



MOBILE APPLIKATION – SAVE UP

Dokumentation zur Projektarbeit
ICT Modul 335

Abgabedatum: Ottenbach, 30.06.2024

Prüfungsbewerber:

Mahir Gönen

Weihermatt 1
8913 Ottenbach
30.06.2024



Inhalt

1	Versionsverzeichnis	3
2	Executive Summary	4
3	Ausgangslage	5
3.1	Projektbeschreibung	5
3.2	Hintergrund und Motivation	5
3.3	Zielsetzung	5
3.4	Rahmenbedingungen	5
3.5	Anforderungen	6
4	Zeitplanung.....	6
4.1	GANTT Diagramm	7
5	Informieren.....	8
5.1	Vorgabe/Anforderungen	8
5.2	Zusätzliche Anforderung	9
5.2.1	Clear-Funktion für alle Einträge	9
5.2.2	Datum als Attribut wählen.....	9
6	Planen.....	10
6.1	Organisation.....	10
6.1.1	Technologieauswahl	10
6.1.2	GitHub.....	11
6.2	Planung der Applikation.....	11
6.2.1	Anforderungsanalyse.....	11
6.2.2	Erstellung von Mockups.....	11
6.2.3	Technische Spezifikationen	12
6.2.4	Zeitplanung und Meilensteine.....	12
6.2.5	Testplanung.....	12
6.2.6	Dokumentation.....	12
6.3	Mockup	13
6.4	Entwurf eines Testplans für Usability-Tests.....	14
7	Entscheiden	14
7.1	Mockup-Design Änderung	14
8	Realisieren	15
8.1	UI-Entwicklung	15
8.2	MVVM-Pattern	16
9	Kontrollieren.....	17
9.1	Testdokumentation	17
10	Auswerten	17

11	Fazit	18
11.1	Persönliches Fazit - Mahir Gönen.....	18
	Anhänge.....	19
I.	Glossar.....	19
II.	Weitere Dokumente	20
III.	Versionsverzeichnis der Anwendung/Pakete	21
12	Quellenverzeichnis	22
13	Tabellenverzeichnis	22
14	Abbildungsverzeichnis.....	Fehler! Textmarke nicht definiert.

1 Versionsverzeichnis

Version	Autor	Datum	Änderung
0.1	Mahir Gönen	29.06.2024	Erstellung des Dokuments
0.2	Mahir Gönen	29.06.2024	Ergänzungen und Überarbeitungen
0.3	Mahir Gönen	30.06.2024	Finalisierung
1.0	Mahir Gönen	30.06.2024	Schlussversion

2 Executive Summary

Das SaveUp-Projekt hatte zum Ziel, eine mobile Applikation zu entwickeln, die Benutzern ermöglicht, kleine, alltägliche Ausgaben zu verfolgen und zu sparen. Nutzer können auf Ausgaben wie Kaffee, Süßigkeiten oder andere kleine Einkäufe verzichten und diese Einsparungen in der App festhalten. Die Anwendung bietet eine übersichtliche Darstellung der gesparten Beträge und motiviert die Nutzer, weiterhin bewusst auf kleine Ausgaben zu verzichten, um grössere finanzielle Ziele zu erreichen.

Die App wurde mit .NET MAUI entwickelt und nutzt das MVVM-Entwurfsmuster, um eine klare Trennung zwischen Benutzeroberfläche, Anwendungslogik und Datenmodell zu gewährleisten. Das Design und die Implementierung basieren auf Mockups, die in Balsamiq erstellt wurden, um eine benutzerfreundliche und intuitive Benutzeroberfläche zu gewährleisten.

Während der Entwicklungsphase wurden mehrere Hauptfunktionen implementiert, darunter die Erfassung und Anzeige von Einsparungen, eine Clear-Funktion zum Löschen aller Einträge und die Möglichkeit, das Datum des Kaufverzichts zu speichern. Aufgrund der begrenzten Zeit konzentrierte sich das Testing hauptsächlich auf Usability-Tests, und einige ursprünglich geplante Funktionen, wie die grafische Darstellung der Daten, wurden nicht umgesetzt.

Die Projektplanung erfolgte nach dem IPERKA-Modell, und die gesamte Entwicklung wurde mithilfe von GitHub dokumentiert. Trotz einiger Herausforderungen, wie der Anpassung der Designstruktur und der Implementierung neuer Funktionen, wurde das Projekt erfolgreich abgeschlossen und bietet eine stabile und benutzerfreundliche Anwendung.

Das SaveUp-Projekt lieferte wertvolle Erkenntnisse und Erfahrungen, die als Grundlage für zukünftige Projekte dienen können. Die gesammelten Feedbacks und die Auswertung der Projektergebnisse halfen dabei, Verbesserungspotenziale zu identifizieren und die Qualität der Anwendung kontinuierlich zu steigern.

3 Ausgangslage

3.1 Projektbeschreibung

Das Projekt "SaveUp" ist eine mobile Applikation, die den Benutzern ermöglicht, kleine, alltägliche Ausgaben zu verfolgen und zu sparen. Die Benutzer können auf Ausgaben wie Kaffee, Süßigkeiten oder andere kleine Einkäufe verzichten und diese Einsparungen in der App festhalten. Ziel der Anwendung ist es, den Benutzern einen Überblick über ihre eingesparten Beträge zu geben und sie zu motivieren, weiterhin bewusst auf kleine Ausgaben zu verzichten, um grössere finanzielle Ziele zu erreichen.

3.2 Hintergrund und Motivation

Die Idee für die SaveUp-App entstand aus der Notwendigkeit, eine einfache Methode zur Verfolgung und Verwaltung von kleinen Einsparungen zu schaffen. Oftmals gehen kleine Beträge, die durch den Verzicht auf alltägliche Ausgaben eingespart werden, im täglichen Leben unter und werden nicht bewusst wahrgenommen. Mit SaveUp können Benutzer diese Beträge sichtbar machen und ihr Sparverhalten nachvollziehen.

3.3 Zielsetzung

Die Hauptziele des Projekts sind:

- Erfassung und Speicherung: Ermöglichen Sie den Benutzern, gesparte Beträge und deren Beschreibungen einfach zu erfassen und zu speichern.
- Übersicht und Motivation: Stellen Sie eine übersichtliche Liste der eingesparten Beträge zur Verfügung, um den Benutzern ihre Fortschritte und Erfolge zu zeigen.
- Benutzerfreundlichkeit: Entwickeln Sie eine intuitive und benutzerfreundliche Oberfläche, die leicht zu navigieren ist.

3.4 Rahmenbedingungen

Für die Umsetzung des Projekts gelten folgende Rahmenbedingungen:

- Die Anwendung wird mit .NET MAUI und C# entwickelt.
- Das Design und die Implementierung erfolgen nach dem MVVM-Entwurfsmuster.
- Die App besteht aus mindestens drei Content Pages.
- Die Anwendung soll auf verschiedenen Geräten lauffähig sein, insbesondere auf mobilen Plattformen.

3.5 Anforderungen

Die grundlegenden Anforderungen an die SaveUp-App umfassen:

- Name und Titel der App: SaveUp
- Content Pages: Mindestens drei Content Pages für verschiedene Funktionen der App.
- GUI Design: Ansprechendes und intuitives Design, unterstützt durch Mock-ups.
- Produktspeicherung: Erfassung von Kurzbeschreibungen und Preisen der gesparten Produkte.
- Menüfunktionen: Bereitstellung von Funktionen zur Speicherung und Anzeige der Liste der gesparten Produkte.
- Design und Layout: Einfache und intuitive Bedienung mit geeignetem Layout und XAML-Styles.
- App-Icon: Ein eigenes App-Icon zur Identifikation der Anwendung.

4 Zeitplanung

Das Projekt wird in mehreren Phasen durchgeführt und es wurden folgender Zeitplan festgelegt:

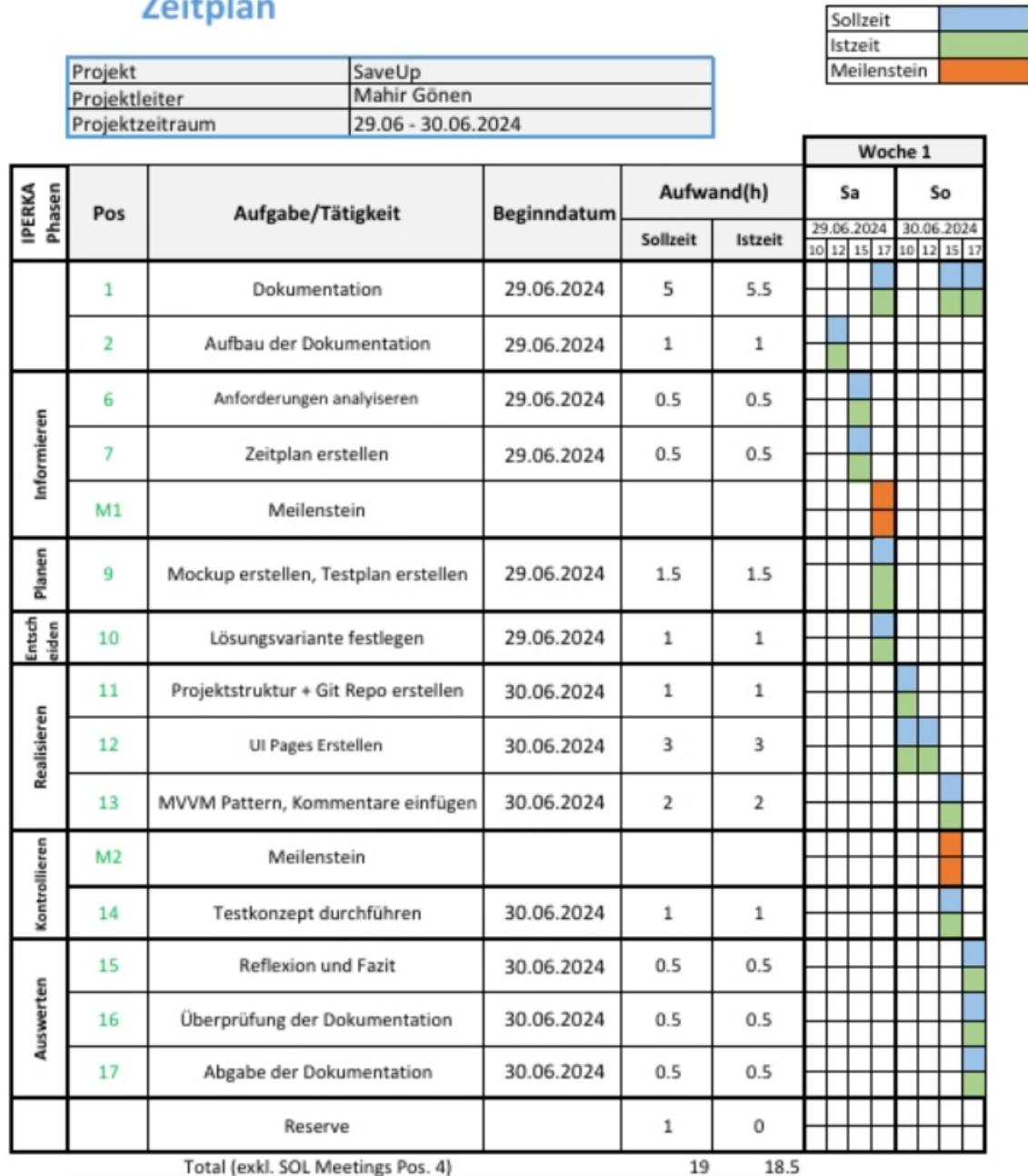
Projektphase	Geplante Zeit
Informieren	1h
Planung, Entwurf, Entscheidung	2.5h
Realisierung	6h
Dokumentation	7.5h
Gesamt	18.5h

Tabelle 1: Grobe Planung

4.1 GANTT Diagramm

Ein GANTT-Diagramm ist nützlich für das SaveUp-Projekt, da es eine visuelle Darstellung des Projektzeitplans bietet. Es hilft dabei, Aufgaben und Meilensteine klar zu strukturieren und die zeitlichen Abhängigkeiten zu erkennen. Durch die Visualisierung können Ressourcen effizient geplant und zugewiesen werden. Zudem erleichtert es die Überwachung des Projektfortschritts und ermöglicht eine klare Kommunikation innerhalb des Teams und mit Stakeholdern.

Zeitplan



5 Informieren

In der Informieren-Phase des IPERKA-Modells geht es darum, alle notwendigen Informationen und Anforderungen für das SaveUp-Projekt zu sammeln und zu verstehen. Für dieses Projekt umfasst die Informieren-Phase:

- Zielgruppenanalyse: Verstehen, wer die Benutzer der App sein werden und welche Bedürfnisse sie haben.
- Anforderungserhebung: Sammlung der funktionalen und nicht-funktionalen Anforderungen, wie z.B. die Pflichtenanforderungen (Content Pages, GUI-Design) und optionalen Erweiterungen.
- Technologie- und Tool-Recherche: Untersuchung der relevanten Technologien und Werkzeuge, wie .NET MAUI, C#, XAML-Styles und MVVM-Entwurfsmuster.
- Best Practices und Beispiele: Sammeln von Best Practices und Beispielanwendungen zur Orientierung und Inspiration für die eigene Implementierung.

Diese Phase bildet die Grundlage für die weiteren Schritte im Projekt und stellt sicher, dass alle Beteiligten ein gemeinsames Verständnis der Ziele und Anforderungen haben.^

5.1 Vorgabe/Anforderungen

Die Hauptziele des SaveUp-Projekts umfassen die Entwicklung und Implementierung von Schlüsselementen, die für eine effiziente und benutzerfreundliche Verwaltung kleiner Einsparungen erforderlich sind:

- Erfassung von Einsparungen: Benutzer sollen eine Kurzbeschreibung und den Preis der gesparten Artikel eingeben und speichern können.
- Anzeigen der Einsparungen: Eine Liste der gesparten Artikel inkl. der Preissumme soll angezeigt werden.
- Speicher- und Aufruffunktion: Bereitstellung von Menüfunktionen zur Speicherung und Anzeige der Liste.
- Intuitive Bedienung: Einfache und intuitive Benutzeroberfläche mit geeignetem Layout und XAML-Styles.

Die Dokumentation folgt dem IPERKA-Modell und ist in entsprechende Abschnitte unterteilt, die die Informationsbeschaffung, Planung, Entscheidung, Realisierung, Kontrolle und Auswertung des Projekts abdecken. Ein Versionsverzeichnis stellt sicher, dass die bereitgestellten Informationen aktuell und relevant sind.

5.2 Zusätzliche Anforderung

Im Laufe der Informationsphase wurden neben den grundlegenden Anforderungen weitere spezifische Bedürfnisse und Wünsche identifiziert, die das Projekt in einer optimierten Form prägen könnten. Diese zusätzlichen Anforderungen sollen sicherstellen, dass die Applikation in ihrer Funktion nicht nur den grundlegenden Bedürfnissen, sondern auch spezielleren Ansprüchen gerecht wird.

5.2.1 Clear-Funktion für alle Einträge

Neben den grundlegenden Anforderungen wurde die Clear-Funktion als optionale Anforderung definiert. Diese Funktion ermöglicht es den Benutzern, alle gespeicherten Einträge mit einem einzigen Befehl zu löschen. Sie bietet eine schnelle und einfache Möglichkeit, die gespeicherten Daten vollständig zu entfernen, was besonders nützlich ist, wenn die Benutzer einen Neustart ihrer Einsparungen vornehmen möchten. Die Implementierung dieser Funktion erhöht die Flexibilität und Benutzerfreundlichkeit der Anwendung.

5.2.2 Datum als Attribut wählen

Eine weitere optionale Anforderung ist die Aufnahme des Datums als zusätzliches Attribut für jeden Eintrag. Dieses Attribut ermöglicht es den Benutzern, genau zu verfolgen, wann der Kaufverzicht erfolgte. Durch die Speicherung des Datums wird es möglich, die Einsparungen chronologisch anzuzeigen und auszuwerten. Dies kann den Benutzern helfen, ihre Fortschritte über bestimmte Zeiträume zu analysieren und Muster in ihrem Sparverhalten zu erkennen.

6 Planen

Die Planungsphase stellt einen entscheidenden Abschnitt dar. Nachdem in der Informationsphase alle erforderlichen Daten gesammelt und ein klares Bild der Anforderungen und Rahmenbedingungen geschaffen wurde, gilt es nun, diese Erkenntnisse in einen strukturierten und realisierbaren Plan zu übertragen.

6.1 Organisation

Um die Architektur und das Zusammenspiel der Komponenten des Applikationssystems optimal zu gestalten, wurde eine sorgfältige Auswahl an Technologien und Methoden getroffen.

6.1.1 Technologieauswahl

Für die Entwicklung der SaveUp-App wurde .NET MAUI (Multi-platform App UI) gewählt. Diese Technologie bietet ein leistungsstarkes und flexibles Framework für die Erstellung von plattformübergreifenden Anwendungen, die auf verschiedenen Geräten, einschliesslich mobiler Geräte, lauffähig sind. Die Entscheidung für .NET MAUI basiert auf seiner umfangreichen Funktionalität, seiner nahtlosen Integration in das .NET-Ökosystem und seiner Fähigkeit, mit einem einzigen Codebase verschiedene Plattformen zu unterstützen.

Benutzeroberfläche

Die Gestaltung der Benutzeroberfläche erfolgt mit XAML (eXtensible Application Markup Language), das in .NET MAUI für die Definition der visuellen Elemente verwendet wird. XAML ermöglicht eine klare Trennung von Design und Logik und unterstützt die Implementierung des MVVM (Model-View-ViewModel) Musters, welches eine effiziente und strukturierte Codeverwaltung fördert. Für die Erstellung von Mockups wurde Balsamiq genutzt, ein benutzerfreundliches Tool, das schnelles und intuitives Prototyping ermöglicht.

UX/UI Design

- Mockups: Zur Visualisierung und Planung der Benutzeroberfläche wurden Mockups in Balsamiq erstellt. Diese Mockups halfen dabei, das Design zu iterieren und sicherzustellen, dass die Benutzerführung intuitiv und ansprechend ist.
- XAML-Styles: Die Verwendung von XAML-Styles ermöglicht ein konsistentes und anpassbares Design der Benutzeroberfläche, wodurch eine einheitliche und ansprechende Benutzererfahrung gewährleistet wird.

6.1.1.1 Anwendungen und NuGet-Pakete

Alle verwendeten Anwendungen und NuGet-Pakete werden hier aufgelistet und beschrieben. Um Komplikationen zu verhindern sind die Versionierungen und vollständigen Beschreibung auf Seite dokumentiert.

Die Applikation läuft unter dem .NET-Framework MAUI 8.0

Folgende NuGet-Pakete werden benötigt für die Applikation:

- CommunityToolkit.Mvvm
- Microsoft.Maui.Controls
- Microsoft.Maui.Controls.Compatibility

6.1.2 GitHub

Mithilfe von GitHub wurde das komplette Projekt durch Versionsverläufe dokumentiert. Durch ein Git-Repository kann der Entwickler in der Realisierungsphase die komplette Programmierarbeit einfacher und unabhängig des Standortes bewältigen. Durch die Commits, die der Entwickler bei allen Änderungen macht, bringt dies eine grosse Transparenz in das Projekt zwischen dem Projektteam und Kunden bei.

6.2 Planung der Applikation

Die Planung der SaveUp-Applikation umfasst mehrere wesentliche Schritte, die sicherstellen, dass das Projekt systematisch und effizient durchgeführt wird. Diese Planung wird durch das IPERKA-Modell unterstützt und beinhaltet folgende Phasen:

6.2.1 Anforderungsanalyse

In dieser Phase werden die funktionalen und nicht-funktionalen Anforderungen der Applikation detailliert erfasst. Dazu gehören:

- Erfassung und Speicherung von Einsparungen
- Darstellung der gesparten Beträge in einer Liste
- Implementierung einer benutzerfreundlichen Oberfläche

6.2.2 Erstellung von Mockups

Zur Visualisierung der Benutzeroberfläche werden Mockups mit Balsamiq erstellt. Diese Mockups dienen als Grundlage für das Design und helfen, die Benutzerführung zu planen und zu testen.

6.2.3 Technische Spezifikationen

Festlegung der technischen Rahmenbedingungen und Auswahl der Technologien:

- Framework: .NET MAUI für plattformübergreifende Entwicklung
- Programmiersprache: C#
- UI-Design: XAML

6.2.4 Zeitplanung und Meilensteine

Erstellung eines GANTT-Diagramms zur Visualisierung des Projektzeitplans. Wichtige Meilensteine umfassen:

- Abschluss der Anforderungsanalyse und Mockups
- Implementierung der Grundfunktionen

6.2.5 Testplanung

Entwicklung eines detaillierten Testplans, der alle Aspekte der Anwendung abdeckt:

- Usability-Tests zur Überprüfung der Benutzerfreundlichkeit

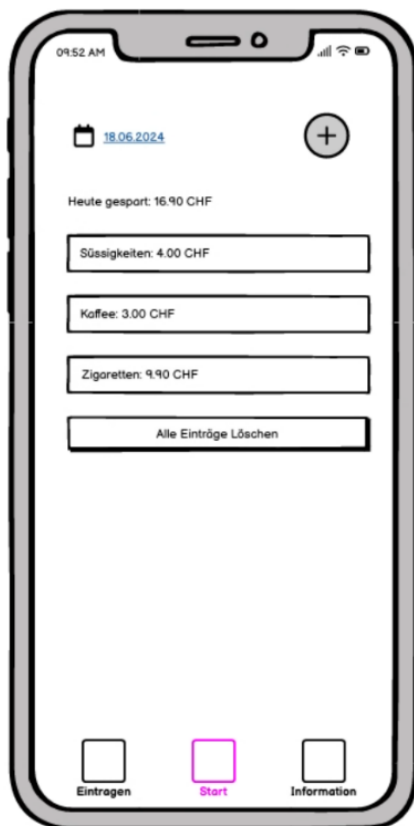
6.2.6 Dokumentation

Erstellung einer umfassenden Projektdokumentation, die alle Phasen und Ergebnisse des Projekts beschreibt. Die Dokumentation folgt dem IPERKA-Modell und enthält:

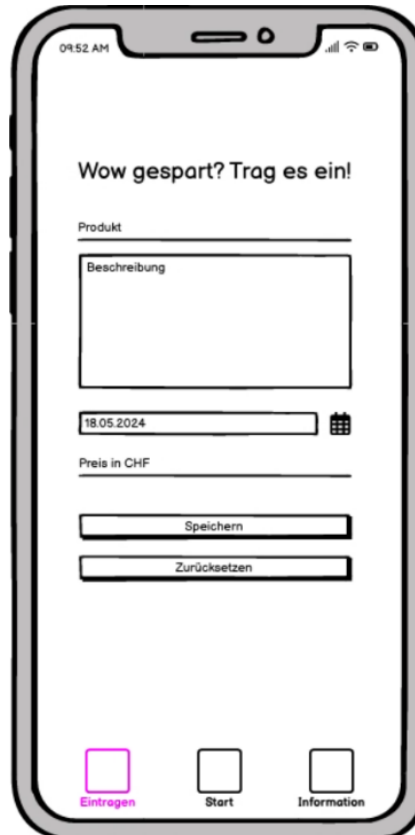
- Ausgangslage und Zielsetzung
- Mockups und Designentscheidungen
- Entwicklungs- und Testprotokolle
- Lessons Learned und zukünftige Verbesserungsvorschläge

Diese strukturierte Planung stellt sicher, dass alle Aspekte der Applikation systematisch angegangen werden und das Projekt erfolgreich abgeschlossen wird.

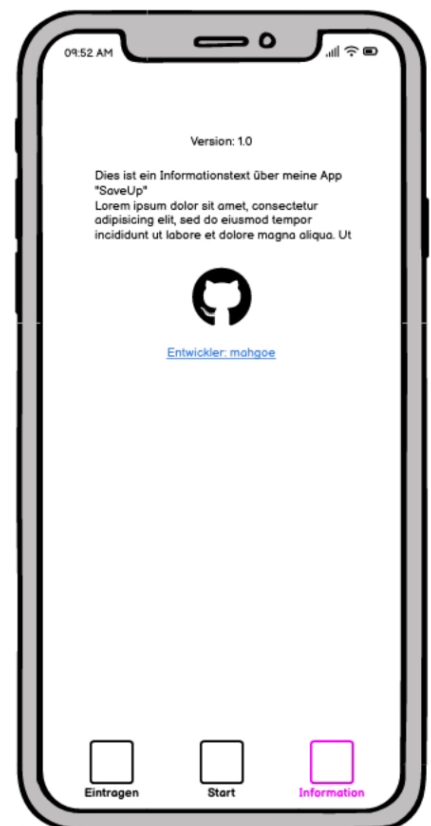
6.3 Mockup



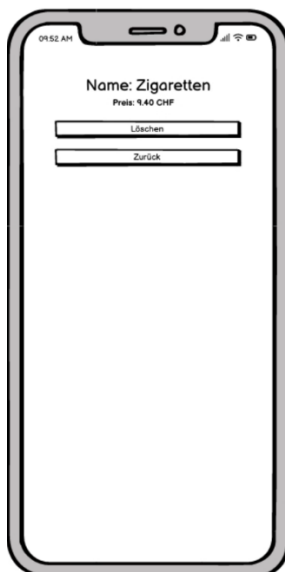
MainPage.xml



EntryPage.xml



InfoPage.xml



PopupPage.xml

6.4 Entwurf eines Testplans für Usability-Tests

Testnummer	Testbeschreibung	Erwartetes Ergebnis
U01	Einsparungen erfassen	Benutzer geben eine Kurzbeschreibung und den Preis eines gesparten Artikels ein und speichern diesen.
U02	Einsparungen anzeigen	Benutzer rufen die Liste der gesparten Artikel auf und überprüfen die Gesamtpreisumme.
U03	Einträge löschen	Benutzer kann alle oder ausgewählte Einträge löschen

Tabelle 2 Testplan

7 Entscheiden

In der Entscheiden-Phase des IPERKA-Modells wird aus den gesammelten Informationen und analysierten Optionen die beste Lösung ausgewählt. Für das SaveUp-Projekt bedeutet dies:

- Design: Festlegung auf ein benutzerfreundliches und intuitives UI/UX-Design basierend auf Mockups aus Balsamiq.
- Funktionen: Auswahl der Pflicht- und optionalen Anforderungen, die umgesetzt werden sollen, wie die Clear-Funktion und Datumseingabe.

Diese Entscheidungen bilden die Grundlage für die nächsten Phasen des Projekts, insbesondere die Umsetzung und das Testen.

7.1 Mockup-Design Änderung

Im Verlauf des Projekts wurden die ursprünglichen Mockups überarbeitet, um die neuen Erkenntnisse und die besonderen Eigenschaften von .NET MAUI zu berücksichtigen. Da MAUI einige Vorteile bietet, die in der anfänglichen Planung nicht vollständig berücksichtigt wurden, war eine Anpassung der Designstruktur notwendig. Zudem wurde auf die JSON-Speicherfunktion verzichtet, was zu einer Änderung der Einstellungsseite führte.

7.1.1 Änderungen im Mockup-Design

Anpassung der Strukturen:

- Navigation und Layout: Verbesserungen im Navigationsfluss und im Layout, um die Vorteile von MAUI optimal zu nutzen und eine bessere Benutzererfahrung zu gewährleisten
- UI-Komponenten: Einsatz von MAUI-spezifischen UI-Komponenten und Styles, die eine konsistente und ansprechende Benutzeroberfläche bieten

Verzicht auf JSON-Speicherung:

Die ursprünglich geplante JSON-Speicherfunktion wurde gestrichen

Stattdessen wurde eine Informationsseite zur App und den Entwicklern eingeführt

7.1.2 Neue Mockup-Elemente

Informationsseite:

- Enthält Informationen zur App, wie Funktionsübersicht und Zielsetzung
- Stellt die Entwickler vor und gibt Einblicke in den Entwicklungsprozess

Überarbeitete Content Pages:

- Eingabeseite: Optimierte Felder für die Eingabe von Kurzbeschreibung und Preis.
- Anzeigeseite: Verbesserte Darstellung der Liste der gesparten Artikel mit klarerer Übersicht.
- Navigationsmenü: Einfacher Zugang zu den Hauptfunktionen der App, inklusive der neuen Informationsseite.

Diese Änderungen verbessern die Benutzerfreundlichkeit und passen die App besser an die technischen Möglichkeiten und die Anforderungen an.

8 Realisieren

Dieser Abschnitt dokumentiert die Umsetzungsphase des Projekts, in der die geplanten Konzepte und Entscheidungen in die Praxis umgesetzt werden.

8.1 UI-Entwicklung

Die UI-Entwicklung für die App umfasst die Gestaltung und Implementierung der Benutzeroberfläche, basierend auf den erstellten Mockups und den spezifischen Anforderungen des Projekts. Die Entwicklung erfolgt in .NET MAUI und verwendet XAML für die Definition der visuellen Elemente.

8.1.1 Implementierungsschritte

1. Setup der Entwicklungsumgebung:

- Installation und Konfiguration von Visual Studio mit .NET MAUI-Unterstützung.
- Einrichtung des Projektordners und der grundlegenden Projektstruktur.

2. Erstellung der Hauptseiten:

- **Eingabeseite:** Implementierung der Felder für Kurzbeschreibung und Preis. Diese Seite ermöglicht den Benutzern die Eingabe und Speicherung ihrer gesparten Artikel.
- **Anzeigeseite:** Darstellung der Liste der gesparten Artikel. Diese Seite zeigt eine Übersicht aller gespeicherten Artikel und die Gesamtsumme der gesparten Beträge.
- **Informationsseite:** Bereitstellung von Informationen zur App und den Entwicklern. Diese Seite wurde anstelle der ursprünglichen JSON-Speicherfunktion integriert und enthält Details über die App und deren Entwickler.

3. Popup-Seite:

- **Löschbestätigung:** Eine Popup-Seite wurde erstellt, die angezeigt wird, wenn ein gespeicherter Artikel ausgewählt wird. Diese Seite bietet Optionen, um den Artikel zu löschen oder zur Hauptseite zurückzukehren. Diese Funktion verbessert die Benutzerfreundlichkeit, indem sie eine einfache und schnelle Möglichkeit bietet, gespeicherte Artikel zu verwalten.

4. Navigationsmenü:

Integration eines einfachen und intuitiven Navigationsmenüs, das den Benutzern schnellen Zugriff auf die verschiedenen Hauptfunktionen der App ermöglicht.

5. Design und Layout:

- Verwendung von XAML-Styles zur Konsistenz und Anpassbarkeit des Designs.
- Sicherstellung einer benutzerfreundlichen und ansprechenden Benutzeroberfläche durch iterative Überarbeitungen und Tests.

Die UI-Entwicklung ist ein entscheidender Schritt, um sicherzustellen, dass die SaveUp-App nicht nur funktional ist, sondern auch eine intuitive und angenehme Benutzererfahrung bietet.

8.2 MVVM-Pattern

Das Model-View-ViewModel (MVVM) Pattern wurde für die Architektur der SaveUp-App gewählt. MVVM trennt die Zuständigkeiten der Benutzeroberfläche (View), der Anwendungslogik (ViewModel) und der Daten (Model) klar voneinander. Dies führt zu einer besseren Wartbarkeit, Testbarkeit und Wiederverwendbarkeit des Codes.

1. Model:

- Stellt die Daten und Geschäftslogik der Anwendung dar.
- Beispiele: Klassen für gespeicherte Artikel, Validierungslogik.

2. View:

- Repräsentiert die Benutzeroberfläche und ist für die Darstellung der Daten zuständig.
- Beispiele: XAML-Dateien, die das Layout und die visuellen Elemente definieren.

3. ViewModel:

- Vermittelt zwischen Model und View.
- Enthält die Logik zur Verarbeitung von Benutzereingaben und zur Aktualisierung der View.

8.2.1 Vorteile des MVVM-Patterns

- **Klare Trennung der Zuständigkeiten:** Erleichtert die Wartung und Erweiterung des Codes.
- **Bessere Testbarkeit:** ViewModels können unabhängig von der Benutzeroberfläche getestet werden.
- **Wiederverwendbarkeit:** Komponenten können leichter wiederverwendet und in anderen Projekten eingesetzt werden.

Durch die Verwendung des MVVM-Patterns wird die SaveUp-App robust, gut strukturiert und zukunftssicher.

9 Kontrollieren

Die Kontrollphase ist entscheidend, um sicherzustellen, dass das entwickelte System den Anforderungen und Erwartungen entspricht. Ein wichtiger Bestandteil dieser Phase ist die Implementierung eines Testprojekts.

9.1 Testdokumentation

Die Tests wurden jeweils vom **Projektmitarbeiter Mahir Gönen** ausgeführt am **30.06.2024**

Testnummer	Testbeschreibung	Erwartetes Ergebnis	Ergebnis	Bemerkung
U01	Einsparungen erfassen	Benutzer geben eine Kurzbeschreibung und den Preis eines gesparten Artikels ein und speichern diesen.	OK	-
U02	Einsparungen anzeigen	Benutzer rufen die Liste der gesparten Artikel auf und überprüfen die Gesamtpreisumme.	OK	-
U03	Einträge löschen	Benutzer kann alle oder ausgewählte Einträge löschen	OK	-

Tabelle 3 Testdokumentation

10 Auswerten

Die Auswertungsphase des Projekts ermöglichte eine umfassende Analyse der Ergebnisse und Erfahrungen. Diese Phase war wichtig, um die Leistung der entwickelten Applikation zu bewerten und wertvolle Erkenntnisse für zukünftige Projekte zu gewinnen.

10.1 Evaluierung der Projektergebnisse

- **Benutzerfreundlichkeit:** Die App wurde als benutzerfreundlich und intuitiv bewertet, mit einfacher Navigation und klarem Design.
- **Funktionalität:** Alle grundlegenden Funktionen arbeiteten wie erwartet und wurden positiv aufgenommen.
- **Stabilität:** Die Anwendung zeigte sich stabil und zuverlässig in den Tests.

10.2 Rückmeldungen und Verbesserungsmöglichkeiten

- **Positives Feedback:** Nutzer lobten die strukturierte Benutzeroberfläche und die einfache Handhabung.
- **Verbesserungspotenzial:** Wünsche nach erweiterten Funktionen wie Kategorisierung und detaillierten Berichten wurden geäußert.

10.3 Erkenntnisse für zukünftige Projekte

- **Nutzerfeedback:** Direkte Rückmeldungen sind entscheidend für die Verbesserung der Nutzererfahrung.
- **Flexibles Design:** Anpassungsfähigkeit im Design ist wichtig für den langfristigen Erfolg.
- **Kontinuierliche Verbesserung:** Regelmässige Überprüfung und Anpassung der App basierend auf Nutzerfeedback.

Die Auswertungsphase lieferte wertvolle Einsichten, die als Basis für zukünftige Innovationen und Verbesserungen dienen werden

11 Fazit

Das SaveUp-Projekt war erfolgreich und hat die gesetzten Ziele erreicht. Die entwickelte App bietet eine benutzerfreundliche Oberfläche und zuverlässige Funktionalität zur Verwaltung von Einsparungen. Nutzerfeedback bestätigte die intuitive Bedienung und den praktischen Nutzen der Anwendung. Wertvolle Erkenntnisse und Verbesserungspotenziale wurden identifiziert, die als Grundlage für zukünftige Projekte dienen werden. Insgesamt hat das Projekt gezeigt, dass eine sorgfältige Planung und iterative Entwicklung entscheidend für den Erfolg sind.

11.1 Persönliches Fazit - Mahir Gönen

Das Projekt war eine wertvolle und lehrreiche Erfahrung für mich. Obwohl ich viel Spass an der Entwicklung hatte, war die verfügbare Zeit begrenzt. Dadurch mussten einige Features auf das Wesentliche reduziert werden, und das Testing konzentrierte sich hauptsächlich auf Usability-Tests anstelle automatisierter UI-Tests. Die grafische Darstellung, die in den ursprünglichen Mockups vorgesehen war, konnte aus Zeitgründen nicht implementiert werden. Trotz dieser Herausforderungen bin ich stolz auf das, was ich erreicht habe, und habe wertvolle Erkenntnisse und Fähigkeiten gewonnen, die mich in meiner weiteren Entwicklung als Softwareentwickler unterstützen werden.

Insgesamt war dieses Projekt eine lohnende Erfahrung, die mir nicht nur tiefe Einblicke in die technischen Aspekte der Softwareentwicklung gegeben hat sondern auch mich innerhalb kürzester Zeit sehr produktiv gemacht hat.

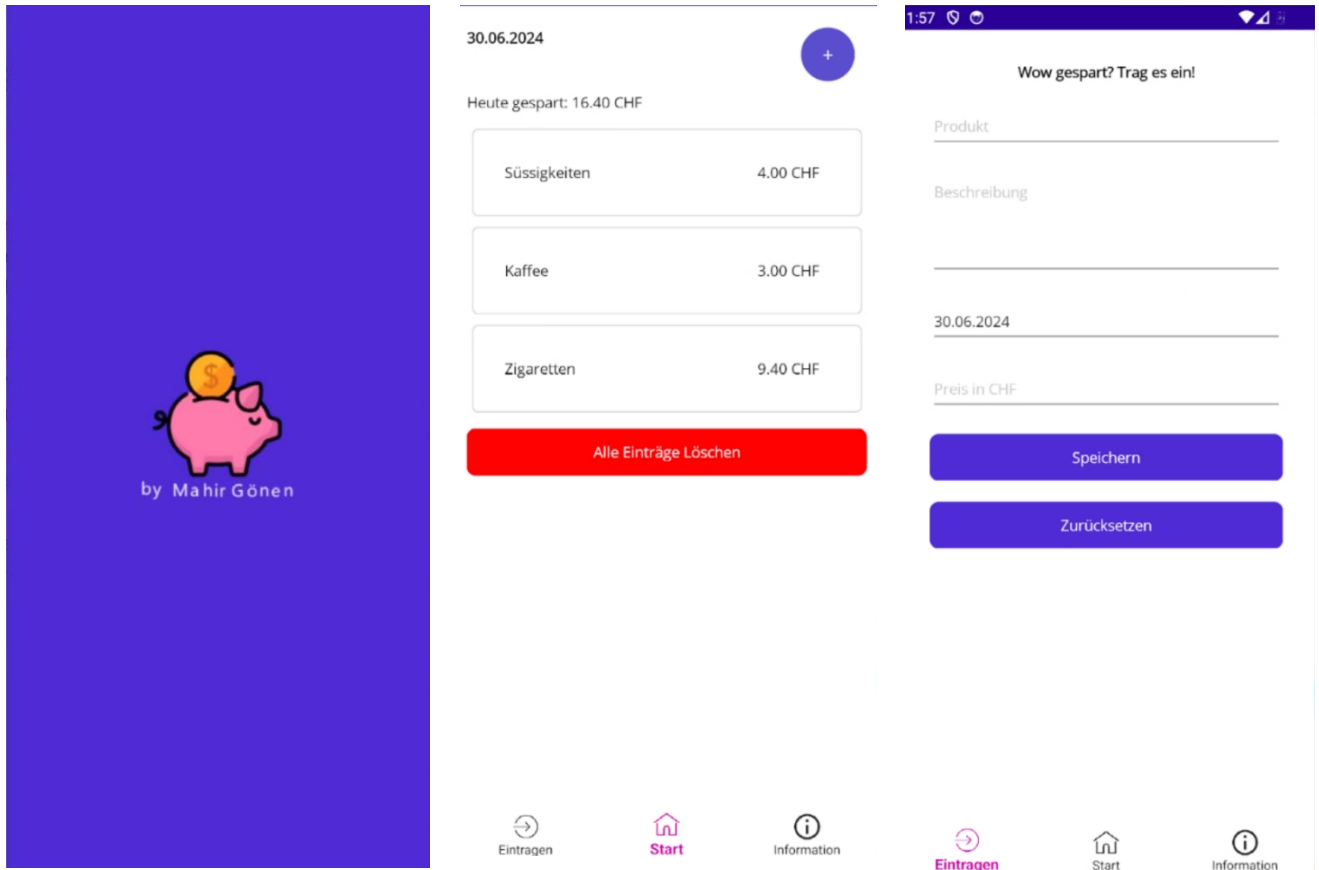
Anhänge

I. Glossar

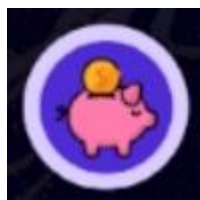
BEGRIFF	BESCHREIBUNG / ERKLÄRUNG
.NET MAUI	Ein Framework zur Entwicklung plattformübergreifender Anwendungen mit einer einzigen Codebasis.
MVVM	Ein Software-Entwurfsmuster, das die Benutzeroberfläche, die Anwendungslogik und das Datenmodell trennt.
BALSAMIQ	Ein Tool zur Erstellung von Mockups und Prototypen für Benutzeroberflächen.
USABILITY-TESTS	Tests zur Bewertung der Benutzerfreundlichkeit und intuitiven Bedienbarkeit einer Anwendung.
MOCKUP	Ein statisches Modell oder Prototyp einer Benutzeroberfläche, das zur Planung und Visualisierung verwendet wird.
GITHUB	Eine Plattform zur Versionsverwaltung und gemeinsamen Entwicklung von Softwareprojekten.
CLEAR-FUNKTION	Eine Funktion, die es Benutzern ermöglicht, alle gespeicherten Daten mit einem einzigen Befehl zu löschen.

Tabelle 4 Glossar

II. Weitere Dokumente



Splash-Screen



MainPage.xml



EntryPage.xml

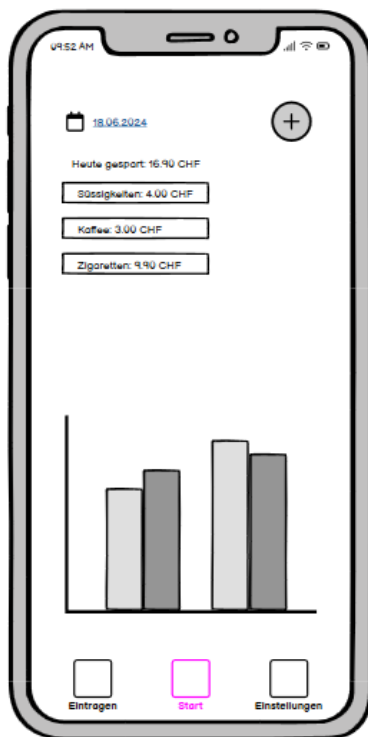


Icon

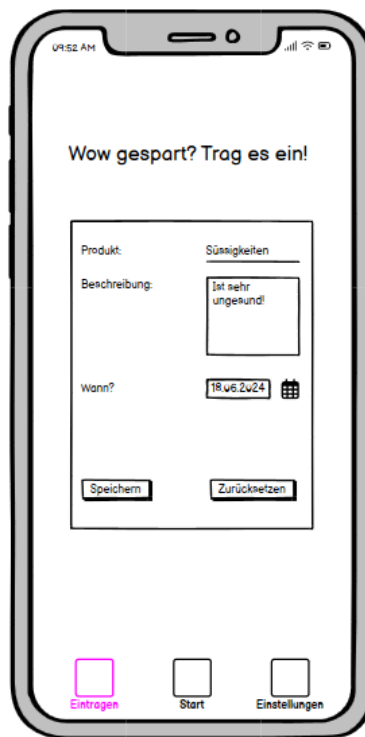
InfoPage.xml

PopupPage.xml

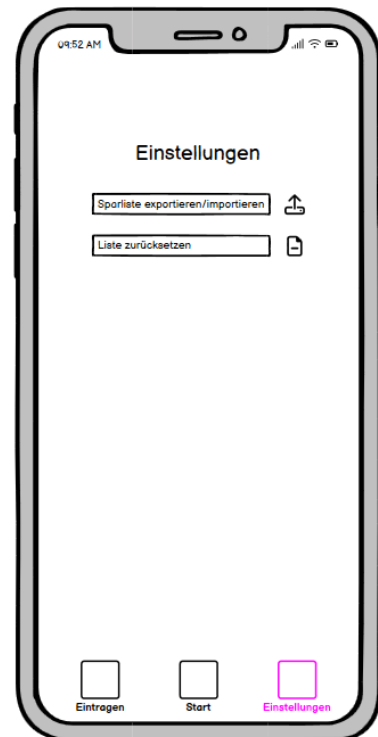
Screenshots des alten Mockups



MainPage



EntryPage



SettingsPage

III. Versionsverzeichnis der Anwendung/Pakete

Anwendung/Paket	Version
.NET MAUI	.NET8.0
CommunityToolkit.Mvvm	8.2.2
Microsoft.Maui.Controls	8.0.0
Microsoft.Maui.Controls.Compatibility	8.0.0
Microsoft.Extensions.Logging.Debug	8.0.0

Tabelle 5 Versionsverzeichnis

12 Quellenverzeichnis

OneNote

Microsoft MAUI Document, Aufgerufen am 29.06.2024 - <https://dotnet.microsoft.com/en-us/apps/maui>

Dotnet MAUI Applcon Tutorial, Aufgerufen am 30.06.2024 - <https://www.youtube.com/watch?v=iigquyXIM0&t=17s>

13 Tabellenverzeichnis

Tabelle 1: Grobe Planung.....	6
Tabelle 2 Testplan	14
Tabelle 3 Testdokumentation	17
Tabelle 4 Glossar.....	19
Tabelle 5 Versionsverzeichnis	22