

Dokumentation

SaveUp-App

Inhaltsverzeichnis

Table des matières

1. Informieren	3
Zielsetzung:.....	3
Rahmenbedingungen:	3
2. Planen	4
Aufgaben:	4
Zeitplan:.....	4
3. Entscheiden	5
Design:	5
Tests:.....	5
4. Realisieren	6
Implementierung:.....	6
5. Kontrollieren.....	6
Tests durchgeführt:.....	6
Usability-Tests:.....	6
6. Auswerten	7
Lessons Learned:	7

1. Informieren

Zielsetzung:

Die SaveUp-App soll Nutzern ermöglichen, ihre gesparten Ausgaben zu verfolgen, um Geld für eine größere private Investition wie einen Urlaub zu sparen. Durch das Erfassen und Speichern von gesparten Kleinbeträgen (z.B. Kaffee, Süßigkeiten) können Nutzer ihren Fortschritt sehen und motiviert bleiben.

Rahmenbedingungen:

Die App wird in .NET MAUI und C# entwickelt.

Sie soll mindestens drei Content Pages umfassen.

Die gesammelten Daten werden in einer CSV-Datei im Downloads-Ordner gespeichert.

Die App wird auf Android veröffentlicht und im Google Play Store verfügbar gemacht.

2. Planen

Aufgaben:

Struktur der App planen: Festlegung der verschiedenen Seiten (Content Pages) und ihrer Funktionalitäten.

GUI-Design und Mockups erstellen: Gestaltung der Benutzeroberfläche und Erstellung von Mockups zur Visualisierung des Designs.

Implementierung der App: Programmierung der App-Funktionalitäten und Integration der Benutzerschnittstellen.

Testen der App: Durchführung von Funktionalitäts- und Usability-Tests, um sicherzustellen, dass die App korrekt und benutzerfreundlich funktioniert.

Dokumentation erstellen: Detaillierte Beschreibung des Projekts und der Implementierungsschritte.

Veröffentlichung der App: Vorbereitung und Upload der App in den Google Play Store.

Zeitplan:

Planungsphase: 1 Stunde

Design und Mockups: 1 Stunde

Implementierung: 4 Stunden

Testen und Fehlerbehebung: 1 Stunde

Dokumentation: 1 Stunde

Veröffentlichung: 1 Stunde

3. Entscheiden

Technologien:

Programmiersprache: C#

Framework: .NET MAUI

Datenpersistenz: CSV-Datei, die im Downloads-Ordner des Benutzers gespeichert wird

Design:

GUI: Die Benutzeroberfläche wird mit XAML gestaltet, um eine klare Trennung von Design und Logik zu ermöglichen.

Entwurfsmuster: Anwendung des Model View ViewModel (MVVM)-Musters zur besseren Strukturierung des Codes und zur Erleichterung der Testbarkeit.

Tests:

Funktionalitätstests: Überprüfung der korrekten Erfassung und Anzeige der gesparten Artikel.

Usability-Tests: Sicherstellung einer intuitiven und benutzerfreundlichen Bedienung der App.

4. Realisieren

Implementierung:

Initialisierung der App: Beim Starten der App wird eine CSV-Datei im Downloads-Ordner erstellt, sofern sie nicht bereits existiert. Dies stellt sicher, dass die Daten persistiert werden können.

Hauptseite: Die Hauptseite der App enthält Buttons, die den Benutzer zu den Seiten zum Hinzufügen von Artikeln und zur Ansicht der gespeicherten Artikel navigieren.

Seite zum Hinzufügen von Artikeln: Der Benutzer kann hier eine Beschreibung und den Preis eines gespeicherten Artikels eingeben. Diese Daten werden dann in der CSV-Datei gespeichert.

Seite zur Ansicht der gespeicherten Artikel: Diese Seite lädt die gespeicherten Daten aus der CSV-Datei und zeigt sie in einer Liste an, damit der Benutzer seinen Fortschritt verfolgen kann.

5. Kontrollieren

Tests durchgeführt:

Test der Funktion zum Hinzufügen von Artikeln: Überprüfung, ob die eingegebenen Daten korrekt in die CSV-Datei geschrieben werden.

Test der Anzeige der gespeicherten Artikel: Überprüfung, ob die Daten korrekt aus der CSV-Datei gelesen und angezeigt werden.

Usability-Tests:

Intuitive Bedienung: Sicherstellung, dass die Benutzeroberfläche einfach und selbsterklärend ist.

Darstellung auf verschiedenen Geräten: Überprüfung, ob die App auf verschiedenen Bildschirmgrößen und Auflösungen korrekt dargestellt wird.

Fehlerbehebung:

Sicherstellen, dass die CSV-Datei korrekt im Downloads-Ordner erstellt und geschrieben wird.

Prüfung und Behebung von Fehlern beim Lesen und Schreiben der CSV-Datei.

Optimierung der Benutzeroberfläche für eine bessere Benutzererfahrung.

6. Auswerten

Zusammenfassung der Ergebnisse:

Die SaveUp-App wurde erfolgreich entwickelt und erfüllt die gestellten Anforderungen. Die App ermöglicht es Nutzern, gesparte Ausgaben einfach zu erfassen und zu verfolgen. Die gesammelten Daten werden in einer CSV-Datei im Downloads-Ordner gespeichert, was eine einfache Verwaltung und Sicherung der Daten ermöglichten. Die App bietet eine intuitive Benutzeroberfläche und eine klare Darstellung der gesparten Beträge.

Lessons Learned:

Effizienz von .NET MAUI: Die Verwendung von .NET MAUI ermöglicht eine effiziente Entwicklung plattformübergreifender Apps, was die Entwicklungszeit reduziert, und die Wartung erleichtert.

Datenpersistenz in CSV-Dateien: Für kleine Datenmengen ist die Speicherung in CSV-Dateien eine einfache und effektive Lösung. Sie erfordert keine komplexe Datenbankverwaltung und ermöglicht dennoch eine schnelle und zuverlässige Datenpersistenz.

Bedeutung einer guten Planung: Eine detaillierte Planung und Strukturierung des Projekts vor Beginn der Implementierung hilft, die Entwicklungszeit zu optimieren und mögliche Fehler zu minimieren.

Diese IPERKA-Dokumentation bietet eine umfassende Übersicht über das Projekt "SaveUp-App" und beschreibt detailliert die einzelnen Schritte und Entscheidungen, die zur erfolgreichen Umsetzung geführt haben.