



SAVEUP



Davud Ponjevic

Inhalt

Einleitung	2
Informieren.....	2
Ziel	2
Anforderungen.....	2
Planen.....	3
Projektplan.....	3
Entscheiden	4
Architektur	4
Werkzeuge	4
Mock-up Design.....	4
Realisieren	5
Projekt Setup.....	5
Model Implementierung	5
ViewModels Implementierung.....	5
Views Implementierung.....	5
Styles und Ressourcen	5
Testing	5
Kontrollieren.....	6
Testing.....	6
Auswerten	7
Fazit	7
Verbesserungspotenzial	7

Einleitung

Die SaveUp-App ist eine mobile Anwendung, die Benutzern hilft, Ausgaben zu erfassen und gesparte Beträge zu verwalten. Diese Dokumentation beschreibt das Projekt unter Verwendung des IPERKA-Modells (Informieren, Planen, Entscheiden, Realisieren, Kontrollieren, Auswerten) und umfasst auch einen Testplan.

Informieren

Ziel

Die App soll Benutzern helfen, ihre Ausgaben zu verfolgen und Ersparnisse für größere private Investitionen (z.B. Ferien) zu verwalten.

Anforderungen

- Name und Titel der App: SaveUp
- Mindestens drei Content Pages
- GUI-Design mit Mock-ups
- Produkterfassung mit Kurzbeschreibung und Preis
- Menüfunktionen zum Speichern und Aufrufen der Listendarstellung
- Einfache, intuitive Bedienung mit passenden Layouts (Styles)
- Strukturierung nach MVVM-Entwurfsmuster
- Verwendung von XAML-Styles für Steuerelemente
- Dokumentation und Testing
- Veröffentlichung in der Google Play Console

Planen

Projektplan

1. Ziele und Anforderungen definieren:

- Erstellen einer detaillierten Liste der funktionalen und nicht-funktionalen Anforderungen.

2. Mock-up Design

- Erstellung eines visuellen Entwurfs (Mock-up) der Benutzeroberfläche.

3. Architektur festlegen

- Entscheidung für die Verwendung von .NET MAUI und das MVVM-Entwurfsmuster.

4. Zeitplan und Meilensteine:

- Festlegen von Meilensteinen und einem detaillierten Zeitplan für die Entwicklung.

5. Ressourcenplanung

- Bestimmung der benötigten Ressourcen (Entwickler, Werkzeuge, Software).

7. ****Dokumentationsplan:****

- Festlegung der zu erstellenden Dokumentation und des Formats.

Entscheiden

Architektur

Die App wird unter Verwendung von .NET MAUI und dem MVVM-Entwurfsmuster implementiert. Dies ermöglicht eine klare Trennung zwischen Logik und UI und unterstützt eine saubere und erweiterbare Architektur.

Werkzeuge

- Visual Studio 2022
- .NET MAUI
- XAML
- C#
- Balsamiq

Mock-up Design

Ein Mock-up wurde erstellt, um die Benutzeroberfläche zu visualisieren. Aufgrund von Zeitbeschränkungen und einer Fehleinschätzung der Umsetzbarkeit wurde das Mock-up jedoch nicht eins-zu-eins in der finalen App umgesetzt.

Realisieren

Projekt Setup

- Erstellen eines neuen .NET MAUI Projekts in Visual Studio.
- Hinzufügen erforderlicher NuGet-Pakete.

Model Implementierung

- Definieren des `SavedItem` Modells zur Speicherung der Ersparnisdaten. Dieses Modell umfasst die Beschreibung des gesparten Produkts, den Preis und das Datum der Ersparnis.

ViewModels Implementierung

- Implementierung des `BaseViewModel`, `MainPageViewModel`, und spezifischer ViewModels zur Verwaltung der Logik und Datenbindung.

Views Implementierung

- Definieren und Implementieren von `App.xaml`, `AppShell.xaml`, und `MainPage.xaml`, um die grundlegende Struktur und Navigation der App bereitzustellen.
- Binding der ViewModels an die Views und Implementierung von Commands für Benutzerinteraktionen wie das Hinzufügen und Löschen von Ersparnissen.

Styles und Ressourcen

- Erstellen von XAML-Styles zur Vereinheitlichung des Designs der Benutzeroberfläche.

Testing

- Durchführung umfassender Tests, um die Funktionalität und Stabilität der App zu gewährleisten. (Im nächsten Schritt genauer)

Kontrollieren

Testing

Ein umfassender Testplan wurde erstellt, um sicherzustellen, dass alle Funktionen der App wie erwartet funktionieren.

Testplan

Testfall-ID	Beschreibung	Erwartetes Ergebnis	Ergebnis
1	Starten der App	App startet ohne Fehler und öffnet Startseite	Pass
2	neuen Artikel kann man auf der separaten Seite einfügen	Soll die separate anzeigen und input Felder sind vorhanden	Pass
3	Artikel anzeigeseite	Diese Seite ist für die Daten	Pass
4	Man sieht die Eingefügten Daten	Man fügt Daten ein und man sieht die	Nein

Der letzte Test funktioniert aufgrund des fehlenden Speicherorts dieser wird aber mit der nächsten Version hinzugefügt.

Auswerten

Fazit

Die SaveUp-App erfüllt die meisten der gestellten Anforderungen. Aufgrund von Zeitbeschränkungen und einer Fehleinschätzung bezüglich der Umsetzbarkeit konnte das Mock-up nicht eins-zu-eins in der finalen App umgesetzt werden. Dennoch bietet die App eine funktionale und benutzerfreundliche Lösung für die Erfassung und Verwaltung gesparter Ausgaben.

Verbesserungspotenzial

- Erweiterung der Funktionen für das Speichern und Laden der Daten in einer Backend-Datenbank.
- Implementierung einer grafischen Darstellung der gesparten Verzichtprodukte.
- Optimierung der Benutzeroberfläche basierend auf dem ursprünglichen Mock-up.