

SaveUp - App

Modul 335



INF EFZ 22
30.06.2024

1	Informieren	3
2	Planen	4
3	Entscheiden	6
4	Realisieren	7
6	Kontrollieren und Auswerten	8
7	Fazit	9

1 Informieren

Ausgangslage:

Das SaveUp Projekt zielt darauf ab, eine plattformübergreifende Anwendung zu entwickeln, die es Benutzern ermöglicht, Produkte mit einer Kurzbeschreibung und einem Preis zu erfassen. Diese Produkte sollen persistent gespeichert und in einer Listenansicht angezeigt werden können.

Ziele:

- **Produkterfassung:** Benutzer sollen Produkte mit einer Kurzbeschreibung und einem Preis erfassen können.
- **Datenpersistenz:** Die erfassten Produkte sollen lokal auf dem Gerät gespeichert werden, ohne die Verwendung einer Datenbank.
- **Listenansicht:** Eine übersichtliche Darstellung der erfassten Produkte in Form einer Liste.
- **Navigation:** Die Anwendung soll eine Navigation zwischen der Startseite, auf der Produkte erfasst werden können, und der Listenansicht unterstützen.
- **Benutzerfreundlichkeit:** Die Benutzeroberfläche soll intuitiv gestaltet sein, um eine einfache Bedienung zu gewährleisten.

Technologie:

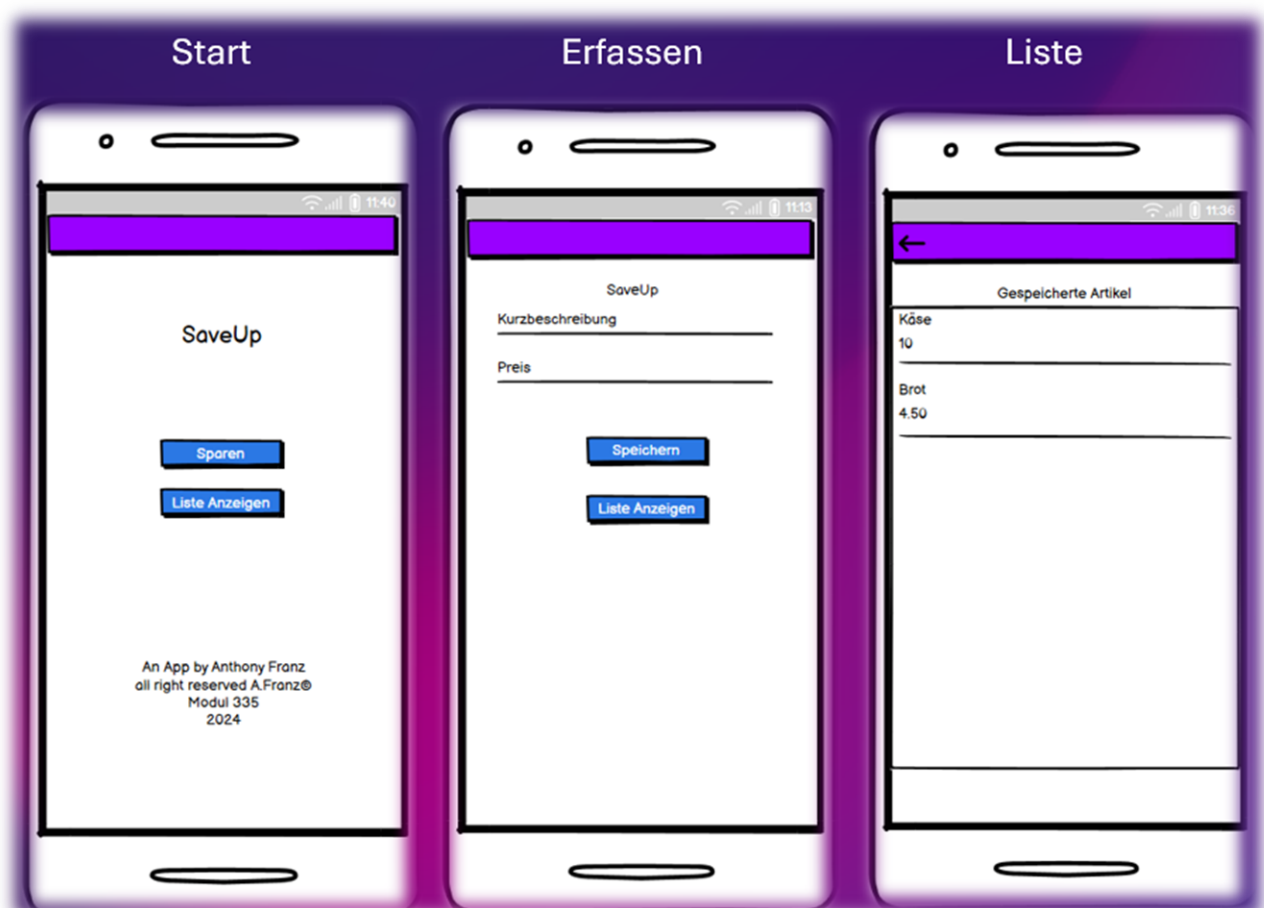
- Nutzung von Microsoft.Maui für die plattformübergreifende Entwicklung.

2 Planen

Mockup:

Das Projekt umfasst folgende Mockups für das Screen Design:

- **StartPage:**
 - Eingabefelder für Kurzbeschreibung und Preis.
 - Buttons zum Speichern und Anzeigen der Liste.
- **MainPage (Erfassen):**
 - Eingabefelder für Kurzbeschreibung und Preis.
 - Buttons zum Speichern und direkten Anzeigen der Liste.
- **ListPage:**
 - Anzeige der erfassten Produkte in einer Liste.
 - Möglichkeit, einzelne Produkte zu löschen.



Arbeitspakete nach Gantt:

Die Entwicklung des SaveUp Projekts wird in folgende Arbeitspakete aufgeteilt:

- **Analyse und Konzeptualisierung:**
 - Anforderungsanalyse
 - Erstellung von User Stories
 - Design der Benutzeroberfläche (UI/UX)
- **Implementierung der Benutzeroberfläche:**
 - Erstellung der StartPage
 - Erstellung der MainPage
 - Erstellung der ListPage
 - Implementierung von Navigation und Buttons
- **Implementierung der Funktionalitäten:**
 - Implementierung der Datenhaltung ohne Datenbank
 - Implementierung der Logik für das Speichern und Laden von Produkten
 - Implementierung der Produktlöschfunktion
- **Testing und Fehlerbehebung:**
 - Erstellung eines Testplans
 - Durchführung von Unit-Tests für Datenhaltung und Logik
 - Integrationstests für die Benutzeroberfläche und Navigation
 - Fehlerbehebung und Optimierung
- **Dokumentation und Abschluss:**
 - Erstellung der IPERKA-Dokumentation
 - Erstellung von Benutzerdokumentation und technischer Dokumentation
 - Projektabschluss und Übergabe

3 Entscheiden

Für die Entwicklung des SaveUp Projekts wurden folgende zentrale Entscheidungen getroffen:

1. Verwendung von Microsoft.Maui als plattformübergreifendes Framework: Microsoft.Maui ermöglicht die Entwicklung der Anwendung für verschiedene Betriebssysteme wie Android, iOS, macOS und Windows mit einer einzigen Codebasis. Dies gewährleistet Effizienz und Skalierbarkeit.
2. Nutzung einer lokalen Dateispeicherung zur Datenpersistenz: Statt einer Datenbanklösung wurde eine lokale Dateispeicherung gewählt, um die Daten der Benutzer auf deren Geräten zu speichern. Dies bietet einfache Implementierung und Wartung, Sicherheit der Daten und Einhaltung von Datenschutzrichtlinien.
3. Implementierung der Benutzeroberfläche basierend auf definierten Mockups: Das Design der Benutzeroberfläche wurde gemäß der vorab erstellten Mockups umgesetzt, um eine intuitive und benutzerfreundliche Anwendung zu gewährleisten.

Diese Entscheidungen sind grundlegend für die effektive Entwicklung und Bereitstellung einer funktionalen und benutzerfreundlichen SaveUp-Anwendung.

4 Realisieren

Die Realisation des SaveUp Projekts basierte auf der Implementierung einer klaren MVVM-Architektur (Model-View-ViewModel), die eine saubere Trennung von Geschäftslogik und Benutzeroberfläche ermöglichte. Hier ist ein Überblick über den Entwicklungsprozess:

1. Architekturwahl und Strukturierung nach MVVM:

- Model: Die Model-Klassen repräsentieren die Datenstruktur der Anwendung. Im SaveUp Projekt umfasst dies insbesondere die `SavedItem` Klasse für die erfassten Produkte.
- View: Die Views in Form von XAML-Dateien (StartPage.xaml, MainPage.xaml, ListPage.xaml) definieren das Aussehen der Benutzeroberfläche.
- ViewModel: Die ViewModel-Klassen enthalten die Geschäftslogik und sind für die Kommunikation zwischen Model und View zuständig. Hier wurde das `MainViewModel` entwickelt, das die Benutzerinteraktionen steuert und die Daten für die Ansicht bereitstellt.

2. Design der XAML-Dateien:

- StartPage.xaml: Diese Seite ermöglicht die Eingabe neuer Produkte mit Feldern für Kurzbeschreibung und Preis sowie Buttons zum Speichern und Anzeigen der Liste.
- MainPage.xaml: Ähnlich wie die StartPage, jedoch mit direkter Anzeige der Liste nach dem Speichern eines Produkts.
- ListPage.xaml: Hier werden die erfassten Produkte in einer CollectionView angezeigt. Jedes Produkt wird mit seiner Beschreibung und dem Preis dargestellt.

3. Integration der ListPage:

- Die ListPage wurde in das Projekt integriert, um eine übersichtliche Darstellung der erfassten Produkte zu ermöglichen.
- Sie verwendet Binding, um die `SavedItems` aus dem `MainViewModel` anzuzeigen, und eine CollectionView, um die Produkte in einem scrollbaren Format darzustellen.

Diese strukturierte Herangehensweise ermöglichte eine klare Trennung der Verantwortlichkeiten innerhalb des SaveUp Projekts, erleichterte die Wartung und Erweiterung der Anwendung und sorgte für eine verbesserte Lesbarkeit und Testbarkeit des Codes.

6 Kontrollieren und Auswerten

Unit-Tests:

- Überprüfung der Datenhaltung (Speichern und Laden von Produkten)
- Überprüfung der Logik für das Löschen von Produkten

Integrationstests:

- Überprüfung der Navigation zwischen StartPage, MainPage und ListPage
- Überprüfung der Benutzeroberfläche auf korrekte Darstellung und Funktionalität der Buttons und Eingabefelder

7 Fazit

Die IPERKA-Dokumentation für das SaveUp Projekt beschreibt die Ausgangslage und Ziele, stellt Mockups für das Screen Design bereit, plant die Arbeitspakete nach Gantt und skizziert einen Testplan mit zugehörigem Testprotokoll. Diese Struktur gewährleistet eine systematische Entwicklung und Qualitätssicherung der Anwendung, die Benutzerfreundlichkeit und Zuverlässigkeit priorisiert.