davud Ponjevic** durchgeführt.

Das Gantt sehen wir am Ende dieses Kapitels mit allen Arbeitspaketen.

#### 3.3 Recherche

Ein wichtiger Bestandteil der Planungsphase war die Analyse ähnlicher Apps und die Recherche von Best Practices. Dies half dabei, Inspiration für die Umsetzung zu gewinnen und bewährte Methoden in das Projekt einfließen zu lassen. Diese Aufgabe wurde von **Sebastian Thurnbauer** und **Davud Ponjevic** gemeinsam erledigt und dauerte **1 Stunde**.

Die Planungsphase legte somit den Grundstein für die technische und organisatorische Umsetzung des Projekts, indem sie klare Ziele, Designs und Methoden vorgab.

#### 3.4 Zeitplan und Aufgabenverteilung

Nachdem das Gantt-Diagramm in der Planungsphase erstellt wurde, haben wir die Aufgabenverteilung vorgenommen. Dies ermöglichte uns, die Verantwortlichkeiten klar zu definieren und den zeitlichen Ablauf des Projekts zu optimieren. Jede Aufgabe wurde basierend auf den jeweiligen Stärken und Präferenzen der Teammitglieder zugewiesen.

##### 3.4.1 Zeitplan

Der Zeitplan diente als Leitfaden für die gesamte Projektlaufzeit. Er umfasste die zeitliche Einordnung aller Phasen – von der Initialisierung bis hin zur abschliessenden Präsentation. Dabei wurden die Dauer jeder Aufgabe und die Meilensteine des Projekts exakt definiert, um den Fortschritt effektiv verfolgen zu können.

##### 3.4.2 Aufgabenverteilung

Die Aufgabenverteilung erfolgte auf Basis des Gantt-Diagramms, nachdem die Gesamtstruktur und der zeitliche Rahmen festgelegt waren.

### 3.5 Grafisch Dargestellt

#### 3.5.1 Mockup



Abbildung 1 Landing Page

Das ist die Landing Page

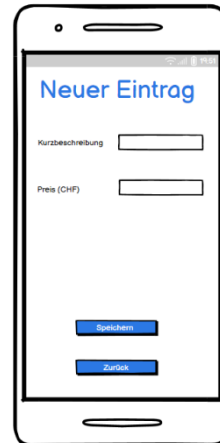
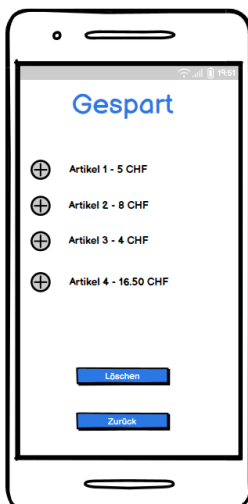


Abbildung 2 Neuer Eintrag

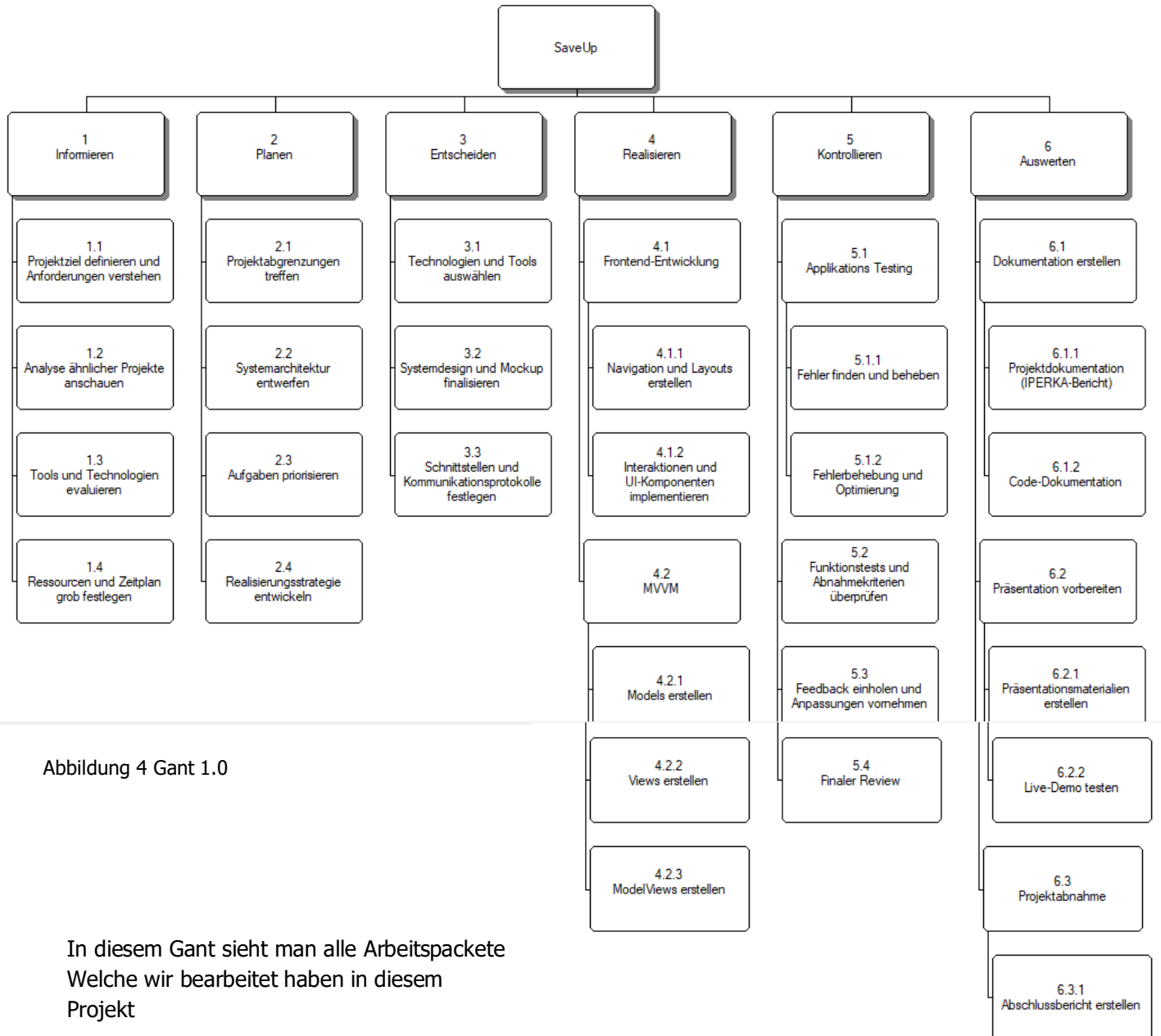
Das ist die neuer Eintrag Seite hier fügt man seine Sachen aus



Das ist die Gespart Seite wo man seine Einträge sehen kann

Abbildung 3 Gespart Seite

### 3.5.2 Gant Diagramm



### 3.5.3 Zeitplan und Aufgabenverteilung

Phase	Aufgabe	Dauer (in Std.)	Verantwortlich
Initialisierung	Projektbesprechung und Zieldefinition	1	Beide
Planung	Mockups/Wireframes/ GANT erstellen	2	Davud Ponjevic
Recherche	Analyse von ähnlichen Apps und Best Practices	1	Beide
Umsetzung: Backend	Models und ViewModels erstellen	3	Sebastian Thurnbauer
Umsetzung: Frontend	Layout und Stile für Content Pages erstellen	2.5	Davud Ponjevic
Funktionale Logik	Bindings und Navigation implementieren	2	Beide
Persistenz	Speicherung in JSON-Dateien	2	Sebastian Thurnbauer
Funktion zum Löschen	Lösch-Funktion für Einträge implementieren	1.5	Sebastian Thurnbauer
Tests	Funktionale Tests und Fehlerbehebung	2	Beide
Dokumentation	IPERKA-Dokumentation und abschließende Analyse	2	Beide
Erweiterungen	Grafische Darstellung der Einsparungen	2	Sebastian Thurnbauer
Präsentation vorbereiten	Live-Demo vorbereiten	1.5	Beide
Feedback und Anpassungen	Iteration basierend auf Feedback	1	Beide

Abbildung 6 Zeitplan

Das war unser Zeitplan, nach dem wir uns gerichtet haben



### 3.5.4 Aufgabenverteilung

<b>1. Informieren</b>	Projektziel definieren und Anforderungen verstehen	<b>Davud Ponjevic</b>
	Analyse ähnlicher Projekte anschauen	<b>Sebastian Thurnbauer</b>
	Tools und Technologien evaluieren	<b>Davud Ponjevic</b>
	Ressourcen und Zeitplan grob festlegen	<b>Sebastian Thurnbauer</b>
<b>2. Planen</b>	Projektabgrenzungen treffen	<b>Sebastian Thurnbauer</b>
	Systemarchitektur entwerfen	<b>Davud Ponjevic</b>
	Aufgaben priorisieren	<b>Davud Ponjevic</b>
	Realisierungsstrategie entwickeln	<b>Sebastian Thurnbauer</b>
<b>3. Entscheiden</b>	Technologien und Tools auswählen	<b>Davud Ponjevic</b>
	Systemdesign und Mockup finalisieren	<b>Sebastian Thurnbauer</b>
	Schnittstellen und Kommunikationsprotokolle festlegen	<b>Davud Ponjevic</b>

Abbildung 7 Aufgabenverteilung

<b>4. Realisieren</b>	<b>Frontend-Entwicklung</b>	
	- Navigation und Layouts erstellen	<b>Sebastian Thurnbauer</b>
	- Interaktionen und UI-Komponenten implementieren	<b>Davud Ponjevic</b>
	<b>MVVM</b>	
	- Models erstellen	<b>Sebastian Thurnbauer</b>
	- Views erstellen	<b>Davud Ponjevic</b>
	- ModelViews erstellen	<b>Davud Ponjevic</b>
	<b>Testing</b>	
	- Fehler finden und beheben	<b>Sebastian Thurnbauer</b>
	- Fehlerbehebung und Optimierung	<b>Davud Ponjevic</b>
<b>5. Kontrollieren</b>	Funktionstests und Abnahmekriterien überprüfen	<b>Sebastian Thurnbauer</b>
	Feedback der Stakeholder einholen und Anpassungen vornehmen	<b>Davud Ponjevic</b>
	Qualitätskontrolle und finaler Review	<b>Sebastian Thurnbauer</b>
<b>6. Auswerten</b>	<b>Dokumentation erstellen</b>	
	- Projektdokumentation (IPERKA-Bericht)	<b>Davud Ponjevic</b>
	- Code-Dokumentation	<b>Sebastian Thurnbauer</b>
	<b>Präsentation vorbereiten</b>	
	- Präsentationsmaterialien erstellen	<b>Davud Ponjevic</b>
	- Live-Demo testen	<b>Sebastian Thurnbauer</b>
	<b>Projektabschluss</b>	
	- Abschlussbericht erstellen	<b>Sebastian Thurnbauer</b>

Abbildung 8 Aufgabenverteilung 2.0

## 4 Entscheidung

### 4.1.1 Auswahl der Technologien

Nach einer sorgfältigen Analyse der Projektanforderungen und möglicher technischer Lösungen haben wir uns für folgende Technologien entschieden;

- **.NET MAUI:**  
Als plattformübergreifendes Framework ermöglicht .NET MAUI die Entwicklung einer einzigen Codebasis für Android, iOS und Windows. Dies spart Entwicklungszeit und gewährleistet eine konsistente Benutzererfahrung auf allen Plattformen.
- **C# und MVVM-Entwurfsmuster:**  
C# wurde als Hauptprogrammiersprache gewählt, da es eine solide Grundlage für die Entwicklung mit .NET MAUI bietet. Das MVVM-Entwurfsmuster wurde verwendet, um eine klare Trennung von Logik und Benutzeroberfläche zu gewährleisten. Dies erleichtert die Wartbarkeit und Erweiterbarkeit der Anwendung.
- **JSON-Dateien für Datenspeicherung:**  
Um die Persistenz der Daten zu gewährleisten, fiel die Entscheidung auf JSON-Dateien. Diese bieten eine einfache, leicht implementierbare und plattformunabhängige Möglichkeit zur Speicherung von Benutzerdaten ohne die Notwendigkeit eines Backends.
- **Entwicklungsumgebung:**  
Die Wahl der Entwicklungsumgebung fiel auf **Visual Studio**, da es umfassende Tools für die Entwicklung mit .NET MAUI bietet, einschliesslich Debugging, Emulatoren und integriertem Git-Support.

### 4.1.2 Gründe für die Auswahl

1. **Effizienz und Flexibilität:** Mit .NET MAUI konnten wir eine plattformübergreifende Lösung entwickeln, ohne separaten Code für jede Plattform schreiben zu müssen.
2. **Erfahrung und Kompetenz:** Wir verfügten bereits über Kenntnisse in C# und dem MVVM-Muster, was die Einarbeitungszeit minimierte.
3. **Einfache Persistenzlösung:** JSON-Dateien boten eine unkomplizierte Möglichkeit, die Anforderungen an Datenspeicherung zu erfüllen, ohne die Komplexität eines Backends einzuführen.
4. **Bewährte Tools:** Visual Studio war die ideale Wahl für die Entwicklung, da es umfassende Unterstützung für .NET-Technologien bietet.

### 4.1.3 Ergebnis der Entscheidungsphase

Mit diesen Entscheidungen haben wir die Basis für die technische Umsetzung des Projekts geschaffen. Die ausgewählten Technologien passten perfekt zu den Anforderungen und ermöglichten eine effiziente und strukturierte Entwicklung der Anwendung.

## 5 Realisieren

Die Realisierungsphase war der Kern des Projekts, in dem die zuvor geplante und definierte Anwendung Schritt für Schritt entwickelt wurde. Sie umfasste sowohl die Implementierung der technischen Grundlagen als auch die Umsetzung der funktionalen und visuellen Anforderungen.

### 5.1 Backend-Entwicklung

Der erste Schritt in der Realisierungsphase war die Erstellung des Backends, das die Datenverarbeitung und -verwaltung der App steuert.

- **Models und ViewModels:**

Die Datenstruktur der App wurde in Form von Models definiert, um die Verzichtprodukte und Einsparungen abzubilden. Die ViewModels dienten als Vermittler zwischen der Benutzeroberfläche und den Daten. Diese Aufgabe wurde von **Sebastian Thurnbauer** übernommen und dauerte **3 Stunden**.

### 5.2 Frontend-Entwicklung

Im nächsten Schritt wurde die Benutzeroberfläche gestaltet und die Grundstruktur der App erstellt.

- **Layouts und Stile:**

Die Hauptseiten der App, wie die Übersicht der Verzichtprodukte und die Einsparungsübersicht, wurden entworfen. Dabei wurde ein klares, benutzerfreundliches Design mit konsistenten Farben und Schriften umgesetzt. Diese Aufgabe übernahm **Davud Ponjevic** und benötigte **2,5 Stunden**.

#### 5.2.1 Funktionale Logik

Die logischen Verbindungen zwischen den Elementen wurden in diesem Schritt implementiert:

- **Bindings und Navigation:**

Die Daten wurden mithilfe von Bindings zwischen den ViewModels und der Benutzeroberfläche verknüpft. Eine intuitive Navigation zwischen den verschiedenen Content Pages wurde eingerichtet. Diese Aufgabe wurde von **beiden** gemeinsam durchgeführt und dauerte **2 Stunden**.

#### 5.2.2 Persistenz

Um die Daten auch nach dem Schliessen der App verfügbar zu machen, wurde eine einfache Persistenzlösung implementiert:

- **JSON-Dateien zur Datenspeicherung:**

Die Verzichtprodukte und Einsparungen werden in JSON-Dateien gespeichert, die lokal auf dem Gerät abgelegt werden. Diese Lösung wurde von **Sebastian Thurnbauer** umgesetzt und nahm **2 Stunden** in Anspruch.

#### 5.2.3 Funktionserweiterungen

In der Realisierungsphase wurden auch erweiterte Funktionen hinzugefügt:

- **Löschfunktion:**

Eine Funktion, mit der Benutzer Einträge aus der Liste entfernen können, wurde implementiert. Diese Funktion wurde von **Sebastian Thurnbauer** entwickelt und benötigte **1,5 Stunden**.

- **Grafische Darstellung:**

Einsparungen wurden in einer grafischen Übersicht visualisiert, um dem Benutzer eine anschauliche Darstellung seiner Erfolge zu bieten. Diese Erweiterung wurde ebenfalls von **Sebastian Thurnbauer** umgesetzt und dauerte **2 Stunden**.

### 5.3 Tests

Nach der Implementierung der Funktionen wurde die App intensiv getestet:

- **Funktionale Tests:**

Jede Funktion wurde überprüft, um sicherzustellen, dass sie korrekt arbeitet. Fehler und Probleme wurden identifiziert und behoben. Die Tests wurden von **beiden** durchgeführt und dauerten **2 Stunden**.

### 5.4 Ergebnis der Realisierungsphase

Die Realisierungsphase schloss mit einer vollständig funktionsfähigen Version der App ab, die alle geplanten Features implementierte. Der strukturierte Ansatz und die klare Aufgabenverteilung ermöglichten eine effiziente Umsetzung und legten die Basis für die abschliessenden Phasen des Projekts.

## 6 Kontrollieren

Die Kontrollphase diente dazu, die Qualität und Funktionalität der entwickelten Anwendung zu überprüfen und sicherzustellen, dass alle Anforderungen des Projekts erfüllt wurden. Sie war ein essenzieller Schritt, um Fehler zu erkennen, die Benutzerfreundlichkeit zu verbessern und das Produkt für die abschliessende Präsentation vorzubereiten.

### 6.1 Funktionale Tests

In dieser Phase wurden alle Funktionen der Anwendung auf ihre korrekte Funktionsweise getestet. Dabei konzentrierten wir uns auf die folgenden Bereiche:

- **Eingabefelder und Validierungen:**  
Überprüfung, ob die Benutzer korrekte Daten eingeben können und ob ungültige Eingaben entsprechend abgefangen werden.
- **Navigation und Benutzerfluss:**  
Sicherstellung, dass die Navigation zwischen den Seiten intuitiv und fehlerfrei funktioniert.
- **Datenpersistenz:**  
Test der JSON-basierten Speicherung, um sicherzustellen, dass die Daten korrekt gespeichert und beim Neustart der App geladen werden.
- **Löschfunktion:**  
Überprüfung, ob Einträge zuverlässig aus der Liste entfernt und die Daten korrekt aktualisiert werden.

Die funktionalen Tests wurden von **Sebastian Thurnbauer** und **Davud Ponjevic** gemeinsam durchgeführt und dauerten **2 Stunden**.

### 6.2 Fehlerbehebung

Während der Tests identifizierten wir kleinere Probleme, die behoben werden mussten:

- **Darstellungsfehler im Frontend:**  
Anpassungen an Layout und Design, um eine konsistente und benutzerfreundliche Oberfläche sicherzustellen.
- **Kleine Logikfehler:**  
Behebung von Unstimmigkeiten in den Bindings und der Navigation.

### 6.3 Feedback und Iterationen

Nachdem die internen Tests abgeschlossen waren, wurde die Anwendung anderen Personen gezeigt, um Feedback einzuholen. Auf Basis des Feedbacks wurden folgende Änderungen vorgenommen:

- Verbesserte grafische Darstellung der Einsparungen, um sie noch übersichtlicher zu gestalten.
  - Kleinere Anpassungen an der Benutzerführung, um die Bedienung noch intuitiver zu machen.
- Die Iterationen basierend auf dem Feedback dauerten **1 Stunde** und wurden von **beiden** gemeinsam durchgeführt.

### 6.4 Ergebnis der Kontrollphase

Am Ende der Kontrollphase war die Anwendung vollständig getestet und optimiert. Alle identifizierten Probleme wurden behoben, und die Benutzerfreundlichkeit wurde durch das Feedback verbessert. Die App war bereit für die Präsentation und erfüllte alle funktionalen und qualitativen Anforderungen.

## **7 Auswertung**

### **7.1 Fazit**

Im gross und ganzen haben wir gut zusammengearbeitet und eine goldene Mitte getroffen für die Abtrennung unserer Aufgaben. Bis auf ein paar kleine Fehler, die wir innerhalb kürzester Zeit beheben konnten, sind wir auf keine grösseren Hindernisse gestossen. Wir hätten vielleicht die Zeit früher einteilen sollen und ein noch besseres Projekt machen, aber wir sind mit unserem Ergebnis zufrieden.