



MODUL 322

SaveUp-APP

Benutzerschnittstellen entwerfen und
implementieren

Autor: Ulu, Ege Tuna
Projektmitarbeiter: Brun, Lenny Ulu, Ege Tuna Atici, Beren Mesic, David
Klasse: Applikationsentwicklung Basel
Schule: IBZ-Basel
Datum: 18.01.2025

Table of Contents

1. EINLEITUNG	3
1.1 AUSGANGSLAGE	3
1.2 ZIELSETZUNG	3
1.3 AUFBAU DER DOKUMENTATION	3
2. PROJEKTPLANUNG.....	3
2.1 ORGANISATION.....	3
2.2 ZEITPLAN	4
2.2.1 AUFGABENVERTEILUNG	4
2.2.2 GANTT-DIAGRAMM.....	4
3. ANFORDERUNGEN.....	6
3.1 PFLICHTANFORDERUNGEN	6
3.2 ZUSÄTZLICHE ANFORDERUNGEN	6
3.3 RANDBEDINGUNGEN.....	6
4. PROJEKTUMSETZUNG	7
4.1 DATENBANKDESIGN	7
4.1.1 DATENSTRUKTUR	7
4.1.2 JSON-SCHEMA	7
4.2 GUI-ENTWICKLUNG	7
4.2.1 MOCKUPS UND WIREFRAMES	7
4.2.2 XAML-STYLES UND LAYOUTS	9
4.3 IMPLEMENTIERUNG DER ANFORDERUNGEN	9
4.3.1 PRODUKTEINGABE	9
4.3.2 ANZEIGE DER PRODUKTLISTE	9
4.3.3 BERECHNUNG DER GESAMTEINSPARUNG	9
4.3.4 PERSISTENZ IN JSON-DATEIEN	9
4.3.5 PAGES	9
5. TEST UND VALIDIERUNG	12
5.1 TESTPLAN.....	12

5.1.1 MANUELLE TESTS.....	12
5.1.2 AUTOMATISIERTE TESTS	12
5.2 TESTERGEBNISSE	12
5.3 FEHLERBEHEBUNG	12
 6. PROJEKTMANAGEMENT.....	 12
 6.1 IPERKA-SCHRITTE	 12
6.1.1 INFORMIEREN	12
6.1.2 PLANEN.....	13
6.1.3 ENTSCHEIDEN.....	13
6.1.4 REALISIEREN.....	13
6.1.5 KONTROLLIEREN.....	13
6.1.6 AUSWERTEN	13
 7. FAZIT UND AUSBLICK	 14
 7.1 LESSONS LEARNED.....	 14
7.2 VERBESSERUNGSPOTENZIALE	14
7.3 AUSBLICK	14

1. Einleitung

1.1 Ausgangslage

Immer mehr Menschen versuchen, durch gezieltes Sparen größere Investitionen zu finanzieren, wie beispielsweise Reisen, technische Geräte oder andere persönliche Ziele. Häufig werden dabei kleine, tägliche Ausgaben reduziert oder ganz vermieden, etwa der Kauf von Kaffee, Snacks oder ähnlichen Produkten. Eine systematische Erfassung dieser Einsparungen fehlt jedoch oft, was dazu führt, dass das volle Potenzial solcher Sparmaßnahmen nicht genutzt wird. Die SaveUp-App bietet eine digitale Lösung, um solche Einsparungen effizient und übersichtlich zu dokumentieren.

1.2 Zielsetzung

Das Ziel dieses Projekts ist die Entwicklung einer .NET MAUI-App, die es Nutzern ermöglicht, Einsparungen in Form von Produkten mit Preis und Datum zu erfassen und übersichtlich darzustellen. Eine intuitive Benutzeroberfläche und einfache Bedienung stehen hierbei im Fokus, sodass die Anwendung sowohl für technisch versierte als auch für weniger erfahrene Benutzer geeignet ist. Neben der Erfassung von Einträgen soll die App eine Summierung der Gesamteinsparungen und zusätzliche Funktionen wie das Löschen von Einträgen bieten.

1.3 Aufbau der Dokumentation

Diese Dokumentation beschreibt den gesamten Entwicklungsprozess der SaveUp-App. Sie umfasst die Analyse der Anforderungen, die detaillierte Projektplanung, die technischen Umsetzungsschritte, die Test- und Validierungsphasen sowie eine abschließende Reflexion der Ergebnisse.

2. Projektplanung

2.1 Organisation

Die Organisation des Projekts basiert auf einem klar definierten Zeitplan und einer detaillierten Aufgabenverteilung. Durch die Verwendung agiler Methoden konnte

2.2 Zeitplan

2.2.1 Aufgabenverteilung

- **Datenbankdesign:** Erstellung eines JSON-Schemas zur Speicherung der Einsparungsdaten.
- **GUI-Design:** Entwicklung der Benutzeroberfläche und Definition der XAML-Styles.
- **Implementierung:** Umsetzung der Produkteingabe, der Listenansicht und der Gesamtsummenberechnung.
- **Testing und Validierung:** Erstellung eines umfassenden Testplans und Durchführung von Tests zur Sicherstellung der Funktionalität.
- **Dokumentation:** Verfassen der Dokumentation und abschließende Reflexion.

Ein Gantt-Diagramm wurde verwendet, um die Aufgaben zeitlich zu planen und zu visualisieren. Der Plan gliedert sich in folgende Phasen:

[illegible]

3. Anforderungen

3.1 Pflichtanforderungen

Die Anwendung muss folgende Anforderungen erfüllen:

- **Produktdatenerfassung:** Benutzer können Produktname, Preis und Datum eingeben.
- **Datenübersicht:** Alle gespeicherten Einträge werden in einer Liste angezeigt.
- **Gesamtsumme:** Die Gesamteinsparung wird dynamisch berechnet und angezeigt.
- **Datenpersistenz:** Alle Daten werden lokal in einer JSON-Datei gespeichert und beim Neustart der App geladen.
- **Benutzerfreundliches Design:** Einfache Bedienung und ein ansprechendes Layout.
- **MVVM-Architektur:** Klare Trennung zwischen Logik und Darstellung.

3.2 Zusätzliche Anforderungen

Optionale Erweiterungen, die zusätzlichen Mehrwert bieten:

- **Löschfunktion:** Einträge können einzeln oder komplett gelöscht werden.
- **Diagramm:** Darstellung der Einsparungen in Form eines Balken- oder Liniendiagramms.
- **Mehrsprachigkeit:** Unterstützung von Deutsch und Englisch.
- **Cloud-Anbindung:** Speicherung der Daten in einer Cloud-Datenbank.

3.3 Randbedingungen

Das Projekt unterliegt folgenden Einschränkungen:

- Die App muss mit .NET MAUI entwickelt werden.
- Es dürfen keine Lösungen direkt aus dem Internet kopiert werden.
- Die Funktionalität muss ohne externe Bibliotheken auskommen.

4. Projektumsetzung

4.1 Datenbankdesign

4.1.1 Datenstruktur

Die Speicherung erfolgt in einer JSON-Datei, um die Persistenz der Daten zu gewährleisten.

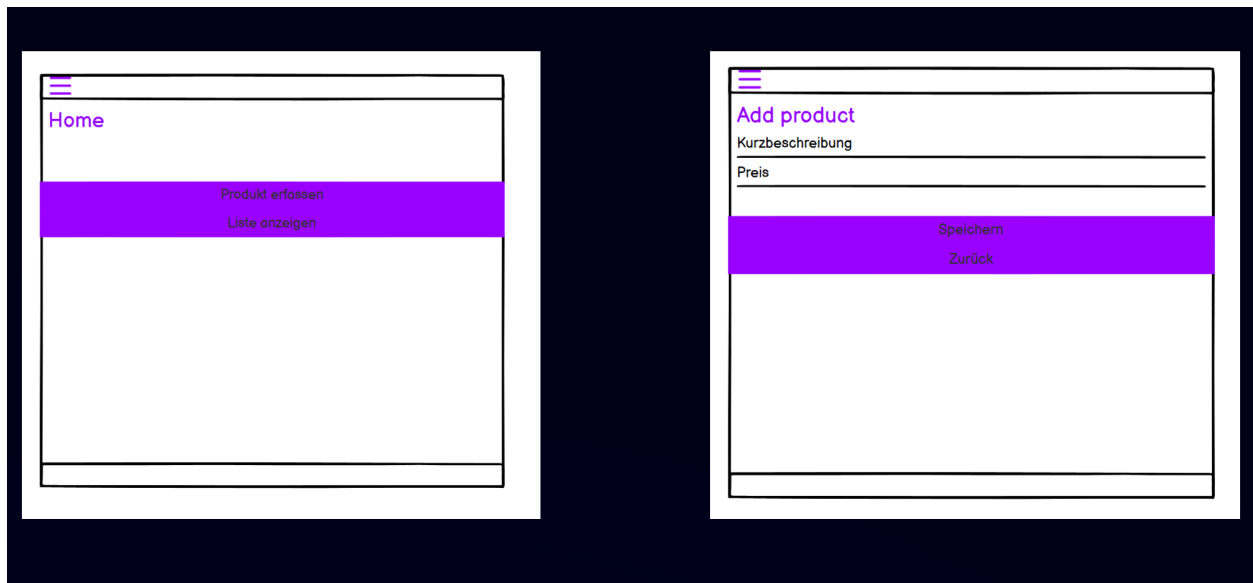
4.1.2 JSON-Schema

Das Schema umfasst Felder für Beschreibung, Preis und Datum. Zusätzlich wurde ein Index eingebaut, um Einträge eindeutig identifizieren zu können.

4.2 GUI-Entwicklung

4.2.1 Mockups und Wireframes

Um die Benutzeroberfläche zu planen, wurden detaillierte Mockups erstellt. Diese zeigen das Layout der Startseite, der Eingabeseite und der Listenansicht.



Home

Produkt erfassen

Liste anzeigen

Add product

Kurzbeschreibung

Preis

Speichern

Zurück

4.2.2 XAML-Styles und Layouts

Die Benutzeroberfläche wurde mit XAML entwickelt. Styles wurden genutzt, um konsistente Farben, Schriftarten und Abstände zu gewährleisten.

4.3 Implementierung der Anforderungen

4.3.1 Produkteingabe

Benutzer können Produktname, Preis und Datum eingeben. Eine Validierung stellt sicher, dass alle Felder korrekt ausgefüllt sind.

4.3.2 Anzeige der Produktliste

Die Liste zeigt alle gespeicherten Produkte an. Filter- und Suchfunktionen erleichtern die Navigation.

4.3.3 Berechnung der Gesamteinsparung

Die Gesamtsumme wird automatisch berechnet und in Echtzeit aktualisiert.

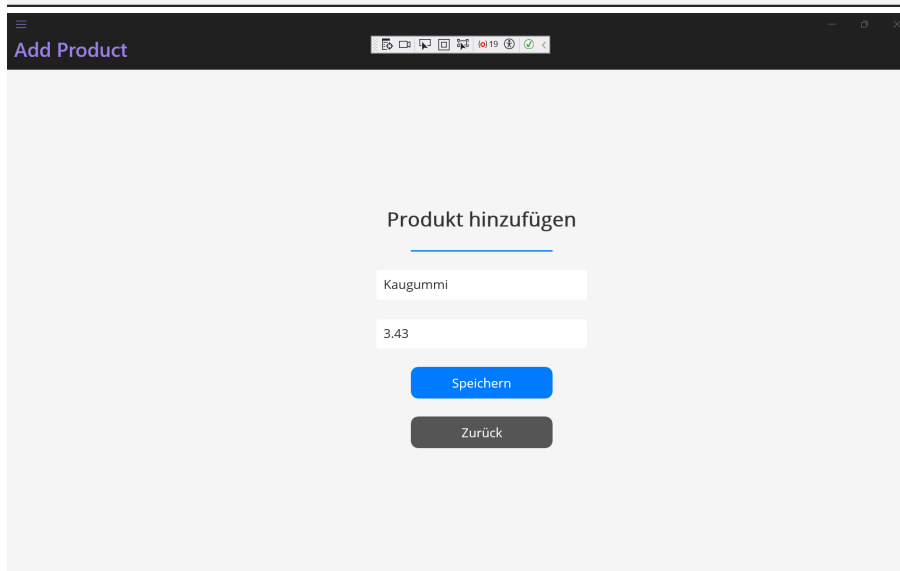
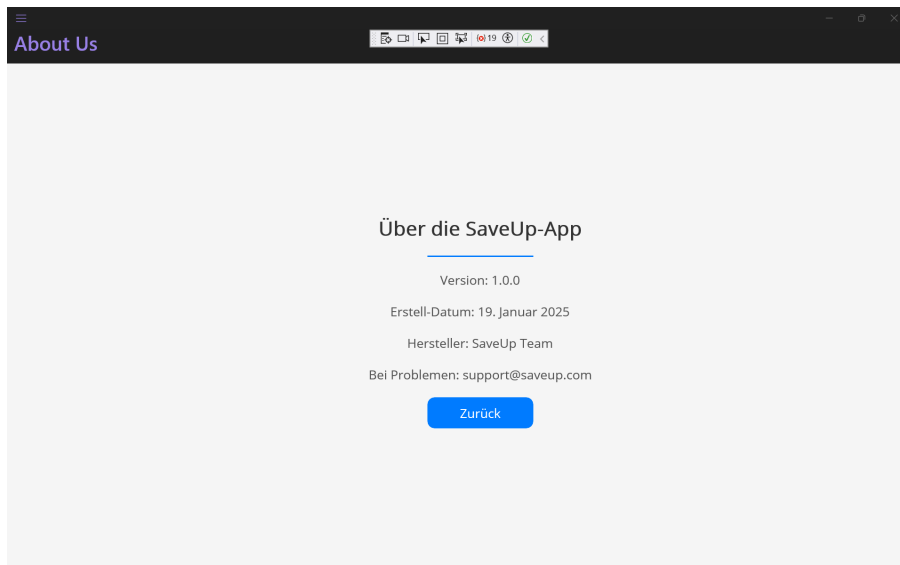
4.3.4 Persistenz in JSON-Dateien

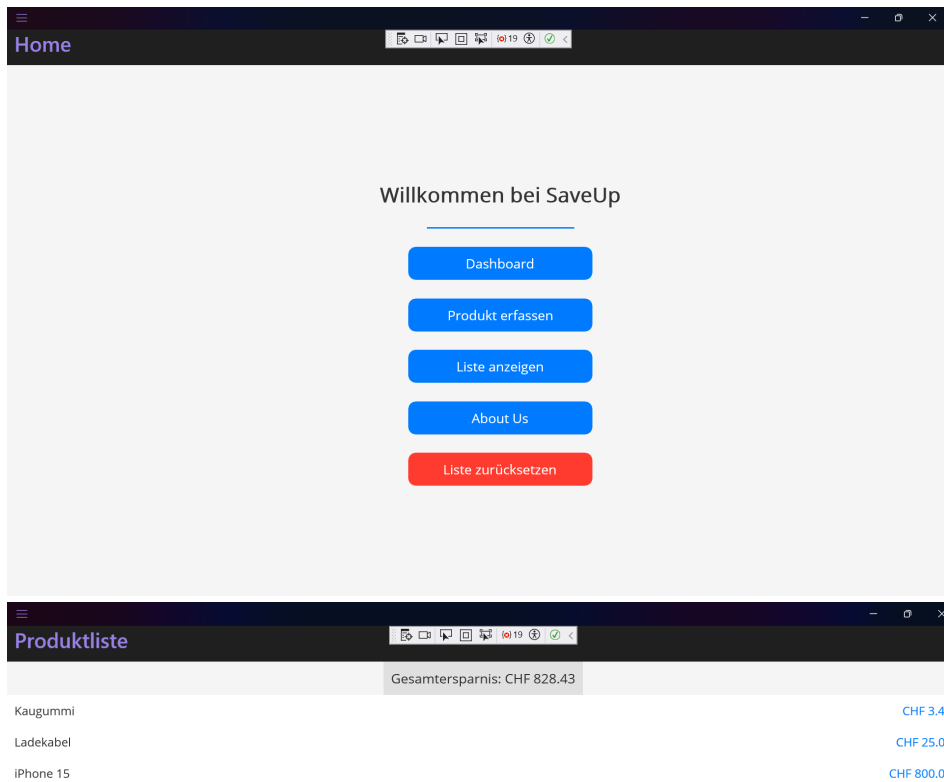
Die Daten werden beim Speichern in einer JSON-Datei abgelegt und beim Neustart der App geladen.

4.3.5 Pages

In dem Projekt gibt es 5 Pages, die hier aufgelistet sind:

- AboutUsPage
- AddProductPage
- DashboardPage
- MainPage
- ProductListPage





5. Test und Validierung

5.1 Testplan

5.1.1 Manuelle Tests

Manuelle Tests wurden für die Kernfunktionen durchgeführt, darunter:

- Hinzufügen neuer Produkte.
- Aktualisierung der Gesamtsumme.
- Löschen von Einträgen.

5.1.2 Automatisierte Tests

Automatisierte Tests wurden mit NUnit durchgeführt. Die Tests konzentrierten sich auf die Berechnung der Gesamtsumme und die Validierung der Eingaben.

5.2 Testergebnisse

Alle Funktionen der App wurden erfolgreich getestet. Kleinere Fehler, wie unzureichende Validierungen, wurden behoben.

5.3 Fehlerbehebung

Fehler, die während der Testphase identifiziert wurden, umfassten:

- Probleme bei der JSON-Speicherung.
- Falsche Berechnungen bei ungültigen Eingaben.

6. Projektmanagement

6.1 IPERKA-Schritte

6.1.1 Informieren

Die Anforderungen wurden analysiert, und passende Technologien wurden ausgewählt.

6.1.2 Planen

Ein realistischer Zeitplan wurde erstellt, und die Aufgaben wurden klar verteilt.

6.1.3 Entscheiden

Die Entscheidung für .NET MAUI und JSON als Speichermedium wurde aufgrund der Flexibilität und Einfachheit getroffen.

6.1.4 Realisieren

Die Umsetzung erfolgte iterativ, um Fehler frühzeitig zu erkennen und zu beheben.

6.1.5 Kontrollieren

Die Tests stellten sicher, dass die App den Anforderungen entspricht.

6.1.6 Auswerten

Die Ergebnisse wurden dokumentiert, und Verbesserungsmöglichkeiten wurden identifiziert.

7. Fazit und Ausblick

7.1 Lessons Learned

Die strukturierte Planung und regelmäßige Tests waren entscheidend für den Erfolg des Projekts.

7.2 Verbesserungspotenziale

Die Erweiterung um mehrsprachige Unterstützung und ein Cloud-Backend wäre sinnvoll.

7.3 Ausblick

In Zukunft könnte die App mit Funktionen wie Barcode-Scans oder einer Diagrammdarstellung der Einsparungen erweitert werden.