**SaveUp App**

Ein Bild, das Clipart, Symbol, Cartoon, Grafiken enthält.

Automatisch generierte Beschreibung

Von Yannick & Tunahan

Contents

[1. Informieren Projektziel 2](#_Toc929325329)

[1.1 Hauptfunktionen: 3](#_Toc2137781342)

[1.2 Pflichtanforderungen: 3](#_Toc1711040782)

[1.3 Optionale Erweiterungen 4](#_Toc495724234)

[2. Planen Ziele: 4](#_Toc332371007)

[2.1 Zeitplan: 4](#_Toc250858008)

[2.2 Ressourcen: 4](#_Toc1001238137)

[2.3 Gannt 4](#_Toc1386390469)

[3. Entscheiden Features: 5](#_Toc85168978)

[3.1 Design: 5](#_Toc936741891)

[4. Realisieren Schritte: 6](#_Toc198603700)

[4.1 Probleme bei der Entwicklung: 6](#_Toc1152824463)

[5. Kontrollieren Tests: 6](#_Toc357060862)

[5.1 Ergebnis: 6](#_Toc382196805)

[6. Auswerten Erkenntnisse: 6](#_Toc1756628595)

[7. Fazit: 7](#_Toc1042920070)

1. Informieren Projektziel

Die App "SaveUp" soll Nutzer motivieren, Geld zu sparen, indem sie kleine Ausgaben wie Kaffee oder Süßigkeiten vermeiden. Alle gesparten Beträge werden dokumentiert und eine Übersicht über die Gesamteinsparungen wird angezeigt.

# 1.1 Hauptfunktionen:

* Produkte erfassen mit Beschreibung, Preis und Datum des Verzichts.
* Liste aller Einsparungen mit Gesamtsumme anzeigen.
* Daten lokal speichern.
* Benutzerfreundliches und modernes Design.

# 1.2 Pflichtanforderungen:

* Mindestens drei Seiten:
  1. Startseite mit Navigation.
  2. Erfassungsseite für gesparte Produkte.
  3. Liste mit allen Einträgen und Gesamtsumme.
* Speicherung der Daten in einer Datei (XML oder JSON).
* Einfache Bedienung und ansprechendes Layout.
* Verwendung von XAML-Styles.
* Strukturierung nach MVVM-Muster.
* Eigenes App-Icon.
* Dokumentation und Testen.

# 1.3 Optionale Erweiterungen

* Löschen der erfassten Einträge.
* Grafische Darstellung der gesparten Verzichtsprodukte .

# 2. Planen Ziele:

* Intuitive App, die alle Anforderungen erfüllt.
* Umsetzung von zwei optionalen Features.
* Stabile und performante App.

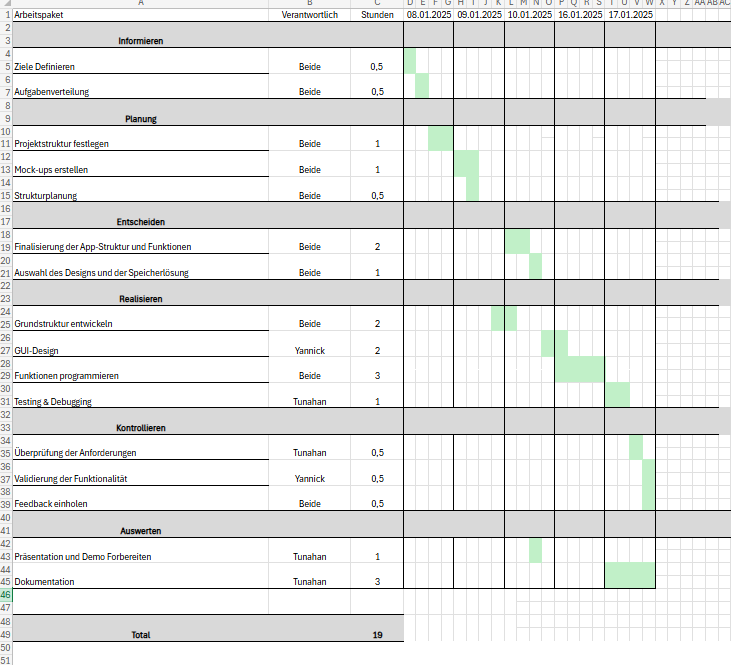
# 2.1 Zeitplan:

* 08.01.2025: Informieren über das Projekt.
* 09.01.2025: Planung für die Projektstruktur und anfangen mit Entwicklung.
* 10.01.2025: Präsentation vorbereiten.
* 16.01.2025: Tests und optionale Features.
* 17.01.2025: Dokumentatsion schreiben.

# 2.2 Ressourcen:

* Visual Studio 2022.
* C#, .NET MAUI.
* Word für Dokumentation.

# 2.3 Gannt



# 3. Entscheiden Features:

* Pflichtanforderungen umsetzen.
* Zusätzlich: Löschfunktion und Diagramme.

# 3.1 Design:

* Fünf Seiten:
  1. Startseite: Navigation.
  2. Erfassungsseite: Eingabe von Daten.
  3. Liste: Anzeige der Einträge mit Gesamtsumme.
  4. Statistik Seite
  5. Settings Page

# 4. Realisieren Schritte:

1. Grundstruktur der App erstellen.
2. Startseite, Erfassungsseite und Liste entwickeln.
3. Daten in JSON-Datei speichern.
4. Löschfunktion erstellen.
5. Tests und Fehlerbehebung.

# 4.1 Probleme bei der Entwicklung:

# 5. Kontrollieren Tests:

* Navigation, Eingaben und Listenansicht testen.
* Feedback von Testnutzern einholen.
* Fehler beheben.

# 5.1 Ergebnis:

* Alle Pflichtanforderungen erfüllt.
* Zwei optionale Features umgesetzt.
* Stabile App.

# 6. Auswerten Erkenntnisse:

* **Erfolge:** Planung führte zu pünktlichem Abschluss. Kenntnisse in .NET MAUI verbessert.
* **Herausforderungen:** Mehrsprachigkeit und Debugging.

# 7. Fazit:

Die Entwicklung der "SaveUp"-App war erfolgreich. Alle Pflichtanforderungen wurden erfüllt, und durch die Umsetzung der Löschfunktion und Diagramme konnten wir den Funktionsumfang erweitern. Die App ist benutzerfreundlich, stabil und optisch ansprechend.

Wir haben wertvolle Erfahrungen mit .NET MAUI gesammelt und Herausforderungen wie Debugging erfolgreich gemeistert. Die strukturierte Planung und gute Teamarbeit waren der Schlüssel zum Erfolg.